FAKULTÄT FÜR INFORMATIK
Faculty of Informatics

APPenger

Evaluation, Development and Usability Analysis for a Tactful Passenger App

Diplomarbeit

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering and Internet Computing

eingereicht von

Manuel Mundorf
Matrikelnummer 1128312

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuer: Thomas Grechenig
Mitwirkung: Stefan Kuschnigg
Christoph Wimmer

Wien, 27. Januar 2019

(Unterschrift Verfasser) (Unterschrift Betreuer)

APPenger

Evaluation, Development and Usability Analysis for a Tactful Passenger App

Master's Thesis

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering and Internet Computing

by

Manuel Mundorf
Registration Number 1128312

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Thomas Grechenig
Assistance: Stefan Kuschnigg
Christoph Wimmer

Vienna, January 27, 2019

Technische Universität Wien, Forschungsgruppe INSO
A-1040 Wien ▪ Wiedner Hauptstr. 76/2/2 ▪ Tel. +43-1-587 21 97 ▪ www.inso.tuwien.ac.at

Statement by Author

Manuel Mundorf
Heigerleinstraße 14, 1160 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

I hereby declare that I am the sole author of this thesis, that I have completely indicated all sources and help used, and that all parts of this work – including tables, maps and figures – if taken from other works or from the internet, whether copied literally or by sense, have been labelled including a citation of the source.

(Place, Date) (Signature of Author)

Acknowledgements

First, I want to thank Thomas Grechenig, Stefan Kuschnigg, Christoph Wimmer and Andreas Mauczka for their scientific supervision, patience and helpfulness in the course of this master's thesis.

Special thanks to my parents, Kornelia and Volker, for their ongoing support during my studies.

Furthermore, I wish to thank my girlfriend, who constantly encouraged and supported me. Finally, I thank all people involved in the usability test of the prototype design.

Kurzfassung

Flugreisen werden immer populärer, allein 2014 wurden in Österreich 4,3 Millionen Flüge durchgeführt [63]. An großen Flughäfen ist es für Reisende jedoch problematisch sich zu orientieren. Aktuelle Fluginformationen werden häufig nur mittels Übersichtstafeln angeboten. Zudem sind Wegweiser und Beschilderungen der vorhandenen Geschäfte und Services oft spärlich. Viele Flughafenbetreiber scheinen die rasante technische Entwicklung der letzten Jahre nicht nutzen zu können, um diese Situation zu verbessern. Die vorliegende Diplomarbeit greift diese Problematik auf und untersucht das Optimierungspotenzial durch die Verwendung einer mobilen Anwendung. Smartphones und Tablets, die heutzutage von den meisten Reisenden mitgeführt werden, bieten eine hervorragende Plattform für ein mobiles Informations- und Orientierungssystem. Die Ziele dieser Diplomarbeit sind die Ermittlung wichtiger Funktionen für Flughafen-Apps, das Aufzeigen von Schwächen in bestehenden Lösungen und die Präsentation einer konkreten Umsetzung anhand eines Prototypen.

Zu diesem Zweck wurde ein Kriterienkatalog definiert, der wichtige Funktionen von Flughafen-Apps umfasst. Basierend auf diesem Kriterienkatalog wurden sieben ausgewählte Flughafen-Apps analysiert und 10 Hauptfunktionen identifiziert. Nahezu jede der getesteten Apps bot diese 10 Funktionen an. Darüber hinaus wurden Anwendungsszenarien entwickelt, anhand derer die ausgewählten Flughafen-Apps ein weiteres Mal analysiert wurden. Ziel der zweiten Analyse war die Bewertung der Anwenderfreundlichkeit auf Basis von Heuristiken. Die gewonnenen Erkenntnisse konnten anschließend verwendet werden, um einen Prototypen zu entwickeln, der die aufgedeckten Schwächen vermeidet. Die Entwicklung des Prototypen folgte einem benutzerorientierten Prozessmodell, beginnend bei der Identifizierung potentieller Anwender und deren Bedürfnisse. Für die erarbeiteten Benutzergruppen wurde in weiterer Folge ein entsprechend abgestimmtes Design entworfen. Anwendertests zur Bewertung der Benutzeroberfläche waren ein wesentlicher Bestandteil des Entwicklungsprozesses.

Die Evaluierung bestehender Flughafen-Apps zeigte, dass diese zum Teil gravierende Designschwächen aufweisen. Bereits der erste Entwurf des entwickelten Prototypen konnte bessere Ergebnisse erzielen als die meisten getesteten Apps. Im Zuge dieser Diplomarbeit wurde ein Prozessmodell vorgestellt, das konkrete Techniken zur Entwicklung neuer Flughafen-Apps präsentiert. Der Kriterienkatalog und das Prozessmodell wurden speziell für Flughafen-Apps entwickelt. Die Verwendung des Modells in anderen Anwendungsbereichen wird im Rahmen der Arbeit diskutiert. Weiterhin konnten auf Basis der Marktanalyse und der Anwendertests des Prototypen Design-Richtlinien extrahiert werden, die bei der Entwicklung neuer Flughafen-Apps unterstützen.

Schlüsselwörter

Flughafen, Mobile Anwendung, Anwenderfreundlichkeit, Prozessmodell, Design-Richtlinien

Abstract

Air travel has become increasingly popular. In Austria, there were 4.3 million flights in 2014 [63]. At airports, typical problems for travelers include orientation and information retrieval regarding flights, available shops or offered services. Most often, this information is provided only by signposts and flight overview panels. This thesis presents ideas and approaches for a mobile application to optimize the time for travelers at airports. Smartphones and tablets, which are carried by most travelers today, would be ideal platforms to implement an information and orientation system for the everyday traveler. The objectives of this thesis are to determine important functions for airport applications, reveal weaknesses of existing solutions and develop a prototype that solves the presented problems.

To this end, a criteria catalog was developed that covers important features of airport applications. Using this catalog, seven selected airport applications were evaluated. As a result, 10 main features implemented by nearly every tested application were determined. Additionally, use case scenarios were developed based on the main features. These scenarios were used to analyze the airport applications a second time. The objective of this second analysis was to assess the usability through defined heuristics. Results were used to develop a prototype that avoids the detected weaknesses. The prototype development followed a user-centered design process. First, potential users and their needs were identified. The resulting user groups contributed significantly to the design process of an appropriate user interface. User tests of the developed design were a central element of this process model.

The evaluation of existing airport applications revealed serious design weaknesses. Even the first draft of the developed prototype achieved better results than most tested applications. As part of this thesis, a process model is introduced, which presents concrete techniques to create new airport applications. The criteria catalog and the process model have been designed for the field of airport applications. The applicability of the presented process modell in different areas of mobile application development is discussed. Finally, relevant airport application design guidelines were presented as a guidance for future airport application development teams.

Keywords

Airport, Mobile Application, Usability, Process Model, Design Guidelines

Contents

List of Figures

List of Tables

List of Abbreviations

AMS	Amsterdam Airport Schiphol. ix, 4, 30, 48–50, 60, 62, 63, 66, 72, 76, 97, 105, 106, 129
ASP	Active Server Pages. 7
CDG	Aéroport Paris-Charles de Gaulle. vii, ix, 4, 30, 44–47, 61, 63, 71, 76, 97, 98, 122–128
CGI	Common Gateway Interface. 7
CRM	Customer Relationship Management. 8
FRA	Frankfurt Airport. ix, 4, 30, 51–53, 61–63, 66, 72, 97, 106, 135, 139
GB	Gigabyte. 89
GSM	Global System for Mobile Communications. 6
GUI	Graphical User Interface. 12, 63, 74, 76, 104
HCI	Human Computer Interaction. 102
HTML	Hypertext Markup Language. 3, 7, 71, 108
HTTP	Hypertext Transfer Protocol. 7
IA	Information Architecture. vi–ix, 11, 19, 74–82, 84–88, 90–96, 101, 107, 152, 155
IS0	International Organization for Standardization. 10, 11, 19, 22, 23
LHR	London Heathrow Airport. vii, ix, 4, 30, 38–40, 60, 61, 63, 71, 75, 97, 106, 112, 114, 115
OS	Operating System. 1, 5–7, 11, 98, 104
PAX	Persons Approximately. 38, 41, 44, 48, 51, 54, 57
PHP	PHP: Hypertext Preprocessor. 7
PPI	Pixel per inch. 89
PRM	Passengers with Reduced Mobility. 31, 40, 104, 111
RP	Retrospective Probing. 91
RQ	Research Question. 105, 106
SIN	Singapore Changi Airport. ix, 4, 30, 41–43, 60, 61, 63, 71, 75, 97, 117, 118, 120, 121
SUS	System Usability Scale. v, vi, 22, 28, 29, 87, 90–92, 94, 96, 102, 107, 158

TOT Time on Task. 30, 87, 91, 94, 96, 97, 104

UCD User Centered Design. 26, 64
UI User Interface. iv, vi, ix, 3–5, 7, 9–20, 22, 26, 36, 64, 71, 74, 82–86, 99–102, 104, 106, 108, 109
UX User Experience. v, 11, 18–23, 27, 99–102, 104, 106
UXD User Experience Design. 5, 22, 23, 64

VIE Vienna International Airport. vii, ix, 4, 30, 54–57, 60, 62, 63, 72, 97, 106, 140

ZRH Zurich Airport. ix, 4, 30, 57–60, 62, 63, 66, 72, 75, 97, 98, 106, 150, 151

1 Introduction

1.1 Problem Description

At large airports it is often a problem for people to get the information they need on time. Most airports try to tackle this problem by using signposts and departure/arrival boards. However, this assistance is often not enough. Travelers waste a lot of their time searching for the right way to their gate or waiting for information. Other assistance, such as information regarding where lounges, bars or restaurants are located, or even about special offers or discounts, would also be helpful to people using the airport. There are a variety of possibilities that could improve this situation. Some of them would be easy to implement, particularly due to the technical developments of recent years. Smartphones and tablets, which are carried by most travelers today, would be ideal platforms for the implementation of an information and orientation system for the everyday traveler.

Even some of the largest airports in the world do not have their own mobile applications, and those that have applications available provide only a limited range of functionality. Furthermore, application usability seems to be almost completely neglected. Most applications for air travelers offer special tracking functions and information regarding flights. Some also provide additional information for specific airports, such as shopping suggestions and airport maps. Only a few exceptions offer a navigation function.
As mentioned, a mobile application could be a potential improvement for travelers. However, several aspects must be considered during the development of such an application. Important issues include the variety of devices and their respective Operating Systems (OSs), different display sizes and varying computational power.

This thesis presents ideas and approaches for an application that could optimize travelers' time at airports. It attempts to demonstrate how such an application could solve the problems previously laid out. The focus is on concept development and the process model, rather than concrete implementation. An adequate concept requires a detailed analysis. Because of this, a criteria catalog for airport applications is elaborated.
While laying out the concept, several aspects are considered. One important issue is usability. A good user experience is crucial for successful applications; thus, only well-designed and user-comprehensible applications with appropriate functionality are used [73]. It is important to develop a system that meets the user's needs. Designing such an application is complex. Today, many mobile design guidelines exist that have emerged from desktop application guidelines. However, they are constantly evolving. Furthermore, constant technical advancements in the mobile sector offer new possibilities, which lead to necessary adjustments in mobile design. The use of mobile applications is very different from the use of desktop applications [83]. Usability does not only cover the look and feel of an application; it also includes the way in which an application notifies its users about current events. The information that should be sent via push notifications and that is requested by the user needs to be defined. As an airport generates a significant portion of its profit through its shops, it also must be determined how much advertisement could be integrated into such an application before the user gets irritated.

In addition to usability, performance is an important aspect as well. The challenge at this point is that there are so many different devices, and the application should perform well on all of them. Moreover, different platforms and display sizes need to be considered. Depending on the application's functionality, security aspects could also become an issue, especially with respect to personal or confidential data. For an application that primarily supports travelers at airports, appropriate offline functionality should be defined, as many travelers will not have the Internet connection necessary to use an online application. [62]

1.2 Motivation

In recent years, development, especially in the field of mobile devices and applications, has progressed immensely. Mobile devices in the form of smartphones and tablets are becoming increasingly ubiquitous in industrialized nations. Thus, it is required that websites, for example, react to this progress and adapt their content to the mobile market [83]. This is also necessary for areas where the information flow is important; one example of this is airports. Flight data or additional information regarding particular airport facilities are crucial for travelers. As mentioned previously, smartphones and mobile applications are excellent platforms for this task.
Meanwhile, some airports already provide a passenger application, but they do not base their applications on scientific research. This thesis is one of the first papers to develop a passenger airport application concept based on scientific research. Many papers to date cover application design and usability in general, but no papers or scientific work that cover these topics with respect to airport applications have been found (see 1.4).
This thesis presents a process model tailored to the needs of airport applications. It focuses on application usability and design regarding the specific requirements for passenger applications. The benefit of such an application for both passengers and airport operators is shown. Especially for first-time fliers, such an application can help to decrease stress and fear. In addition, this application should facilitate the journey for frequent fliers.
Moreover, such an application could generate more revenue for an airport operator because more travelers feel comfortable when using it and may fly more regularly. Supplementary to this, the application would allow shops and bars to advertise special offers, which could also increase turnover.

1.3 Goal

The thesis explores the extent to which recent technical advancements could be used to support travelers at airports, in addition to the ways they are currently being used. Furthermore, appropriate new approaches to support travelers at airports are discussed. On the one hand, features which primary aid travelers are discussed, while, on the other hand, in such a thesis, the interests of the airport operators should not be neglected. Of course, features that could potentially generate revenue for an airport are also considered. The objective is to discover whether an application for smartphones and tablets is a good solution in this area.

This thesis presents a process model to create a new airport application and applies it for the implementation of APPenger.
It is therefore important to determine who the potential users are and to understand the market. This means an analysis of existing airport applications is necessary. In addition, a criteria catalog

needs to be drawn up, which lists important properties and features of such an application.
This thesis illustrates that many analyzed airport applications use similar layouts and patterns. The analysis also determines that certain features exist that can be seen in nearly every tested application.

Research questions to be answered throughout the thesis are:

- RQ1: Which criteria are important for an airport app? / What are appropriate approaches, and in which way could an app support travelers?
- RQ2: How do existing airport apps fulfill the criteria catalog?
- RQ3: What would an app that fulfills the criteria catalog look like?

The practical part of this master's thesis is divided into two phases. The first phase contains an analysis of existing products on the market. For that, various applications from some of the largest airports and some smaller airports are compared. The largest airports are defined by their 2012 passenger volume [102]. A criteria catalog is developed, which lists important features and properties of an airport application. Both layout and design, as well as functionality and the corresponding workflow, are compared.

An Apple iPod touch 3G is used as a test device. The analysis is based on ISO/IEC 25010 standard characteristics for quality in use. Thus, the criteria of usability (i.e. effectiveness, efficiency and satisfaction), flexibility and safety are considered [56].

In the second phase, an airport application concept is developed. It is based on the analysis findings and avoids detected weaknesses. As part of the concept, a prototype called APPenger is developed, which illustrates meaningful scenarios, contains important features and presents some of the discussed approaches. It is possible to play through defined use cases via HTML mockups.

Furthermore, as part of the development, a user group analysis is necessary. This is important because of the significant effects of the design. The layout of an application is crucial for the success of a mobile application. For this reason, the design is based on various scientific works and papers to improve the user experience. In addition to this, the User Interface (UI) design development utilizes a user-centered process model [106]. That means it is an iterative process, and after each iteration, a user test is performed. For the prototype, two different UI approaches are designed and potential users take part in an A/B test [22] to compare them.
After the best prototype design is found, it is compared against the airport applications from the market analysis. This shows that the presented process model leads to appropriate usability and improvements.

1.4 State of the Art

To determine the current state of the art, official iOS mobile applications of the 30 largest airports were sought out; this was restricted to iOS applications because the testing device was an Apple iPod Touch. Just 16 airports offer an application directed at passengers. Two of these are available on the App Store in their corresponding country (Denver International Airport and Suvarnabhumi Airport – Bangkok). Another 11 airports provide a mobile version of their homepage, but one of them does not display arrival or departure information. The other three airports provide no mobile

version of their website.

Thus, 14 airports offer an application for iOS users. The following seven airport applications are used for the analysis: London Heathrow Airport (LHR), Aéroport Paris-Charles de Gaulle (CDG), Frankfurt Airport (FRA), Singapore Changi Airport (SIN), Amsterdam Airport Schiphol (AMS), Vienna International Airport (VIE) and Zurich Airport (ZRH).

During the development of APPenger, particular attention was paid to usability. Many papers have focused on the topic of mobile applications and their usability. Kascak, Rébola, and Sanford 2014 in [61] demonstrated the integration of universal design principles in mobile design.

Some papers have tried to define usability and identify important characteristics, e.g. Moumane and Idri 2017 in [77]], while others have demonstrated the potential of mobile applications and the Internet of things; see Alnahdi and Liu 2017 in [9]. Furthermore, many studies in recent years have been concerned with the design of mobile UIs. Dumas, Solórzano, and Signer 2013 in [26] as well as Alkhafaji et al. 2017 in [7], wrote design guidelines, while others have described good UI design for mobile applications [14, 73, 12, 48]. Ligman et al. 2016 in [65] presented a technique to automatically validate the UI design of a mobile application.

Moreover, some papers have discussed the development process and the architectural side of mobile applications [21, 62]. Huang et al. 2017 in [47] illustrated a software design in which users could delete unused features at runtime.

Nielsen and Budiu 2013 in [83] argue that usability on mobile devices is restricted due to the smaller display size and other device limitations. They explain techniques that could help in this context, focusing particularly on iPhone and iPad development and design guidelines. Haaksma, Menno, and Karreman 2018 in [34] attempt to define usability from another point of view. They determine the concepts of usability and user experience from the user's perspective and do not restrict their study to software; rather, they also consider devices of daily life.

2 Theory

The first part of this chapter examines the fundamentals of mobile applications. It presents an overview of the most important mobile OSs, measured at the market share, and discusses the advantages and disadvantages of different application types. The second part describes theory regarding UI design and actions. It contains principles of form and function and presents the concepts of usability and user experience. Section 2.2 defines the term 'design' more precisely. It becomes clear that there are many definitions for this term, as well as also some process models to develop a well-designed application. It is shown that design is essentially a graphical representation of natural language with the objective to explain the usage and satisfy the users. Over the years, design patterns for UI elements and workflows have arisen. These patterns support developers while implementing applications to archive a certain degree of usability, but they do not guarantee a good design.
The last part of this chapter describes a more comprehensive concept, the User Experience Design (UXD). It starts before the user purchases the software, covers the installation, usage, support, and does not end with the uninstallation of an application, but rather includes the memories on the product. The section illustrates the relation between UI design and UXD and that both aspects are important to user satisfaction.

2.1 Mobile Application

This section begins with a short introduction of the most frequently used mobile OSs. The focus is on Apple's iOS and Google's Android. Together, these two operating systems had a market share of 94% in 2013 [29]. Other mobile OSs exist, including Windows Phone and BlackBerry OS, but their market share is very low; thus, they are not included in this examination.
The second part of this section describes different application types: native, web and hybrid. The choice of application type decisively affects the development process, as well as the look, feel and performance of an application. Section 2.1.2 presents the technical differences and describes the advantages and disadvantages of each type.

2.1.1 Mobile Operating Systems

Google Android

Android was developed by Google Inc. and launched on the 21st of October, 2008. The current version is Android 8.1 (as of May 2018). The first mobile device with Android was HTC's Dream smartphone. Since January 2010, Google has provided its own mobile devices with the Nexus series.
Android has a flexible design; therefore, many smartphone manufacturers use it on their devices, but install their own themes on top of it. Android is based on a Linux kernel. The kernel manages the memory and offers a hardware abstraction layer. Android also provides a Java virtual machine (VM), but this VM differs from those on desktop computers by the underlying virtual processor architecture. On Android devices, a wide range of applications are preinstalled; it contains a web browser, email client, media player, the Play Store client and much more. Most applications

for Android are written in Java. Google offers its 'Play Store' as a marketplace for Android applications. Registered developers can publish their applications via the Play Store, but Android users can also install applications from elsewhere. [53]
Android has the advantage that, in principle, everyone could develop mobile applications for this platform; no special hardware is necessary, and the required development tools are free of charge. The major part of the Android platform is published under the Apache license, so it is available to everyone for free. Exceptions are the Linux kernel and some preinstalled Google applications. In the second quarter of 2014, Android had a market share of around 85% [29].
Due to the openness of Android and the comfortable possibilities to distribute applications, Google was often criticized for the relative ease of spreading malware over the Play Store. Since the beginning of 2012, however, all applications have been checked for malware before publication, so Android applications are much safer now. Google's potential to control smartphones is also often criticized. Even after delivery, Google can access the mobile phone via remote control, and thus transfer private data.

Apple iOS

iOS was developed by Apple Inc. and launched on the 9th of January, 2007. It was presented in the MacWorld conference, together with the first iPhone version. The first iOS version had hardly any more functions than a conventional GSM phone of that time. The current version is 11.4 (as of May 2018). In March 2008, Apple released the software developer kit for iPhone OS 2.0, concurrent with the App Store (the marketplace of iOS applications). Until version 4.2.1, the OSs for iPads and iPhones were separated. In iPhone OS 3.0, push notifications were introduced. With iOS version 7.0, UI design was revolutionized. [107]
One of the main distinctions between iOS and Android is that iOS is exclusively available on Apple's mobile devices. In addition, Apple provides a wide range of preinstalled software on its devices, e.g. web browser, email client and calendar. All applications run in a so-called 'sandbox' and can only read and write local files within their sandbox. So, direct access to the file system or the command line is not allowed. Like Android, iOS is based on a Unix kernel. [52]
Like Google, Apple checks all applications for malware before they are published. End users can only download applications via the App Store. This centralized process around the publication and retrieving of applications often draws skepticism because of the restricted methods to get software. The non-admission of applications implies censorship to critics. [72]

2.1.2 Application Types

As previously mentioned, every mobile OS manufacturer provides a marketplace for corresponding applications. They facilitate the publishing of applications for software developers, and it becomes easy for users to find, install, update or delete such third-party software.
From a technical viewpoint, these mobile applications can be distinguished, in principle, into two types, as well as some mixed forms. One is *native applications*, which are programmed in the native programming language of the OS. They are executed on the device. The second is *mobile web applications*, which are executed on a web server or in a browser, rather than directly on the mobile device. [55]

Native Applications

Native applications are developed specifically for a particular platform and can typically only be executed on this system. It is usually not possible to install native applications on a different

OS. They offer an OS-specific look and feel and can use the device's hardware, like camera or sensors. Both Apple and Google provide software developer kits, which support programmers developing new applications. Applications written for iOS are developed in Objective-C or Swift, while Android applications are developed in Java and, for some specific parts, C/C++ is used. One advantage of this kind of application is that, depending on the functionality, they do not typically need an active Internet connection. [28] Because these applications are optimized for a particular OS, they are more suitable for complex or computationally intensive applications. Furthermore, after such an application is installed on a device, it is easy for users to find it again and use it. [112]

Mobile Web Applications

Unlike native applications, web applications are executed on a web server and are typically displayed in a web browser. For such an application, an active Internet or Intranet connection is necessary. The advantage of such applications is that there is no restriction to a specific OS. Thus, they are, in principle, executable on every platform, though a special runtime environment is required.
In principle the operation of all web applications is equivalent. Each interaction of the user with the application triggers a Hypertext Transfer Protocol (HTTP) request. The web server accepts this request and forwards it to the program. Thereafter, the program generates or loads the Hypertext Markup Language (HTML) source code of a website and returns it to the browser of the user (HTTP response). A web application does not necessarily presuppose the use of a browser. Other programs can also transmit requests, but these programs must handle the response of the web server, too.

There are essentially two web application architectures: standalone applications and integrated ones. Standalone applications are independent binaries or scripts that are interpreted by an independent binary. This binary needs to be started anew for each request. This kind of application is also known as a Common Gateway Interface (CGI) program. [33]
In contrast to this, integrated web applications are part of the web server or a script that is interpreted by the web server. It is not necessary to start it for each request cycle. Examples include PHP, Perl, Python, Java Servlets, JavaServer Pages and Active Server Pages (ASP).NET.

Figure 2.1: Native Applications vs. HTML5 Applications [36]

Mobile web applications should behave identical to native applications so that users do not perceive them as websites. Web applications should provide a UI that is integrated optically and ergonomically into the mobile device.
Advantages of web applications include platform independence and low maintenance. If the logic has to be changed, the changes need only be deployed on one central point. Additionally, web applications have some security advantages because vulnerabilities can be fixed directly. Moreover, if a web application is compromised, usually no application on the user system is endangered. Web applications do not need to be installed and can be used directly via a web browser. [112]
However, for users, it is more complex to locate the respective application again. Furthermore, additional charges may accrue because of the necessary Internet connection. Regarding security, web applications have weaknesses that native applications do not. Because web applications require a connection to the web server, this can lead to vulnerabilities to, e.g., man-in-the-middle attacks.

In addition, web applications should perform well for all web browsers, but, unfortunately, display problems frequently occur, especially due to the varying interpretations of JavaScript. Hence, browser switches often need to be used, sometimes even for different browser versions. [83]

Hybrid Applications

Hybrid applications are a mixed form of native and web applications. The idea is to combine the advantages of both types. An essential benefit is that, in principle, one implementation can be used for all mobile platforms. Therefore, no parallel software development for the different ecosystems is necessary. Because of the combination of native and web technologies, hybrid applications can address various system functions and can use the device's hardware. However, hybrid applications partially use the same technology as web applications, and therefore must interact with the interlayer of the browser. This results in performance problems for computationally intensive functions or applications, e.g. complex games. Due to the platform-independent implementation of hybrid applications, both the platform-specific look and feel and corresponding interaction design cannot be easily integrated into such an application. [28]
Hybrid application solutions are typically used in the fields of mobile business, mobile marketing and mobile commerce. This application technology is also used for Customer Relationship Management (CRM) systems. The combination of in-house CRM system and hybrid application is a beneficial solution for mobile data exchange between the employees and the management. Data can be synchronized via the CRM system or a separate calendar.

2.2 (Mobile) UI Design

Before discussing mobile UI design, the term 'design' must be generally defined. The UI is the first contact users have with an application (apart from the installation). It is the part of the application that directly communicates with the user. Unfortunately, this part of a software project is often neglected. Nielsen in [82] demonstrated that, among 863 projects, usability costs of the project's budget were between 8% and 13%; this included development, implementation and testing.
In *The Design of Everyday Things* [86], Norman points out that, in some situations, costs dominate. He states that "in businesses, purchasing departments make decisions for large companies" [86, p. 241]. He further describes that "the purchaser is probably interested primarily in price, perhaps in size or appearance, almost certainly not in usability" [86, p. 241]. He emphasizes that the needs of eventual users are important, but to the business, they seem to be neglected. In these cases it is required to design an UI not only for product users, but also for other groups involved in product lifecycle, like the purchasers.
This differs from software for private customers, where the UI or product design is more important. Customers can freely decide which software they want to buy and use. Even more important is the design for websites and mobile applications. The webpage *internetlivestats.com*, which counts the current number of registered websites, reports that there are approximately 1.8 billion (as of May 2018). This number reveals that there is a very wide range of offers for almost every use case and that just the UI design will decide if a user stays on the website or not.
In the first quarter of 2018, Google's Play Store listed 3,800,000 applications, while Apple's App Store listed 2,000,000 [87]. Using application stores makes it is easy for users to search for, download and install applications on their mobile phones. If they do not like an application, deleting it and installing a new one is simple. This is why UI design and usability are so important: to win users over for continuous use.
Now the question arises, what is a well-designed UI and how can one achieve it? In this chapter,

the term *design* is defined more precisely. Concepts and guidelines to develop a well-designed UI are presented.

Design concerns everyone; all artificial things are designed. Design is a term that has existed for a long time; thus, there are many definitions. They differ, in some cases significantly, depending on the field of use. Furthermore, many scientific papers and books regarding the topic of *design* have been published.

The Oxford dictionary lists five definitions for design, two of which are presented below:

> "The general arrangement of the different parts of sth. that is made, such as a building, book, machine etc." [104, p. 340]

> "In the field of Drawing/Plan/Model: "the art or process of deciding how sth. will look, work, etc. by drawing plans, making models, etc." [104, p. 340]

> "Industrial Design is the professional service of creating products and systems that optimize function, value and appearance for the mutual benefit of user and manufacturer." [10]

According to these definitions, design is, in general, a kind of art, a craft or a professional service. Thus, it is something that could be learned to a certain degree, and its result could finally be sold. Additionally, design is described as a process, so it is something that evolves and changes.

Considering now the topic of this thesis, design must be defined in the context of UI development. The following are some definitions for UI design:

> "User Interface Design is a process of visually guiding the user through a product's interface via interactive elements and across all sizes/platforms. (...) User Interface Design is the look and feel, the presentation and interactivity of a product." [64]

> "User Interface (UI) Design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions. UI brings together concepts from interaction design, visual design, and information architecture." [42]

> "User interface design isn't a subjective visual art about pixels and aesthetics but rather a principled objective communication skill to explain tasks to users" [74, p. 3]

Given just these few definitions of UI design, one may quickly recognize that it is user-centered, and not centered on the software itself. User interface design provides guidance for users, anticipates user needs, and communicates with the user. In other words, the UI is the result of the design process, which has the objective to graphically reveal to the user as clearly and intuitively as possible the functionality of a software and thus lead him or her through the application.

There are many process models and rules for developing UIs. The following guidelines describe iterative processes and consider the user in the design process. In 2010, the International Organization for Standardization (IS0) defined in the *Ergonomics of human-system interaction* (ISO

9241) standard a part for *Human-centered design for interactive systems* (Part 210) [59]. This contains the following process model:

1. The design is based upon an explicit understanding of users, tasks and environments.
2. Users are involved throughout design and development.
3. The design is driven and refined by user-centered evaluation.
4. The process is iterative.
5. The design addresses the whole user experience.
6. The design team includes multidisciplinary skills and perspectives.

It starts by understanding tasks, environments and users' needs. Furthermore, it describes the process as iterative, so the user is involved. A peculiarity of the IS0 standard is that it is directed to multidisciplinary skills and perspectives within the design team. Therefore, different opinions and views can be discussed among the team. In 2014, Allanwood and Beare presented in their book *User Experience Design* [8, p. 131] a process model, which was inspired by the IS0 standard:

1. Plan the human centered design process
2. Understand and specify the context of use
3. Specify the user requirements
4. Produce design solutions to meet users requirements
5. Evaluate the designs against requirements
6. Iterate where appropriate
7. Finish if the designed solution meets the requirements

This model focuses on users and describes an iterative process as well. It is more specific than the IS0 standard because it defines concrete tasks. It is striking that all presented guidelines contain parts for usability and User Experience (UX). Therefore, it is obvious that both usability and UX design are important for the product's quality. Nevertheless, Haaksma, Menno, and Karreman in [34] illustrate that the actual relation between them is unclear in the literature. Section 2.3 covers the topic of UX design and describes it in further detail.

McKay, in his book *Ui Is Communication: How to Design Intuitive, User Centered Interfaces by Focusing on Effective Communication* [74], views UI from a different perspective. He compares UI design with natural language and describes it as communication – as a dialog between the user and the application. In the course of this discussion, he defines five principles that should be considered during the development of an UI:

1. UI is Communication
2. Explain tasks clearly and concisely, as you would in person
3. Every UI element can be evaluated by what it communicates and how effectively it does that job

4. Be polite, respectful, and intelligent
5. If a UI feels like a natural, professional, friendly conversation, it is probably a good design

McKay's principles do not substantially diverge from the models presented above. Although he does not describe the engineering process, these principles are also user centered. The satisfaction of users and satisfying users' needs is important; thus, UX is also included in these principles.

No step-by-step instructions for designing great UIs can be found in the literature. Such universal guidelines for the development of software UIs do not seem to exist. However, some approaches, e.g. Marcus, Schieder, and Cantoni in [71] recommend starting with an analysis of the provided information and defining the Information Architecture (IA), a logical structure of the application.

While the steps for beginning and proceeding with UI design have to be defined for each project, many sources exist that describe graphical UI elements. In regard to the topic of this chapter, the following sub-chapter focuses on *mobile* UI design. Both Apple and Google provide interface guidelines for application developers. These guidelines describe, for each OS, the layout and the look and feel of an application, and more precisely of UI elements. They define UI elements that should be used to ensure a uniform behavior of all applications on the specific platform.
The W3C publishes a best practice guide [103] that contains 60 recommendations for the behavior and design of a web application. Likely not entirely coincidental, this resembles the guide and the guidelines of Apple and Google. They define concrete patterns for the UI design. The categorization of patterns is essentially identical for both platforms. User interface patterns evolved for decades and were already used for desktop applications. They were partially adapted for the field of mobile applications and are presented in greater detail in the following section.

2.2.1 Design Patterns

This sub-chapter covers the topic of design patterns. Since Graphical User Interfaces (GUIs) have existed for such a long time, many studies and papers regarding UI design have been published (see e.g. [14, 73, 26, 79]). It has been revealed that some UI elements are used in many different applications and that some are more suitable than others. These findings were used to develop design patterns to support designers and developers in implementing an attractive GUI. Design patterns are continuously advanced and adapted for various devices and areas of application, as are the patterns presented below, which partially find their use only in the mobile sector. Because there is a very wide range of design patterns, the following are just a select few that are applicable to the topic of this thesis.
Apple and Google describe the design patterns in their UI guidelines [49, 54] more specifically. They delineate the concepts and patterns more generally and precisely define the various UI elements.

Navigation Pattern

In her book [79], Neil divides navigation patterns into two classes: persistent and transient. Persistent navigation patterns describe navigation elements that are permanently visible. Transient navigation patterns cover navigation elements that can be moved in and out the display.
Some examples for persistent navigation patterns include springboard, cards, dashboard and tab menu. In [79] *springboard* is defined as a view on which all menu entries appear equally important. A structured hierarchy is difficult to realist; one possibility is to use different font sizes. This

pattern is often used for the application selection on smartphones.
Google defines the *Grid List* UI element for this navigation type [54]. It essentially describes the organization of content in a grid with one or two line titles. The same element is used for the *gallery* navigation. The *gallery* pattern is often used to present live content, such as news stories or photos. The content is typically ordered in a grid. "The Gallery pattern works best for showing frequently updated, highly visual content where no hierarchy is implied" [79, p. 19]. Apple calls this pattern *collection views* [49].
The *card* pattern is based on a card deck metaphor. Thereby, different content is presented via separate cards. Cards provide an elegant way to display content for browsing. Thus, this pattern is often used for mobile browsers. Apple does not use the card metaphor; it is instead called pages, views or modals [49]. The website *ui-patterns.com* [101] recommends using this pattern only if the content could be visually presented and is thereby easy to distinguish.
In [79], Neil presents the *dashboard* pattern as a basic overview page. It is similar to the springboard and displays a snapshot of the most relevant information. Neil recommends using a dashboard when it makes sense to use key metrics or data as launch points to an application. It is crucial not to overload the dashboard. Neither Apple's nor Google's UI guidelines contain the dashboard pattern, but the various UI elements in the guidelines could be used to design a dashboard.
The *tab menu* creates a context for content. It is important to distinguish between tab bar (which contains application items) and toolbar (which contains tools for actions on the current screen) [101]. Tabs separate content into sections using a flat navigation structure. Google recommend using this pattern for applications with few top-level views to quickly switch between them [54].

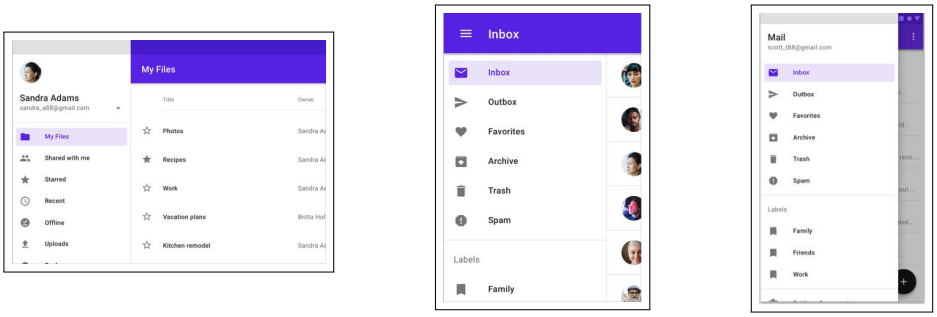


Figure 2.2: Navigation Drawer Styles: Permanent (Left), Dismissible (Middle) and Modal (Right) [54]

The *side drawer* pattern describes a transient navigation menu that can be swiped into and out of the visible area of the display. Google defines in its guidelines [54] three different types of navigation drawers: permanent, dismissible and modal (see Figure 2.2). The permanent navigation drawer is always visible and is only recommended for desktop applications. The dismissible navigation drawer can open or close the menu. This drawer is designed as an inlay style, on which the content will be moved and the menu is shown beside the content. The modal navigation drawer can also toggle the visibility of the menu but is designed as an overlay; thus, the menu is shown in front of the content. This type is recommended for tablets and required for mobile devices. Apple does not provide a definition for a side drawer menu, and instead uses *temporary views* which are modal views designed as an overlay or *popover*.
The *toggle menus* pattern describes a very similar menu style – the menu can be opened and closed, and both overlay and inlay style can be used, although overlay is more common. The main difference from a side drawer is that the toggle menu is activated by a separate button and will not be swiped into the display [79].

Forms

Another category of patterns are forms, which are views for special tasks. Neil presents in *Mobile Design Pattern Gallery: UI Patterns for Smartphone Apps* [79] patterns for sign-in, registration, multi-step, checkout, search forms and long forms.
Sign-in or log-in forms should require a minimal number of inputs: username, password, command button, password help and an option to register. Neil does not recommend getting creative with the sign-in screen, but rather suggests using standard designs to make it easy for users to log in.
Toxbue [101] recommends keeping *registration* forms short, preferably one screen with the command button in plain sight, to avoid frustrating users. In addition, one should clearly state field entry requirements and use inline feedback to speed up the process and let the user know what is going on. Registration should be thought of as a flow.
The *multi-step* pattern describes how to guide mobile users through a series of steps; it shows users where they are and where they can go. Google provides a special UI component for this pattern: *steppers*. "Steppers display progress through a sequence by breaking it up into multiple logical and numbered steps" [54]. Apple does not directly provide a UI element for multi-step views, but there are other elements that could be used for this purpose, e.g. the page control.
In [79], Neil provides five tips for implementing the *checkout* pattern: 1. Include sign in, register and guest options; 2. Streamline the flow; 3. Provide timesaving shortcuts; 4. Offer express checkout; 5. Forget the web.
Some searches require multiple inputs to generate results. Like the other form patterns, *search forms* should have only the essential or most requested fields and provide sensible defaults. Search forms should be kept short and simple. Google and Apple provide search bars for simple searches [49, 54]. The *long form* pattern describes the need to view a subset of data that is not easily displayed on a single page. Toxbue [101] recommends using this pattern when navigating to a second page of data takes too much attention away from the content.

Tables

Neil also presents in her book [79] some patterns for tables. Perhaps the best-known pattern for tables is the *basic table*. It builds a common representation for a large amount of data in many applications. Google describes it as follows: "A data table contains a header row at the top that lists column names, followed by rows for data" [54]. In the basic table pattern, the column width is variable; in contrast, in the *fixed column* pattern, the column width cannot be changed. Apple uses the term *table* in its guidelines for a single-column list of multiple rows presenting data [49]. A classical table UI element is not provided; one would need to use a collection view instead.
There are some extensions of the basic table pattern, such as *overview plus data*, *grouped rows* and *editable table*. These patterns expand the basic table with some additional functions, like the direct editing of cell content. Tables are a classic form to present data. Neil [79] hints that it is always a challenge to find viable ways to display data on small screens, and it is important to determine what data the user really needs.

Search Patterns

Google defines two search patterns: *persistent search* and *expandable search* (see Figure 2.3). Persistent search displays the input field inside of an inset search box. The user can touch the microphone icon to initiate a voice search or can click the input field to enter the desired term. When the field is in focus, it expands to show historical search suggestions, and the onscreen

keyboard will appear. In contrast, expandable search shows only a magnifying glass icon in the toolbar. Clicking the icon will open the search text field with the microphone icon [54].

Apple defines similar search designs, but also provides a *Scope Bar* UI element (see Figure 2.3). A scope bar is only available in conjunction with a search bar and helps users refine the scope of a search. It is a graphical element (a bar) used to filter search results [49]. Neil [79] also presents the search result pattern, which describes various possibilities to present search results. Results can either be displayed on the same screen or on a dedicated result view. "Results may be displayed in a table or list, on a map or satellite image, or as thumbnails" [79, p. 95]. Furthermore, results can be displayed so that each dataset is shown on a card, slide or tile. Neil recommends, in [79], labeling results with the number of returning items, using lazy loading instead of paging, and applying a reasonable default sort order.

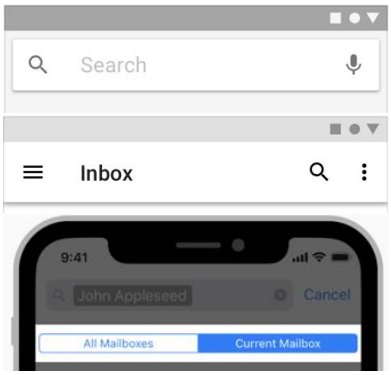


Figure 2.3: Search Patterns: Persistent Search (Top), Expandable Search (Middle), Scope Bar (Bottom)

Sort Patterns

Ordering date is often used for tables and lists. Google defines the *sorted column* pattern to order data in tables. Users can click on a column header to activate the sort function. This column will sort the data and a sort state icon appears. [54]
Neil presents, in her book [79] , some sort patterns. The *onscreen sort* pattern can provide a simple one-tap solution. As the name suggests, the sort options are shown permanently and directly on the screen. In contrast, the *sort overlay* pattern defines a separate overlay for the sort options. Another pattern is the *sort form*. In this pattern, sort options are presented on a separate screen.

Filter Patterns

A large set of data can be so unmanageable as to be almost useless. Users often need to filter or refine the results to make sense of the data. Neil states that "Filters let users select criteria to reduce data sets to their most manageable, relevant results" [79, p. 104].
The previously mentioned sort patterns, *onscreen*, *overlay filter* and *filter form*, work for filtering as well. Apple provides a special scope bar to filter search results (see Figure 2.3). In addition to these, Neil [79] has introduced the *gesture-based filter* pattern. The results of this filter dependent on a location, such as a map, graph or chart.

Tools

The first tool pattern presented here is the *toolbar*. A toolbar looks similar to a navigation bar or a tab bar, but it does not enable navigation possibilities. Instead, a toolbar provides user controls that act on the current screen's content. Apple defines in its guidelines [49] that a toolbar appears at the bottom edge of a screen. Google, in [54], less strictly defines the location of a toolbar, but the style and transitions of a toolbar are defined when the sheet is moved. Neil [79] recommends following the OS specifications for toolbar placement and behaviors. It is advisable to choose icons that are familiar to the user and that are easy to recognize.
The *inline actions* pattern refers to action buttons that are in line with the object they affect, as opposed to being at screen level like the toolbar. Normally, one should not assign multiple actions

to a UI element – it can overload the screen and the user. The *multi-state* button is an exception. This type of button acts as both an action trigger and a feedback mechanism. Both Apple and Google provide several button types for this purpose, such as switch, toggle and customizable buttons [49, 54].

Invitations Pattern

Invitations patterns should facilitate working with the application. They describe different approaches to clarify the usage of UI elements. Whether or not an application requires a tutorial for onboarding, there are likely opportunities throughout the application inviting users to engage with a specific feature. Invitations are also helpful to introduce new features within a specific flow, as opposed to just displaying a dialog on launch with a list of new features.
A *tip* can be implemented anywhere in the application, making it more contextually relevant than a dialog. It is often used to draw attention to features that could be easily overlooked. Normally, tips are shown for a short time and typically only upon first use of the application. Google provides several UI elements for this: *tooltip*, *chip*, *snackbar* and *toast*. Aside from tooltips and toasts, all of these elements can contain action buttons [54]. Apple also provides the *alert view*, which describes a modal dialog with at least one button to confirm the message [49].

Feedback Patterns

For applications, it is important to provide user feedback. This is especially true for actions that are computationally intensive and therefore need some time, so that the user does not assume the application has crashed. For this purpose, the *system status* pattern was developed.

Figure 2.4: Apple Feedback UI Elements: Refresh Content Control/Activity Indicator (Left), Progress Bar (Middle) and Network Activity Indicator (Right) [49]

Regarding progress display, Apple distinguishes between four UI elements: *progress bar*, *refresh content control*, *network activity indicator* and *activity indicator* (see Figure 2.4). The activity indicator can be used in a toolbar or a main view. It blocks further user interactions and does not predict when the activity will finish. Apple states in its UI guidelines that "a network activity indicator spins in the status bar at the top of the screen as networking occurs" [49]. The progress bar, in contrast, is used to display the progress of a task or process that has a known duration. The refresh control performs a user-triggered content refresh and looks similar to an activity indicator. [49]
Google defines just a progress indicator UI element. It can occur linearly or circularly and each representation can be determinate or indeterminate [54]. The determinate progress indicator presents the concrete progress of a task or process, whereas the indeterminate indicator just signals that a process is running.
The system should also provide user feedback when exceptions occur. Error messages are often used for this (*error message* pattern). Additionally, Neil [79] recommends presenting success feedback and introducing the *confirmation* pattern. This provides users with feedback when an operation is successful. This feedback does not necessarily occur via text; e.g. on iOS, the successful sending of an e-mail is signaled acoustically.

Affordance Patterns

In 1988, Norman introduced the term 'affordance' in the field of interaction design in the first edition of his book *The Design of Everyday Things* [86]. He described it as a relation between the properties of an object and the ability of a user to recognize how the object can be used. So, affordance patterns deal with the design of UI elements to signal the user if and in which way the UI element can be used. Neil [79] defines affordance as "the quality of an object that lets a user know that the object can be used to perform an action" [79, p. 298]. In the field of mobile application tabs, swipe/flick and drag are often used. These navigation elements allow a clear and structured application.

Anti-Patterns

In addition to the design pattern, anti-patterns have also emerged over the years. They have proved to be impractical or eventually lead to a bad user experience. Neil, in her book [79], presents some examples of anti-patterns: *Novel Notions (novel UI elements)*, *Needless Complexity*, *Metaphor Mismatch*, *Idiot Boxes*, *Chart Junk* and *Oceans of Buttons*. In addition, Toxbue presents on *ui-patterns.com* [101] several anti-patterns, pointing out that anti-patterns sneak in very quickly and designs must take care of them. He recommends user tests to avoid bad designs and anti-patterns.

Meanwhile, some of the patterns previously presented are so natural that they are often not perceived as patterns. They are common UI elements and intuitively used by developers and designers. The main advantage of design patterns is that, in general, users can quickly orient in new mobile applications and use them in a well-known manner. Design patterns do not automatically guarantee a good UI, but they help to reach a certain quality standard.

2.3 Usability and User Experience (UX)

This chapter describes the concepts of usability and UX and discusses the differences between both, as well as approaches to achieve high usability and good UX. In the previous chapter, design elements and patterns for the development of UIs were presented. These are the basic tools for great usability.
Usability covers more graphical aspects and workflows, while UX embraces the understanding of users' needs and goals, as well as user satisfaction [34]. As mentioned in the previous chapter, determining users and identifying their needs and goals are the first steps in the development of a new application. In the course of this development, besides users' goals, the environment and the manner in which an application is used are also important. In his study [45], Hoober discovered that 75% of all smartphone users operate their devices with only a thumb. Nielsen and Budiu [83] revealed that mobile applications are used sporadically and that there are essentially only two scenarios for using: either users are in a hurry and quickly need information, or they use the application as a pastime, e.g. to bridge waiting.
All of these aspects influence UX and must be considered during the development. Nielsen and Norman formulate it as follows:

> "User experience encompasses all aspects of the end-user's interaction with the company, its services, and its products." [84]

Nielsen and Norman describe UX as a comprehensive concept in which users occupy the center stage and usability is considered a part of the product. Thus, UX and usability must be distin-

guished. User experience appears to be an umbrella term for multiple product quality characteristics, and usability is either part of it or contributes to it. The actual relation between both concepts is unclear in the literature. The ISO 9241 standard definition of UX is as follows:

> "A person's perceptions and responses resulting from the use and/or anticipated use of a product, system, or service." – ISO 9241-210 [59]

This definition suggests that UX encompasses a wide range of user-related aspects: emotions, beliefs, preferences and feelings. It starts before using a product for the first time and includes the memories after deletion. User experience is heavily dependent on users' expectations. If users expect a complex software, they will be satisfied with complex workflows. However, depending on the context, a workflow with more clicks could be more intuitive and thus improve customer satisfaction. Nielsen and Budiu advise against immediate advance registration at the first start of an application. This procedure will eventually scare users because it could lead to a feeling of distrust or observation.
Allanwood and Beare, in [8], argue that "The most fundamental human needs are the requirement for food, exercise and safety. If safety is violated, then all other motivating forces vanish" [8, p. 62]. Users must have confidence in the security of their experience; otherwise, they will feel uneasy. It is important that experiences promote a feeling of security and trust, especially if tasks rely on extrinsic motivation. Thereby, Allanwood and Beare advocate the importance of good UX in preserving trust and the perception of credibility among users.
As defined above by Nielsen and Norman, UX encompasses much more than usability. For example, the business model could influence the UX. A workflow redesign by the *Nielsen Norman Group* of the Wall Street Journal application has shown that most negative App Store ratings refer to in-application advertising [83]. Store ratings are critical to the success of an application. Allanwood and Beare [8, p. 51] discovered that 66% of users download an application based on recommendations. This reinforces the assertion that UX starts even before the user opens an application for the first time.
Norman has described this in an interview as follows:

> "User experience is really the whole totality... It's the total experience that matters. And that starts from when you first hear about a product... experience is more based upon memory than reality. If your memory of the product is wonderful, you will excuse all sorts of incidental things." [85]

As mentioned, it is assumed that high usability contributes to good UX. Neil explains that "Alignment, labels, fonts, button placement, contrast, and colors all affect the usability of mobile forms" [79, p. 90]. So, usability mainly refers to UIs; it defines measurable quality criteria.
Still, Richter and Flückiger point out, in *Usability Engineering kompakt* [95], that usability is more than the quality of an UI. Moreover, they state that software applications and products exhibit a high degree of usability if they can be learned easily and used efficiently. Furthermore, it is important that users can achieve the desired goals in a satisfactory manner. At that, it is somewhat difficult to define measurable criteria and metrics for e.g. learnability or the degree of satisfaction. To improve learnability, numerous publications recommend using known UI elements. Apple and Google define the previously mentioned UI guidelines. McKay recommends, in [74], using standard interaction designs and UI elements for the development of new UIs, although the standard interactions do not work for special cases.
The IS0 standard 9241-11 defines usability as follows:

> "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." [58]

This definition describes the concept well but does not specify usability in quantifiable or measurable terms. Some researchers claim that this is what makes usability difficult to define. Allanwood and Beare, in [8, p. 82], describe usability as a measure of how easily users can achieve their goals. It is a subjective measure because levels of usability depend on the user and the context of use. Sometimes, for efficiency, it is not only the numbers of clicks that are crucial, but rather the degree of frustration and the time exposure for users.

Specific to the field of mobile devices, like smartphones, Nielsen and Budiu [83] identified four usability hurdles: small displays; problematic input operations, particularly when typing; delay for downloads and badly designed UIs. In particular, for websites, Nielsen and Budiu recommend reducing the provided features such that only relevant functions remain for mobile usage. This chiefly offers a significantly better usability. Additionally, they advise removing secondary information from the main page, and instead presenting an overview including links to the corresponding subpages. An advantage of this approach is that more visitors will be satisfied and data transfer will be increased; thus, greater business profit can be generated. For a user-friendly information structure, Nielsen and Budiu advise employing a mini-Information Architecture (IA) based on a user-oriented structuring model. The mini-IA describes the structure of information presented for a single topic.
To master small displays, both Apple and Google define fixed sizes for their UI elements. In 2006, a relevant study [91] showed that the minimum size of clickable elements on touch displays should not be less than 1 x 1 cm. Google defines in its guidelines [54] a button height of 48 dpi, which corresponds to a height of 0.76 cm on a Nexus 6P with 515 dpi and a resolution of 2560 x 1440 px [100], it is $^1/_4$ smaller than what the study recommends. Apple defines the size of icons even smaller; on an iPhone 6 Plus, the icons of the toolbar, tab bar and settings are just over 0.5 cm [49]. Both companies appear successful with their guidelines, because both recommendations have existed for several years.

Not only is the element size important to accomplish good usability, but also the perceived affordance (see Section 2.2.1). Affordance describes the use property of an object and corresponds figuratively to the invitation to do something with said object. According to Nielsen and Budiu [83] an UI element has affordance if the gesture is not only possible, but is also recognized by the system and is treated like an input or command. Furthermore, users must realize that they can perform it. To facilitate it for users, all clickable navigation elements should look similar. The same applies to gesture-controlled actions. If an application utilizes new gestures that are not common in other applications, it is highly likely that the user will forget these unique gestures. Therefore, Nielsen and Budiu suggest using a small number of gestures, and preferably those that are familiar to the user. In addition, redundancies should be integrated in a UI to avoid mistakes. Apple and Google define in their guidelines typical gestures for the corresponding platform [49, 54].
If, for a particular reason, it makes sense to incorporate new gestures or extraordinary design elements, it is proper to provide help for users. This could be done through tooltips or complete tutorials.
Nielsen and Budiu [83] warn that extremely exceptional designs will not achieve added value – constant interaction techniques are certainly better suited.
Antoine de Saint-Exupéry claims that "Perfection is achieved not when there is nothing more to add, but when there is nothing left to take away" [8, p. 84]. This statement illustrates that it is important to concentrate on the essentials. To achieve high usability, developers must know the

exact scope of an application to accurately tailor the features and design to the users' needs.
On *usabilitynet.org*, another aspect is highlighted: "usability should not be confused with 'functionality', however, as this is purely concerned with the functions and features of the product and has no bearing on whether users are able to use them or not. Increased functionality does not mean improved usability" [105] .

Based on this introduction, it is clear that good usability and UX are elaborate, and they utilize a noticeable part of the project's budget. As already mentioned, Nielsen revealed in [82] that usability costs ranged from 8% to 13% of the studied projects' budgets. This raises the question: why are usability and UX so important?
Norman presents an example in *The Design of Everyday Things* [86]: typically, in large companies, the selection for a purchase of new machines is based on price and a list of required features. Usability, training costs and maintenance are not typically considered. Ordinarily, there are no requirements regarding understandability or usability of a product. Those aspects can eventually cost the company a significant amount of money through wasted time, increased need for service calls and training. Norman [86] additionally explains, "if people complained strongly enough, usability could become a requirement in the purchasing specification, and that requirement could trickle back to the designers" [86, p. 241].
This example illustrates that usability could appear unimportant at first glance, but is profitable on closer consideration. In contrast to that example, the purchasing process in the market for mobile applications is slightly different. The purchaser is most often also the user. Allanwood and Beare, in [8], point out that spending 10% of the projects budget on UX improve conversion by 83%. Furthermore, they indicate that 66 % of users download an application based on recommendations. This illustrates that it is important to satisfy the user to increase return on investment, and one important aspect for user satisfaction is a good UX. Moreover, not only do the publishers of mobile applications compete against each other, but mobile websites could also offer the needed functionality. "If one can motivate people to install the app and to use it, one has gained a great advantage against mobile websites and other competitors", Nielsen and Budiu [83, p. 61] explain. They also state that it is important to spark users' curiosity during their first experiences with an application to win them over for continuous use. Therefore, it is beneficial to speak the user's language in the sense of context correspondence, and thus know how and where the application is typically used. Other aspects that affect user satisfaction are splash or launch screens. Nielsen and Budiu have observed that most splash screens contain no information regarding the application load time or the progress. Apple and Google both recommend avoiding such screens [49, 54].
Some applications must deal with big data, and they use the splash screen to load them. In this case, it is more user-friendly to consider partial or pre-load strategies. With frequent use, such screens can annoy and frustrate users. Nielsen and Budiu [83] suggest leading the user directly to the characteristic area of the application.

This chapter has presented and discussed the concepts of usability and UX, explained the relation between them and depicted aspects that must be considered. The next chapter presents a user-centered process model and evaluation methods for usability and user experience.

3 Methodology

In the beginning of this chapter, the user-centered design process model is presented. It describes an agile and iterative design process with the objective to improve the UX of the developed product. In the course of this, the eight phases of that process model are considered, starting at concept finding via developing design evaluation and leading to the deployment of the solution.
Subsequently, software evaluation in general is illuminated. Therefore, the ISO/IEC standards 25010 and 25022 [56, 57] that define generally software evaluation are presented in Section 3.2. There are many different methods for software evaluation. The choice of a concrete technique depends heavily on two criteria: the project phase and the desired goal. For example, if the project is in the implementation phase and the code quality should be evaluated, automated unit tests and code analysis tools could be used. If the project is in the design phase of the UI, iterative design cycles including user-testing techniques or heuristic and metric analyses are possible choices.
Sections 3.2.1, 3.2.2 and 3.3.1 introduce concrete evaluation methods and techniques. These methods can be used for evaluating existing software to elaborate requirements and define the functional scope, as well as to evaluate the developed product.
The criteria catalog method, described in Section 3.2.1, provides the ability to define characteristics of a software system depending on the objective of investigation and ends up with a comparable result. The use case technique presented in Section 3.3.1 allows an analysis of real life scenarios, rated by a heuristic, which is described in Section 3.2.2. These two methods offer a systematic evaluation form and supply meaningful and comparable results.
This chapter concludes by presenting the topic of user testing. This is especially important for an iterative design process with the objective of improving the UX. Therefore, two concrete techniques are presented: the System Usability Scale (SUS) and A/B tests. After potential users test a system via defined scenarios, they can rate its usability by the use of SUS. It is a standardized questionnaire to collect data and provide a general statement about the ease of use of a given system.
An A/B test is a method to evaluate the effectiveness of different variants.

3.1 User Centered Design - Process Model

In this section, one possible process is described which is needed to archive effective usability and high UX. The following process is presented in *User Experience Design* [8, p. 108 - 131] by Allanwood and Beare and is based on the process definition in the IS0 standard 9241-210 [59].

In a typical software project, multiple things are happening in parallel and some tasks are repeated until the results are satisfactory. Allanwood and Beare in [8] identify specific tasks that are common to a UXD approach: 1. "The creation of a concept brief that will outline a strategy and include the project scope and objectives" [8, p. 108]; 2. "The bringing together of a multidisciplinary design team" [8, p. 108]; 3. "The identification of users and their needs, including the tasks and environments of the design" [8, p. 108]; 4. "User research, including an understanding of the environment in which users will experience the design and how the design addresses the whole user experience" [8, p. 108]; 5. "Initial design and development of one or more creative approaches"

[8, p. 108]; 6. "User-centered evaluation followed by design selection and refinement" [8, p. 108]; 7. "Further user evaluation until satisfactory evaluation results are achieved" [8, p. 108]; and 8. "Deployment of the design and continuous evaluation in its live state" [8, p. 108].

Figure 3.1 presents a design process life cycle for a UXD approach. The process life cycle starts by understanding and analyzing the context of use and the users. Allanwood and Beare mention that it is important to answering fundamental questions during comprehensive user research, such as: Who will use the product? What is the objective of the user? What is the context of use and what are the surroundings? The design team must interpret the information to elaborate user requirements and develop initial design ideas.

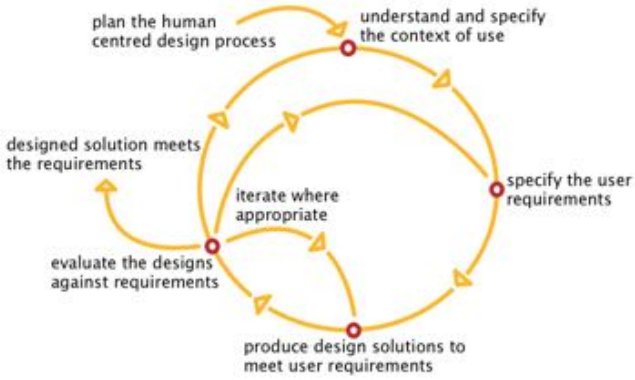


Figure 3.1: Design Process Model [8]

As a next step, it is important to define concrete user groups. With the intention to create a design that 'works for everyone, everywhere', design teams frequently create something that 'doesn't work very well for anyone, anywhere' [8, p. 114]. A better tactic is to focus on accommodating user characteristics that are similar in what is called a *persona*. Allanwood and Beare define a persona as "a fictional description of a model user based on high-quality user research of actual users in the target user group" [8, p. 114]. It is usually necessary to create multiple personas. Scenarios can be used to bring them to life. They feature personas in the role of a user. These scenarios can help a design team view the system from the user's perspective. Allanwood and Beare explain that "scenarios are written in third person narrative style and usually start by placing the persona in specific context with a problem to solve" [8, p. 118].
One significant advantage of a UXD approach is the amount of beneficial information that is available when specifying the design requirements. It brings UX at the forefront of everyone's minds. Allanwood and Beare recommend starting the design process by defining a list of initial design requirements. They can be specified in relation to the user, system and product. [8, p. 121]
Based on the requirements, an initial design is developed. This design is then assessed, the assessment triggers improvements, and the design is assessed once again [8, p. 110]. Figure 3.1 illustrates possible iteration paths in the life cycle. In principle, there are two loops in the design process: the feedback loop and the design iteration loop. Involving users in the design process to receive constructive feedback is an effective way to successfully complete the process [8, p. 111].

This illustrated process is one possible model, but there other process models, as mentioned in Section 2.2. This model is based on an international standard, the IS0 standard 9241-210 [59], and was solidified by Allanwood and Beare in [8, p. 108 - 131]. User experience design can be combined with modern agile process models and defines only a part of the whole project.

3.2 Software Evaluation

One of the first steps in the aforementioned process model is to identify users and their needs. One possible approach is to evaluate similar existing products to determine the state of the art. To

ensure an objective analysis, the evaluation should be based on standards. The ISO/IEC 25010 [56] and ISO/IEC 25022 [57] are both international standards that contain definitions for and measurements of software quality. ISO/IEC 25010 describes characteristics and properties by which the quality of software can be determined and measured. The ISO/IEC 25022 standard contains concrete measurement methods for these properties.
Software quality is divided into three categories: internal software quality, external software quality and quality in use. Internal software quality can mainly be determined during the development phase of a software because it covers properties such as code quality and software structure. External software quality is characterized by the behavior of the system as compared to the requirements and the specified behavior. It also covers properties such as the number of failures detected during the test phase.
ISO 25010 [56] defines quality in use as follows:

> "Quality in use is a measure of the system's quality in a real or simulated operational environment. It is determined by the quality of the software, hardware, operating environment, and the characteristics of the users, tasks and social environment. All these factors contribute to quality in use." [56, p. 10]

In principle there are three types of users/perspectives:

1. End user perspective – Quality in use is mainly a result of functional suitability, reliability, operability and performance efficiency
2. Maintainer perspective – Quality in use is a result of maintainability
3. Perspective of the person who is porting the software – Quality in use is a result of portability

Thus, for the evaluation, the type of quality and from which point of view the quality will be assessed must be defined. The objective of the present research is to evaluate the usability and features of various airport applications. Therefore, the evaluation concentrates on quality in use from the end user's perspective. Figure 3.2 displays the quality characteristics of software products, defined by ISO/IEC 25010. The green and orange marked properties are evaluated in Chapter 4. Those in green are the more functional properties, whereas the orange properties describe non-functional features. A criteria catalog is one possible tool to evaluate the functional scope of an application. What a criteria catalog is and how it could be developed is discussed in Section 3.2.1. The red categories are not evaluated in this study, because these topics mainly cover internal software quality.

In the ISO/IEC 25010, quality in use is divided into the following subcategories: usability in use, flexibility in use and safety. Usability covers the topics effectiveness in use, efficiency in use, satisfaction in use and usability in user compliance [56].
These categories are assessed in Chapter 4. Quality in use can be evaluated by observing representative users performing representative tasks in a realistic context of use. The results can be obtained by observing operational use of the product or by simulating a realistic working environment. The assessment should match the context of use as closely as possible. [57]
This evaluation technique analyzes, inter alia, the non-functional features (colored orange in Figure 3.2). That could be achieved through use case scenarios, which are described in Section 3.3.1. To compare the use cases evaluation, it is necessary to define a heuristic to unify the results. This heuristic is presented in Section 3.2.2.

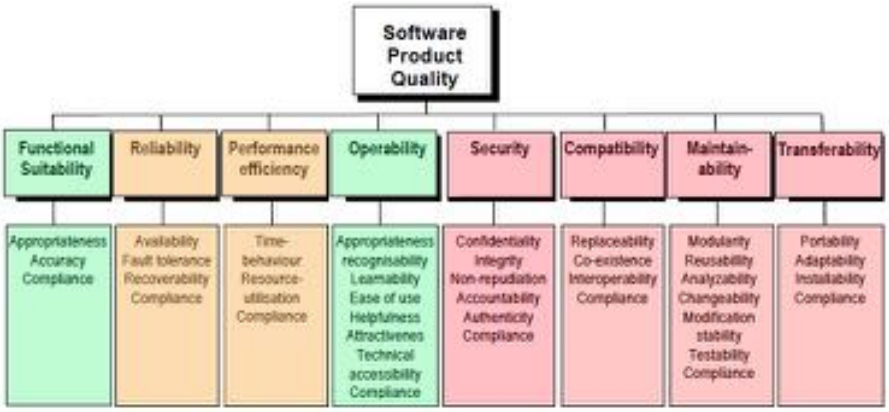


Figure 3.2: Software Product Quality Model [56, p. 14]

3.2.1 Criteria Catalog

A criteria catalog is a tool that could be used in the field of software engineering to evaluate systems or applications. It allows a standardized form of assessment and provides a clear and comparable result. A criteria catalog defines characteristics of a software system that seem to be important for the corresponding study. These characteristics depend on the objective of the investigation; e.g. for a usability evaluation, the defined criteria will differ from those of a security or code quality evaluation. [56, 18]
Braubach, Pokahr, and Lamersdorf, in [18], observe that one of the main problems in defining a criteria catalog is that, in general, there are two opposing requirements. First, the catalog should be universal and applicable to all applications of the study. Second, the catalog should be highly concrete to ensure meaningful results when evaluating specific applications. The first requirement, universality, is important to achieving the comparability of evaluation results by providing a stable set of criteria. The second requirement is essential for obtaining specific and precise evaluation results.

Derived from the software quality model presented in the previous chapter, a criteria catalog can now be defined. The ISO/IEC 25010 standard [56] and the included quality model are broadly formulated, and the standard encourages extending and defining the concrete categories. As mentioned previously, the study presented in Chapter 4 evaluates the quality in use for various airport applications. ISO/IEC 25010 [56] divides quality in use into the categories effectiveness, efficiency, satisfaction and usability in user compliance. Here, effectiveness in use describes "the degree to which specified users can achieve specified goals with accuracy and completeness in a concrete context of use" [56, p. 22]. Efficiency in use is defined as "the degree to which specified users expend appropriate amounts of resources in relation to the effectiveness achieved in a specified context of use" [56, p. 22]. Satisfaction in use describes "the degree to which users are satisfied in a specified context of use" [56, p. 22]. Usability in user compliance measures the "adherence to standards or conventions relating to usability in use" [56, p. 23].

To measure these categories the ISO/IEC 25022 standard defines properties and measurements. These measurements are used for this evaluation. Two of the categories, satisfaction in use and usability in user compliance, can only be meaningfully investigated via a field study with a suitable number of subjects [57].
Thus, the criteria catalog contains only characteristics of the categories effectiveness and efficiency. Table 3.1 displays the defined measurement criteria, assigned to the corresponding quality

model categories.

Functional Suitable	**Operability**
Effectiveness • Task completion	Efficiency • Task time • Number of clicks

Table 3.1: Criteria to Software Quality Model Assignment

These properties are used to assess a representative feature set, which is defined and presented in Section 4.1.1.

3.2.2 Heuristic Evaluation

This sub-chapter presents an additional metric for a systematic evaluation. Due to the time and resource restrictions of a master's thesis, a heuristic evaluation is particularly suitable to inspect different applications. This form of evaluation reveals many usability or design problems in a relatively short time. Wilson, in [108], defines a heuristic as a common sense rule or a simplified principle. A list of heuristics is intended as an aid or mnemonic device for the evaluators. Wilson defines a heuristic evaluation as follows:

> "A heuristic evaluation ... is a type of user interface (UI) or usability inspection where an individual, or a team of individuals, evaluates a specification, prototype, or product against a brief list of succinct usability or user experience (UX) principles or areas of concern." [108, p. 2]

Rosenbaum, Rohn, and Humberg, in [97], argue that the heuristic evaluation method is one of the most common methods in user-centered design (UCD) for identifying usability problems. Wilson in [108] revealed, as a weakness of heuristic evaluations, that they are not suitable for complex interfaces, because these UIs typically offer several methods to achieve a task. In contrast, mobile application UIs are relatively simple systems, so this form of evaluation is applicable. Furthermore, Wilson mentions that different evaluators often detect dissimilar usability problems, and they may weigh them differently, which is known as the 'evaluator effect' [43]. To average the weights, they recommend at least three to five evaluators. In the course of this thesis, only one evaluator inspects the applications due to the limited resources. Therefore, the ratings are not averaged, but the findings are weighed in the same manner. This form of evaluation provides no solution for the detected usability problem, but that is not the objective of the assessment in Chapter 4.

3.3 Usability Testing

As presented in the previous chapters, there are numerous possibilities to determine the quality and usability of software. This chapter covers the topic of usability testing, which is another technique to analyze the quality in use and UX of a product. Depending on the chosen method, usability testing allows contact with the end user and offers the opportunity to receive direct feedback.

On their website [75], MeasuringU define the core idea of usability testing as follows:

> "The core idea behind usability testing is having real people (users) try to accomplish real tasks on your software, websites, mobile apps, or devices. Through observing both what users do and say we are able to both quantify the experience using metrics and understand the problems in the experience." [75]

Roy, Pattnaik, and Mall in [98] classify usability testing into two categories based on the type of data collection. The first is performance-based evaluation, in which task success, task completion time and number of clicks are measured. The second is questionnaire-based evaluation, in which participants fill out a provided questionnaire.
Performance-based evaluation techniques have been presented in the previous chapter. MeasuringU divide user tests into two groups: moderated labor testing and unmoderated or remote user testing. Labor testing provides direct contact with end users and allows observing them while using the product. In contrast, unmoderated testing facilitates administrating tasks and questions to users around the world; thus, one can quickly collect large amounts of data.

As described in ISO/IEC 25010 [56], user tasks should be representative tasks in a realistic context of use and the evaluation should be designed to match the context of use as closely as possible. To this end, the next subsection presents the use case technique to define such representative tasks.
Then, Section 3.3.2 presents a metric to measure the usability of defined use cases. Finally, Section3.3.3describes a technique that allows the comparison of different layouts for equivalent functionality or to test the satisfaction of real users with new features.

3.3.1 Use Cases

As mentioned, quality in use could be measured by simulating a realistic working environment or by observing operational use of the product, and the evaluation should be designed to match the context of use as closely as possible [57].
For this purpose, in 1986, Jacobson formulated textual, structural and visual modeling techniques for specifying use cases [60]. In general, a use case briefly and precisely describes a scenario for the software, website, mobile application or device, and thus defines the features and implementation criteria for that use case [76].
Use cases can be derived from different sources. Rupp and Group describe in *Requirements-Engineering und -Management* [99] an approach in which the use cases derive from the requirements. They arise as a part of the system analysis and describe the functional scope.
In another approach, the use cases arise from the evaluation of applications similar to the target system. Therefore, they cover the functional scope of competitor products. This procedure offers test scenarios for the new system and provides direct comparison options to the competitor products.
An indeterministic approach for the development of use cases is to thinking about typical application scenarios of the system. This could be done as preparatory work for the requirements of engineering, but it hides the potential to forget requirements.

3.3.2 The System Usability Scale (SUS)

As previously mentioned, there are various possibilities to collect the data of an evaluation. The SUS is a method in which users perform defined use cases and answer a standardized questionnaire afterwards. Brooke originally created it in 1986 [19]. The objective is to evaluate a wide variety

of products and services. The U.S. Department of Health and Human Services describes the SUS on its website [39] as "a reliable tool for measuring the usability". It consists of a 10-item questionnaire with five response options each, ranging from strongly agree to strongly disagree. The 10 questions are as follows:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

The U.S. Department of Health and Human Services on [39] points out that the SUS has become an industrial standard and it is referenced in over 1,300 articles and publications. An advantage of the SUS is that it is an easy scale. Furthermore, it has high validity and can be used on small sample sizes with reliable results. As a result, it can effectively differentiate between usable and unusable systems.
However, one must consider when using the SUS that the scoring system is slightly complex and it is not diagnostic. Its use is in classifying the ease of use of a tested product or service. The U.S. Department of Health and Human Services on [39] warns that there is a temptation, when looking at the scores to interpret them as percentages. Although they are on a scale of 0-100, they are not percentages.

The SUS is a Likert scale [104]. This scale is based on a questionnaire in which statements are made and the respondent then indicates the degree of agreement or disagreement with the statements.
Brooke, in [19], states that the SUS "is generally used after the respondent has had an opportunity to use the system being evaluated, but before any debriefing or discussion takes place". He recommends that respondents be asked to record their immediate responses to each question, rather than be given time to think about the questions. Furthermore, Brooke argues that all questions should be checked. He states that "if a respondent feels that they cannot respond to a particular question, they should mark the center point of the scale" [19, p. 189 et. seq.].
As mentioned, the scoring system is slightly complex. The SUSyields a single number representing a composite measure of the overall usability of the system being studied. The score for a single question is not meaningful on its own. Brooke describes, in [19], the calculation of the SUS score as follows:

> "... sum the score contributions from each question. Each question's score contribution will range from 0 to 4. For question 1,3,5,7, and 9 the score contribution is the scale position minus 1. For items 2,4,6,8 and 10, the contribution is 5 minus the scale position. Multiply the sum of the scores by 2.5 to obtain the overall value. SUS scores have a range of 0 to 100." [19, p. 189 et. seq.]

3.3.3 A/B Test

In software engineering, not only is it often challenging to determine the perfect time to introduce new features, but it is also difficult to predict how users will react and whether the features fulfill their requirements. A solid statement about that can only be made if real users have tested the new features and no assumptions need to be made. A/B testing is a technique that is often used in software and web design to evaluate the usability of two potential variants.
Chopra, in [22], defines A/B testing as follows:

> "At its core, A/B testing is exactly what it sounds like: you have two versions of an element (A and B) and a metric that defines success. To determine which version is better, you subject both versions to experimentation simultaneously. In the end, you measure which version was more successful and select that version for real-world use." [22]

The A/B test is a technique for testing; in addition to that, one must define the goal and metrics for the result. In practice A/B tests, the participants are split into two groups. Each group tests one design. Chopra, in [22], explains that certain elements are typically tested: buttons (wording, size, color and placement), headlines, form length, layout and style of websites, as well as product pricing and promotion offers. The objective is to determine website changes that will increase or maximize an outcome of interest.
Chopra argues that it is crucial to test both versions simultaneously and split traffic between these two versions. To produce significant results, it is necessary to obtain a certain amount of traffic or participants for the test.

A similar technique to A/B testing is multivariate testing or multinomial testing, which may contrastingly test more than two versions at the same time or use more controls. The website *optimizely.com* describes that "The goal of multivariate testing is to determine which combination of variations performs best out of all possible combinations" [89].

4 Evaluation of Existing Applications

The previous chapter has presented several evaluation methods and techniques. Section 4.1 takes these descriptions and puts them into practice in developing the concrete evaluation form for the airport application analysis.
This chapter begins with a criteria catalog presented in Section 4.1.1, which is based on a study of the Vienna International Airport (VIE) and has been extended for this thesis to form the sum of all features of the various airport applications that will be evaluated in this chapter. It contains features with pure information features such as *airport news* and *arrival* or *departure* schedules, features that improve the orientation of travelers such as the *airport map* and features with the intention to increase the airport's revenue, e.g. *shop & restaurant information*. For each criteria completion, Time on Task (TOT) and number of clicks are measured.

Based on this criteria catalog analysis, eight scenarios have arisen. They cover features that are implemented by at least five out of the seven evaluated applications. These use cases are used to analyze the usability of tested airport applications, and they define the functional scope of the prototype. Additionally, they are reused in the user testing of the prototype. They cover topics for local arrivals who want to *retrieve public transportation information* or *parking information*, as well as features for travelers who want to relax and *retrieve lounge information*.
Section 4.1.3 presents the heuristic guidelines used to assess the usability of the use cases. They form a rating schema for the use cases in terms of usability. It defines ten categories of problems, from *visibility of system status* to *error prevention* and ending with *help and documentation*. Each detected problem is linked to one of three severity degrees.

In the second part of this chapter (Section 4.2), the developed evaluation form is used to analyze the selected airport applications. As previously mentioned, the following seven applications are assessed:
London Heathrow Airport (LHR), Aéroport Paris-Charles de Gaulle (CDG), Frankfurt Airport (FRA), Singapore Changi Airport (SIN), Amsterdam Airport Schiphol (AMS), Vienna International Airport (VIE) and Zurich Airport (ZRH).

4.1 Evaluation Form

4.1.1 Criteria Catalog

This set of criteria is based on a study of the Vienna International Airport [24] and was adapted for this thesis. It contains all features of the various airport applications.
The objective is to obtain a profound and accurate evaluation for the different airport applications and their features. The combination of the feature set and the defined metrics is presented in Table 4.1.

This feature set covers a wide range of functionality. It contains features with pure information features, features that improve the orientation of travelers at the airport, features that help with

	Arrival	Departure	Public Transportation Information	Parking Information	Shop and Restaurant Information	Services	Maps	Wi-Fi Information	Airport News	Couponing	Baggage Information	Hotel Information	Weather Information (Local)	Weather Information (Destination)	PRM Service Information	Lounges	Security and Customs Information	Security Check-In Time	My Flight	Book/Reserve Parking	Indoor Navigation	Check-In	Social Media Function	Push Notification
Completion Time																								
Clicks																								

Table 4.1: Criteria Catalog

tracking flights and features that intend to increase the revenue of the airport.
The following table presents an overview of the various features and a brief description for each:

Criteria	Description
Arrival/Departure	The application provides arrival and departure information for flights.
Public Transportation Information	The application offers information for ground/public transportation.
Parking Information	The user can retrieve information for parking possibilities, utilization rate or pricing.
Shop and Restaurant Information	It is possible to receive information for shops and restaurants such as detail descriptions, location or special offers.
Services	The application provides an overview of services available at the airport, such as medical scientist or money exchange.
Maps	The application contains an airport map.
Wi-Fi Information	The application displays free Wi-Fi spots.
Airport News	The application presents current airport news.
Couponing	The user can receive coupons via the application.
Baggage Information	The application provides information about the passenger's baggage.
Hotel Information	It is possible to obtain information regarding hotels or reserve hotel rooms via the application.
Weather Information (Local)	The application provides weather information for the airport.
Weather Information (Destination)	Weather information from the destination airport can be retrieved.
Passengers with Reduced Mobility (PRM) Service Information	The application offers special information for PRM.
Lounges	The application presents details such as the location of lounges.
Security and Customs Information	It is possible to receive security and customs information via the application, such as security regulations for hand baggage or exemption limits for air traveling.

Security Check-In Time	The application displays (in real time) the queuing time for the security check-in.
Travel Agents	The location (and details) of travel agents at the airport can be retrieved.
My Flight	The application provides an overview of the passenger's flights.
Book/Reserve Parking	It is possible to book/reserve a car parking via the application.
Indoor Navigation	The application provides an active/real indoor navigation.
Check-In	It is possible to check in via the application.
Social Media Function	The application provides social media functions, such as sharing via Facebook or Twitter.
Push Notification	The application informs the user about (important) news via push notifications.

Table 4.2: Criteria Description

Metrics

As mentioned in Section 3.2.1, the properties task completion, time on task and number of clicks are measured for the above defined criteria. The value for *Task completion* could be either 1 or 0, where 1 means the application contains the feature and that the task could be fulfilled, while 0 means the feature is not implemented or could not be completed.
Task time is the time in seconds that is needed to complete a task. Each task is measured three times and the corresponding average is calculated. The optimal path is used for this. The value for *clicks* is the number of clicks that are needed to achieve the task's goal, using the optimal path.

4.1.2 Use Cases

The criteria catalog analysis evaluates task completion, number of clicks and time on task. All of these properties are important and suitable in comparing different applications, but to determine the usability of an application, this analysis must be extended. Thus, the following eight use cases are used to assess usability of the main features.
As mentioned in Section 3.3.1, there are different approaches to develop use cases. For this thesis, the use cases derived from the criteria catalog evaluation. First, all applications were assessed using the criteria catalog. This analysis revealed that 10 features are implemented by at least five out of the seven tested applications. Based on these 10 features, eight use cases are derived. The objective of the use case analysis is to detect usability problems, identify positive design approaches and provide the ability to compare the usability of tested applications.
In the next step, presented in Chapter 5, a prototype called APPenger is developed, which exclusively implements these 10 main features. The use cases represent test scenarios for the prototype usability testing as well. Additionally, they offer comparability with the rated airport applications. Section 4.1.3 presents the heuristics by which the uses cases are assessed.

Figure 4.1 presents the use cases for the evaluation:

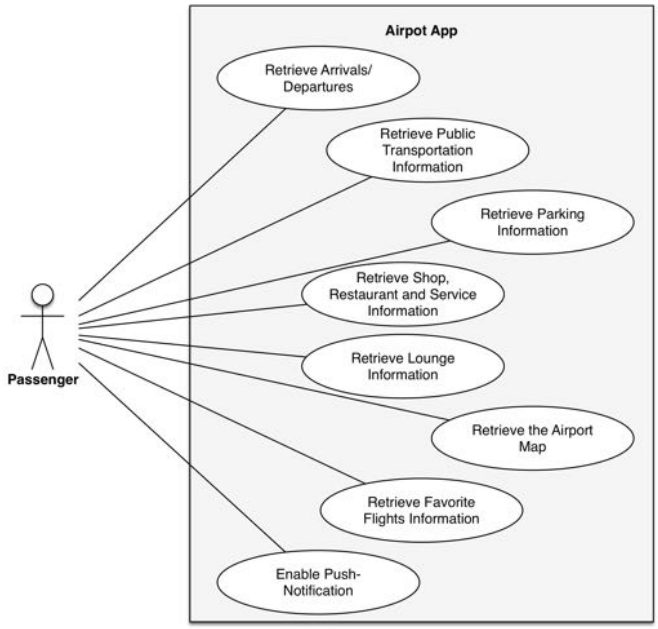


Figure 4.1: Use Case Diagram

Use Case 1: Retrieve Arrivals/Departures

Number	1
Name	Retrieve Arrivals/Departures
Actor(s)	User
Precondition	Internet connection
Postcondition	The passenger received information regarding arrival/departure flights
Description	The passenger navigates to a provided flight schedule view. He/she can select a flight to get detail information.

Use Case 2: Retrieve Public Transportation Information

Number	2
Name	Retrieve Public Transportation Information
Actor(s)	User
Precondition	Internet connection
Postcondition	The passenger received a timetable for the public transportation options.
Description	The passenger navigates to a provided public transportation schedule view.

Use Case 3: Retrieve Parking Information

Number	3
Name	Retrieve Parking Information
Actor(s)	User
Precondition	Internet connection
Postcondition	The passenger received current car parking information.
Description	The passenger opens the car parking information and selects the desired information category to receive detailed information.

Use Case 4: Retrieve Shop, Restaurant and Service Information

Number	4
Name	Retrieve Shop, Restaurant and Service Information
Actor(s)	User
Precondition	Internet connection
Postcondition	The passenger receives information regarding shops, restaurants or services.
Description	The passenger would like to get detailed information for a specific shop. He/she has the ability to retrieve it for the desired shop via the application.
Alternative	The passenger is looking for a restaurant or bar. He/she opens a detailed view of the bar or restaurant and receives information such as opening hours, location and a brief description.
Alternative	The passenger needs information for airport services. He/she has the ability to retrieve the offered airport services via the application.

Use Case 5: Retrieve Lounge Information

Number	5
Name	Retrieve Lounge Information
Actor(s)	User
Precondition	Internet connection; the airport offers a lounge or located airlines provide lounges
Postcondition	The passenger received detail information for the desired lounge.
Description	The passenger navigates to a lounge overview, select the desired lounge and get detail information for it.

Use Case 6: Retrieve the Airport Map

Number	6
Name	Retrieve the Airport Map
Actor(s)	User
Precondition	Internet connection or a local saved airport map
Postcondition	The passenger sees the airport map.
Description	The passenger has the ability to open an airport map view via the application.

Use Case 7: Retrieve Favorite Flights Information

Number	7
Name	Retrieve Favorite Flights
Actor(s)	User
Precondition	Already added flights to the user's favorite flights; Internet connection
Postcondition	The passenger received detailed information for his/her flights.
Description	The passenger has already added his/her favorite flights. He/she can open an overview to select the desired flight and get detailed information.

Use Case 8: Enable Push Notification

Number	8
Name	Enable Push Notification
Actor(s)	User
Precondition	Internet connection; Push notifications are enabled for this application in the security settings of the mobile device
Postcondition	Push notifications for a specific flight are enabled.
Description	The passenger wants to activate push notifications for a flight. He/she has the ability to select the desired flight and enable the notifications.

4.1.3 Heuristic Guideline

Table 4.3 provides a list of heuristics from Nielsen [80] that highlight potential usability problem categories. Based on these heuristics, other studies and books (e.g. [46, 108]) define their evaluation procedures for UIs.

No	Guideline	Explanation
1	Visibility of system status	The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
2	Match between system and the real world	The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
3	User control and freedom	Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
4	Consistency and standards	Users should not have to wonder whether different words, situations, or actions mean the same thing.
5	Error prevention	Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
6	Recognition rather than recall	Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
7	Flexibility and efficiency of use	Accelerators – unseen by the novice user – may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
8	Aesthetic and minimalist design	Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
9	Help users recognize, diagnose, and recover from errors	Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
10	Help and documentation	Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Table 4.3: Usability Heuristics for User Interface Design (Nielsen 1995, [80])

To measure the usability of applications by using the heuristic, a severity rating must be established. This offers the ability to assess the identified usability problems and to provide comparable results of the reviews. Based on the evaluation in [46], three severity grades are defined: *minor (1)*, the value for cosmetic issues like untypical icons or overlapping text fields; *serious (2)*, meaning that presented information or options differ to some extent from the user's expectations; and *critical (3)*, the value if a task could not be completed or it is not possible to continue.
The result of this evaluation is a list of usability problems for the different use cases rated by the defined schema.

4.2 Application Evaluation

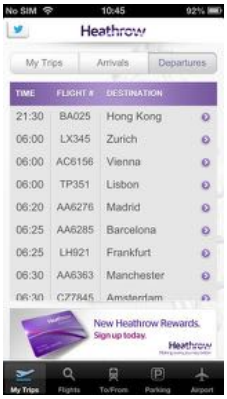
Each evaluation begins with a brief overview containing a picture of the main screen, as well as information such as the name and publisher of the application and the evaluation date. After that, the criteria catalog is used to assess the airport applications. This catalog supplies a measurable and comparable first impression of the applications. Every task is measured three times to calculate an average value. Thereafter, to compare the applications' usability, they are reviewed by the use cases; this describes a walkthrough of the defined scenarios. Every investigation ends with a summary of the use case results.

The following table presents the technical data of the used test device:

Property	**Value**
Device	Apple iPod touch (4th generation)
System chip	Apple A4 (1 GHz)
Graphics processor	Yes
System memory	256 MB RAM
Capacity	32GB
Operating System	iOS 6.1.6
Size	Height: 4.4 inches (111.0 mm) Width: 2.32 inches (58.9 mm) Depth: 0.28 inch (7.2 mm)
Wireless	802.11b/g/n Wi-Fi (802.11n 2.4GHz only) Bluetooth 2.1 + EDR
Display	3.5-inch (diagonal) widescreen Multi-Touch display 960-by-640-pixel resolution at 326 pixels per inch

Table 4.4: iPod touch (4th generation) - Technical Specifications [51]

4.2.1 London Heathrow Airport (LHR)



Name	Heathrow Airport Guide
Publisher	Heathrow Airport Limited
Version	4.2
Last Update	17.03.2015
Price	free
Evaluated on	16.06.2016

Figure 4.2: LHR Home [66]

Basic Airport Data

The London Heathrow Airport (LHR) is with 85 million persons approximately (PAX) and 472,067 movements in 2015, one of the largest airports in the world and is the largest in Europe. It has four terminals, two runways and 613 check-in counters. The 12.27 km^2 hold 177 aircraft stands. Eighty located airlines fly to 185 destinations in 84 countries. [67]

Criteria Catalog

	Arrival	Departure	Public Transportation Information	Parking Information	Shop and Restaurant Information	Services	Maps	Wi-Fi Information	Airport News	Couponing	Baggage Information	Hotel Information	Weather Information (Local)	Weather Information (Destination)	PRM Service Information	Lounges	Security and Customs Information	Security Check-In Time
Completion	1	1	1	1	1	0	1	0	1	0	1	0	1	1	0	1	0	0
Time (in Sec.)	1	1	14.3	25	9.1	-	9.1	-	4	-	2	-	7.7	7.7	-	33.3	-	-
Clicks	1	1	4	10	4	-	2	-	1	-	1	-	2	2	-	5	-	-

	My Flight	Book/Reserve Parking	Indoor Navigation	Check-In	Social Media Function	Push Notification	Sum
Completion	1	1	0	0	1	1	15
Time (in Sec.)	1	25	-	-	2	4.8	147
Clicks	1	12	-	-	2	5	53

Table 4.5: LHR Application Criteria Catalog

Use Case Analysis

The detailed description of the use case analysis can be found in the Appendix (see Section 9.2.1).

1) Retrieve Arrivals/Departure:
After starting the application, the home screen with three tabs '*My Trips*', '*Arrival*' and '*Departure*' appears. '*My Trips*' is the default tab. The user must click on '*Arrival*' or '*Departure*' to open the corresponding schedule. Clicking on a list entry opens the desired flight details.

2) Retrieve Public Transportation Information:
The user can retrieve public transportation information via the '*To/From*' menu entry. This view presents a search form in which the desired time and origin must be inserted. Furthermore, the user can select the transportation type, e.g. bus or train. After a click on '*Find Journeys*', the possible connections are shown.

3) Retrieve Parking Information:
The menu entry '*Parking*' provides parking information. The user must enter the terminal and parking duration and click on '*Get my quote*'. The London Heathrow Airport (LHR) application offers an info screen with parking possibilities where it is also possible to reserve a parking lot.

4) Retrieve Shop, Restaurant and Service Information:
The LHR application offers shop and restaurant information via the map; special service information is not available. The user must open a flight detail view and click on *Terminal* to open the map. There, he/she can browse it or search for specific shops or restaurants via the input field on top of this view.

5) Retrieve Lounge Information:
Like in use case 4, the lounge information is retrievable via the map.

6) Retrieve the Airport Map:
As mentioned in use case 4, the user must navigate to a flight detail view and click on '*Terminal*' to open the map.

7) Retrieve Favorite Flights Information:
Favorite flights are presented directly on the home screen. The user can select a flight to open the corresponding detail view.

8) Enable Push-Notification:
While adding a flight to one's favorite flights, a confirmation dialog appears, on which it is possible to enable the notifications for said flight.

Summary

The LHR application supports all eight use cases but provides no airport service information. It fulfills 15 of the 24 criteria; to achieve all of them, 139 seconds and 53 clicks are needed. Furthermore, 12 usability problems could be detected by the heuristic evaluation. Eight of the usability problems are minor, and four are serious. Table 4.6 illustrates a summary of the heuristic evaluation for the use cases.
The LHR map displays seating, which is helpful for elderly passengers or PRMs. Every flight could be shared via e-mail, Facebook or Twitter. LHR also offers a *Heathrow Airport Twitter* feed to publish current news. The application provides no offline function.

1	**Visibility of System Status**	
	▶ The application starts with a splash screen but without progress information	**minor**
3	**User control and freedom**	
	▶ Some views provide a *Back* button others not	**minor**
4	**Consistency and standards**	
	▶ The layout of the *Back* button differs on different views	**minor**
7	**Flexibility and efficiency of use**	
	▶ On the *Parking* view, the time has to be selected via a drop down box	**minor**
	▶ On the *Parking* view, the drop down box for the terminals provides only four entries but it require scrolling down	**minor**
	▶ Shop and restaurant information is only available via the map	**minor**
	▶ The selection in the drop down box must be confirmed	**serious**
	▶ To get local weather information, one must open the flight details of an arrival flight	**serious**
	▶ To open the map, one must navigate via the flight details	**serious**
8	**Aesthetic and minimalist design**	
	▶ The map shows too many result icons after a search for one specific shop	**minor**
9	**Help user recognize, diagnose and recover from errors**	
	▶ Search for a flight number of another day's results in an error dialog instead of an empty result list	**serious**
	▶ The caption of an error dialog for parking is: *http://m.secure.heathrow.com*	**minor**

Table 4.6: LHR Heuristic Evaluation

4.2.2 Singapore Changi Airport (SIN)

Name	iChangi
Publisher	Changi Airport Group (Singapore) Pte Ltd
Version	1.67
Last Update	28.12.2015
Price	free
Evaluated on	19.06.2016

Figure 4.3: SIN Home [68]

Basic Airport Data

Singapore Changi Airport (SIN) is one of the largest airports in Asian. It managed 66 million PAX and 346,300 movements in 2015. The 100 located airlines fly to 330 destinations in 80 countries. The Singapore Changi Airport (SIN) offers three terminals with 310 check-in counters. The 13 km^2 airport area contains three runways and 180 aircraft stands take place. [31]

Criteria Catalog

	Arrival	Departure	Public Transportation Information	Parking Information	Shop and Restaurant Information	Services	Maps	Wi-Fi Information	Airport News	Couponing	Baggage Information	Hotel Information	Weather Information (Local)	Weather Information (Destination)	PRM Service Information	Lounges	Security and Customs Information	Security Check-In Time
Completion	1	1	1	1	1	0	1	1	1	1	1	1	0	0	0	1	1	0
Time (in Sec.)	1	1	5	5	3	-	5	7.7	1	2	5	9.1	-	-	-	6.3	6.3	-
Clicks	1	1	4	4	3	-	4	3	1	2	2	3	-	-	-	3	3	-

	My Flight	Book/Reserve Parking	Indoor Navigation	Check-In	Social Media Function	Push Notification	Sum
Completion	1	0	0	0	0	1	15
Time (in Sec.)	1	-	-	-	-	3	61.4
Clicks	1	-	-	-	-	2	35

Table 4.7: SIN Application Criteria Catalog

Use Case Analysis

A detailed description of the use case analysis can be found in the Appendix (see Section 9.2.2).

1) Retrieve Arrivals/Departure:
The SIN home screen presents a search form for flights. The user can insert the desired data to search for the corresponding flight. Alternatively, he/she can swipe the view to the left and open the current arrival schedule. If he/she swipes the view once more, the current departure list appears. The user can click on the desired flight to open the flight detail view.

2) Retrieve Public Transportation Information:
The public transportation information is retrievable via the sidebar menu. The user must select '*Airport Info*' and click '*To & From Airport*'. This view presents all information categories regarding local transportation.

3) Retrieve Parking Information:
The user can receive parking information via the '*To & From Airport*' view. The '*Parking*' view offers general car parking information.

4) Retrieve Shop, Restaurant and Service Information:
The SIN application provides shop and restaurant information via the sidebar menu entry '*Shop & Dine*'. On this view, the user must select the shop category to open a shop list or select the terminal to get the restaurant list. On the next view, he/she can select a list entry to open the detail view.
The SIN application offers no special information for services.

5) Retrieve Lounge Information:
Lounge information is available via the '*Attraction/Facilities*' sidebar menu entry. On this view, the user must click '*Smoking Area (Lily Pads)*' to open a list of lounges and their locations.

6) Retrieve the Airport Map:
The user can open the airport map by clicking on '*Terminal Map*' in the sidebar menu. On this screen, he/she can select the desired terminal to receive the available terminal areas. After clicking an area, the map is presented.

7) Retrieve Favorite Flights Information:
Favorite flights are presented directly on the home screen. There the user can click on a flight to open the flight detail view.

8) Enable Push Notification:
Push notifications for favorite flights are automatically enabled. The SIN application provides multiple notification types. The user can globally enable or disable the different types for the application in the settings menu.

Summary

Fifteen criteria are satisfied by the SIN application; 35 clicks and 61.4 seconds are necessary to achieve these criteria. The application supports all eight use cases but offers no airport service information. Additionally, 10 usability problems were detected, of which six are minor and four are serious problems. Table 4.8 illustrates a summary of the heuristic evaluation for the use cases.

1	**Visibility of System Status**	
	▶ The application starts with a splash screen but without progress information	**minor**
3	**User control and freedom**	
	▶ The symbol to add a flight is a heart	**minor**
	▶ One of the grouping options for restaurants is named *By Shop*	**minor**
4	**Consistency and standards**	
	▶ On the flight detail view the *Gate* button open the map and mark the gate; the *Check-In row* button leads to an advertising view	**serious**
7	**Flexibility and efficiency of use**	
	▶ Settings do not automatically save	**serious**
	▶ Lounge information is hidden under categories such as *Smoking areas*, *Powder Room* or *Orchid garden*	**serious**
	▶ Navigating to the *Home* screen requires up to six clicks	**minor**
	▶ The time range to search for a flight is restricted to +/- one day of the current date	**serious**
8	**Aesthetic and minimalist design**	
	▶ The layout of the flight schedule seems to be overloaded	**minor**
	▶ The flight detail view presents the information *Schedule* and *Estimated*, even if both show the same time	**minor**

Table 4.8: SIN Heuristic Evaluation

4.2.3 Aéroport Paris-Charles de Gaulle (CDG)

Name	My Airport – Official Aéroports de Paris service
Publisher	Aéroports de Paris
Version	7.3
Last Update	03.06.2014
Price	free
Evaluated on	17.06.2016

Figure 4.4: CDG Dashboard [92]

Basic Airport Data

With 64 million PAX and 514,049 movements in 2015, Aéroport Paris-Charles de Gaulle (CDG) is the second largest airport in Europe and one of the 10 largest in the world. The airport area of 35 km^2 contains three terminals, four runways and 314 aircraft stands. The passengers can check in on 420 counters and the 319 located airlines fly to 319 destinations. [1]

Criteria Catalog

	Arrival	Departure	Public Transportation Information	Parking Information	Shop and Restaurant Information	Services	Maps	Wi-Fi Information	Airport News	Couponing	Baggage Information	Hotel Information	Weather Information (Local)	Weather Information (Destination)	PRM Service Information	Lounges	Security and Customs Information	Security Check-In Time
Completion	1	1	1	1	1	1	1	1	0	0	1	0	0	0	1	1	1	0
Time (in Sec.)	10	10	4	4	5	5	3	5	-	-	5	-	-	-	5	11.1	3	-
Clicks	6	4	3	3	4	3	2	3	-	-	3	-	-	-	3	4	2	-

	My Flight	Book/Reserve Parking	Indoor Navigation	Check-In	Social Media Function	Push Notification	Total Sum
Completion	1	0	0	0	1	1	15
Time (in Sec.)	2	-	-	-	5	4	81.1
Clicks	2	-	-	-	3	3	48

Table 4.9: CDG Application Criteria Catalog

Use Case Analysis

A detailed description of the use case analysis can be found in the Appendix (see Section 9.2.3).

1) Retrieve Arrivals/Departure:
The user can use the search field at the bottom of the home screen to retrieve arrival or departure information, or he/she can click '*Schedule & airline*' to open an advanced search view. There, he/she can select the desired departure or destination and the date. Furthermore, the flight number can be inserted. After clicking on '*Search*', the corresponding schedule is presented. The user can select a flight to get the detailed information.

2) Retrieve Public Transportation Information:
The Aéroport Paris-Charles de Gaulle (CDG) application provides public transportation information via the menu entry '*Orientation*'. This screen contains the category '*Access Information*'. There, the user can choose between the options '*By public transportation*', '*By car or motorcycle*' or '*By taxi*'. After clicking '*By public transportation*', general information is shown.

3) Retrieve Parking Information:
The user can retrieve parking information via the '*Service & Shopping*' menu entry. This view contains a link to car parking. The car parking information is a list of different categories regarding parking. The user can select '*Parking space available*' to view current available car lots.

4) Retrieve Shop, Restaurant and Service Information:
The '*Service & Shopping*' view provides a '*Shopping*' link, which opens the shop list. The user can click on a shop to obtain detailed information. The '*Service & Shopping*' view also offers a '*Services*' link, which opens a list of service categories. The user can click on a list entry to open the detail view.
Restaurant information is retrievable via the '*Orientation*' view. On this screen, the user can select '*Point of interest search*'. That view contains a '*Bars and restaurants*' link, which opens the bars and restaurant list.

5) Retrieve Lounge Information:
To retrieve the lounge information, users must open the '*Point of interest search*' view and select '*Formalities*'. This opens a list of categories. One of these categories is '*Lounge*'; this leads to a list of all available lounges.

6) Retrieve the Airport Map:
The link to the airport map is contained in the '*Orientation*' view. The user must select '*See the maps*' to open the airport overview map. The CDG provides no indoor map – only the airport overview.

7) Retrieve Favorite Flights Information:
One favorite flight is directly presented on the home screen. To retrieve the list of all favorite flights the user can open the '*My space*' view. This screen provides all bookmark categories, including '*My flights*'.

8) Enable Push Notification:
Every flight detail view contains a toggle button to enable or disable notifications. After the user enables notifications, the flight is automatically added to the user's favorite flights.

Summary

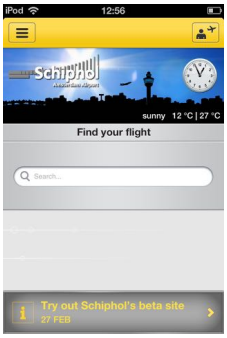
The CDG application satisfies 15 criteria on the criteria catalog. To achieve all of these, 48 clicks and 81.1 seconds are required. It is possible to submit all eight use cases. The heuristic evaluation revealed 16 usability problems, of which seven of them are serious problems. Table 4.10 illustrates a summary of the heuristic evaluation for the use cases.

1	**Visibility of System Status**	
	▶ The application starts with a splash screen but without progress information	**minor**
	▶ The flight detail view provides a *Call* option but offers no feedback after clicking on it	**minor**
	▶ The *Point of Interest* view loads some seconds but it provides no progress feedback	**minor**
2	**Match between system and the real world**	
	▶ The menu entry *Map* provides an overview of the airport area; at this point the user would expect an indoor map	**serious**
	▶ The *Point of Interest* view contains the category *Formalities*, which groups gates and terminals; the category name is confusing	**minor**
	▶ The public transportation information is located under *Orientation*	**minor**
3	**User control and freedom**	
	▶ The bookmark button looks like a toggle button but it provides no undo functionality	**serious**
4	**Consistency and standards**	
	▶ The flight schedule offers a menu titled *Sort*, but this menu contains only filter options	**serious**
	▶ The bookmark icon looks liken a 'add user' symbol	**minor**
	▶ The confirmation button caption of the date selection dialog is 'validate'	**minor**
	▶ The notification button is a toggle button in contrast to the bookmark button	**serious**
	▶ The bookmark for search results adds the search screen to the bookmarks, rather than the results list	**serious**
5	**Error prevention**	
	▶ The flight detail view offers the ability to enable notifications for that flight. For some flights, a dialog occurs with the message: "Notifications are not available for this flight"	**serious**

7	**Flexibility and efficiency of use**	
	► Every bookmark must be confirmed	**minor**
8	**Aesthetic and minimalist design**	
	► The flight schedule list provides too much information, it looks overloaded	**minor**
9	**Help user recognize, diagnose, and recover from errors**	
	► Incomprehensible error message for deactivating notifications: "A problem occurred while registering to flight notifications. Please try later."	**serious**

Table 4.10: CDG Heuristic Evaluation

4.2.4 Amsterdam Airport Schiphol (AMS)



Name	Schiphol Amsterdam Airport
Publisher	Schiphol Nederland B.V.
Version	3.3.3
Last Update	03.06.2015
Price	free
Evaluated on	19.06.2016

Figure 4.5: AMS Home [20]

Basic Airport Data

The Amsterdam Airport Schiphol (AMS) is, with 64 million PAX and 450,679 movements in 2015, Europe's third largest airport. The airport area covers 27.9 km^2 and contains for one terminal, six runways and 207 aircraft stands. The 109 located airlines connect to 322 destinations. Amsterdam Airport Schiphol (AMS) provides 311 check-in counters. [32]

Criteria Catalog

	Arrival	Departure	Public Transportation Information	Parking Information	Shop and Restaurant Information	Services	Maps	Wi-Fi Information	Airport News	Couponing	Baggage Information	Hotel Information	Weather Information (Local)	Weather Information (Destination)	PRM Service Information	Lounges	Security and Customs Information	Security Check-In Time
Completion	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	0	0
Time (in Sec.)	3	4	4	6.3	6.3	10	5	10	-	3	5	6.3	1	7.7	-	7.1	-	-
Clicks	2	3	2	3	3	4	2	4	-	2	3	5	1	4	-	4	-	-

	My Flight	Book/Reserve Parking	Indoor Navigation	Check-In	Social Media Function	Push Notification	Sum
Completion	1	1	0	1	1	1	19
Time (in Sec.)	3	4	-	16.6	7.7	5	115
Clicks	2	4	-	5	5	4	62

Table 4.11: AMS Application Criteria Catalog

Use Case Analysis

A detailed description of the use case analysis can be found in the Appendix (see Section 9.2.4).

1) Retrieve Arrivals/Departure:
The AMS application displays a search field on the home screen. Users can enter a search term to find the desired flight. Alternatively, he/she can open the sidebar menu and click '*All flights*'. This opens the current flight schedule. The user can select a flight to get detailed information.

2) Retrieve Public Transportation Information:
General public transportation information is available via the menu entry '*Trains*'.

3) Retrieve Parking Information:
The user can retrieve parking information via the '*Parking*' menu entry in the sidebar.

4) Retrieve Shop, Restaurant and Service Information:
The AMS application provides shop information and current offers via the menu entry '*Shop*'. This view presents the top 10 offers. Furthermore, this view groups the offers by category. The user can switch between categories to view the different offers. Service information is also available via the menu. The user can click '*Facilities*' to retrieve a list of services and facilities. He/she can click on a list entry to open the detail view.
Restaurant information can be retrieved via the airport map. The user must click on '*Map*' in the sidebar menu. This view offers a search list that contains bars and restaurants.

5) Retrieve Lounge Information:
Like the restaurant information in use case 4, lounge information is also retrievable via the map.

6) Retrieve the Airport Map:
The airport map can be opened by clicking the '*Map*' entry in the sidebar menu.

7) Retrieve Favorite Flights Information:
One favorite flight is retrievable via an icon in the upper right corner of the home screen. To receive a list of all favorite flights, the user can open the '*My flight*' list via the corresponding menu entry. He/she can click on the desired flight to open the detail view.

8) Enable Push Notification:
The AMS application provides, on every flight's detail view, a '*Push notification*' toggle button to enable or disable flight notifications. After activation, the flight is automatically added to the user's favorite flights.

Summary

The AMS application supplies all eight use cases. Furthermore, it fulfills 19 of the 24 criteria, which require 115 seconds and 62 clicks to achieve. Eight usability problems were revealed, of while three are serious and one critical. Table 4.12 illustrates a summary of the heuristic evaluation for the use cases.

1	**Visibility of System Status**	
	► The application starts with a splash screen but without progress information	**minor**
	► Menu entry *At Schiphol* opens an empty screen without further information	**serious**
2	**Match between system and the real world**	
	► The flight detail view contains a *Things to do* link, which sounds like a checklist but provides a list of bars and family corners	**minor**
3	**User control and freedom**	
	► The date selection button on the flight search screen is easy to overlook	**minor**
5	**Error prevention**	
	► To register or log in leads to a connection error dialog	**serious**
6	**Recognition rather than recall**	
	► The side menu could be opened from every screen, but this is not apparent	**serious**
7	**Flexibility and efficiency of use**	
	► The flight detail view contains a link to weather information; this could presented directly, instead of through a separate link	**minor**
9	**Help user recognize, diagnose, and recover from errors**	
	► Login via Facebook results in a cryptic error message	**critical**

Table 4.12: AMS Heuristic Evaluation

4.2.5 Frankfurt Airport (FRA)

Name	Frankfurt Airport (FRA Airport)
Publisher	Fraport AG - Frankfurt Airport Services Worldwide
Version	2.4
Last Update	30.07.2015
Price	free
Evaluated on	18.06.2016

Figure 4.6: FRA Live View [111]

Basic Airport Data

The site of Frankfurt Airport (FRA) covers 21.6 km^2. That includes two terminals, four runways and 221 aircraft stands. In 2015, Frankfurt Airport (FRA) managed 61 million PAX and 469,026 movements. Because of this, it is the fourth largest airport in Europe. The 104 located airlines connect to 297 destinations in 104 countries. FRA provides 410 check-in counters for its passengers. [6]

Criteria Catalog

	Arrival	Departure	Public Transportation Information	Parking Information	Shop and Restaurant Information	Services	Maps	Wi-Fi Information	Airport News	Couponing	Baggage Information	Hotel Information	Weather Information (Local)	Weather Information (Destination)	PRM Service Information	Lounges	Security and Customs Information	Security Check-In Time
Completion	1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	1	0	1
Time (in Sec.)	1	2	6.3	5	2	2	5	5	1	4	-	6.3	-	.	6.3	3	-	3
Clicks	1	2	3	4	2	2	1	4	1	2	-	3	-	-	3	2	-	1

	My Flight	Book/Reserve Parking	Indoor Navigation	Check-In	Social Media Function	Push Notification	Sum
Completion	1	1	1	0	0	1	18
Time (in Sec.)	1	14.3	25	-	-	3	95.2
Clicks	1	5	7	-	-	2	46

Table 4.13: FRA Application Criteria Catalog

Use Case Analysis

A detailed description of the use case analysis can be found in the Appendix (see Section 9.2.5).

1) Retrieve Arrivals/Departure:
The FRA application presents a permanent menu at the bottom edge. This menu contains a '*Flight*' entry. The user can select the desired flight to open the detail view.

2) Retrieve Public Transportation Information:
The user must click '*Airport Guide*'. This view presents different information categories. To retrieve public transportation information, the user must select *Driving direction*, then click '*Bus and train*'. This view offers general information regarding local transportation.

3) Retrieve Parking Information:
Parking information is provided via the '*Airport Guide*', as well. On this view, the user must select '*Parking*', then '*Parking options at airport*'. This view presents the various parking options. The user can choose e.g. '*Terminal 1*' to open the detail view.

4) Retrieve Shop, Restaurant and Service Information:
Like use cases 2 and 3, shop, service and restaurant information is retrievable via the '*Airport Guide*' view. The user can choose between the categories '*Restaurants*', '*Shops*' and '*Services*' to open the desired list. He/she can click on a list entry to open the detail view.

5) Retrieve Lounge Information:
The FRA application offers lounge information via the '*Airport Guide*' view. The user must select '*Lounges*' to receive the lounge list. He/she can select the desired lounge to open the detail view.

6) Retrieve the Airport Map:
The airport map can be opened via the '*Map*' menu entry. The FRA map is the only one out of the evaluated airport applications that provides active navigation functions.

7) Retrieve Favorite Flights Information:
Favorite flights are directly presented on the home screen. The user can select the desired flight to open the detail view.

8) Enable Push Notification:
Push notifications are automatically enabled for every favorite flight. The FRA application provides global notification settings in the sidebar menu.

Summary

The FRA application supports all eight use cases. The heuristic evaluation determined six usability problems, of which half of them are serious problems. Table 4.14 illustrates the summary. The application complies with 18 of the 24 criteria; to achieve them, 95.2 seconds and 46 clicks are required, in total.

1	**Visibility of System Status**	
	▶ The application starts with a splash screen but without progress information	**minor**
2	**Match between system and the real world**	
	▶ To start the navigation on the navigation view, one must click on the *Get Direction* button; the *Show on Map* button only shows the destination on the map	**serious**
4	**Consistency and standards**	
	▶ The *Save For Later* button is a toggle button, but it does not appear as such	**minor**
7	**Flexibility and efficiency of use**	
	▶ The side menu *Favorites & More* can only be opened from the home screen	**serious**
	▶ It is only possible to search for flights by destination, not by date or time	**serious**
8	**Aesthetic and minimalist design**	
	▶ The information labels overlap in the flight schedule view	**minor**

Table 4.14: FRA Heuristic Evaluation

4.2.6 Vienna International Airport (VIE)

Name	Vienna Airport
Publisher	Flughafen Wien AG
Version	1.4.2
Last Update	26.10.2015
Price	free
Evaluated on	14.06.2016

Figure 4.7: VIE Dashboard [4]

Basic Airport Data

The Vienna International Airport (VIE) is the largest airport in Austria. It managed 30 million PAX and 226,811 movements in 2015. Its 10 km^2 area contains three terminals, two runways and 96 aircraft stands. Vienna International Airport (VIE) offers 128 check-in counters, and the 75 located airlines fly to 181 destinations in 75 countries. [2]

Criteria Catalog

	Arrival	Departure	Public Transportation Information	Parking Information	Shop and Restaurant Information	Services	Maps	Wi-Fi Information	Airport News	Couponing	Baggage Information	Hotel Information	Weather Information (Local)	Weather Information (Destination)	PRM Service Information	Lounges	Security and Customs Information	Security Check-In Time
Completion	1	1	1	1	1	1	1	0	0	0	1	0	1	0	0	1	1	0
Time (in Sec.)	7.7	7.7	16.7	2	20	33.3	7.1	-	-	-	25	-	1	-	-	33.3	7.1	-
Clicks	2	2	3	1	4	4	2	-	-	-	5	-	1	-	-	4	3	-

	My Flight	Book/Reserve Parking	Indoor Navigation	Check-In	Social Media Function	Push Notification	Sum
Completion	1	0	0	0	0	1	13
Time (in Sec.)	1	-	-	-	-	1	162.9
Clicks	1	-	-	-	-	1	33

Table 4.15: VIE Application Criteria Catalog

Use Case Analysis

A detailed description of the use case analysis can be found in the Appendix (see Section 9.2.6).

1) Retrieve Arrivals/Departure:
If no flight has been added to favorite flights, the VIE application provides links to arrival and departure schedules on the home screen. The user can click the desired one to open the corresponding flight list. He/she can click on a list entry to open the detail view.
Alternatively, the user can open the schedules via the sidebar menu.

2) Retrieve Public Transportation Information:
Public transportation information is retrievable via the '*Arrival & Parking*' menu entry. This view presents different information categories regarding local transportation. The user can select '*CAT, train & bus to the airport*' to open the local arrival schedule.

3) Retrieve Parking Information:
The home screen of the VIE application presents, in the lower left corner, current parking information. The user can click on this to open a list with the workload of every parking spot. He/she can select a parking spot to check the parking fee.

4) Retrieve Shop, Restaurant and Service Information:
Shop and restaurant information is available via the '*Shops & Restaurant*' menu entry. The user can select the desired category to retrieve a corresponding list view. He/she can click on a list entry to open the details.

5) Retrieve Lounge Information:
The VIE application provides lounge information via the airport map. The user can choose the desired lounge from the list in the upper right corner. The selected lounge is marked on the map.

6) Retrieve the Airport Map:
The airport map is retrievable via the '*Airport map*' menu entry.

7) Retrieve Favorite Flights Information:
Favorite flights are presented directly on the home screen as a card layout. The user can switch between the different flights by swiping through the cards.

8) Enable Push Notification:
Push notifications are automatically enabled for favorite flights.

Summary

The VIE application satisfies 13 criteria from the criteria catalog, and it is possible to submit all eight use cases. To fulfill these criteria, 33 clicks and 162.9 seconds are required. The heuristic evaluation revealed nine usability problems, of which six are minor and three are serious problems. Table 4.16 illustrates a summary of the heuristic evaluation for the use cases.

1	**Visibility of System Status**	
	► The application starts with a splash screen but without progress information	**minor**
	► The *Search for Flights* view contains a date selection input field. One must select the date from a drop down menu. The drop down box contains a confirm button, but the selection is automatically taken without the confirmation	**minor**
3	**User control and freedom**	
	► It is not possible to directly undo the addition of a flight	**serious**
4	**Consistency and standards**	
	► The search view contains a field '*Enter flight directly*'. This field looks like a drag and drop field, but it is a link to another search view	**serious**
	► Some views, e.g. *Parking possibilities* mix German and English	**minor**
5	**Error prevention**	
	► The destination search field provides no auto-completion	**minor**
7	**Flexibility and efficiency of use**	
	► It is not possible to open flight details via the flight schedule; rather, they are accessible via the *Add departure* or *Add arrival* menu entry	**serious**
	► Adding a favorite flight requires confirmation via a dialog	**minor**
8	**Aesthetic and minimalist design**	
	► The short description of a shop provides too little information, and it does not contain a link to detailed descriptions	**minor**

Table 4.16: VIE Heuristic Evaluation

4.2.7 Zurich Airport (ZRH)

Name	Flughafen Zürich / ZRH
Publisher	Flughafen Zuerich AG
Version	1.1.2
Last Update	26.02.2015
Price	free
Evaluated on	19.06.2016

Figure 4.8: ZRH Welcome Screen [5]

Basic Airport Data

Zurich Airport (ZRH) is the largest airport in Switzerland, and with 9,2 km^2, is comparable to Vienna International Airport. The airport area consists of for two terminals, three runways and 194 aircraft stands. In 2015, Zurich Airport (ZRH) managed 36 million PAX and 265,095 movements. It contains 182 check-in counters, and the 74 located airlines fly to 169 destinations in 60 countries. [3]

Criteria Catalog

	Arrival	Departure	Public Transportation Information	Parking Information	Shop and Restaurant Information	Services	Maps	Wi-Fi Information	Airport News	Couponing	Baggage Information	Hotel Information	Weather Information (Local)	Weather Information (Destination)	PRM Service Information	Lounges	Security and Customs Information	Security Check-In Time
Completion	1	1	1	1	1	1	0	0	0	0	1	1	0	1	1	1	1	0
Time (in Sec.)	1	1	2	3	2	2	-	-	-	-	2	2	-	3	2	7.1	2	-
Clicks	1	1	2	3	2	2	-	-	-	-	1	2	-	2	2	3	2	-

	My Flight	Book/Reserve Parking	Indoor Navigation	Check-In	Social Media Function	Push Notification	Sum
Completion	1	1	0	1	0	1	16
Time (in Sec.)	1	10	-	10	-	5	55.1
Clicks	1	4	-	5	-	5	38

Table 4.17: ZRH Application Criteria Catalog

Use Case Analysis

A detailed description of the use case analysis can be found in the Appendix (see Section 9.2.7).

1) Retrieve Arrivals/Departure:
Current arrival and departure flights are shown direct on the home screen. The user can select the desired flight to open the detail view. Alternatively, the permanent bottom menu contains a '*Flights*' link that opens the flight schedule.

2) Retrieve Public Transportation Information:
The user can retrieve public transportation information via the menu entry '*More*'. This view offers an '*Access & Parking*' link, which provides the current schedule for trains and buses.

3) Retrieve Parking Information:
The '*Access & Parking*' view contains a second tab, '*Parking*', which displays the current parking availability.

4) Retrieve Shop, Restaurant and Service Information:
The ZRH application presents shop and restaurant information via the '*Shopping*' menu entry. This leads to a list view where the user can select the desired entry to retrieve detailed information.
To obtain service information, the user must open the airport '*Guide*'. On this view, he/she can select '*Services*'. This opens a list of available services. The user can click on a list entry to open the detail view.

5) Retrieve Lounge Information:
Lounge information is retrievable via the *Service* list on the '*Guide*' view. Users must select '*Day rooms & lounges*', which presents the only lounge entry '*Swiss Arrival Lounge*'.

6) Retrieve the Airport Map:
The ZRH application provides no airport map.

7) Retrieve Favorite Flights Information:
Favorite flights are presented on the home screen. This view contains brief information on the flights. The user can click on a flight to retrieve detailed information.

8) Enable Push Notification:
Push notifications are automatically enabled for favorite flights.

Summary

The ZRH application supports seven out of eight use cases and fulfills 16 of the 24 criteria. To achieve them, 38 clicks and 55.1 seconds are required. Table 4.18 illustrates the summary of the heuristic evaluation, which determined five usability problems; four of them are minor, and one is serious.

1	**Visibility of System Status**	
	▶ The application starts with an empty home screen and without progress information	**minor**
6	**Recognition rather than recall**	
	▶ The *Guide* menu provides no breadcrumbs	**minor**
	▶ On the *Flight details* view, it is possible to click on the airline symbol; this open a new screen with more detail information. The layout of the airline symbol does not appear to be clickable	**serious**
7	**Flexibility and efficiency of use**	
	▶ The flight search offers no input field for date or time it is only possible to search for a destination	**minor**
8	**Aesthetic and minimalist design**	
	▶ Adding a flight to the favorites automatically enables push notifications. An information dialog appears, which requires confirmation	**minor**

Table 4.18: ZRH Heuristic Evaluation

4.3 Discussion

Figure 4.9 presents the evaluation results. The blue bars represent the number of fulfilled criteria; in total, 24 criteria have been rated. The yellow bars represent the sum of measured time in seconds for all tested criteria. The purple bars indicate the necessary amount of clicks in total to accomplish the goal of all tested features.
It is clear that the VIE application requires the most time to achieve all covered criteria – 162.9 seconds – although it covers the fewest criteria. However, using the VIE application requires the fewest number of clicks. This indicates that there is no direct correlation between covered criteria, necessary time and number of clicks. The ZRH application, for example, fulfills 16 criteria, but with 55.1 seconds it is also the fastest one.

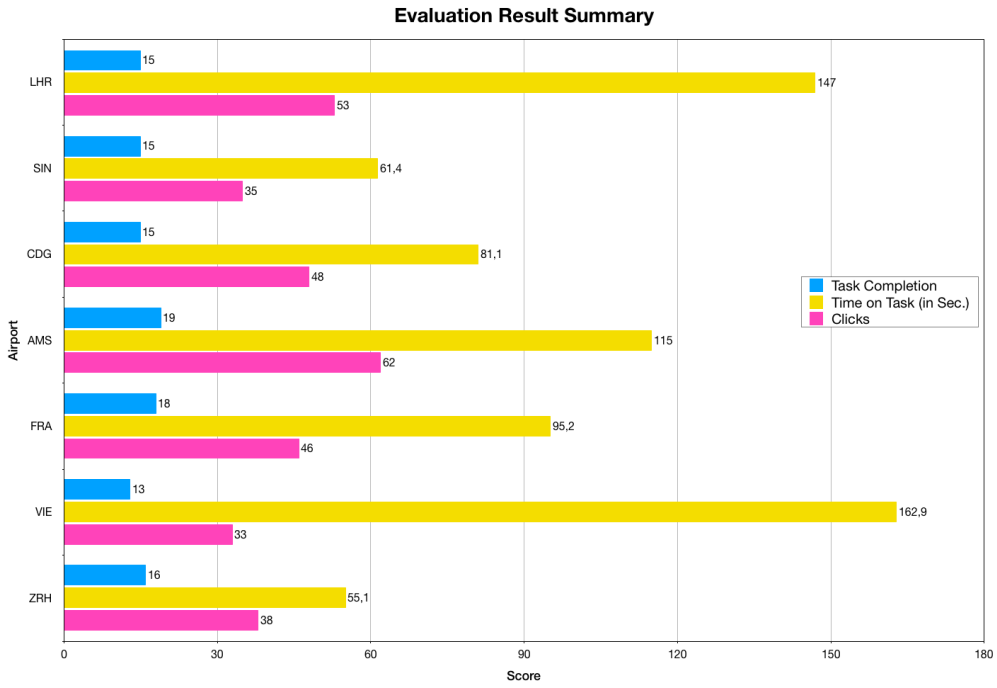


Figure 4.9: Evaluation Result Summary

Figure 4.10 depicts the average time and average clicks per feature. As mentioned, VIE covers the fewest features and requires the most time to achieve the goals: an average value of 12.53 seconds per feature. However, the median is 7.7 seconds and four criteria require more than 20 seconds. This is followed by the LHR application, with 9.8 seconds per feature and an identical median value. In contrast to the VIE application, LHR requires the highest number of clicks to achieve a feature, on average. The SIN application implements 15 features, and it took 61.4 seconds and 35 clicks to accomplish them all. This means an average time per feature of 4.09 seconds and an average number of clicks of 2.47. This indicates that it is important to integrate the implemented features well, and not just implement a wide range of features. Similar results were measured for the ZRH application, which provides 16 features. It is possible to fulfill all of them in 55.1 seconds and 38 clicks. On average, a feature goal could be achieved in 3.44 seconds and with 2.38 clicks. Furthermore, ZRH has the lowest median for the time per feature rating, at just 2 seconds. Two features (*Book/Reserve Parking* and *Check-In information*) require around 10 seconds, which increases the average value.
The AMS application offers 19 features, which is the most out of all of the applications. However,

on average, it took the third longest time to fulfill a feature, at 6.05 seconds. Additionally, it required the second highest number of clicks. The FRA application demonstrates that it is possible to integrated a variety of feature in a comfortable way. This application implements 18 features, achieves the third best value for average clicks, and is just 0.23 seconds slower, on average, than the SIN application. In addition, it offers the second highest median value for feature time, at 3.5 seconds.

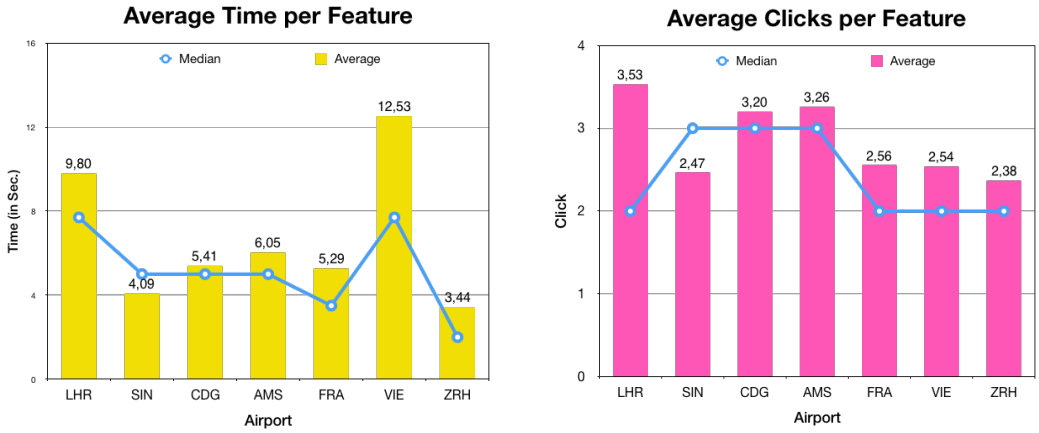


Figure 4.10: Average Time and Clicks per Feature Summary

Figure 4.11 displays the results of the use case analysis. The use cases have arisen from the criteria catalog evaluation and cover those features that are implemented by at least five out of the seven tested applications. The objective is to assess the usability of these features and to discover usability faults of each application. In Figure 4.11, the detected issues are grouped by minor, serious and critical defects.

The LHR application includes 12 usability defects, of which eight are minor. These eight mainly cover inconsistencies, e.g. some views contain a back button, while others do not. In addition, the application has some small design bugs, e.g. the headline of an error dialog is '*http://m.secure.heathrow.com*'. Furthermore, some features are solved rather complicated, shop and restaurant information is only available via the map, and in the parking view, time must be selected via a drop-down element. Regardless of these defects, the desired tasks could be fulfilled. Additionally, the LHRapplication contains four serious usability issues. Three out of these four treat complex workflows and hidden features; for example, the airport map is only retrievable via the flight detail view, and local weather information is only presented in the detailed view of arrival flights. The fourth serious usability problem concerns searching for flight numbers from a different day. This leads to an error dialog instead of an empty results list.
The SIN application fulfills as many criteria as the LHR application, and eight use cases are covered. However, it contains two fewer minor usability problems than the LHR application. Four out of the six detected minor usability issues tackle unusual icons or crowded views, e.g. the '*add to favorite flights*' icon is a heart, and the flight schedule view is overloaded. The four serious usability problems essentially concern hidden information and features, similar to the LHR application. Lounge information can be found under categories such as '*Smoking areas*', '*powder room*' and '*orchid garden*'. Additionally, similar buttons behave differently; for example, the '*Gate*' button marks a gate on the map but the '*Check-in row*' button leads to an advertising website. Furthermore, the flight search is restricted to +/- one day.

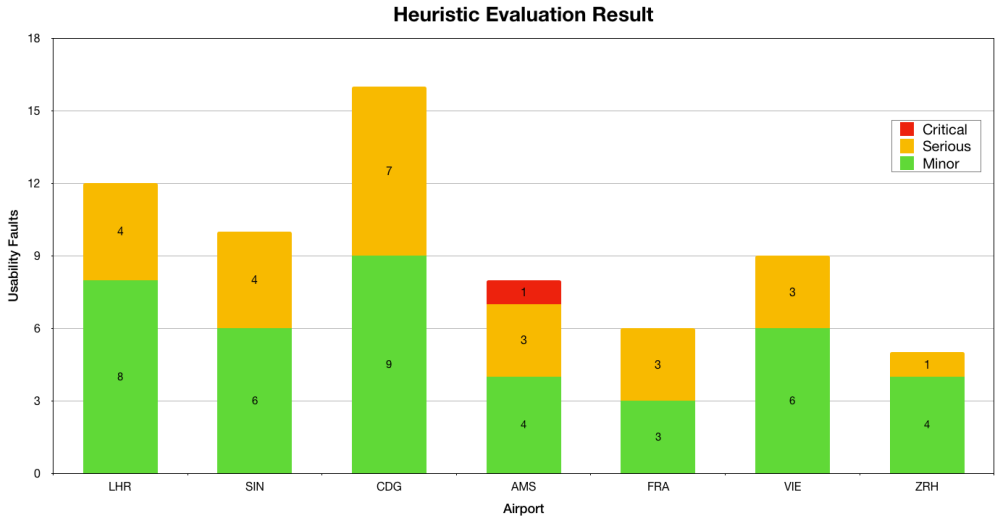


Figure 4.11: Heuristic Evaluation Result Summary

The CDG application contains 16 detected usability issues; thus, it has the most discovered problems. As illustrated in Figure 4.11, nine issues are minor and seven are serious. The minor problems mainly involve unusual icons, confusing naming or missing feedback. For example, the '*add flight*' icon resemble an 'add user' icon, and the '*Formalities*' category in the '*Point of Interest*' menu groups information for gates and terminals. The '*Point of Interest*' view requires some seconds to load, but there is no feedback that the view is loading. The seven serious usability problems mostly concern inconsistency or unexpected behavior. For example, the airport map shows an overview of the airport area, rather than an indoor map; the '*bookmark*' button looks like a toggle button but it is not; and the bookmark for a flight search result adds the search view to the bookmarks instead of the results list.
The AMS application contains, with eight usability issues, the third fewest. Similar to the other applications, the four minor issues involve bad naming and laborious operation realizations. For example, the flight detail view contains a menu entry '*Thinks to do*', which presents a list of bars and shops. Furthermore, the weather information is not presented directly in the flight details; rather, one must click on a link to retrieve this information. The three serious issues concern the hidden side menu, the empty '*At Schiphol*' view and the error that occurs on every registration attempt. The AMS application is the only evaluated airport application for which a critical usability issue was detected. A login via Facebook account always leads to an error dialog with cryptic messages; thus, it provides no help for users.
The FRA app application fulfills 18 criteria, which is the second highest, and with six usability issues it contains the second fewest number of problems. The three minor usability issues concern overlapping labels and missing feedback, e.g. the '*Save for Later*' button is a toggle button but it does not appear as such. The three serious problems involve missing search functionality, unexpected behavior of the navigation, and that the side menu can only be opened on the home screen.
The VIE application fulfills 13 criteria, thus contains the fewest features in the evaluation. Furthermore, nine usability problems could be detected. The six minor issues mainly concern bad naming and laborious task realizations, e.g. the '*Parking possibilities*' view mixes German and English languages. Additionally, a user must provide confirmation when adding a flight to his or her favorite flights. The three serious usability issues cover unexpected behavior and missing functionality. It is not possible to directly undo the addition of a flight to favorite flights; for this, one must open the *My Flights* view and delete the desired flight there. Furthermore, it is not possible to open the flight detail view via the schedule overview.

The ZRH application implements 15 out of the 24 criteria, and just five usability issues were detected. Four of these problems are minor and consist of the splash screen not including progress information, missing breadcrumbs in the '*Guide*' menu, missing search functions and needless confirmation dialogs. The serious usability problem involves missing feedback; the flight detail view provides a link to airline information, but it is not apparent that the airline icon is clickable.

Many of the presented usability problems could be corrected through simple measures. Bad naming and mixing German and English on views are issues that lead to the impression of unprofessionalism. An iterative design process could prevent such minor defects because the GUI would be reviewed more often, which, at the end, would result in a higher quality.

Table 4.19 summarizes the number of clicks necessary for each use case.

	Airport						
Use Case	LHR	SIN	CDG	AMS	FRA	VIE	ZRH
1) Retrieve Arrivals/Departures	2	2	5	2	2	2	1
2) Retrieve Public Transportation Information	4	4	3	2	3	3	2
3) Retrieve Parking Information	9	4	3	3	4	2	3
4) Retrieve Shop, Restaurant and Service Information	6	4	3	3	4	4	5
5) Retrieve Lounge Information	5	3	4	5	2	4	5
6) Retrieve the Airport Map	2	4	2	2	1	2	-
7) Retrieve Favorite Flights information	0	1	0	1	0	0	0
8) Enable Push-Notification	1	2	2	0	0	0	0

Table 4.19: Summary Use Case Clicks

The summary of use case 7 assumes that flights are already added to '*Favorite Flights*'. Use case 8 counts the number of additional clicks to enable push notifications while adding flights. The clicks for use cases 1 - 5 count the steps until the corresponding detail view.

The use case evaluation and the summary presented in Table 4.19 reveal that there are three particularly important features. These tasks could be fulfilled by one or two clicks in almost every assessed application; the use cases are 1, 7 and 8. Apart from the SIN application, every other evaluated application provides the airport map with a maximum of two clicks (use case 6).

Use cases that cover features which potentially generate revenue for the airport generally require at least three or four clicks. This indicates that every evaluated application defines similar main and secondary features. Furthermore, it shows that the airport applications are primarily developed for supporting customers and travelers.
In addition, the evaluation points out that it is important not to neglect the secondary features. For example, the FRA application integrates the variety of features noticeably better than e.g. LHR.
In Chapter 5, the knowledge collected from the assessment is used to develop a prototype that covers the eight use cases but avoids the previously presented faults.

5 APPenger

In the previous chapter, existing airport applications were tested and rated using the defined evaluation form. Both positive and negative aspects could be determined. In the current chapter, these results are used to develop a prototype, called APPenger, which combines the positive approaches of the evaluated applications, implements their main features and thus illustrates a solution that avoids the revealed weaknesses.

First, a user group analysis is conducted to determine potential users. This is then used to evolve personas and the corresponding requirements.
During the user group analysis, eight groups could be identified. They range from private opportunity travelers to frequent business fliers. Physical impairments are considered, such as visually impaired travelers, as well as frequency of smartphone usage. From these eight user groups, six personas are elaborated in Section 5.1.2.
Subsequently, in Section 5.2, the requirements for the prototype are defined. There, the system idea is presented as well: primarily, the application should be a system that supports air travelers through important information regarding their flights. After that, Tables 5.9 and 5.10 list the requirements for APPenger.
In Section 5.3, two designs are developed. They differ essentially in the main menu, which also influences the resulting workflows. Both designs are evaluated with a usability test of potential users in the last section of this chapter. The analysis reveals that both designs achieve exceptional results. The discussion and detailed outcomes are presented in Section 5.5.3.

5.1 Identifying the Users

5.1.1 User Group Analysis

In this chapter, the potential user groups are identified as groundwork for requirements of engineering and design development of the prototype in the following chapters. As specified in ISO 9241 part 210 [59], User Centered Design (UCD) starts with a comprehensive understanding of users' needs and goals. Therefore, it is especially important to identify target users at an early stage of the project to define and develop the functional scope and a corresponding UI layout. Allanwood and Beare define in *User Experience Design* [8] one of the central questions for a UXD process: "Who will use it?" ?" This question refers to the potential user groups. Furthermore, they recommend narrowing the field to 'primary users' during the design phase of a project. As a result, the properties of identified users are defined relatively roughly in this chapter. These properties are then refined and personalized in Section 5.1.2.

McKay in [74] defines properties and criteria to classify different user groups. Maguire and Bevan in "User requirements analysis - A review of supporting methods" [70], define quite similar properties, e.g. general computer skills, task knowledge, user goals, frequency of use, context, age and physical abilities. McKay highlights that the age property is often used to imply some capability or handicap. These assumptions often lead to inexact user descriptions. Therefore, in

the following analysis, the concrete user properties are defined and Austria is selected as 'primary marked'.

In 2015, *Statistic Austria* published in a study [63] the 2014 volume of air travel in Austria. This study revealed that 5.5 million citizens of Austria took, in total, 18.3 million holiday trips, with 35.1% completed by airplane. Furthermore, 1.3 million citizens took a total of 4.1 million business trips, of which 29.3% were completed via airplane. Hence, the two user groups *private traveler* and *business traveler* are identified.
Additionally, the family statistic [90] shows that 2,370,000 families live in Austria, 1,390,000 of them with at least one child. Therefore, the *families with child* user group is identified.
An article on *nachrichten.at* [78] reports that 1.6 million people in Austria between ages 16 and 64 suffer with a disability. The prevalent permanent impairments are agility problems, with 13% of Austrian's population [78]. This is a significant portion; thus, its own user group will represent it.
Moreover, an article [78] revealed that further 3.9% of the citizens are visually impaired. *Der Standard* reports in [93], that, in total, 3.1 million people wear glasses or contact lenses in Austria. Visual disability is a handicap that should be considered during the development of an application design, which is why *visually impaired* impaired is identified as a user group.

So far, five user groups could be identified:

1. Private travelers
2. Business travelers
3. Families with child
4. Agility impaired
5. Visually impaired

Especially important for a user group analysis for airport applications is the general usage of smartphones. *Futurezone.at* published a study [27], which states that 61% of Austrians use a smartphone. As mentioned at the beginning of this chapter, user groups should also be distinguished by their general computer skills and the frequency of use. This results in three further user groups:

6. Technically versed travelers who use their smartphones frequently
7. Travelers who seldom use smartphones
8. Frequent fliers

The properties of the identified users overlap in some cases; for example, a frequent flier could be both a private traveler and a business traveler. These matches are not dissolved here.

5.1.2 Personas

Defining user groups or market segments is not the same as creating personas. A discussion of user categories requires using ranges to summarize the attributes of groups. Allanwood and Beare, in

[8], argue that such statistics are impersonal and it is difficult to take them into account during the design phase. In contrast, a *persona* is based on a person. It is derived from data of the user group analysis to highlight details and important characteristics of a group. Thereby, a story results that is much easier to use and to remember. Thus, it is more likely that they will be considered during the whole design phase and beyond.
In what follows, the user groups that were worked out in the previous chapter are used to define personas for them.

> "Great design requires clear target users" [74, p. 255]

"A persona is a fictional description of a model user based on high-quality user research of actual users in the target user group" [8, p. 114]. The notion of personas was introduced by Cooper and popularized in his book *The Inmates Are Running the Asylum: Why High-tech Products Drive Us Crazy and How to Restore the Sanity* [23]. One big advantage of these technique is that it gives a face to the target users. It provides information about users to the development team in ways that other artifacts cannot.

In Section 5.1.1, eight user groups were identified. In the following section, personas are defined based on these user groups. In addition to Section 5.1.1, the AMS Traffic Review 2015 [32], ZRH Facts and Figures 2015 [3] and the facts and figures 2015 of Frankfurt Airport [6] are used as a foundation for the personas.

The ZRH investigations, as well as the AMS study, reveal that men fly slightly more than women. Thus, three male and two female personas are created. As mentioned in Section 5.1.1, on average, every fourth flight is for family leisure. Therefore, the Schultz family is defined, which consists of two parents and two children (see Table 5.6).
The AMS and ZRH studies have shown that $^1/_3$ of all flights are business trips. To cover this group, Michael is created (see Table 5.1). He is a business consultant and must frequently travel for his job, so he covers the group of frequent fliers as well. ZRH divided the passengers into five age groups: 16-25, 26-34, 35-44, 45-64, 65+. This grouping is also used for the definitions of the personas; for every age group, at least one persona is constructed.
As the youngest persona, at 16 years, Alexander is designed (see Table 5.3). He seldom flies but is very familiar with his smartphone. In contrast to Alexander, as the oldest persona, George is created (see Table 5.5). He is 73 years old and, as age appropriate, is a bit visually impaired; he uses his mobile phone mainly to talk to his family, and he travels frequently since his retirement. Jasmine represents the group of opportunity fliers (see Table 5.2). She flies two to three times a year to visit her family and friends. In addition, she often picks up friends at the airport. To consider the agility impaired passenger group, Stephanie is defined (see Table 5.4).). She has to use a wheelchair and flies approximately one time per year.
The FRA report breaks down the local arrival and departure possibilities. Based on this, for each persona the preferred arrival/departure type is defined.

Name	Michael	**Birthday**	25. February 1965
Marital status	Married	**Occupation**	Business Consultants

Is often on the road because of his job
Flies at least two times per month
His business trips are long distance flights with several stops as well as short non-stop flights
Michael prefers the business class
He often use his smartphone for professional reasons
Prefers to talk to his customers on the phone instead of writing messages
The most frequently used application is the customer relationship management application of his company
Besides for his job he seldom uses a smartphone
Michael has no physical impairments; he regularly trains in a fitness center
In general, Michael travels by private car to the airport and rent a parking space during his journey

Table 5.1: Persona: Michael

Name	Jasmine	**Birthday**	31. May 1983
Marital status	Unmarried	**Occupation**	Student

Source: [13]

Studying Social Work
Moved away from her family to study
Uses her smartphone often for writing messages to her family and friends
She earns some money as a plus-size model
Does not have a boyfriend and it is difficult for her to become acquainted with men
Uses many dating applications
Likes hiking and takes many pictures with her mobile
Flies two to three times a year to visit her family and friends
She is also visited by friends two to three times per year and she picks them up at the airport
Jasmine has no car and uses the public transportation to and from the airport

Table 5.2: Persona: Jasmine

Name	Alexander	**Birthday**	12. November 2001
Marital status	Unmarried	**Occupation**	Pupil

Is in his first year of high school
Alexander is rather shy and withdrawn and has only few friends at his school
Plays many computer games
Found some friends via the Internet and computer games, but he has never seen them in real life
Hears music and writes messages to his Internet friends with his smartphone
Uses the mobile phone often as a pastime
Flies only with his family or on school trips
Flies approximately one times per year

Table 5.3: Persona: Alexander

Name	Stephanie	**Birthday**	18. April 1976
Marital status	Married	**Occupation**	Disability pension

Source: [25]

She suffers from a rare bone disease
Sits in a wheelchair
Flies one time per year to see a specialist
Her other motor skills are also limited
Uses her tablet to read newspapers via her newspaper application and to call friends
Always takes a taxi to and from the airport

Table 5.4: Persona: Stephanie

Name	George	**Birthday**	6. January 1945
Marital status	Unmarried	**Occupation**	Pension

Frequently travels since his retirement
Has a glasses because he is farsighted
Has limited motor skills due to his age
Flies two to three times per year
Occasionally uses the smartphone; he calls his family on his trips
Usually travels by bus to the airport

Table 5.5: Persona: George

Name	Schultz		
Father	Peter	**Mother**	Christine
Daughter	Megan	**Son**	Tom

- The Schultzs fly only for holidays
- They fly approximately every second year
- They prefer long-distance flights for their holidays
- Use their own car to and from the airport

Name	Peter
Birthday	1. September 1978
Marital status	Married
Occupation	Office clerk

- Likes to meet with friends
- Spends much time of his leisure time watching TV
- Slightly clumsy
- Has 'sausage fingers'
- Uses his smartphone infrequently

Name	Christine
Birthday	28. March 1981
Marital status	Married
Occupation	Housewife

- Is primarily occupied by her children and her man
- Occasionally teaches piano
- Uses her mobile phone regularly
- Texts with her family and friends

Name	Tom
Birthday	21. October 2014
Marital status	Unmarried
Occupation	-

- Attends kindergarten
- Is a very clever and open-minded child
- Does not have his own mobile phone
- Likes to play with Christine's mobile phone

Table 5.6: Persona: Schultz Family

5.2 Requirements

Oestereich wrote in his book [88] that it is important to define a system idea at an early stage of the project. Thus, this chapter begins with the system idea of the prototype. Oestereich recommends formulating the system idea like for backside of a finish product packaging. He mentioned that 5 to 20 sentences are enough, but they must be concrete, clear and unambiguous. The system idea should contain system requirements and the main features.
IEEE, in [16], defines the term 'requirement' as follows:

> A requirement is:
>
> 1. a condition or capability needed by a user to solve a problem or achieve an objective
> 2. a condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document
> 3. a documented representation of a condition or capability as in (1) or (2)
>
> – IEEE [16]

Rupp and Group, in [99], define a process model to elaborate requirements for a system. Like Oestereich in [88], they recommend starting with the system idea and defining the scope; they call it *specification level 0*. Rupp and Group define in their book [99] five specification levels, presented in Table 5.7.

Specification-level	Description
0	Describes the idea of the system, the aim and the scope. Could also be called the vision, mission statement or business goals.
1	Describes the use cases and the business processes. Contains functional descriptions and feature lists.
2	Describes the operational concept, the process model. Contains interface requirement specifications.
3	Refine the level 2 requirements and contains technical conditions. This level also contains requirements regarding the architecture of the system.
4	Describes component and module requirements. Contains conditions for parts of the system and technical details.

Table 5.7: Specification Level by Rupp and Group [99]

The process model defined by Rupp and Group is based on a use case analysis. This is also used for this master's thesis. The use cases from the application evaluation in Section 4.2 could be reused for that. The model essentially consists of four steps:

1. Define the system idea and the scope. (Specification level 0)
2. Analyze the system by means of use cases. This will produce the main requirements. (Specification level 1)

3. Refine the main requirements (Specification levels 2 & 3)

4. Architecture step - Define the system architecture and determine the requirements for the different modules. (Specification level 4)

Rupp and Group highlighted that the degree of detail differs on each project. It mainly depends on the desired conditions and requirements that should be specified for the development. In this master's thesis, no requirements on specification levels 3 and 4 are elaborated. The developed prototype provides just HTML-based UI mockups and it focuses on workflow optimization.

5.2.1 System Idea

APPenger is a mobile device airport software that supports air travelers with important information regarding their flights via smartphone or tablet. It contains departure and arrival times, flight detail information and an airport map for orientation. Favorite flights could be saved to quickly and easily relocate them.
In addition, the application provides a public transportation schedule and current parking information. News regarding favorite flights is distributed via push notifications. As a bonus, APPenger supplies information about the offered services of the airport, as well as current information for shops, restaurants and lounges.
The application requires an active Internet connection for current news and information.

5.2.2 Specifications

The functional scope derives from the main features of the previous application evaluation. Table 5.8 displays the summary of the criteria catalog evaluation of these 10 main features.

		Arrival	Departure	Public Transportation Information	Parking Information	Shop and Restaurant Information	Services	Maps	Lounges	My Flight	Push-Notification	Sum
LHR	Time	1	1	14.3	25	9.1	-	9.1	33.3	1	4.8	98.6
	Clicks	1	1	4	10	4	-	2	5	1	5	33
SIN	Time	1	1	5	5	3	-	5	6.3	1	3	30.3
	Clicks	1	1	4	4	3	-	4	3	1	2	23
CDG	Time	10	10	4	4	5	5	3	11.1	2	4	58.1
	Clicks	6	4	3	3	4	3	2	4	2	3	34

AMS	Time	3	4	4	6.3	6.3	10	5	7.1	3	5	53.7
	Clicks	2	3	2	3	3	4	2	4	2	4	29
FRA	Time	1	2	6.3	5	2	2	5	3	1	3	30.3
	Clicks	1	2	3	4	2	2	1	2	1	2	20
VIE	Time	7.7	7.7	16.7	2	20	33.3	7.1	33.3	1	1	129.8
	Clicks	2	2	3	1	4	4	2	4	1	1	24
ZRH	Time	1	1	2	3	2	2	-	7.1	1	5	24.1
	Clicks	1	1	2	3	2	2	-	3	1	5	20

Table 5.8: Summary Feature List

IEEE defines in ISO/IEC 29148-2011 [17] the content and characteristics of a requirement specification. The requirement should be concrete, unambiguous, complete, without contradiction, rated by importance, verifiable, modifiable and traceable. The next section contains the specifications for the APPenger prototype.

Main Requirements

Nr	Description
1	The application must be developed for smartphones and tablets.
2	The application must be able to provide information for one airport.
3	The system must be able to provide flight arrival information to the user.
4	The system must be able to provide flight departure information to the user.
5	The application must offer the option to receive detailed information for each flight. This information contains flight number, flight status, schedule, gate and terminal.
6	The application must be able to provide public transportation information, containing schedule and departure or arrival locations.
7	The application must offers car parking information, which contains parking lot availability.
8	The application must supply shops and restaurants information, including opening hours, location and a brief description.
9	The application must offers information regarding airport services to the user. This information must include opening hours and a brief description of the service.
10	The system must be able to provide information for lounges to the customer, containing opening hours, a brief description and access determinations.
11	The application must contain an airport map.
12	It must be possible to mark flights as favorite flights.

13	The application must offer the opportunity to receive an overview of favorite flights.
14	Information regarding the favorite flights must be distributed via push notifications.

Table 5.9: APPenger Main Requirements

Detail requirements

Nr	Description
1.1	The application must be developed for a portrait format usage.
3.1	The application must provide a view that presents the current arrival flights.
3.2	The arrival flights view should present brief information regarding each flight, containing flight number, arrival time, gate and terminal.
3.3	The user has the ability to search for arrival flights by entering the date.
3.4	The user has the ability to search for arrival flights by entering the origin.
3.5	The search functions for the requirements 3.3 and 3.4 should be combinable.
3.6	The user has the ability to search for arrival flights by entering the flight number.
4.1	The application must provide a view that presents the current departure flights.
4.2	The departure flights view should present compact information regarding each flight, containing flight number, departure time, gate and terminal.
4.3	The user has the ability to search for departure flights by entering a date and time.
4.4	The user has the ability to search for departure flights by entering the destination.
4.5	The search functions for the requirements 4.3 and 4.4 should be combinable.
4.6	The user has the ability to search for departure flights by entering the flight number.
11.1	The map must be an indoor map.
11.2	The map must display shop and restaurant locations.
12.1	The application must offer the ability to add flights to one's favorite flights and to remove them.
13.1	The favorite flights overview must provide the option to open a flight detail view for each flight.

Table 5.10: APPenger Detailed Requirements

5.3 Prototype Design

In this chapter, the UI design of APPenger is developed. Nielsen and Budiu, in [83], recommend starting by identifying the information that should be contained in the application. After that, the information should be classified and grouped by importance. They called the resulting structure information architecture (IA). The following section describes the information categories for the prototype and defines two different architectures. The main difference between IA1 and IA2 is the menu design; IA1 uses a bottom menu whereas IA2 provides a home screen that contains the application menu.
Then, Section 5.3.2 presents the GUI mock-ups and describes the used UI elements used for the different screens.

5.3.1 Information Architecture (IA)

It is important to organize data in a way that makes sense for users. As mentioned, the provided information is structured first.Allanwood and Beare, in [8], also stated that "Users will need to find, retrieve, aggregate, create, edit and contextualize information with ease, and the experience will need to be researched and designed with this in mind" [8, p. 142]. Derived from the findings of the previous chapter, the following nine information categories are identified:

1. Arrival
2. Departure
3. Favorite Flight(s)
4. Public Transportation
5. Airport Map
6. Parking
7. Shop and Restaurant
8. Airport Service
9. Lounge

Due to the limited size of a mobile device screen, Nielsen and Budiu in [83] argue that important information should be presented on primary screens and less important information should be moved to second- or third-level screens. For the prototype, the first five categories are the more important ones and 6 - 9 contain less important information. In the following section, two different IAs are tested. The first one, IA 1, has no central/main view. *Arrival*, *departure*, *favorite flight(s)*, *airport map* and *public transportation* are at the same hierarchical level. Figure 5.1 presents the information structure and Figure 5.2 illustrates the connections between them. It is possible to navigate from one primary view to another. Some information categories are grouped in one view, like *arrivals* and *departures*. This approach is targeted at users who use their smartphones frequently and those who are familiar with typical application structures. It allows faster navigation between the different information categories. It is designed for users such as Michael, Jasmine and Alexander.

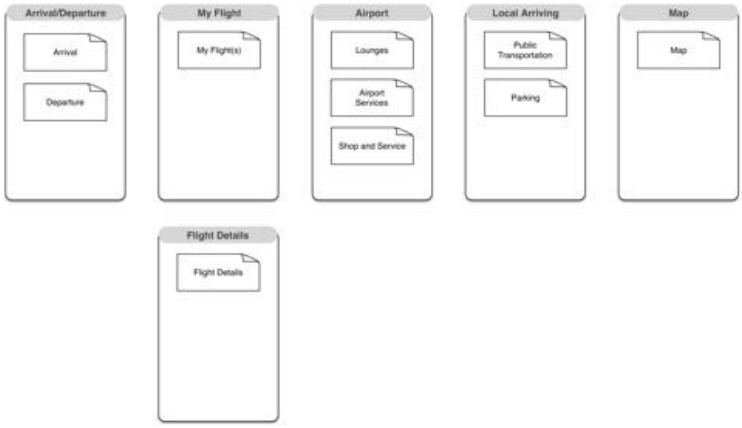


Figure 5.1: IA 1 View Overview

As mentioned in Section 4.3, the use case evaluation revealed that there are essentially three important features, namely *retrieve arrival/departure information*, *retrieve favorite flights* and *push notifications*. The approach of IA 1 ensures that these features and information are retrievable through a maximum of one click.
Like the applications of ZRH, SIN and LHR, IA 1 presents the arrival and departure flights directly after application launch. Favorite flights could be opened with one click.

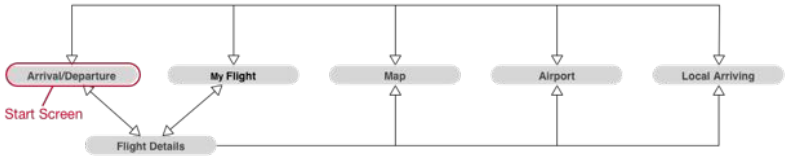


Figure 5.2: IA 1 Connection Overview

The second approach, IA 2, aims at users who seldom use their smartphones. The intention is to support users with stricter workflows. This approach should satisfy users like Stephanie, George and the Schultz family. Each information category has its own view; the home menu view is used to connect all other views. Figure 5.3 presents the different views structured by presented information, and Figure 5.4 illustrates the connections between them. This structure has three hierarchical layers: on top is the home screen, the second layer contains all overviews with one view for each identified information category, and the third layer is the flight detail view.

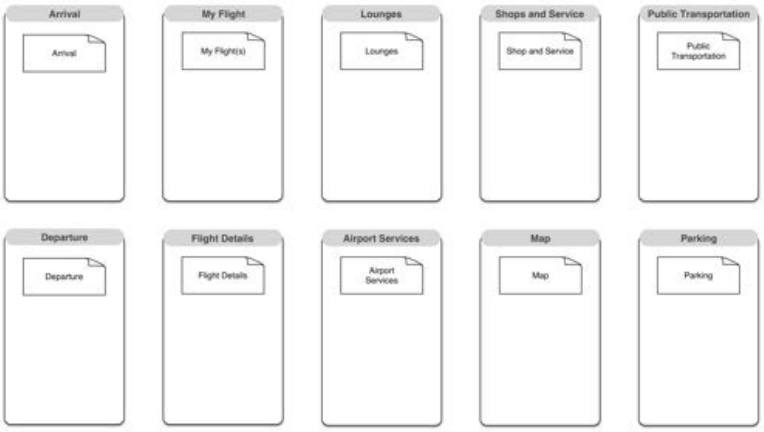


Figure 5.3: IA 2 View Overview

The structure of IA 2 is inspired by the AMS and CDG applications. Both offer links to the main information categories via a central menu view.

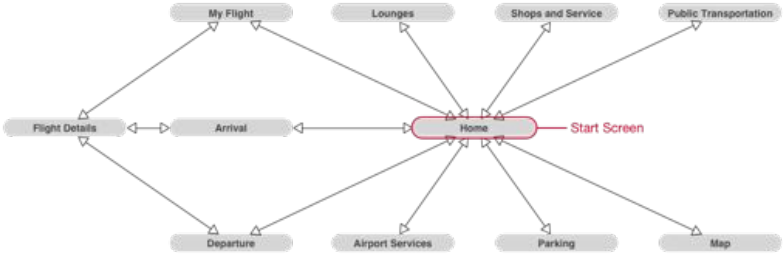


Figure 5.4: IA 2 Connection Overview

Both IA 1 and IA 2 offer fast navigation to all provided features. The main difference between these two approaches is the navigation concept, and thus the information groupings. The following chapter presents initial GUI mockups for both architectures.

5.3.2 User Interface Mockups

Information Architecture (IA) 1

IA 1 structures the information in five top-level views. It uses a tab menu bar, a persistent navigation pattern (see Section 2.2.1), to distinguish these views. The menu bar at the bottom edge allows a quick navigation between the different screens.

Figure 5.5: Home (IA 1)

IA 1 offers the arrival/departure schedule after launching. Current flights are presented in a list view. Each entry contains basic information regarding the flight. For arrival flights, date, time, origin airport, arrival terminal and gate, status and flight number are presented. Additionally, it offers the ability to add desired flights to *favorite flights* via a '+' icon. The list is ordered by date and time. Furthermore, this view provides filter options using the onscreen filter pattern (see Section 2.2.1). The list can be filtered by date, origin/destination airport and flight number. The *Arrival/Departure* view is split up into two tabs (see Section 2.2.1) that can be switched between via the buttons on the top of the screen.

Each arrival/departure list entry is a link to the corresponding flight detail view. This view contains, aside from the basic information of the list view, landing information, flight duration and a weather forecast for the destination airport. Furthermore, the flight status is presented graphically.

Figure 5.6: Flight Details (IA 1)

Figure 5.7: Favorite Flights (IA 1)

The *favorite flight* overview presents the flights in a list view as well. The entries contain the same information as the arrival/departure view, but all favorite arrival and departure flights are presented in one list and an icon distinguishes them. Furthermore, the flight time and destination or origin airport are displayed in a lager font. Past favorite flights are moved to a separate history list. These two lists are split into two tabs (see Section 2.2.1) that can be switched between via the buttons on the top of the screen.

Figure 5.8: Favorite Flights History (IA 1)

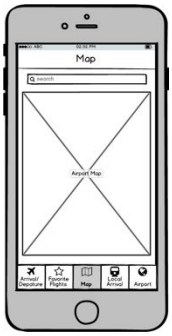


Figure 5.9: Map (IA 1)

The indoor map presents the airport graphically. It offers a full text search function as an onscreen search (see Section 2.2.1).

Figure 5.10: Local Arrival/Public Transportation (IA 1)

The *local arrival* view is divided into two tabs (see Section 2.2.1) that can be switched between via the buttons on the top of the screen.
The *Public Transportation* tab contains a local arrival and departure schedule of trains and buses. Each list entry displays time, local stage, destination or origin station and transportation type. The list is ordered by time. Furthermore, this view provides filter and search options using the onscreen filter pattern (see Section 2.2.1). The schedule can be filtered by date, type and flight number. The flight number filter filters the list by possible public transportation connections matching the corresponding flight schedule.
The second tab, *Parking*, lists the parking areas. The workload is presented graphically by a color-coded bar and free parking lots are displayed textually.

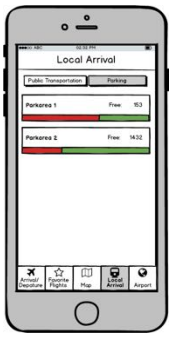


Figure 5.11: Local Arrival/Parking (IA 1)

Figure 5.12: Shops/Bars & Restaurants (IA 1)

Shop & Restaurant, *Service* and *Lounge* is information are summarized in the *airport* view. These categories are divided into a tab menu (see Section 2.2.1) on the top of the screen.
Each tab contains a list of corresponding entries. The entries display name, location and state information. The list is ordered by name. Furthermore, this view offers an on-screen full text search (see Section 2.2.1). Each list entry is also a link to the detailed description of that entry.
The detail screen is divided into three areas. The first area contains the logo, name and a color-coded status. The second area offers opening hours and a link to the map. The third area displays a textual description.

Figure 5.13: Shop details (IA 1)

Table 5.11 illustrates the necessary clicks for the different use cases.

Use Case	IA 1
1) Retrieve Arrivals/Departures	1/1
2) Retrieve Public Transportation Information	1
3) Retrieve Parking Information	2
4) Retrieve Shop, Restaurant and Service Information	1/2
5) Retrieve Lounge Information	2
6) Retrieve the Airport Map	1
7) Retrieve Favorite Flights information	1
8) Enable Push Notification	0

Table 5.11: IA 1 Clicks

Information Architecture (IA) 2

The main difference between IA 1 and IA 2 is the used navigation element. IA 2 uses a springboard (see Section 2.2.1) instead of a permanent tab menu. The springboard is used as a home screen, which is the top-level view in this hierarchy. Each determined information category is represented by its own menu entry. This architecture is more restrictive and the navigation possibilities are strictly predefined.

Figure 5.14: Home (IA 2)

The *home* screen presents a springboard (see Section 2.2.1) to navigate between the different information views.

Figure 5.15: Arrivals (IA 2)

The *arrival flight* view is essentially the same as in IA 1. Current flights are presented in a list view. Each entry contains basic information regarding the flight. It displays date, time, origin airport, arrival terminal and gate, status and flight number. Additionally, each entry provides an icon to add said flight to *favorite flights*. The list is ordered by date and time. Furthermore, this view offers onscreen filter capabilities (see Section 2.2.1) for date, origin airport and flight number.
List entries are links to the corresponding flight details. This view is also the same as for IA 1. It presents, in addition to the basic information of the list, landing information, flight duration and a weather forecast. Furthermore, the flight status is presented graphically.

Figure 5.16: Flight Details (IA 2)

The *departure flight* view looks equivalent to the arrivals, but contains information regarding departure flights.
Favorite flights and *Map* use the same views as IA 1 (see Sections 5.7 and 5.9).

In IA 2, the *Public Transportation* and *Parking* are divided into two separate views. The views themselves look similar to the ones in IA1. *Public transportation* presents a schedule of local arrival and departure trains and buses. Each list entry displays time, local stage, destination or origin station and transportation type. The list is ordered by time. Furthermore, this view provides onscreen filter options (see Section 2.2.1) by date, type and flight number. The flight number filter filters the list by possible public transportation connections matching the corresponding flight schedule.

Figure 5.17: Public Transportation (IA 2)

For IA 2 the *Shop*, *Restaurant*, *Service* and *Lounge* information categories are split up into separate views.

Figure 5.18: Shops (IA 2)

Like in IA1, the shop overview is presented as a list view. Each entry provides name, location and state. The list is ordered by name. Furthermore, this view offers an onscreen full text search (see Section 2.2.1). Each entry is also a link to the corresponding detail view.
The detail view is divided into three areas. The first area contains the logo, name and a color-coded status. The second area offers opening hours and a link to the map. The third area displays a textual description.

Figure 5.19: Shops Details (IA 2)

Table 5.12 presents the necessary clicks for the different use cases.

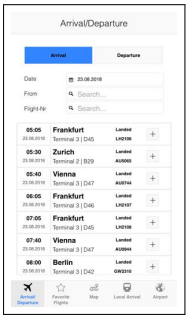
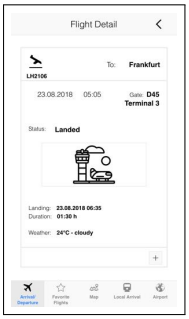
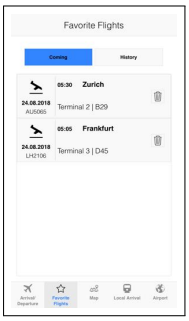
Use Case	IA 2
1) Retrieve Arrivals/Departures	1/1
2) Retrieve Public Transportation Information	1
3) Retrieve Parking Information	1
4) Retrieve Shop, Restaurant and Service Information	2/2
5) Retrieve Lounge Information	2
6) Retrieve the Airport Map	1
7) Retrieve Favorite Flights information	1
8) Enable Push Notification	0

Table 5.12: IA 2 Clicks

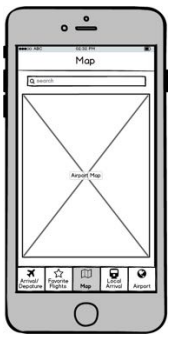
5.4 APPenger - Final Prototype UI

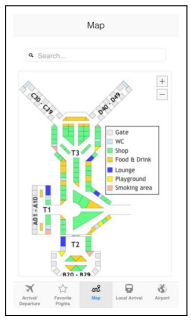
The introduced wireframes of the previous chapter were already quite detailed. Tables 5.13 and 5.14 present screenshots of the final prototype UI. These are shown alongside the wireframes. The finally implemented design is fairly similar to the wireframes.

Information Architecture 1

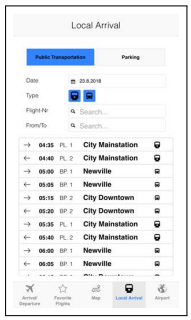
Wireframe	Screenshot	Comment
		Home - In contrast to the wireframe, the final UI is more colored. Furthermore, the arrival and departure buttons are slightly bigger. Besides that, this view and the prototype UI as a whole are unpretentious. Each list entry is a link to the corresponding flight details and it offers the possibility to add a flight directly to the favorites.
		Flight Details - The flight details presents flight states as images; the gate number and terminal are presented in bold font. The weather information is at the bottom edge of the details, rather than in the lower right corner.
		Favorite Flights - The final layout is nearly the same as in the wireframe, but the terminal and gate information are switched. Each list entry is a link to the corresponding flight details and the user can directly remove a favorite flight on this view.

Favorite Flights History - After a favorite flight has landed, it will automatically be moved to the history list, so that the coming list will not be overloaded. On this view, it is also possible to open the details or remove a flight from the favorites.

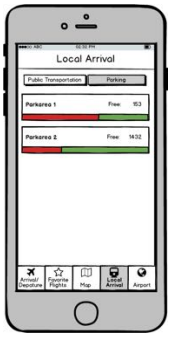


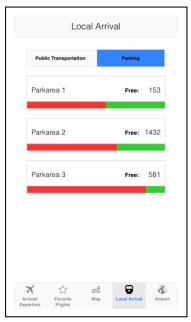


Map - The map view presents an indoor schema of the airport. Gates, toilets, shops, bars and smoking areas are color coded. Furthermore, users can zoom in on the map and search via a search bar at the top of the screen.

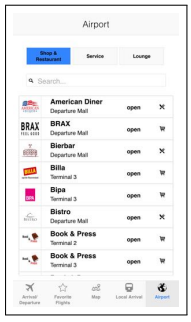


Public Transportation - This view presents the public transportation options. It offers filter options for date, type, flight number and origin/destination. The flight number filter presents connections that match with the flight time so that travelers arrive at the airport early enough for departure flights and they will get the connection for arrival flights.





Parking - This presents the current workload as a color-coded bar and illustrates the concrete number of free parking spots per car lot.



Shop List - This list presents all shops and restaurants at the airport. Users can also switch to the service or lounge list. Each list entry contains logo, name, location and status. Furthermore, each entry is a link to the detail view.

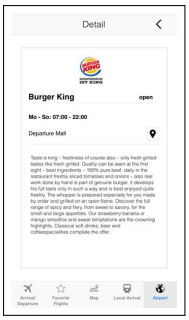
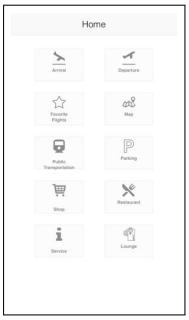
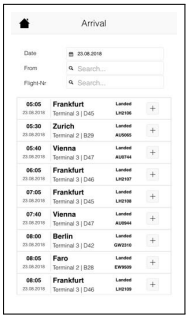
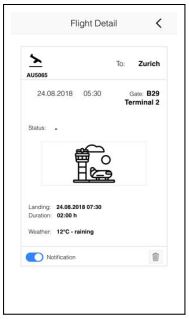
		Shop/Restaurant details - The detail view presents loge, state, opening hours, a brief description and a link to the location on the map.

Table 5.13: Wireframe and Final Prototype UI Comparison IA 1

Information Architecture 2

Wireframe	Screenshot	Comment
		Home - The final home screen looks quite similar to the wireframe; just some icons differ.
		Arrivals - The arrival view of IA 2 is quite similar to IA 1, but there is no menu at the bottom edge; instead, there is a home button in the upper left corner.
		Favorite Flight details - This view presents favorite flight details. These flights have the ability to disable or enable flight notifications and to remove the flights from favorites.

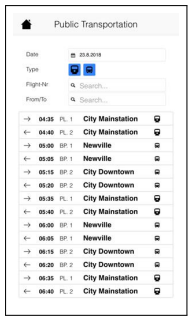
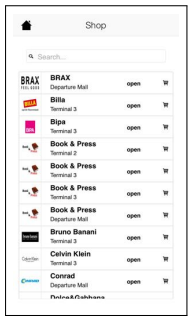
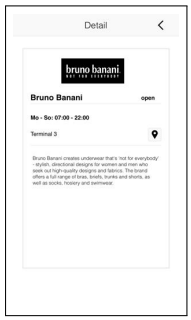
		Public Transportation
		Shop List - In contrast to IA 1, shop, restaurant, service and lounge lists are divided into separate views. Via the search bar at the top of this screen, one can filter by name. Each list entry is a link to the corresponding detail view.
		Shop details - The detail view provides just a back button in the upper right corner to the list view; thus, it is not possible to navigate directly back to the home screen. Furthermore, it offers a link to the location on the map.

Table 5.14: Wireframe and Final Prototype UI Comparison IA 2

Search Features

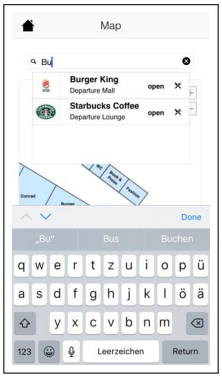


Figure 5.20: IA 2 Map Search

The prototype provides live auto-completion and search suggestions for the search bar on the map view. After selecting the desired result, the map jumps to the corresponding location and presents a brief description.

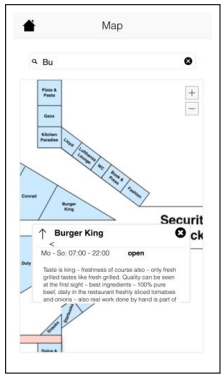


Figure 5.21: IA 2 Map Search Result

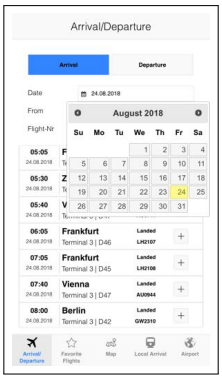


Figure 5.22: IA 1 Date Picker

Each date field offers a date picker to select the desired date. Furthermore, every search input field is a live search, so results are presented immediately . The blue bar at the top of the list represents the loading process; thus, the user gets feedback that the application is still working.

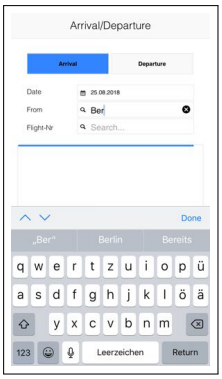


Figure 5.23: IA 1 Arrival Search

5.5 Prototype Evaluation

This chapter consists of the evaluation of the prototype. It describes the test plan and the test execution, and ends with the test results. Wimmer, in [109], argues that the quality of evaluation methods can be assessed using four criteria: completeness, objectivity, reliability and validity. Furthermore, he cautions that the overall quality of a product's usability is difficult to verify. The main cause for this is, according to Wimmer, that completeness and objectivity are nearly impossible to achieve. It is not possible to determine whether all problems are detected, because there is no perfect usability and therefore no perfect evaluation method. On the other hand, objectivity is also difficult to achieve because every classification of errors allows interpretations. Nevertheless, studies have revealed that usability testing has found many real problems reliably.
Wimmer in [109] presents four types of usability testing:

1. Exploratory Testing - is mainly a formative method that can be used early in the design process. The goal of this technique is a qualitative statement to improve the user interface.
2. Assessment Testing - formative and summative form of testing that can be applied during the implementation phase. It is typically used to receive quantitative data (user performance) and to check that the design meets the requirements.
3. Validation Testing - mainly a summative method in which the finished product is tested against standards, definitions and requirements.
4. Comparison Testing - can be used in each phase of the design process to compare different design concepts or to compare it with competitors.

Wimmer, in [110], explains that mobile usability testing is especially challenging because of the technical restrictions of the platform, the ergonomics and the logging on the device.

To determine whether the prototype provides an improvement over the airport applications tested in Section 4.2, the assessment testing technique is used. Twenty-four representative users evaluate the prototype. The objectives of this assessment are to identify usability problems, collect qualitative and quantitative data and to ascertain the participant's satisfaction with the product. Therefore, the test users are divided into two groups, and each group tests one IA. To compare APPenger with the airport applications, participants must fulfill tasks, equivalent to the use cases

in Section 4.1.2. The TOT and the necessary clicks are measured. Furthermore, participants must answer a SUS questionnaire to evaluate their satisfaction. The concrete test plan is presented in Section 5.5.1. Afterwards, in Chapter 6, the better IA design is compared against the evaluation results of Section 4.3. The U.S. Department of Health and Human Services mentions on its website [37] that usability testing is also helpful to identify changes required to improve user performance and satisfaction. So, it is possible to analyze the performance to see whether it meets the usability objectives.

The U.S. Department of Health and Human Services recommends in [38] introducing a pilot test. This is important to test equipment and materials with a volunteer participant. They advise running the pilot test 1-2 days prior the first test session. This allows the facilitator to handle any technical issues, or change the scenarios or other materials. A pilot test offers the opportunity to test the equipment, provides practice for facilitators, conveys a good sense of whether the questions and scenarios are clear to the participants, and allows the facilitator to make any last-minute adaptations.

5.5.1 Test Plan

Scope and Purpose

This section describes the test plan for the APPenger usability test. This test covered both variants IA 1 and IA 2. Therefore, an A/B test was carried out so that half of the participants would test IA 1 and the other half IA 2. The aims of usability testing include determining and validating user performance measures, identifying potential design concerns, and improving end-user satisfaction. Furthermore, the test results should provide a comparison to the evaluation of airport applications presented in Chapter 4.
The U.S. Department of Health and Human Services defines, in their usability test plan template, [40] the following objectives:

- Determine design inconsistencies and usability problems within the user interface. Potential sources of error may include:
  - Navigation errors – failure to locate functions, excessive clicks to complete a function, failure to follow recommended screen flow
  - Presentation errors – failure to locate and properly act upon desired information in screens, selection errors due to labeling ambiguities
  - Control usage problems – improper toolbar or entry field usage
- Exercise the application under controlled test conditions with representative users. Data will be used to access whether usability goals regarding an effective, efficient, and well-received user interface have been achieved
- Show that the users can fulfill all given tasks independently
- Determine the subjective impressions and preferences of the participants to measure end-user satisfaction

Participants

Nielsen in [81] outlines the number of participants needed based on a number of case studies. It shows that just five test participants can find almost as many usability problems as a larger number of test users would. For quantitative studies (aiming at statistics, not insights), Nielsen recommends at least 20 users to obtain statistically significant numbers.
Section 5.1.1 has defined eight user groups and six personas. Due to the limited resources of a master's thesis, the usability test was restricted to the two largest user groups according to the statistics in Section 5.1.1:

1. Private Travelers
2. Technically versed travelers who use their smartphones frequently

Participants were not design experts, because that would distort the test results. As mentioned, both IA 1 and IA 2 were tested using an A/B test. Therefore, the 24 test participants were divided into two groups, so that each architecture was tested by 12 participants. The tests were planned as a *between-subject-design*, so each participant tested only one architecture to avoid learning effects.

Location and Schedule

The usability tests took place in an undisturbed environment and were appointed individually. Each test session was scheduled for 60 minutes. Between two sessions, a break of 30 minutes was planned to reset the environment, to briefly review the session and to introduce a buffer for sessions that may have ended late.

Equipment

The test mobile device was an iPhone 6s with a 4,7" LCD multi-touch display. The display has a resolution of 1334 x 750 pixels by 326 pixels per inch (PPI). The processor of the iPhone 6s is an A9 chip (64-bit architecture, 1.85 GHz) and an integrated M9 monitor co-processor [50].
The back-end server was a Lenovo ThinkPad T470p with a core i7 (64-bit architecture, quad core 2.8 GHz) processor, a 250 GB hard disk and 16 GB RAM.

Scenarios

The U.S. Department of Health and Human Services in [41] notes that "typically, for a 60 min. test, you should end up with approximately 10 (+/-2) scenarios for desktop or laptop testing and 8 (+/- 2) scenarios for a mobile/smartphone test" [41]. As previously mentioned, eight scenarios are defined for the usability test. These scenarios are equivalent to the use cases of the airport applications evaluations in Section 4.2. For detailed information on each scenario, see the description in the Appendix, Section 9.4.3.

S1: Retrieve Arrivals/Departures

S2: Retrieve Public Transportation Information

S3: Retrieve Parking Information

S4: Retrieve Shop, Restaurant and Service Information

S5: Retrieve Lounge Information

S6: Retrieve the Airport Map

S7: Retrieve Favorite Flights Information

S8: Enable Push Notification

Learning effects of participants can distort the test results. To compensate for this, counterbalancing was used [69]. MacKenzie, in [69], states that counterbalancing is performed by presenting the scenarios to participants in different orders:

Test person (TP)	**Scenario order** 1	2	3	4	5	6	7	8
TP1	S6	S7	S4	S5	S8	S3	S1	S2
TP2	S8	S6	S7	S4	S1	S2	S3	S5
TP3	S2	S3	S1	S7	S4	S6	S5	S8
TP4	S7	S4	S5	S1	S6	S8	S2	S3
TP5	S5	S2	S8	S6	S3	S7	S4	S1
TP6	S1	S5	S3	S8	S2	S4	S7	S6
TP7	S4	S8	S2	S3	S5	S1	S6	S7
TP8	S3	S1	S6	S2	S7	S5	S8	S4
TP9	S6	S8	S2	S7	S5	S1	S4	S3
TP10	S7	S6	S3	S4	S2	S5	S8	S1
TP11	S6	S7	S4	S5	S8	S3	S1	S2
TP12	S8	S6	S7	S4	S1	S2	S3	S5

Table 5.15: Scenario Order Definition

This order is given by the Latin Square [30] in Table 5.15 and was used for both the IA 1 tests and the IA 2 tests.

Metrics

Scenario Completion
Each scenario was designed as a typical task for airport applications. Participants needed to enter defined data to obtain the desired results. A scenario was completed when the participant indicated the scenario's goal had been achieved.

Subjective Evaluations
Questionnaires were used to evaluate the subjective impressions of participants regarding ease of use and satisfaction. This was done during debriefing at the conclusion of a session.
In this study, the SUS questionnaire (presented in Section 3.3.2) was used. In addition to that, each participant needed to answer two open questions to determine the best and worst aspects of the software.

1. What do you think is the best aspect of this software?
2. What do you think needs most improvement?

Scenario Completion Time (time on task)
The time to complete each scenario was automatically recorded by the back-end system of the prototype.

Clicks
The necessary clicks to complete each scenario were automatically recorded by the back-end system of the prototype.

5.5.2 Test Execution

Each session of the usability testing was carried out according to the same pattern:

1. Greeting
2. Describing the purpose of the test
3. Hinting that the application will be tested and not the test participants
4. Hinting that the application is a prototype and may contain errors
5. Pre-test questionnaire (demographical data)
6. Presenting and describing the scenario/task list
7. Test execution by the participant
8. Post-test questionnaire (SUS and open questions)
9. Debriefing

The moderating technique of these sessions was planned as a Retrospective Probing (RP). This means that questions about the participant's thoughts and actions were asked after the session was completed. Romano Bergstrom in [96] points out that researchers often use RP in conjunction with other methods, e.g. researcher takes notes and follows up with additional questions at the end of a session.

5.5.3 Test Results

The usability tests were carried out between the 2th February and 18th March 2018. Twenty-four participants evaluated the APPenger prototype. As mentioned previously, the user testing was designed as an A/B test so that half of the participants would test IA 1 and half IA 2. The results are presented below in the respective subsections for IA 1 and IA 2.

Results IA 1

Eleven male testers and one female tester evaluated IA 1. The age structure was as follows: one tester was between 16 and 25, eight between 26 and 35, and three between 36 and 45. Every tester was able to achieve the goal of each scenario. Table 5.16 presents the usability test result of IA 1. It contains the TOT in seconds and the number of clicks that were needed to satisfy the scenarios.

		Test person (TP)												
		TP1	TP2	TP3	TP4	TP5	TP6	TP7	TP8	TP9	TP10	TP11	TP12	Average
S1	Time	15	22	31	42	25	15	33	42	54	31	49	35	32.83
	Clicks	5	10	10	9	8	5	3	8	11	11	14	10	8.67
S2	Time	20	14	46	16	12	15	21	37	77	9	13	20	25.00
	Clicks	4	5	5	5	6	4	5	5	7	4	4	4	4.83
S3	Time	4	3	11	17	13	20	25	2	23	13	19	4	12.83
	Clicks	2	2	2	3	2	2	3	2	3	3	5	2	2.58
S4	Time	15	20	11	21	27	23	10	19	57	80	19	18	26.67
	Clicks	6	3	6	5	3	6	6	4	6	7	7	4	5.25
S5	Time	12	19	6	6	11	16	56	22	47	35	46	63	28.25
	Clicks	6	8	3	3	3	3	8	3	3	7	4	5	4.67
S6	Time	2	9	2	2	4	27	7	3	72	31	8	5	14.33
	Clicks	1	2	1	1	1	1	1	1	1	2	1	1	1.17
S7	Time	5	23	6	26	10	12	2	12	12	15	12	14	12.42
	Clicks	2	7	2	2	2	2	2	2	2	2	1	2	2.33
S8	Time	18	38	17	13	42	54	61	55	75	66	49	35	43.58
	Clicks	5	1	1	3	2	2	2	9	2	2	2	3	2.83

Table 5.16: Usability Test Results IA 1

As mentioned previously, every test participant had to complete a SUS questionnaire. The results of IA 1 are displayed in Table 5.17.

	Testperson (TP)												
	TP1	TP2	TP3	TP4	TP5	TP6	TP7	TP8	TP9	TP10	TP11	TP12	Average
SUS	100	100	95	85	80	92.5	77.5	92.5	97.5	92.5	85	95	91.04

Table 5.17: SUS Results IA 1

In addition to the measured data and the questionnaire, participants had the opportunity to provide feedback on the prototype. Table 5.18 presents the positive aspects of the application, as well as the points that could be improved.

Positive aspects	Improvements
Easy to use, helpful	Favorite flight icon ('star') should be used after a flight is added, rather than the 'flag'
User-friendly, good handling	The 'plus' icon is misleading for adding a flight to the favorites, a 'star' would be better
Simple, easy to understand	The parking information is hard to find

Contains many different options (trains, stores, schedule, etc.)	'Local Arrival' icon should also contain a car symbol
The subdivision of the main menu categories is intuitive	A time filter for the flight schedule would be helpful
Good logical structure	Multilingualism
Simple design	Too much information directly on the home screen
The airport map is a very useful feature	Shop information should contain special offers and coupons
The presentation of arrival and departure flights is well structured and gives a good overview	More color coding e.g for arrival/departure flights
The parking view is clear and easy to understand	Font is too small
Displaying the flight status is very helpful	

Table 5.18: IA 1 Open Question Results

Results IA 2

Five male and seven female testers evaluated IA 2. Three testers were between 16 and 25 years, five between 26 and 35, and four between 46 and 65. Every participant was able to achieve the goal of each scenario. Table 5.19 presents the usability test result of IA 2. It contains the TOT in seconds and the number of clicks needed to fulfill the scenarios.

		Test person (TP)												
		TP1	TP2	TP3	TP4	TP5	TP6	TP7	TP8	TP9	TP10	TP11	TP12	Average
S1	Time	24	20	51	31	66	42	35	37	210	62	74	90	61.83
	Clicks	10	10	23	8	10	8	11	12	10	11	5	5	10.25
S2	Time	22	7	16	18	14	19	14	22	74	10	21	44	23.42
	Clicks	7	5	4	5	4	4	4	4	9	4	4	4	4.83
S3	Time	54	4	2	2	3	8	5	8	10	2	5	7	9.17
	Clicks	3	1	1	1	1	1	1	1	1	1	1	1	1.17
S4	Time	22	6	12	18	16	13	15	14	25	42	100	78	30.08
	Clicks	6	6	5	6	2	7	8	2	2	2	14	2	5.17
S5	Time	5	4	18	16	4	11	8	9	7	39	31	27	14.92
	Clicks	2	2	6	3	2	2	2	2	2	4	2	2	2.58
S6	Time	3	2	2	8	1	7	4	3	10	2	4	6	4.33
	Clicks	1	1	1	1	1	1	1	1	1	1	1	1	1.00
S7	Time	14	5	3	17	25	15	15	2	51	20	114	17	24.83
	Clicks	2	2	1	2	2	2	4	10	2	2	6	2	3.08
S8	Time	36	85	56	39	26	55	32	28	240	68	18	72	62.92
	Clicks	2	14	6	2	2	11	10	4	11	4	2	5	6.08

Table 5.19: Usability Test Results IA 2

Table 5.17 displays the SUS questionnaire results of IA 2.

	Testperson (TP)												
	TP1	TP2	TP3	TP4	TP5	TP6	TP7	TP8	TP9	TP10	TP11	TP12	Average
SUS	97.5	90	95	92.5	82.5	92.5	85	97.5	75	95	92.5	90	90.42

Table 5.20: SUS Results IA 2

In addition to the measured data and the questionnaire, participants had the opportunity to provide feedback on the prototype. Table 5.21 presents the positive aspects of the application in addition to the points that could be improved.

Positive aspects	Improvements
Easy to understand	The add icon for favorite flights should be a 'star', not a 'plus'
Easy to use	It was difficult to recognize how to add a favorite flight
User friendly, good handling	Time filter for flight schedule
Self-explanatory	Font is too small

Very clear menu (home screen)	Color coding for e.g. arrivals/departures
understandable for the elderly	colored icons would be helpful
All necessary information at a glance	Parking information should contain the location of parking lots and the distance to terminals
Flight details are useful	
Parking information is well structured	
'Favorite flights' is a helpful feature	
The used icons are very helpful and make the application easy to use	
Performance	

Table 5.21: IA 2 Open Question Results

6 Result Comparison

The usability tests yielded in excellent results. Bangor, Kortum, and Miller in [15] present a scale for SUS scores. It indicates that a score greater than 85 is an *excellent* result, and the tested system will be highly acceptable to users. Both designs, IA 1 and IA 2, reached a score higher than 90 out of 100. It was possible for every participant to archive the goal of each scenario. Table 6.1 presents an overview of clicks and TOT averages for the different scenarios of IA 1 and IA 2. It is clear that retrieve flight information (S1), retrieve favorite flights (S7) and add favorite flights (S8) using IA 1 were faster, on average, than using IA 2. The optimal path required the same number of clicks for both designs. One reason for that could be that participants of IA 2 used the search functions more often, so for S1 and S8, more clicks were needed. In general, the flight number search function was rarely used.
Retrieving public transportation (S2) was achieved in a similar time and with the same number of clicks for both designs. Using IA 1 to retrieve parking information (S3), one more click was necessary, because it is a sub-view of local arrivals. This is also recognizable in the results comparison. As mentioned, the local arrival icon of IA 1 needs some improvements, because the participants needed more time to fulfill this scenario. It is evident that scenario order played an important role in this case. Participants who tested S2 (retrieve public transportation) before S3 (retrieve parking information) had no problems finding the corresponding view; however, the other way around often led to difficulties.
Using IA 1, retrieve the airport map (S6) took much more time, although participants used nearly the same number of clicks. The optimal path in both designs provided the airport map in one click via the main menu.
For both designs, scenario S4 (retrieve Shop, Service and Restaurant information) was fulfilled in nearly same amount of time. Retrieve lounge information (S5) required around 14 seconds longer, on average, using IA 1. One reason for this could be that lounge information is presented as sub-view of the*airport* menu entry.

		Scenario								
		S1	S2	S3	S4	S5	S6	S7	S8	Average
Time	IA 1	32.83	25.00	12.83	26.67	28.25	14.33	12.42	43.58	24.49
	IA 2	61.83	23.42	9.17	30.08	14.92	4.33	24.83	62.92	28.94
Clicks	IA 1	8.67	4.83	2.58	5.25	4.67	1.17	2.33	2.83	4.04
	IA 2	10.25	4.83	1.17	5.17	2.58	1.00	3.08	6.08	4.27

Table 6.1: Scenario Result Comparison

As mentioned, both designs accomplished excellent results, but Table 6.1 reveals that IA 1 barely prevailed. For SUS, it scored one point more than IA 2, and to fulfill a scenario, participants needed an average of four seconds and 0.2 clicks fewer. Therefore, as the IA with the better result, it was chosen for comparison against the results of the airport application evaluation in Section 4.2. To this end, IA 1 needed to be evaluated in the same way as the airport applications in Section 4.2. Thus, the criteria catalog evaluation for the 10 main features was carried out for IA 1. Here, the

optimal path for the various features were measured three times and the average was calculated. Table 6.2 summarizes the results of the TOT for the main features.

Features	Airport							APPenger
	LHR	SIN	CDG	AMS	FRA	VIE	ZRH	(IA 1)
Arrivals	1	1	10	3	1	7.7	1	1
Departures	1	1	10	4	2	7.7	1	2
Public Transportation	14.3	5	4	4	6.3	16.7	2	1.4
Parking Information	25	5	4	6.3	5	2	3	2.9
Shops and Restaurants	9.1	3	5	6.3	2	20	2	1.3
Services	-	-	5	10	2	33.3	2	2.1
Lounges	33.3	6.3	11.1	7.1	3	33.3	7.1	2.8
Maps	9.1	5	3	5	5	7.1	-	1.5
My Flights	1	1	2	3	1	1	1	2
Push-Notification	4.8	3	4	5	3	1	5	3.3
Sum	98.6	30.3	58.1	53.7	30.3	129.8	24.1	20.3

Table 6.2: Summary Feature List (Time in Seconds)

One reason for the difference from Table 6.1 is that it was the participants' first time using the prototype, and they had no previous opportunity to view the application in more detail. Furthermore, the scenarios are more detailed than the tasks of the criteria catalog, e.g. the arrival/departure scenario fabricates a specific date and flight number, so participants did not just need to open the departure or arrival list like for the criteria catalog evaluation. The same is true for shops and lounges; test participants had to find a specific shop or lounge. Almost everyone used the search bar and opened the corresponding detail view.
The comparison of *Arrivals* and *Departures* reveals that, for APPenger as well as for LHR, SIN, FRA and ZRH, corresponding information was retrievable in one or two seconds, respectively. All of these applications present arrivals or departures directly on the start screen. APPenger achieved the best test results for *Public Transportation*, *Lounges* and *Maps*. The airport map was also positively regarded by participants. Five airport applications offered the favorite flight in one second; these applications directly provide information regarding the next favorite flight via the home screen. That could be a possible improvement for APPenger, but no participant missed the information or perceived the additional click as disturbing. In conclusion, APPenger provides information quickly and overall achieves the best time for all features.

Features	Airport							APPenger
	LHR	SIN	CDG	AMS	FRA	VIE	ZRH	(IA 1)
Arrivals	1	1	6	2	1	2	1	1
Departures	1	1	4	3	2	2	1	1
Public Transportation	4	4	3	2	3	3	2	1
Parking Information	10	4	3	3	4	1	3	2
Shops and Restaurants	4	3	4	3	2	4	2	1
Services	-	-	3	4	2	4	2	2
Lounges	5	3	4	4	2	4	3	2
Maps	2	4	2	2	1	2	-	1
My Flights	1	1	2	2	1	1	1	1
Push-Notification	5	2	3	4	2	1	5	2
Sum	33	23	34	29	20	24	20	14

Table 6.3: Summary Use Case Clicks

A similar picture is drawn by the analysis of necessary clicks for the various criteria. Each one was fulfilled using less or an equivalent number of clicks with APPenger. Table 6.3 presents the criteria catalog click summery compared against the APPenger results.

It is recognizable that the prototype has been optimized for these 10 features. In total, it is clear that less clicks are needed to fulfill all of these tasks – in the extreme case, compared to CDG, 20 clicks less. Even the ZRH application, which achieved similar results for the time comparison, requires a total of six more clicks to accomplish all feature goals, although the ZRH application does not provide an airport map. APPenger and the tested airport applications achieve similar results for *Arrivals*, *Departures*, *Maps* and *My Flights*; all of these features are retrievable in one or two clicks in nearly every application. Other features, such as *Shops and Restaurants*, *Services* or *Public Transportation*, are not efficiently integrated in the tested airport applications, and more clicks are necessary to achieve them.
However, it is important to note that the evaluated applications in Section 4.2 partly offer much more functionality than the developed prototype. It is clear that a multitude of features could not be integrated in an application as well as the 10 features of the prototype. Still, during the usability tests, no feature was missed by the participants. They requested just some extensions to the provided information: 1. the parking view should offer more details and 2. the shop or restaurant detail view should present current offers or coupons.
The comparison of APPenger and the evaluated airport applications has revealed that even the first prototype achieved high usability results and significantly better ratings for time and clicks. This demonstrates that the presented process model for airport applications could lead to measurable improvements and will improve usability and the UX.

6.1 Airport App Design Guidelines

The airport application analysis revealed that the majority of tested applications use similar patterns, and they also provide the same basic set of features. Furthermore, the APPenger prototype and its evaluation revealed potential improvements. Table 6.4 presents guidelines that are extracted from the results and findings of these evaluations or elaborated from the literature.

No.	Guideline	Description and Examples
	External Influences	
1	Consider the variety of mobile devices	Different mobile OSs, computational power, display resolutions and screen sizes will affect the usability. These aspects must be considered during the design and development process.
2	Consider the environment	It is important to regard the surrounding environment. Airport applications are typically used in artificial light and in a hurry. These factors influence the colors, icons and font sizes.
	User related	
3	Be tactful	Avoid disturbing the user, e.g. by advertisement or multiple unnecessary notifications.

4	Provide user feedback	After user actions, offer feedback about the current state; e.g. introduce a progress bar instead of a loading icon.
5	Consider impaired users	A wide range of users will potentially be physically or visually impaired. These users have different needs for application usability, like certain button or font sizes.
6	Determine the user's needs	It is important to design an application for the user's needs. That includes the functional scope, layout, workflow and context-relevant wording.
	Function scope	
7	Avoid splash screens	Even airport applications must load a lot of data. They should avoid splash screens; instead, partially loading and updating data in the background is preferable.
8	Implement the main features	Airport applications should provide the following features: arrival and departure information, public transportation and parking information, an indoor airport map, information regarding shops, restaurants, services and lounges, a favorite flight section and push notifications.
9	Offer filter and search capabilities	The list views present a large amount of data. Therefore, it is important that users can search for information and filter the provided data.
10	Provide favorite flight information via the home screen	The presented evaluations have shown that favorite flights information on the home screen improves usability.
11	Avoid needless functions	Avoid integrating a variety of unnecessary features to keep the application smart and usable.
12	Offer personalization	To improve the UX, customization features are recommended, such as the favorite flight function or also color, font and button size options. This supports impaired users as well.
	Design	
13	Use known UI elements	Use common and known UI elements to increase learnability and user satisfaction. Furthermore, avoid unknown or application-specific gestures; they will likely go unused or be forgotten.
14	Design clear, structured views	Do not overload the views. Design a clear layout and use overview lists that link to corresponding detail views.
15	Avoid deep navigation hierarchies	A hierarchical depth of 2-3 layers should not be exceeded to ensure the effective usage.

16	Use color coding	The layout should consider color coding to improve learnability and user efficiency. For example, use different colors for arrivals and departures or color the various information categories differently.

Table 6.4: Airport app Design Guidelines

The overall objective of these guidelines is to provide optimal usability and UX. They are divided into four categories: *external influences*, *user related*, *functional scope* and *design*. The first two are more related to the analysis and planning phase of a project, while the last two are more related to the concrete application development, including feature and layout recommendations. Some of the presented guidelines are also important for other application topics; user feedback, using known UI elements and considering different devices are essential to the usability of any application.
Guidelines 4 and 14 are crucial for airport applications because the main task of such an application is essentially to present a large amount of data in a clear manner. The airport application analysis and the prototype evaluation have shown that presenting an overview list with a link to the corresponding details is an often used and user-friendly pattern to present such large amounts of data. Furthermore, filter and search options for these lists are essential to improve usability.
As illustrated in Chapter 4, almost all tested airport applications implement the 10 presented main features; they are also included in the guidelines (No. 8).

7 Discussion

One of the main assumptions of this thesis was that current UIs of airport mobile applications have usability weaknesses and could be improved. Many studies have confirmed that current mobile application UIs often contain usability faults. Furthermore, they also agree that a good user experience, and consequently a correspondingly well-designed UI, are important factors for the success of an application (see e.g. [11, 12, 35, 21]).
This thesis has presented a process model with concrete techniques to improve the UX for airport applications and developed an appropriate prototype, called APPenger. This chapter examines the APPenger usability test results and discusses the findings in the context of related scientific work.

As previously mentioned, both APPenger designs, IA 1 and IA 2, achieved excellent scores. The feedback of all test participants was consistently positive; the high usability and clear design were particularly praised. Both IAs were perceived as easy and intuitive, and the logical structure was comprehensible for participants. Furthermore, some test participants highlighted the *airport map*, *favorite flights* and *parking information* as especially useful features.
Nevertheless, the usability tests also revealed improvement potential. The small font size was mentioned several times, and following this, the arrivals and departures lists were noted as slightly overloaded. It can be assumed that different users will perceive the optimal font size differently. Therefore, personalization settings would be helpful. Kascak, Rébola, and Sanford in [61] describe universal design principles. Their approach essentially considers more user feedback by sound or editable font and button sizes. The universal design principles are very similar to the mobile UI designs presented in Section 2.2, but they are more precise in terms of feedback and user settings. The current prototype version contains no user settings, but it seems suitable for that issue.
Another point that was criticized by some participants was the '*add favorite flight*' icon. APPenger uses a 'plus' icon, whereas some test users expected a 'star' icon instead. Further possible improvements include color coding for e.g. arrival and departure flights or a time filter for public transportation and flights.
In addition, *parking information* was discovered as improvable. It should contain more details such as the distance to terminals. Moreover, current offers and coupons should be integrated in the shop details as a refinement.
Especially for IA 1, the local arrival menu icon was not meaningful enough and should be extended by a 'car' or 'parking' symbol. As mentioned in Chapter 6, some test users had problems finding the parking information. The results revealed that the scenario order was important in this case. Participants who had to retrieve public transportation before retrieving parking information had no problems finding the corresponding view; the other way around often led to difficulties.

In addition to these improvement possibilities, the previous chapter has shown that the developed prototype could prevail against established airport applications regarding the necessary number of clicks and the time needed to achieve task goals. Both are measurements for the criteria of efficiency, which are defined in the ISO/IEC 25010 standard [56] as important characteristics for usability. However, the ISO/IEC 25010 standard points out that additional criteria such as user satisfaction and effectiveness are also significant for the perception of usability. Haaksma, Menno, and Karreman in [34] analyzed the relation between usability and User Experience (UX) from the

user's perspective. They discovered that regarding usability, participants focused more on effectiveness and efficiency than on satisfaction and regarding UX, they paid much attention to the way that the product worked. Therefore, it is necessary to understand users' interpretation of these concepts and what they perceive as high usability and good UX to develop a suitable application.

In Section 6.1, 16 airport application design guidelines were presented. These contain suggestions regarding the environment, functional scope, user-related suggestions and design proposals. It is clear that they are not only applicable to airport applications. Some concepts, such as 'provide user feedback', 'consider the environment', 'use known UI elements' and 'design clear structured views' are regularly mentioned in the literature; see e.g. Nielsen's *10 Usability Heuristics for User Interface Design* [80]. Others are also inspired by related work. As mentioned, Kascak, Rébola, and Sanford in [61] indicated the creation of a universal design. They highlight features such as 'impaired users' and 'personalization', as well.
In contrast, other guidelines arose from the airport application evaluation, such as 'implement the main features', 'avoid needless functions' and 'avoid deep navigation hierarchies'. Furthermore, additional suggestions such as 'use color coding', 'provide favorite flight information via the home screen', 'offer filter and search capabilities' and 'avoid splash screens' directly arose from the usability tests of APPenger. In conclusion, the guidelines developed from the various assessments are more airport application related, whereas guidelines elaborated from the literature are applicable to mobile UI development in general.

7.1 Comparison to Related Work

Similar to this thesis, Baillie and Morton in [14] highlight that the development of a mobile application that considers design principles yields better usability results than ignoring them. Baillie and Morton used simple Human Computer Interaction (HCI) design principles in their paper: constraints, consistency, affordance, visibility and feedback. Just like in this thesis, two different UI designs were developed, and the resulting prototypes were compared through a use case evaluation and SUS questionnaire. The procedure of the use case evaluation was identical to the usability testing in Section 5.5. To avoid learning effects, they varied the task order for different participants. Similar to the results of this master's thesis, the prototype that considered HCI design principles achieved better ratings. They assumed that it was because of "focusing on the task to be completed and assessed this rather than the aesthetics of the design" [14, p. 107].

As previously mentioned, Haaksma, Menno, and Karreman in [34] analyzed the relation between usability and UX from the user's perspective. They illustrated that the exact relation between both concepts is quite unclear in the literature. For their study, they introduced a grid with four quadrants (low usability/low UX, high usability/low UX, low usability/high UX, high usability/high UX). Participants had to place products about which they had rather strong feelings into this grid. The results could lead to the assumption that usability and UX are strongly related. However,14% of the products were placed in the high usability/low UX and 11% in the low usability/high UX quadrants. This indicates that both concepts are relevant for the product's quality but need not necessarily be related. In the debriefing, the most commonly mentioned relationships were that "usability can influence UX or that UX consists of many different qualities, including usability" [34, p. 128].
Both the concepts of and relations between usability and UX were also discussed in this thesis, but from a literature point of view. It is necessary to understand users' interpretation of these concepts

and what they perceive as high usability and good UX.

Equivalent evaluation techniques for a different application topic were used in the paper of Alkhafaji et al. [7]. They implemented a smart and ubiquitous learning environment with respect to cultural heritage. Alkhafaji et al. provided an application and wearables to facilitate the learning process. Similar to the present thesis, they combined questionnaires, observation and a brief interview to determine the results. Furthermore, 26 participants tested their solution. What is interesting is that, although it was a completely different topic, the findings of the application evaluation were similar to the ones of this master's thesis. For example, they found that the surrounding environment should not be neglected, users' knowledge, especially regarding technology and smartphone usage, could obstruct the experiences, and the differences of the various devices are complex to handle during the development of an application. In the end, Alkhafaji et al. provided a list of guidelines for designing mobile location-based learning services. These guidelines centered on the users and should lead to an enjoyable learning process. They also contained user feedback, personalization, usability and the surrounding environment, like the presented guidelines in this thesis (see Section 6.1).

The comparison with other scientific research has shown that many other authors have identified the usability problem in the field of mobile applications as well. It has also shown that several potential process models and guidelines were developed to improve this situation. Furthermore, some advanced scientific papers and articles cover new concepts. Nevertheless, especially in the field of mobile airport applications, usability faults are still common. The usability test result comparison has shown that the illustrated process model of this master's thesis could lead to significant improvements regarding efficiency and effectiveness. Even the first prototype draft achieved better results than most tested airport applications.

7.2 Limitations and Future Work

As mentioned, APPenger is limited to the main features of airport applications. These features were determined by a sample size of seven tested applications. It remains to be seen whether these features will fulfill the needs of all users or if they should be extended. Analyzing further airport application could yield in an adaption of the main features. Owing to the limited resources and time limitations of this master's thesis, further evaluations were not possible. Because of technical development of mobile devices in recent years, their computational power has increased immensely. Complex and computationally intensive tasks could also be done on smartphones. Especially in the field of indoor navigation, large deficits are recognizable. In the future, improvements are expected. Augmented reality, for example, would be possible so that navigation and special offers or advertisement of shops could be combined. The amount of advertisement in an airport application was not investigated in this thesis, but it should be for such extensions.
As discussed, refinements of input dialogs, as well as the automatic adaption of them according to the environment, are currently being researched. The integration of such techniques would also be helpful for airport applications, and similar improvements are expected in the coming years.
As previously mentioned, the introduced process model and the presented techniques were defined for airport applications, but they seem to be applicable for other application developments as well. This should be researched through future work. Furthermore, the defined airport application design guidelines are the first in that topic. Follow-up research is necessary to confirm and improve them.

In the future, developers and project leaders will likely notice the importance of solid process models in improving user experience and satisfying users' needs. The cognition on UI design is changing and evolving. As demonstrated, automated UI and usability approaches are in research. It is assumed that future professional mobile applications will consider usability design during the development process. Therefore, significantly better UI designs are expected for mobile applications in general and for airport applications in particular.

8 Conclusion

At the beginning of this master's thesis, in 2.2, the term '*design*' was illuminated more closely and different definitions were presented. It was demonstrated that design is not a kind of art, but rather a graphical communication skill to explain tasks to users.

After that, commonly used design patterns and graphical user interface (GUI) elements were introduced. Both Apple and Google provide development guidelines for these user interface (UI) elements. They are the market leaders for mobile operating systems (OSs); most smartphones use Google's Android or Apple's iOS. It was pointed out that optics is not the only factor for successful mobile applications. The whole user experience (UX) must considered. The UX starts from when users hear about an application and includes the installation and usage. It is more based upon memory than reality. If the memory of a product is great, many incidents can be excused [85].

In this thesis, a process model for airport applications was introduced to achieve high usability and a great UX. It illustrates specific techniques, starting in the analysis phase up to the first application design draft. This process model was exemplarily applied for the development of APPenger, a tactful passenger airport application. Competitor products on the market were evaluated during the analysis phase. Therefore, a criteria catalog, including 24 criteria and features, was developed (see Table 8.1). The catalog evolved by aggregating all features of the various tested airport applications. They were rated by efficiency (clicks) and Time on Task (TOT).

	Arrival	Departure	Public Transportation Information	Parking Information	Shop and Restaurant Information	Services	Maps	Wi-Fi Information	Airport News	Couponing	Baggage Information	Hotel Information	Weather Information (Local)	Weather Information (Destination)	PRM Service Information	Lounges	Security and Customs Information	Security Check-In Time	My Flight	Book/Reserve Parking	Indoor Navigation	Check-In	Social Media Function	Push Notification
Completion																								
Time																								
Clicks																								

Table 8.1: Criteria Catalog

Based on the criteria catalog evaluation, 10 main features could be determined. They are implemented by nearly every tested application. As aforementioned, the criteria catalog evaluation measured the necessary time and clicks to achieve a task goal. However, these values are not meaningful regarding usability. To assess the usability of these main features, the following eight use cases were developed:

1. Retrieve Arrivals/Departures
2. Retrieve Public Transportation Information
3. Retrieve Parking Information
4. Retrieve Shop, Restaurant and Service Information

5. Retrieve Lounge Information
6. Retrieve the Airport Map
7. Retrieve Favorite Flights Information
8. Enable Push Notification

All competitor applications were tested according to these use cases and rated by a heuristic created by Nielsen (see Section 4.1.3). Some of the evaluated airport applications contain serious usability weaknesses. In each application, 5 to 16 usability faults were detected.

The first research question (Research Question (RQ)1) could be answered after the analysis phase: *Which criteria are important for an airport app? / What are appropriate approaches, and in which way could an app provide support for travelers?*
As mentioned, the features that are implemented by almost every evaluated airport application are identified as the most important ones. These include flight arrival and departure information, parking and public transportation details, an indoor airport map and information regarding shops, restaurants, services and lounges. It is also important to allow travelers to save their favorite flights and thus retrieve corresponding details quickly. These functions offer information regarding flights and the airport to travelers quickly and easily. The usability tests of APPenger, which implemented these exact features, revealed that potential users did not miss further functionality. While evaluating the various airport applications, it was clear that functions that could potentially generate revenue, such as advertisements or shop information, are gently integrated.

Figure 8.1 presents the answer to RQ2: *How do existing airport apps fulfill the criteria catalog?* It is the summary of the application evaluation in Chapter 4. One can see that each application fulfills between 15 and 19 out of the 24 criteria. Although the Amsterdam Airport Schiphol (AMS) application offers the most functionality, it also required the greatest number of clicks and the third longest time to achieve the goal of all tested tasks.

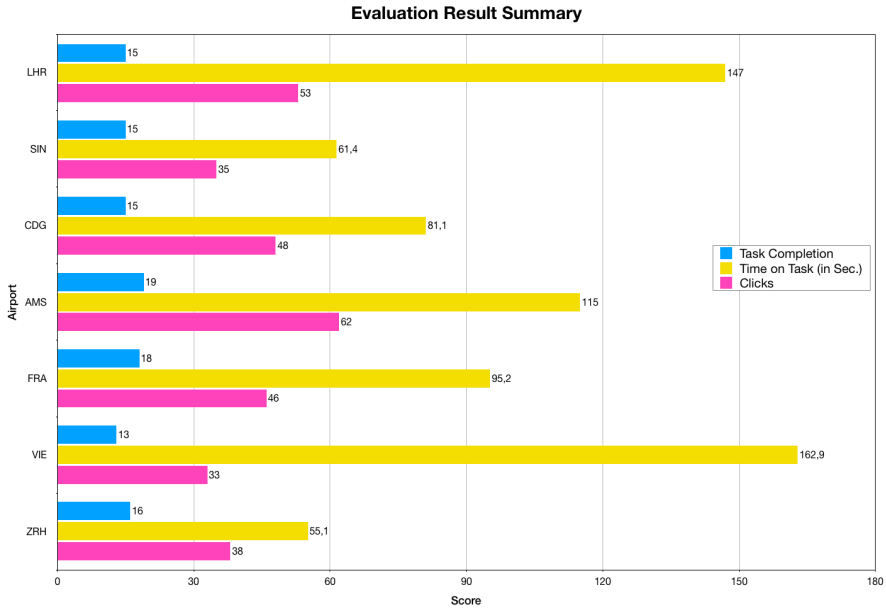


Figure 8.1: Airport App Evaluation Result Summary

The reason for that is that AMS neglected proper usability for several features. As mentioned, good user experience and a well-designed UI are crucial to the success of an application. Various usability weaknesses could be determined while evaluating the airport applications. For example, every application started with a splash screen without presenting the progress. Other typical problems included bad wording and unnecessary deep information hierarchies. The Zurich Airport (ZRH) application presented shop, restaurant and service details in a hierarchical list view. For some entries, it was necessary to navigate through four layers.
On the other hand, some positive aspects and well-constructed designs were discovered. For example, the applications of London Heathrow Airport (LHR), Frankfurt Airport (FRA) and Vienna International Airport (VIE) directly offered information regarding the next favorite flight via the home screen. This allowed a quick navigation to the corresponding details. Multiple well-designed approaches could be extracted and reused for the prototype development.

The findings were used in Chapter 5 to prototypically develop an airport application. This prototype demonstrates how an application could look that fulfills the main features of the criteria catalog, and thus presents an answer for RQ3: *What would an app that fulfills the criteria catalog look like?* The development process started by identifying potential users. To this end, a user group analysis was carried out and corresponding personas were defined. Based on the evaluation findings and the developed personas, two different designs evolved, which should support travelers at airports.
The development of APPenger followed a user-centered design process: it began by creating a brief concept, which contained the functional scope and objective of the project. Afterwards, users and their needs, including the tasks, were identified. During the next step, concepts regarding the UI design as well as the whole UX were elaborated. After that, the initial design and different creative approaches were implemented. The prototypical designs were used for a user-centered evaluation, followed by design selection and refinement. Further evaluations and refinements were carried out until satisfactory results were achieved. Finally, developed software was deployed and continuously improved.
This thesis applied the described process until the first usability evaluation. Additionally, the improvement suggestions were captured and discussed in Chapter 6. Refinements and further iterations were not performed due to the limited resources of a master's thesis.

Two different design approaches were developed. They differ especially for the menu layout; Information Architecture (IA) 1 offers a permanent menu at the lower edge, whereas IA 2 includes a separate menu screen. Figure 8.2 presents screenshots of both home screens for the developed prototype designs. The objective of APPenger was to present information regarding flights to users in a quick and compact manner. Search functions should help to find desired flights faster. Adding and storing favorite flights is another important feature. Furthermore, an indoor airport map and detailed information regarding shops, restaurants, services and lounges provide a comprehensive overview of options at the airport. Public transportation and parking information are also included.

APPenger implements the 10 main features previously discussed. The workflows are optimized in a way that all detail views can be achieved by a maximum of three clicks. To open overviews or lists, a maximum of two clicks are needed.

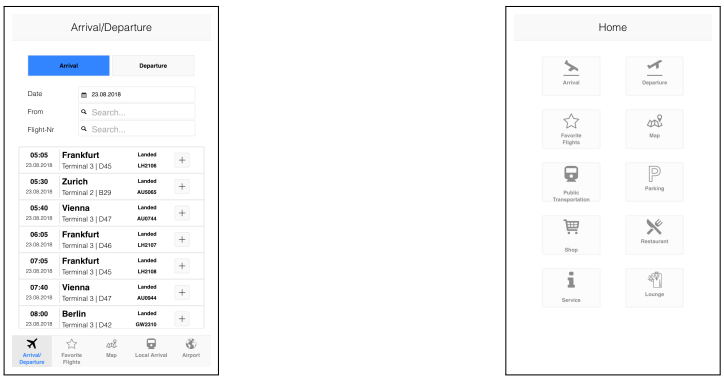


Figure 8.2: Screenshot Home Screen IA 1 (Left) and IA 2 (Right)

Both designs were evaluated through a usability test with 24 participants. The test was designed as an A/B test so that 12 participants tested IA 1 and the other half IA 2. Each test participant assessed one design. The presented use cases were also used as scenarios for the usability test. That offers the ability to compare the two prototype designs against each other and to compare the prototype against evaluated airport applications of the market analysis. In addition, participants rated the prototype designs, as part of the usability test, using a System Usability Scale (SUS) questionnaire. Both designs achieved excellent results – over 90 out of 100 points – in which IA 1 just prevailed over IA 2.
Even the first draft of the prototype achieved excellent results. In comparison to the evaluated airport applications, it accomplished the best ratings for time and clicks.

Based on the application evaluations and the usability test results of the prototype, 16 airport application design guidelines could be extracted. They were divided into four categories: *external influences* (1 - 2), *user related* (3 - 6), *functional scope* (7 - 12) and *design* (13 - 16).

1. Consider the variety of mobile devices
2. Consider the environment
3. Be tactful
4. Provide user feedback
5. Consider impaired users
6. Determine the user's needs
7. Avoid splash screens
8. Implement the main features
9. Offer filter and search capabilities
10. Provide favorite flight information via the home screen
11. Avoid needless functions
12. Offer personalization
13. Use known UI elements
14. Design clear, structured views

15. Avoid deep navigation hierarchies
16. Use color coding

There are currently no other airport application design guidelines, making the guidelines proposed in this work the first in that topic. Follow-up research is necessary to confirm and improve them. The presented process model and the corresponding results have shown that a solid market analysis offers a significant advantage to the development of new mobile applications. Heuristic evaluations and use case analysis could quickly discover usability weaknesses in competitor products. This knowledge can be used to avoid the recurrence of such faults. It was also shown that it is important to determine potential users and to satisfy their needs. APPenger offers less functionality than other evaluated airport applications, but usability test participants were sufficiently satisfied. The defined criteria catalog and the elaborated use cases were specifically developed for airport applications, but, in general, the presented process should be applicable to the development of mobile applications regarding other topics as well. Furthermore, the implementation of an HTML-based prototype is recommended to test complex workflows and tasks for almost every development of new mobile applications. It can be used to provide a usable application quickly and enable discussion of workflows and usability aspects. Moreover, refinements and improvements could be integrated easily.

9 Appendix

9.1 Complete Criteria Catalog

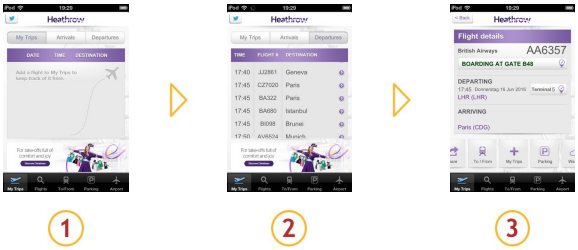
Airport		Arrival	Departure	Public Transportation Information	Parking Information	Shop and Restaurant Information	Services	Maps	Wi-Fi Information	Airport News	Couponing	Baggage Information	Hotel Information	Weather Information (Local)	Weather Information (Destination)	PRM Service Information	Lounges	Security and Customs Information	Security Check-In Time	My Flight	Book/Reserve Parking	Indoor Navigation	Check-In	Social Media Function	Push Notification	Sum	Average
LHR	Completion	1	1	1	1	1	-	1	-	1	-	1	-	1	1	-	1	-	-	1	1	-	-	1	1	15	
	Time	1	1	14.3	25	9.1	-	9.1	-	4	-	2	-	7.7	7.7	-	33.3	-	-	1	25	-	-	2	4.8	147	9,80
	Clicks	1	1	4	10	4	-	2	-	1	-	1	-	2	2	-	5	-	-	1	12	-	-	2	5	53	3,53
SIN	Completion	1	1	1	1	1	-	1	1	1	1	1	1	-	-	-	1	1	-	1	-	-	-	-	1	15	
	Time	1	1	5	5	3	-	5	7.7	1	2	5	9.1	-	-	-	6.3	6.3	-	1	-	-	-	-	3	61.4	4.09
	Clicks	1	1	4	4	3	-	4	3	1	2	2	3	-	-	-	3	3	-	1	-	-	-	-	2	37	2.47
CDG	Completion	1	1	1	1	1	1	1	1	-	-	1	-	-	-	1	1	1	-	1	-	-	-	1	1	15	
	Time	10	10	4	4	5	5	3	5	-	-	5	-	-	-	5	11.1	3	-	2	-	-	-	5	4	81.1	5.41
	Clicks	6	4	3	3	4	3	2	3	-	-	3	-	-	-	3	4	2	-	2	-	-	-	3	3	48	3.20
AMS	Completion	1	1	1	1	1	1	1	1	-	1	1	1	1	1	-	1	-	-	1	1	-	1	1	1	19	
	Time	3	4	4	6.3	6.3	10	5	10	-	3	5	6.3	1	7.7	-	7.1	-	-	3	4	-	16.6	7.7	5	115	6.05
	Clicks	2	3	2	3	3	4	2	4	-	2	3	5	1	4	-	4	-	-	2	4	-	5	5	4	62	3.26
FRA	Completion	1	1	1	1	1	1	1	1	1	1	-	1	-	-	1	1	-	1	1	1	1	-	-	1	18	
	Time	1	2	6.3	5	2	2	5	5	1	4	-	6.3	-	-	6.3	3	-	3	1	14.3	25	-	-	3	95.2	5.29
	Clicks	1	2	3	4	2	2	1	4	1	2	-	3	-	-	3	2	-	1	1	5	7	-	-	2	46	2.56
VIE	Completion	1	1	1	1	1	1	1	-	-	-	1	-	1	-	-	1	1	-	1	-	-	-	-	1	13	
	Time	7.7	7.7	16.7	2	20	33.3	7.1	-	-	-	25	-	1	-	-	33.3	7.1	-	1	-	-	-	-	1	162.9	12.53
	Clicks	2	2	3	1	4	4	2	-	-	-	5	-	1	-	-	4	3	-	11	-	-	-	-	1	33	2.54
ZRH	Completion	1	1	1	1	1	1	-	-	-	-	1	1	-	1	1	1	1	-	1	1	-	1	-	1	16	
	Time	1	1	2	3	2	2	-	-	-	-	2	2	-	3	2	7.1	2	-	1	10	-	10	-	5	55.1	3.44
	Clicks	1	1	2	3	2	2	-	-	-	-	1	2	-	2	2	3	2	-	1	4	-	5	-	5	38	2.38

Table 9.1: Complete Criteria Catalog

9.2 Use Case analysis

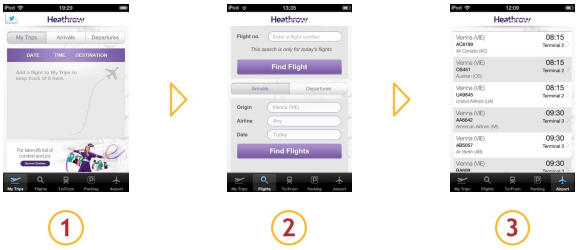
9.2.1 London Heathrow Airport (LHR) Use Case analysis

1) Show Arrivals/Departures



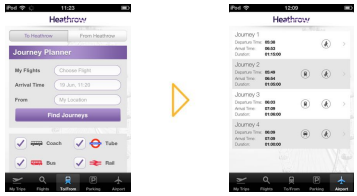
1. The LHR app provides the current arrival and departure schedule as a tab on the home screen.
2. Click *Departures* to open the departure timetable.
3. Click on a specific flight to open the flight detail view.

1) Show Arrivals/Departures - Alternative



1. Alternatively, click on *Flights* at the bottom menu.
2. Select an origin and click on *Find Flights*
3. This view presents a list of arrival flights, containing, inter alia, flight number, airline and terminal.

2) Show Public Transportation Information

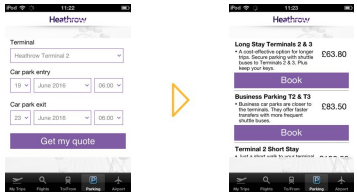


1 2

1 Public transportation information are available via the *To/From* menu entry at the bottom bar.

2 Select time and origin to get the possible journeys.

3) Show Parking Information

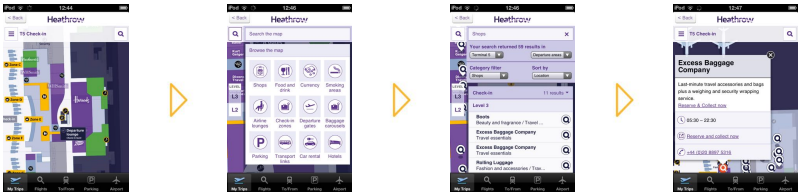


1 2

1 Receive the parking information by clicking the *Parking* menu entry.

2 Edit the duration and get the parking fee information.

4) Show Shop, Restaurant and Service Information - Shops



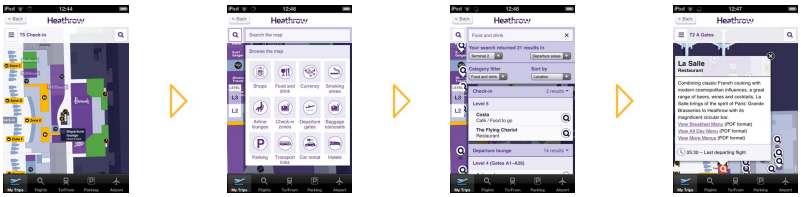
1 Shop information are only available via the airport map.

2 At the upper right corner the map provides a search function. Select *Shop* to get the shop list.

3 Choose the desired shop.

4 The selected shop will be marked on the map with a short description.

4) Show Shop, Restaurant and Service Information - Restaurants



1 Search for restaurants is only possible via the airport map.

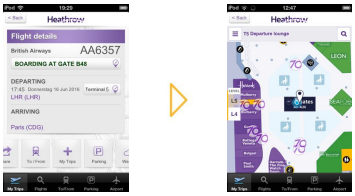
2 Open the search field at the upper right corner.

3 Select *Food and drink* to get the restaurant and bar list.

4 The chosen restaurant will be shown on the map with a brief description.

4) Show Shop, Restaurant and Service Information - Services
The LHR app provides no special service information.

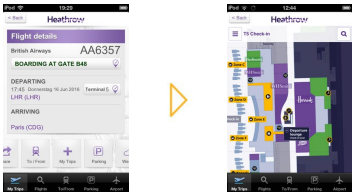
5) Show Lounge Information



1 The LHR app offers the lounge information via the flight detail view.

2 Click on the *Terminal* button to get the location of the lounge.

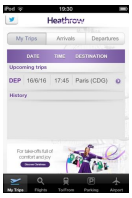
6) Show the Airport Map



1 The map is only available via the flight details.

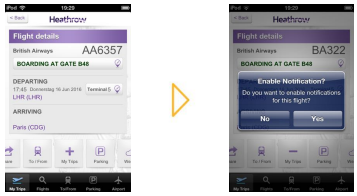
2 Click on the map icon to open the map.

7) Show Favorite Flights



1 The LHR app presents the favorite flights direct at the home screen.

8) Enable Push-Notification for flights



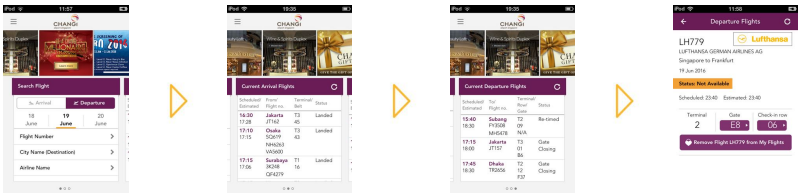
1 2

1 While adding a flight to *My Trips* via the '+' icon a confirm dialog occurs.

2 On this dialog it is possible to enable the notifications.

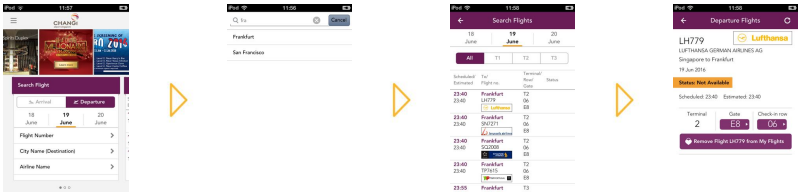
9.2.2 Singapore Changi Airport (SIN) Use Case analysis

1) Show Arrivals/Departures - Arrivals



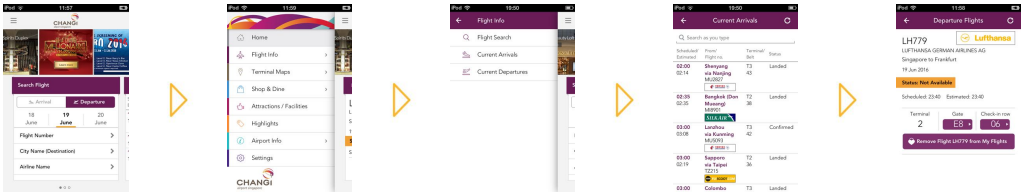
1. The SIN app presents a search form as home screen. Wipe it to the left to see the current arrival flights.
2. Swipe it one more to get the current departure list.
3. Select a flight to open the flight details.
4. This view contains information like flight number, airline and baggage belt.

1) Show Arrivals/Departures - Alternative 1



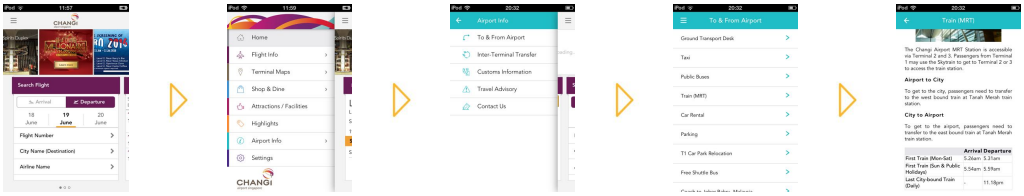
1. Click on *City Name (Destination)* to search for a flight.
2. For example enter 'fra' as search term and select *Frankfurt* as destination.
3. Select a flight to open the detail information.
4. The flight detail view provides the same information as the compact view at the home screen.

1) Show Arrivals/Departures - Alternative 2



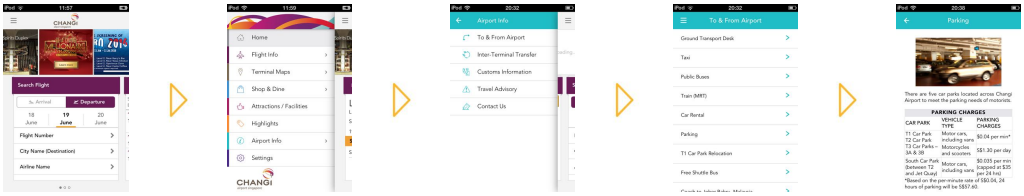
1. The SIN app provides a sidebar menu.
2. Select *Flight Info* to choose between *Current Arrivals* or *Current Departures*.
3. Click on *Current Arrivals* to open a list of currently arrival flights.
4. Select a flight to open the detail information.
5. The flight detail view.

2) Show Public Transportation Information



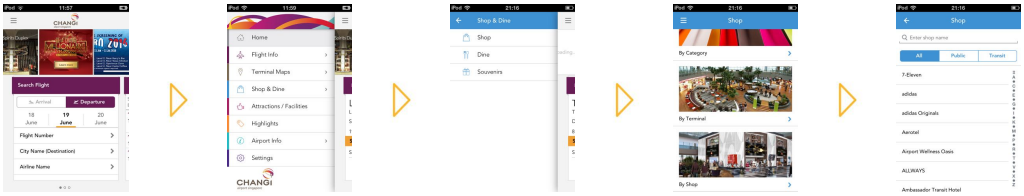
1. The public transport information are available via the sidebar menu.
2. Select *Airport Info*.
3. Click on *To & From Airport*.
4. This view contains all information categories regarding public transportation, taxis and car parks.
5. Click on *Train (MRT)* to open the MRT station information.

3) Show Parking Information



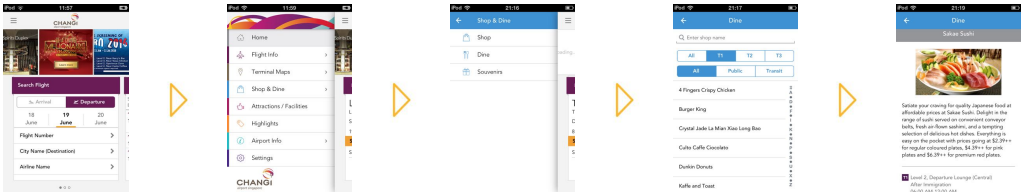
1. The public transport information are available via the sidebar menu.
2. Select *Airport Info*.
3. Click on *To & From Airport*.
4. This view contains all information categories regarding public transportation, taxis and car parks.
5. Click on *Parking* to open the car park information.

4) Show Shop, Restaurant and Service Information - Shops



1. Shopping information are available via the sidebar menu
2. Click on *Shop & Dine*.
3. Select *Shop* to open the grouping option view.
4. This screen provides three grouping options: *By Category*, *By Terminal* and *By Shop*.
5. Click on *By Shop* to open the shop list. Select a shop to see shop details.

4) Show Shop, Restaurant and Service Information - Restaurants



1 Shopping information are available via the sidebar menu

2 Click on *Shop & Dine*.

3 Select *Dine* to open the grouping option view. This view contains the same grouping options like for shops.

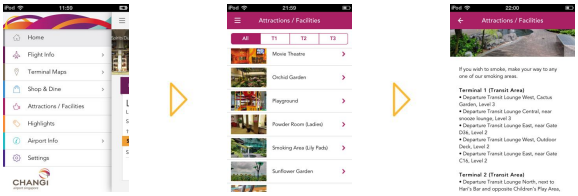
4 Click on *By Terminal* to open the restaurant list grouped by terminal.

5 Select a restaurant to see detail information.

4) Show Shop, Restaurant and Service Information - Services

The SIN app provides no special service information.

5) Show Lounge Information

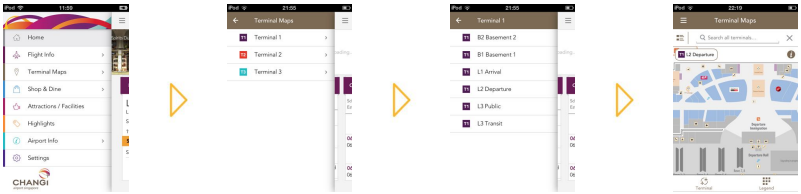


1 Select *Attractions/Facilities* on the sidebar menu.

2 Click on *Smoking Area(Lily Pads)*.

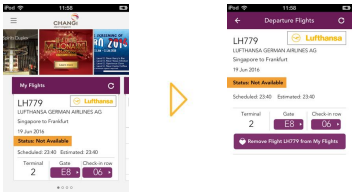
3 This view contains a list of lounges and their locations.

6) Show the Airport Map



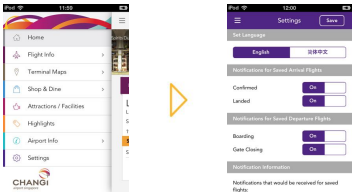
1. The airport map is retrievable via the sidebar menu.
2. Click on *Terminal Map* to get a selection for the three terminals.
3. Select *Terminal 1* to see the different areas within the terminal.
4. Click on *L2 Departure* to open the overview map of that.

7) Show Favorite Flights



1. Added flights are presented at the home screen in a card layout.
2. The flight detail view provides the same information as the compact view at the home screen.

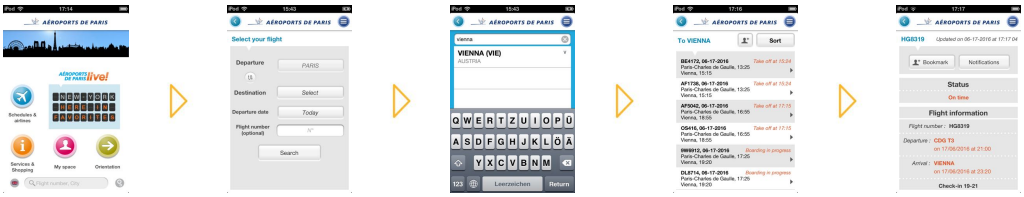
8) Enable Push-Notification



1. Push-Notifications are automatically enabled for added flights.
2. The SIN app provides a setting menu where different notification types could be enabled or disabled.

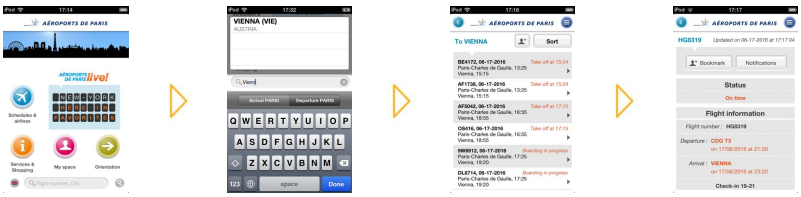
9.2.3 Aéroport Paris-Charles de Gaulle (CDG) Use Case analysis

1) Show Arrivals/Departures



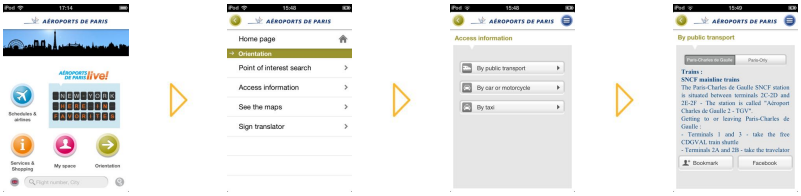
1. Open the *Schedule & airlines* view via the icon on the home screen.
2. Select *Schedule* to open the search view.
3. Click on *Destination* to select the destination.
4. This will open a list view of corresponding flights.
5. Click on a specific flight to open the flight details.

1) Show Arrivals/Departures - Alternative



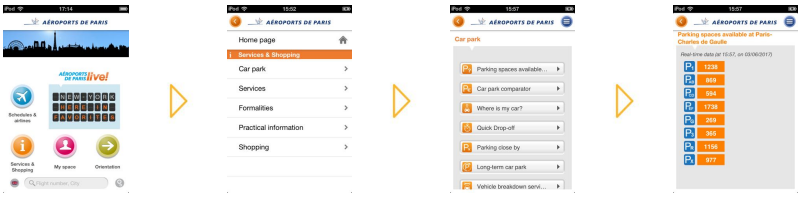
1. The CDG app provides direct at the bottom of the home screen a text search field for flights.
2. It is possible to search for flight numbers or cities of departure or arrival flights.
3. Searching for a city will open a list view of corresponding flights.
4. Click on a specific flight to open the flight details.

2) Show Public Transportation Information



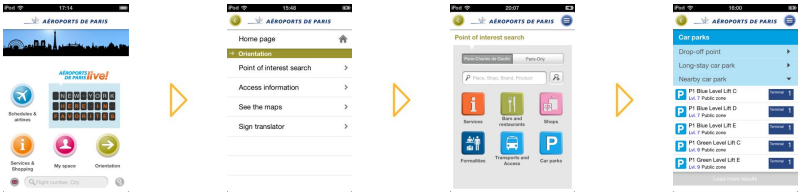
1. The access information are available via the *Orientation* icon on the home screen.
2. Click on *Access Information* to open another list view of access possibilities
3. Select *By public transport* to open the detail view.
4. CDG provides public transportation information in a flowing text style.

3) Show Parking Information



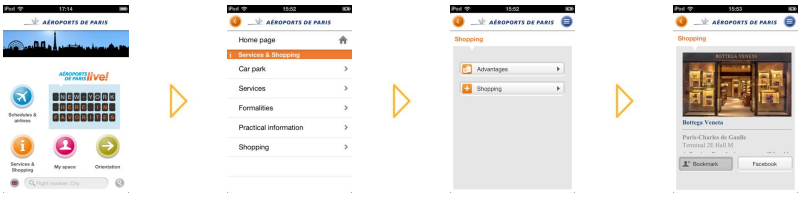
1. Open *Service & Shopping* to navigate to the *Car park* list entry.
2. Click on *Car park* to open the view regarding parking information.
3. Select the desired information category to open the detail view.
4. The entry *Parking space available* show the number of free parking lots for each car park.

3) Show Parking Information - Alternative



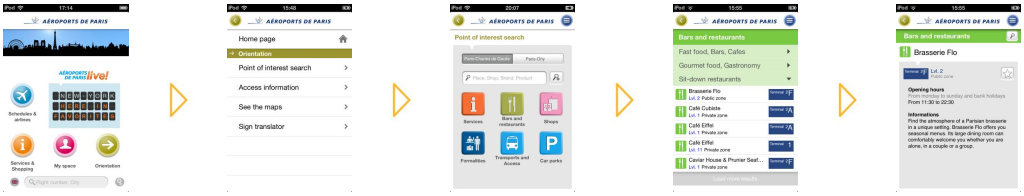
1. Alternatively, the CDG app offers parking information via the *Orientation* icon.
2. Click on *Points of interest search* to open the search overview.
3. Select the *Car parks* icon.
4. This screen provides an overview of the car parks locations.

4) Show Shop, Restaurant and Service Information - Shops



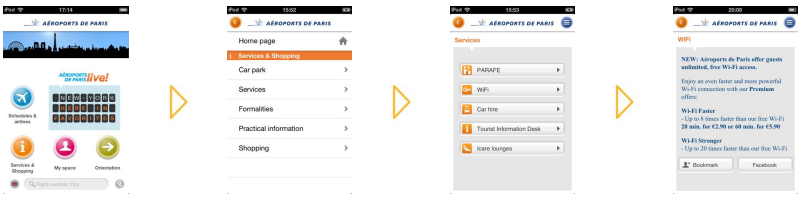
1. Shopping information are available via the *Services & Shopping* icon.
2. Click on *Shopping* to see the possible information categories.
3. Select *Shopping* to open the shop list.
4. This screen provides detail information for each shop, containing the location and a short description.

4) Show Shop, Restaurant and Service Information - Restaurant



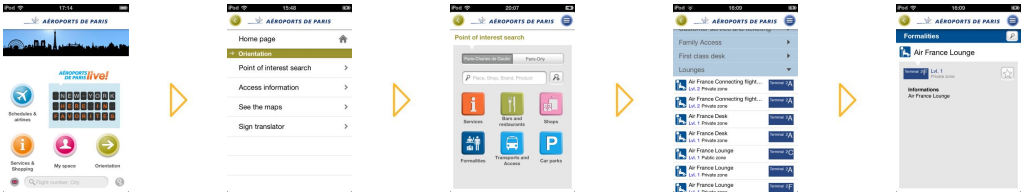
1. Restaurant information are available via the *Orientation* icon.
2. Click on *Point of interest search* to open the search view.
3. Select *Bars and restaurants* to open the list of restaurants and bars.
4. Choose the desired bar or restaurant to see the details.
5. This screen provides detail information, containing opening hours, location and a brief description.

4) Show Shop, Restaurant and Service Information - Services



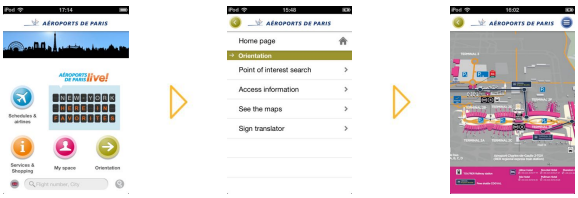
1. Service information are available via the *Services & Shopping* icon.
2. Click on *Services* to see the possible information categories.
3. Select *WiFi* to open the detail view.
4. This screen provides detail information for the CDG WiFi offers.

5) Show Lounge Information



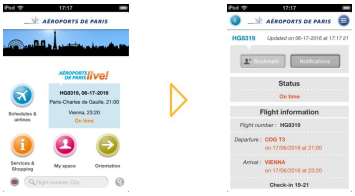
1. Lounge information are available via the *Orientation* icon on the home screen.
2. Click on *Point of interest search* to open the search view.
3. Select *Formalities* via the icon.
4. This view contains a category *Lounges*.
5. Choose a specific lounge to open the detail view.

6) Show the Airport Map



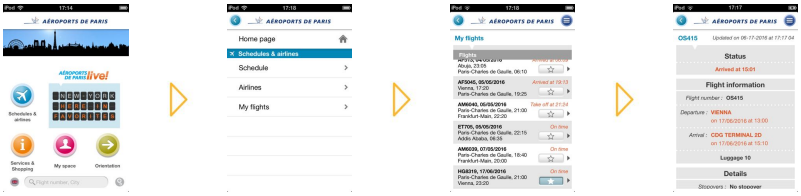
1. To navigate to the map view click on the *Orientation* icon on the home screen.
2. Select *See the maps* to open the airport overview map.
3. The CDG app provides only an overview of the airport area without much detail information.

7) Show Favorite Flights



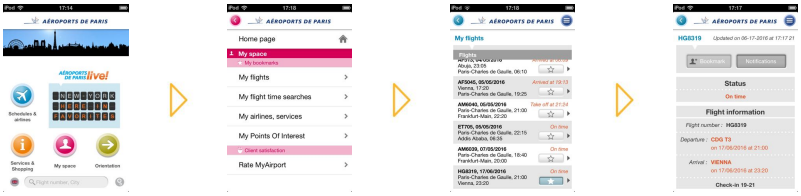
1. The CDG app provides the possibility to add one favorite flight to the home screen.
2. Click on the flight to open the flight details.

7) Show Favorite Flights - Alternative 1



1. Alternatively, click on the *Schedules & airlines* icon at the home screen.
2. That opens a list view which contains an entry *My flights*.
3. Select *My flights* to show the list of all added flights. That flight which is picked on the home screen is marked with a star.
4. Click on the desired flight to see the flight details. This view contains also the baggage information.

7) Show Favorite Flights - Alternative 2



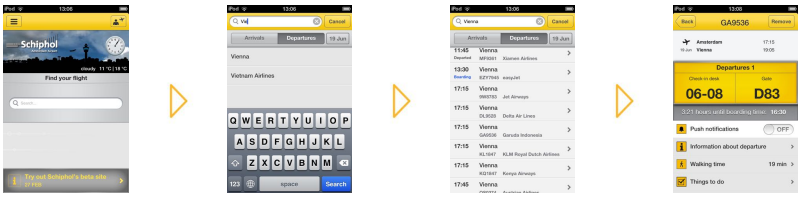
1. The third possibility to get information for favorite flights is, to click on the *My space* icon at the home screen.
2. This view shows all categories of bookmarks. Click on *My flights* to open the list of added flights.
3. Select a flight to see the details.
4. The flight detail view.

8) Enable Push-Notification

1. Every flight detail view contains a toggle button to enable or disable the notifications.
2. It needs approximately three seconds to enable the notifications for a flight, during this time the CDG app shows a loading dialog.
3. After the notifications are activated the flight is automatically added to *My flights*, too.

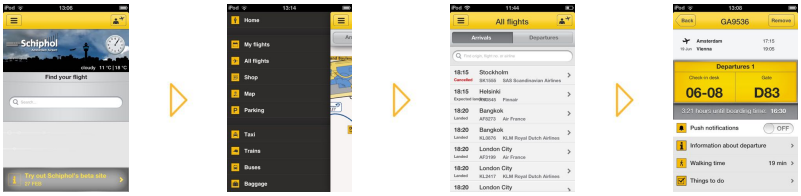
9.2.4 Amsterdam Airport Schiphol (AMS) Use Case analysis

1) Show Arrivals/Departures



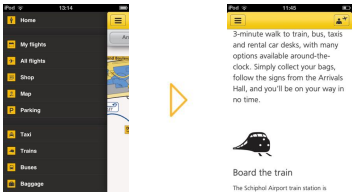
1. The AMS app home screen shows always a search field for flights, the current time and weather information.
2. Enter a destination name and click *Search*.
3. Select a flight of the result list.
4. The flight details containing e.g. flight number, check-in desk and gate.

1) Show Arrivals/Departures - Alternative



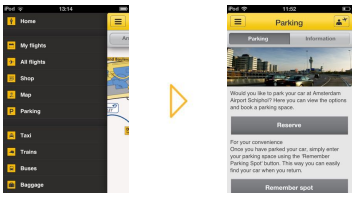
1. Alternatively, open the sidebar menu.
2. Click on *All flights* to open a list of current flights.
3. Select a flight to get detail information.
4. The flight detail view.

2) Show Public Transportation Information



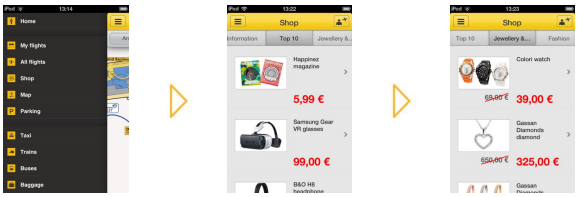
1. Open the sidebar menu and select *Trains*.
2. This view contains information regarding train connection.

3) Show Parking Information



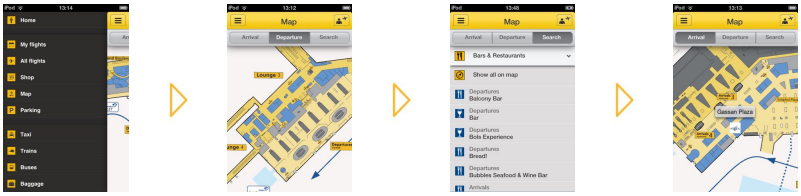
1. Open the sidebar menu and select *Parking*.
2. At this view it is possible to reserve a parking lot and to the save parking position.

4) Show Shop, Restaurant and Service Information - Shops



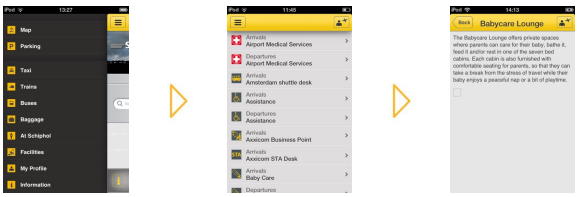
1. Shops and their current offers are available via the sidebar menu.
2. Click on *Shop* to open the shop top ten list.
3. Switch between the categories to see corresponding shop offers.

4) Show Shop, Restaurant and Service Information - Restaurants



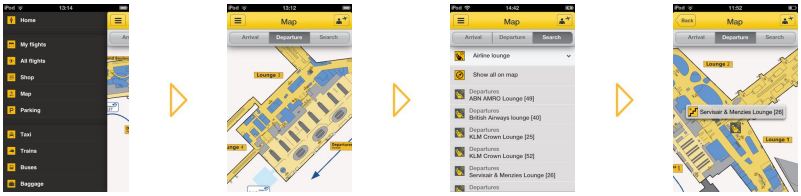
1. Open the sidebar menu and click on *Map*.
2. Click *Search* in the upper menu bar.
3. This opens a list of shops, restaurants and other services of the airport.
4. Select the desired one to show it on the map.

4) Show Shop, Restaurant and Service Information - Services



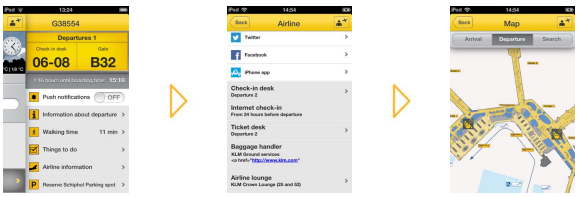
1. Open the sidebar menu and click *Facilities*.
2. This opens a list of the airport services and facilities.
3. Select one to see the detail view.

5) Show Lounge Information



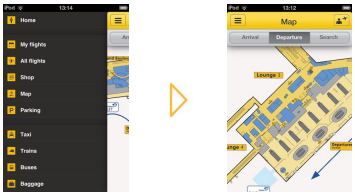
1. Open the sidebar menu and click on *Map*.
2. Select *Search* in the upper menu bar.
3. This opens a list which contains, inter alia, *Airline lounge*.
4. Select the desired one to show it on the map.

5) Show Lounge Information - Alternative



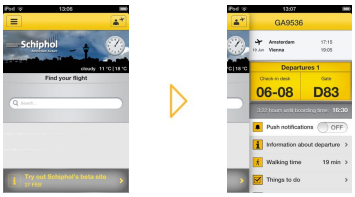
1. Open the next favorite flight via the main screen icon. Select *Airline information*.
2. Click *Airline lounge* to open the map.
3. This view shows the map with the marked airline lounges.

6) Show the Airport Map



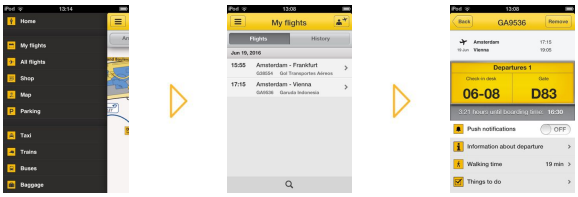
1. Open the sidebar menu and select *Map*.
2. This view presents the airport map and provides a search function.

7) Show Favorite Flights - Alternative



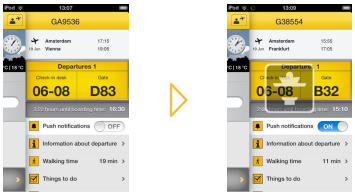
1. The icon in the upper right corner of the home screen opens the next favorite flight.
2. It shows the same information like the flight detail view.

7) Show Favorite Flights - Alternative



1. Alternatively, open the sidebar menu and click *My flights*.
2. This view presents all added flights in a list.
3. Select the desired flight to open the detail view.

8) Enable Push-Notification



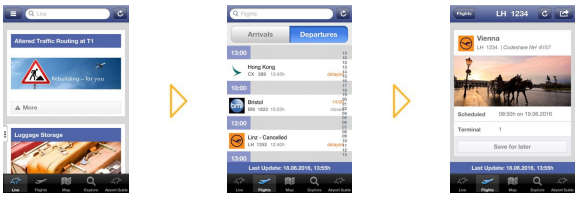
1 2

1 Every flight detail view offers a push notification toggle button.

2 This button enables or disables the push notifications.

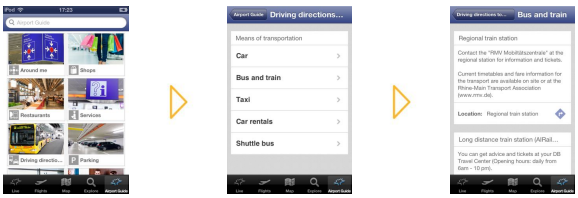
9.2.5 Frankfurt Airport (FRA) Use Case analysis

1) Show Arrivals/Departures



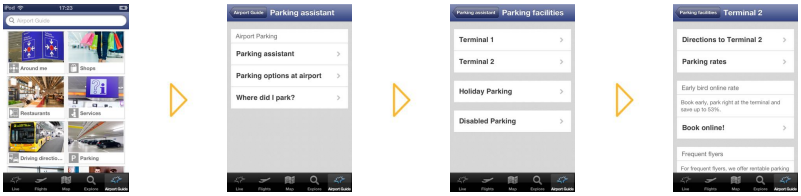
1. The FRA *Live* view presents current information and news.
2. Click on *Flights* at the bottom menu and search for the desired flight.
3. Select the flight to open the detail view.

2) Show Public Transportation Information



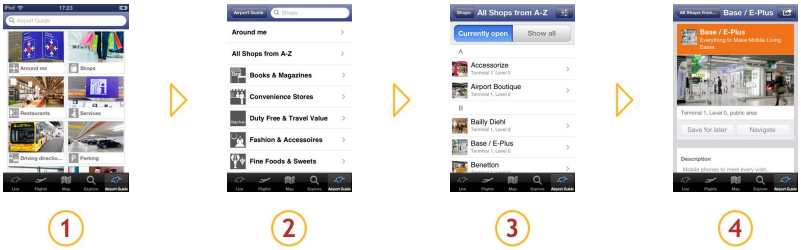
1. Click *Airport Guide* at the bottom menu.
2. Select *Driving directions* to get the public transportation overview.
3. Click on *Bus and train* to open the detail view.

3) Show Parking Information



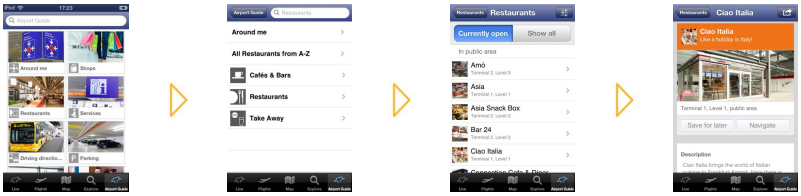
1. Parking information are available via the *Airport Guide* menu entry.
2. Click on *Parking* to see the parking assistant view.
3. Select *Parking options at airport* to get the parking facilities.
4. Choose a terminal to open the detail view.

4) Show Shop, Restaurant and Service Information - Shops



1. Shop information are available via the *Airport Guide* menu entry.
2. Click on *Shops* to see the shop overview.
3. Select *All Shops from A-Z* to get the shop list.
4. Click on a shop to see the details.

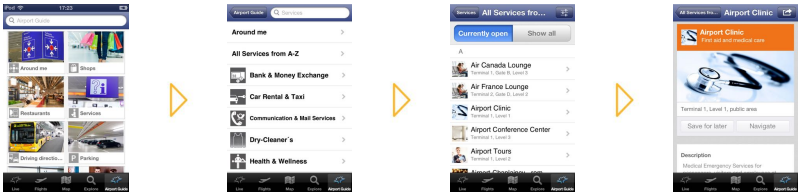
4) Show Shop, Restaurant and Service Information - Restaurants



1 2 3 4

1. Restaurant information are retrievable via the *Airport Guide* menu entry.
2. Click on *Restaurants* to see the restaurant overview.
3. Select *Restaurants* to get the restaurant list.
4. Chose a restaurant to open the detail view.

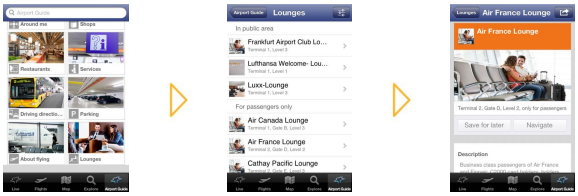
4) Show Shop, Restaurant and Service Information - Services



1 2 3 4

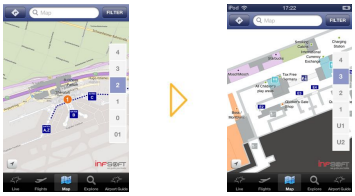
1. Service information are available via the *Airport Guide* menu entry.
2. Click *Services* to see the service overview.
3. Select *All Services from A-Z* to get the service list.
4. Click e.g. on *Airport Clinic* to receive the clinic details.

5) Show Lounge Information



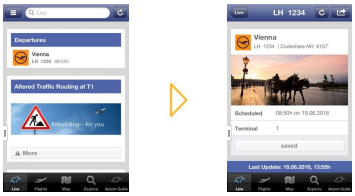
1. Open the *Airport Guide* to select *Lounges*.
2. This view presents a list of all lounges.
3. Select the desired lounge to get detail information.

6) Show the Airport Map



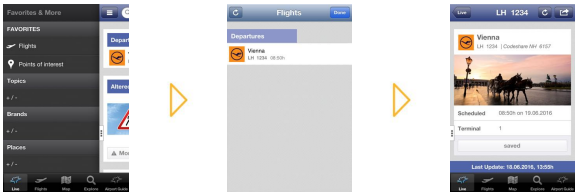
1. Click on the *Map* icon at the bottom menu to open the map.
2. Change the zoom level by clicking on the selection bar at the right side.

7) Show Favorite Flights



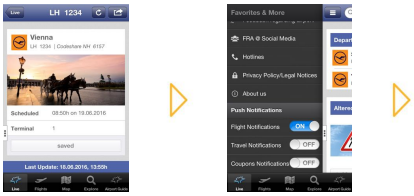
1. Favorite flights are presented on top of the *Live* screen.
2. Click on the flight to get the flight detail view.

7) Show Favorite Flights - Alternative



1. The second possibility to get information regarding favorite flights is via the *Flights* menu entry.
2. This view presents all added flights in a list.
3. Select the corresponding flight to show detail information.

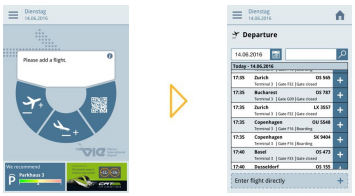
8) Enable Push-Notification



1. Push-Notifications are automatically enabled for every favorite flight.
2. The FRA app provide global push notification settings to enable or disable them.

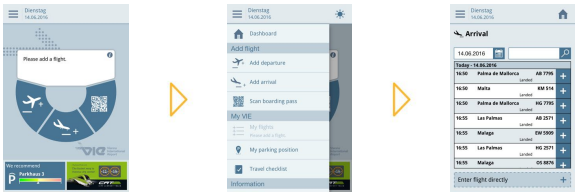
9.2.6 Vienna International Airport (VIE) Use Case analysis

1) Show Arrivals/Departures



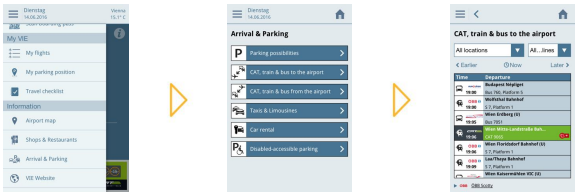
1. The VIE app provides the arrival or departure schedule via one click from the *Dashboard*.
2. Click on departure icon to open the departure schedule.

1) Show Arrivals/Departures - Alternative



1. Alternatively, the sidebar menu contains *Add arrival* and *Add departure* entries.
2. This view presents all added flights in a list.
3. Click on *Add arrival* to open the arrival schedule.

2) Show Public Transportation Information

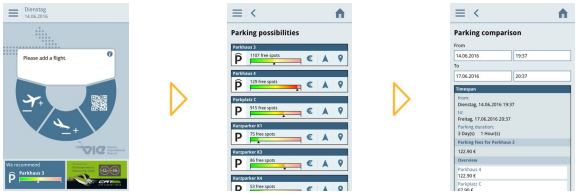


1 Open the sidebar menu and select *Arrival & Parking*.

2 Click on *CAT, train & bus to the airport* to open the schedule.

3 This view provides the current timetable for public transportation.

3) Show Parking Information

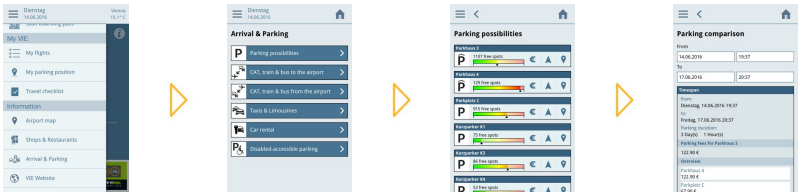


1 Click on the car park icon at the bottom left corner of the *Dashboard* to open the *Parking possibilities*.

2 Select a car park.

3 On this screen you can check the parking fee for your parking duration.

3) Show Parking Information - Alternative

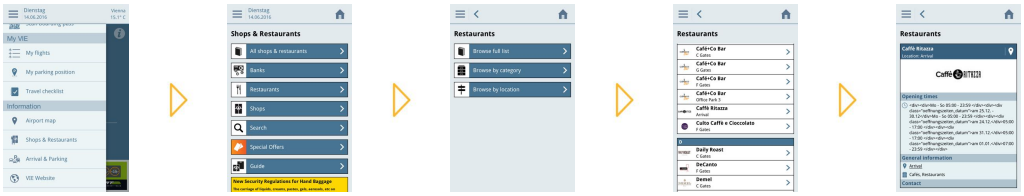


1. Alternatively, click *Arrival & Parking* in the sidebar menu.
2. Select *Parking possibilities* to open the car park overview.
3. Select a car park to open the detail view.
4. On this screen you can check the parking fee for your parking duration.

4) Show Shop, Restaurant and Service information - Shops

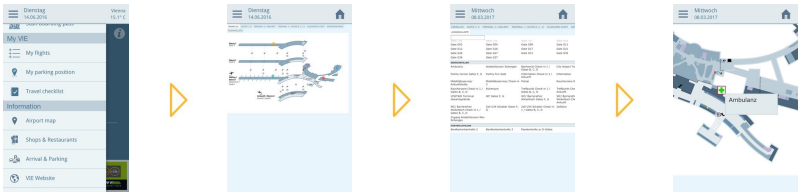
1. Select *Shops & Restaurants* in the sidebar menu.
2. Click on *Shops* to open shop browsing options.
3. Choose *Browse full list* to see the shop list.
4. Search for the desired shop and open it.
5. The Shop detail view contains the shop logo, opening hours, a short description and a link to the map.

4) Show Shop, Restaurant and Service information - Restaurants



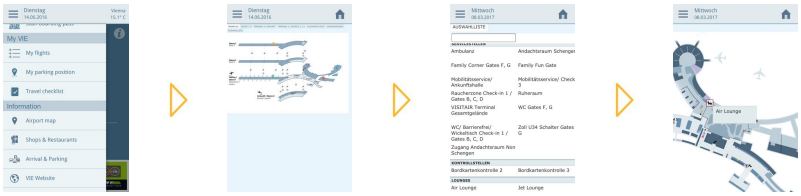
1. Select *Shops & Restaurants* in the sidebar menu.
2. Click on *Restaurants* to open restaurant browsing options.
3. Choose *Browse full list* to see the restaurant list.
4. Search for the desired restaurant and open it.
5. The Restaurant detail view contains the restaurant logo, opening hours, a short description and a link to the map.

4) Show Shop, Restaurant and Service information - Services



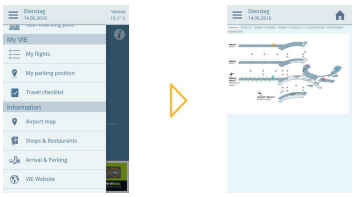
1. Open the sidebar menu and click *Airport map*.
2. Open the selection list at the upper right cornet.
3. Scroll to the services and click the desired one.
4. The selected service is shown on the map.

5) Show Lounge information



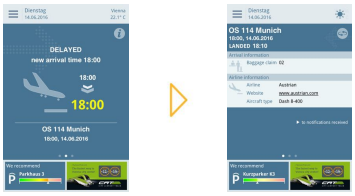
1. Open the sidebar menu and click *Airport map*.
2. The selection list at the upper right cornet contains an entry *Lounges*.
3. Click on the desired lounge.
4. The selected lounges is shown on the map.

6) Show the Airport Map



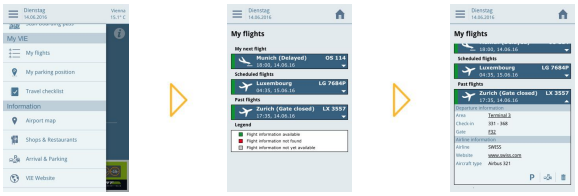
1. The airport map is retrievable via the sidebar menu.
2. It is possible to select different gates and terminals to get a more detailed view.

7) Show Favorite Flights



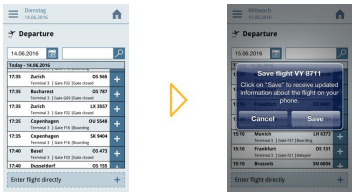
1. Favorite flights are presented direct at the *Dashbord* in a card layout.
2. Click on the information icon to open the flight details. This view contains, inter alia, flight status, airline and baggage claim.

7) Show Favorite Flights - Alternative



1. The second possibility to get information regarding favorite flights is via the *My flights* menu entry.
2. This view presents all added flights in a list.
3. Select the desired flight to get detail information.

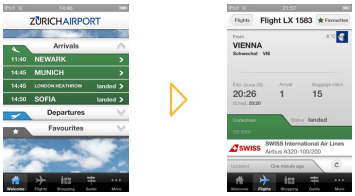
8) Enable Push-Notification



1. A flight could be added via the departure or arrival schedule to *My flights*.
2. Push notifications are automatically enabled for each saved flight. These notifications could not be disable.

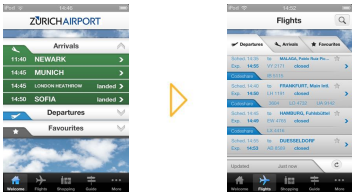
9.2.7 Zurich Airport (ZRH) Use Case analysis

1) Show Arrivals/Departures



1. Current arrival and departure flights are shown direct at the *Welcome* view.
2. Click on a flight to get detail information.

1) Show Arrivals/Departures - Alternative

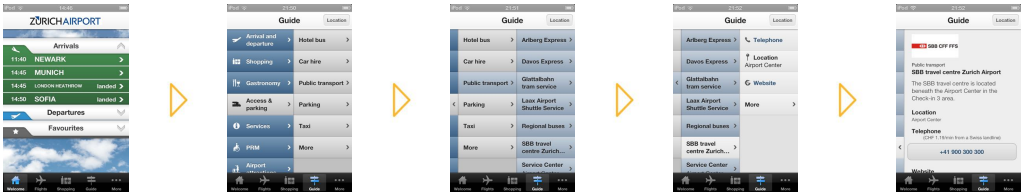


1. Alternatively, click *Flights* on the bottom menu.
2. This view shows the arrival or departure flights with some details and provides search functions.

Case 2) Show Public Transportation Information

1. Click *More* on the bottom menu.
2. Select *Access & Parking*.
3. This view contains the current schedule for trains and buses.

2) Show Public Transportation Information - Alternative

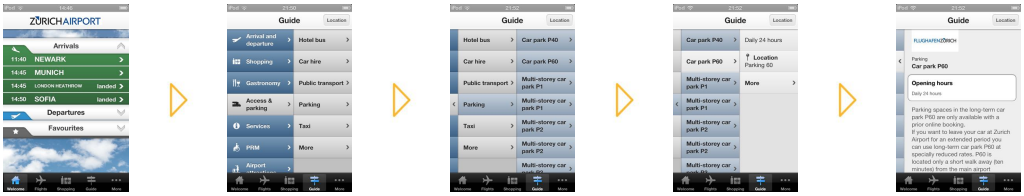


1. Open the airport *Guide* via the bottom menu.
2. Select *Access & Parking* to see the local transportation possibilities.
3. Open *Public Transportation*.
4. Click e.g. on *SBB travel center Zurich* to get compact information.
5. Choose *More* for more details.

Case 3) Show Parking Information

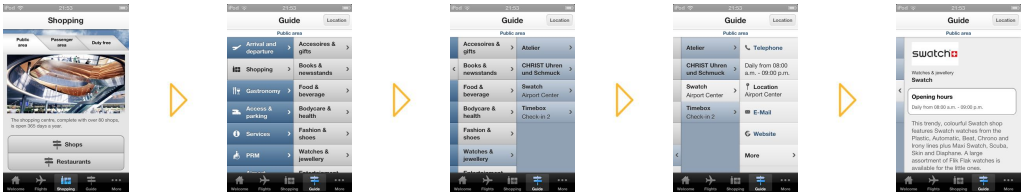
1. Click *More* on the bottom menu.
2. Select *Access & Parking* to see the *Public transportation* tab.
3. Choose the *Parking* tab to get the current parking availability.

3) Show Parking Information - Alternative



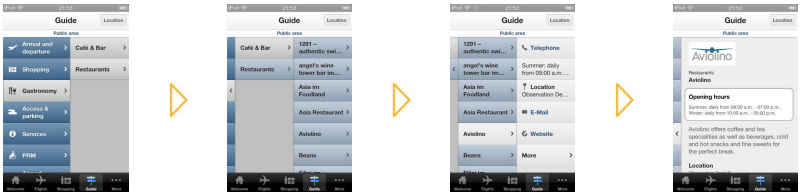
1. Open the airport *Guide* via the bottom menu.
2. Select *Access & Parking*.
3. Click *Parking* to get the parking possibilities.
4. Choose a car park for short information.
5. Click *More* for the car park detail view.

4) Show Shop, Restaurant and Service Information - Shops



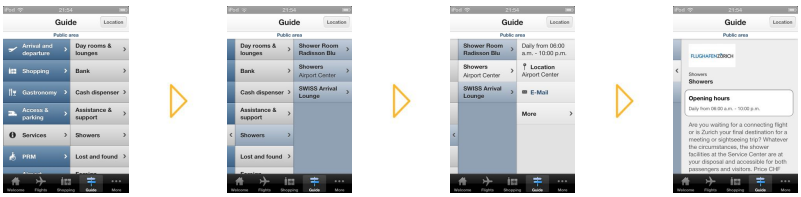
1. Open the *Shopping* view via the bottom menu.
2. Click *Shops* will link to the airport *Guide* view.
3. Choose a shop category to get the shop list
4. Select a shop for compact information.
5. Click *More* for more details.

4) Show Shop, Restaurant and Service Information - Restaurants



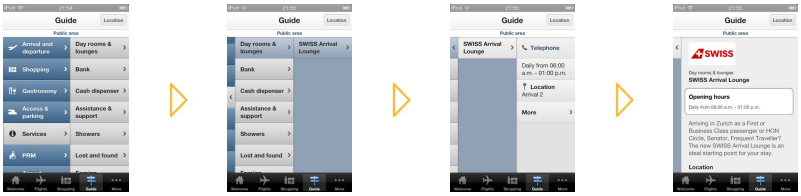
1. Restaurant information are available via the airport *Guide*.
2. Select *Restaurants* to get the restaurant categories.
3. Choose a restaurant for compact information.
4. Click *More* to open the restaurant detail view.

4) Show Shop, Restaurant and Service Information - Services



1. Open the airport *Guide* via the bottom menu.
2. Click *Services* to get the offered services.
3. Select e.g. *Showers* to see the different shower rooms. Choose one to the compact information.
4. Click *More* for more details.

5) Show Lounge Information



1. To get lounge information open the airport *Guide*.
2. Click *Services*.
3. Choose *Day rooms & lounges* and *Swiss Arrival Lounge* to get compact information.
4. Select *More* to open the lounge detail view.

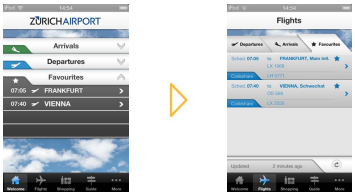
6) Show the Airport Map

The ZRH app offers no airport map.

7) Show Favorite Flights

1. The ZRH app presents favorite flights at the *Welcome* screen. This view contains compact information including the baggage claim

7) Show Favorite Flights - Alternative

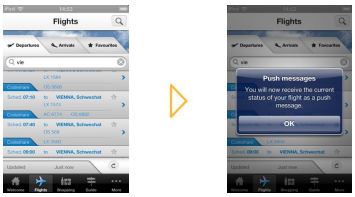


1 2

1 If there are more than one favorite flight added, the ZRH app presents the flights as a list on the home screen.

2 Select *Flight* at the bottom menu and open the *Favorites* for more details.

8) Enable Push-Notification



1 2

1 Push notifications are automatically enabled for added flights. Click on the star icon to add a flight to the *Favorites*.

2 A confirm dialog informs about the activated notifications.

9.3 Initial GUI Mock-Ups

9.3.1 Information Architecture (IA) 1

Figure 9.1: Prototype IA 1 - Home

Figure 9.2: Prototype IA 1 - Flight details

Figure 9.3: Prototype IA 1 - Favorite Flights

Figure 9.4: Prototype IA 1 - Favorite Flights History

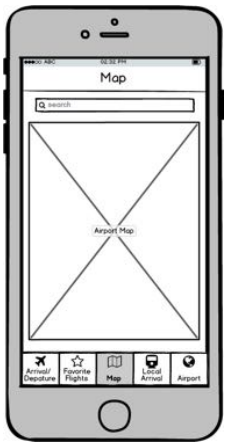


Figure 9.5: Prototype IA 1 - Map

Figure 9.6: Prototype IA 1 - Local Arrival/Public Transportation

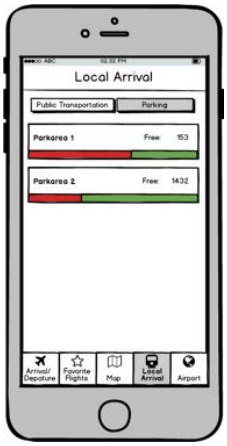


Figure 9.7: Prototype IA 1 - Local Arrival/Parking

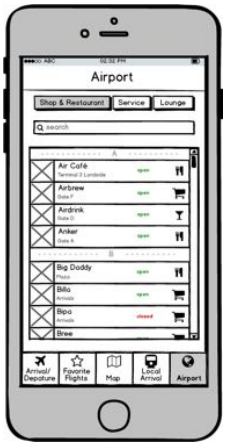


Figure 9.8: Prototype IA 1 - Shops/Bars & Restaurants

Figure 9.9: Prototype IA 1 - Shop details

Figure 9.10: Prototype IA 1 - Airport Services

Figure 9.11: Prototype IA 1 - Airport Lounges

9.3.2 Information Architecture (IA) 2

Figure 9.12: Prototype IA 2 - Home

Figure 9.13: Prototype IA 2 - Arrivals

Figure 9.14: Prototype IA 2 - Departures

Figure 9.15: Prototype IA 2 - Flight Details

Figure 9.16: Prototype IA 2 - Favorite Flights

Figure 9.17: Prototype IA 2 - Favorite Flights History

Figure 9.18: Prototype IA 2 - added Flight Details

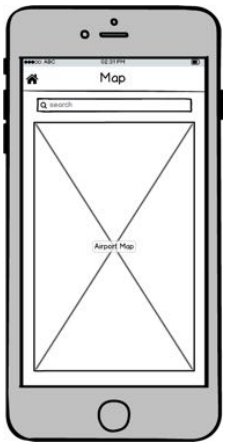


Figure 9.19: Prototype IA 2 - Map

Figure 9.20: Prototype IA 2 - Public Transportation

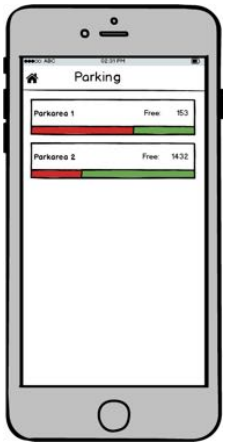


Figure 9.21: Prototype IA 2 - Parking

Figure 9.22: Prototype IA 2 - Shops

Figure 9.23: Prototype IA 2 - Shops Details

Figure 9.24: Prototype IA 2 - Restaurants

Figure 9.25: Prototype IA 2 - Service

Figure 9.26: Prototype IA 2 - Lounge

9.4 Usability Testing Form

Date: _____ Time: _____

Location: _____

IA 1
IA 2

9.4.1 Demographical Data

Age	16 - 25	26 - 35	36 - 45	46 - 65
Gender		Female		Male
Flight frequency (last year)	0	1	2	> 2
Frequency of Smartphone usage	One time per Week	every two days	every day	Multiple times a day

	1	2	3	4	5	6	7	8
Scenario order								

9.4.2 System Usability Scale

		Strongly disagree				**Strongly agree**
1.	I think that I would like to use this system frequently	1	2	3	4	5
2.	I found the system unnecessarily complex	1	2	3	4	5
3.	I thought the system was easy to use	1	2	3	4	5
4.	I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5
5.	I found the various functions in this system were well integrated	1	2	3	4	5
6.	I thought there was too much inconsistency in this system	1	2	3	4	5
7.	I would imagine that most people would learn to use this system very quickly	1	2	3	4	5
8.	I found the system very cumbersome to use	1	2	3	4	5
9.	I felt very confident using the system	1	2	3	4	5
10.	I needed to learn a lot of things before I could get going with this system	1	2	3	4	5

Score: _____

What do you think is the best aspect of this software?

What do you think needs most improvement?

9.4.3 Scenarios

Number	S1
Name	Retrieve Arrivals/Departures
Username	S1
Description	You want to see the flight details of the **Departure Flight** with the flight number **PA0413** at **April the 1th, 2018**.

Number	S2
Name	Retrieve Public Transportation Information
Username	S2
Description	You're searching for a **Train** to the airport at **April the 1th, 2018**. For that you open the corresponding view to see the public transportation schedule.

Number	S3
Name	Retrieve Parking Information
Username	S3
Description	Assume that you are currently on the way to the airport by car and looking for the car park availability. Open the corresponding view to see the current workload.

Number	S4
Name	Retrieve Shop, Restaurant and Service Information
Username	S4
Description	You're searching for detail information of the **Tommy Hilfiger** shop at **Terminal 2**.

Number	S5
Name	Retrieve Lounge Information
Username	S5
Description	You want to see detail information for the **Austrian Lounge** in **Terminal 3**.

Number	S6
Name	Retrieve The Airport Map
Username	S6
Description	You want get an overview of the airport and for that you open the **airport map**.

Number	S7
Name	Retrieve Favorite Flights
Username	S7
Description	You want to see the detail information of you already added **Favorite Flight** to Frankfurt on April the 10th, 2018.

Number	S8
Name	Enable Push-Notification
Username	S8
Description	You want to add the flight to **Zurich today at 21:30** o'clock to your favorite flights and enable the notifications for that flight.

Bibliography

References

[7] Alaa Alkhafaji et al. “Guidelines for designing a smart and ubiquitous learning environment with respect to cultural heritage”. In: *2017 11th International Conference on Research Challenges in Information Science (RCIS)*. IEEE, 2017, pp. 334–339.

[8] Gavin Allanwood and Peter Beare. *User Experience Design*. 1st ed. Fairchild Books, 2014. ISBN: 978-2-940496-6.

[9] Amany Alnahdi and Shih-Hsi Liu. “Mobile Internet of Things (MIoT) and Its Applications for Smart Environments: A Positional Overview”. In: *2017 IEEE International Congress on Internet of Things (ICIOT)*. IEEE, 2017, pp. 151–154.

[11] Bo Andersson. “Handheld Computing from a Designer’s Perspective: A 10–year Review – 2001–2010”. In: *45th Hawaii International Conference on System Sciences*. IEEE, 2012.

[12] Nurul Zakiah binti Ayob, Ab. Razak Che Hussin, and Halina Mohamed Dahlan. “Three Layers Design Guideline for Mobile Application”. In: *International Conference on Information Management and Engineering*. IEEE, 2009.

[14] Lynne Baillie and Lee Morton. “Designing Quick & Dirty Applications for Mobiles: Making the Case for the Utility of HCI Principles”. In: *Journal of Computing and Information Technology* (2010), pp. 103–107.

[15] Aaron Bangor, Philip Kortum, and James Miller. “Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale”. In: *Journal of Usability Studies* 4 (2009), pp. 114–123.

[16] IEEE Standard Board. *IEEE Std. 610.12-1990. IEEE Standard Glossary of Software Engineering Technology*. Tech. rep. IEEE, 1990.

[17] IEEE Standard Board. *ISO/IEC/IEEE 29148. Systems and software engineering - Life cycle processes - Requirements Engineering*. Tech. rep. IEEE, 2011.

[18] Lars Braubach, Alexander Pokahr, and Winfried Lamersdorf. “A Universal Criteria Catalog for Evaluation of Heterogeneous Agent Development Artifacts”. In: *Distributed Systems and Information Systems*. 2008.

[19] John Brooke. *Usability Evaluation in Industry*. London: Taylor and Francis, 1986, pp. 189–196.

[21] Andre Charland and Brian Leroux. “Mobile Application Development: Web vs. Native”. In: *Communications of the ACM* (2011), pp. 49–53.

[23] Alan Cooper. *The Inmates Are Running the Asylum: Why High-tech Products Drive Us Crazy and How to Restore the Sanity*. 1st ed. Carmel, Indiana; USA: Sams Publishing, 1999.

[24] *Die bereits angebotenen Services in der App Ihres Flughafens*. Flughafen Wien AG. Postfach 1; A-1300 Wien Flughafen, 2014.

[26] Bruno Dumas, Maria Solórzano, and Beat Signer. “Design Guidelines for Adaptive Multimodal Mobile Input Solutions”. In: *15th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM Digital Library, 2013.

[33] Shishir Gundavaram. *CGI Programming on the World Wide Web*. O'Reilly Media, Inc., 1996.

[34] Tim R. Haaksma, D. T. De Jong Menno, and Joyce Karreman. "Users' Personal Conceptions of Usability and User Experience of Electronic and Software Products". In: *IEEE Transactions on Professional Communication* 61.2 (2018), pp. 116–132.

[35] Rachel Harrison, Derek Flood, and David Duce. "Usability of mobile applications: literature review and rationale for a new usability model". In: *Journal of Interaction Science* (2013).

[43] Morten Hertzum and Niels Ebbe Jacobsen. "The evaluator effect: A chilling fact about usability evaluation methods". In: *International Journal of Human-Computer Interaction* (2001), pp. 421–443.

[46] Kasper Hornbaek et al. "Use Case Evaluation (UCE): A Method for Early Usability Evaluation in Software Development". In: *Human-Computer Interaction – INTERACT 2007*. 2007, pp. 578–591.

[47] Jianjun Huang et al. "UI driven Android application reduction". In: *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2017, pp. 286–296.

[48] Zahid Hussain et al. "User Interface Design for a Mobile Multimedia Application: An Iterative Approach". In: *First International Conference on Advances in Computer-Human Interaction*. IEEE, 2008.

[56] *ISO/IEC 25010 - Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) - Software and quality in use models*. International Organization for Standardization. 1, ch. de la Voie-Creuse; CP 56 - CH-1211 Geneva 20, 2011.

[57] *ISO/IEC 25022 - Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) – Measurement of quality in use*. International Organization for Standardization. 1, ch. de la Voie-Creuse; CP 56 - CH-1211 Geneva 20, 2012.

[58] *ISO/IEC 9241-11:1998 - Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability*. International Organization for Standardization. 1, ch. de la Voie-Creuse; CP 56 - CH-1211 Geneva 20, 1998.

[59] *ISO/IEC 9241-210:2010 - Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems*. International Organization for Standardization. 1, ch. de la Voie-Creuse; CP 56 - CH-1211 Geneva 20, 2010.

[60] Ivar Jacobson. *Object-Oriented Software Engineering. A Use Case Driven Approach*. 1st ed. Amsterdam, Netherland: Addison-Wesley Longman, 1992.

[61] Ljilja Ruzic Kascak, Claudia B. Rébola, and Jon A. Sanford. "Integrating Universal Design (UD) Principles and Mobile Design Guidelines to Improve Design of Mobile Health Applications for Older Adults". In: *2014 IEEE International Conference on Healthcare Informatics*. IEEE, 2014, pp. 343–348.

[62] Jennifer Kim. "Design and Evaluation of Mobile Applications with Full and Partial Offloadings". In: *7th International Conference on Grid and Pervasive Computing*. Springer-Verlag, 2012.

[63] Dr. Peter Laimer. *Urlaubs- und Geschäftsreisen – Kalenderjahr 2014*. Tech. rep. STATISTIK AUSTRIA, 2015.

[65] Joe Ligman et al. “Improving Design Validation of Mobile Application User Interface Implementation”. In: *2016 IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft)*. IEEE, 2016, pp. 277–278.

[70] Matin Maguire and Nigel Bevan. “User requirements analysis - A review of supporting methods”. In: *IFIP* 17^th *World Computer Congress*. Kluwer Academic, 2002, pp. 133–148.

[71] Aaron Marcus, Theresa Karolina Schieder, and Lorenzo Cantoni. “The Travel Machine: Mobile UX Design That Combines Information Design with Persuasion Design”. In: *A. Marcus (Ed.): DUXU/HCII 2013, Part IV, LNCS 8015*. Springer-Verlag, 2013, pp. 696–705.

[73] Jörg H. Mayer and Timm Weitzel. “Appropriate Interface Designs for Mobile End–User Devices–Up Close and Personalized Executive Information Systems as an Example”. In: *45th Hawaii International Conference on System Sciences*. IEEE, 2012.

[74] Everett N. McKay. *Ui Is Communication: How to Design Intuitive, User Centered Interfaces by Focusing on Effective Communication*. 1st ed. Burlington, Massachusetts: Morgan Kaufmann, 2013.

[77] Karima Moumane and Ali Idri. “Software quality in mobile environments: A comparative study”. In: *2017 4th International Conference on Control, Decision and Information Technologies (CoDIT)*. IEEE, 2017, pp. 1123–1128.

[79] Theresa Neil. *Mobile Design Pattern Gallery: UI Patterns for Smartphone Apps*. 2nd ed. Sebastopol, California: O’Reilly & Associates, 2014.

[83] Jakob Nielsen and Raluca Budiu. *Mobile Usability: Für iPhone, iPad, Android, Kindle*. 1st ed. Berkeley: mitp, 2013.

[85] Donald A. Norman. *Don Norman interviewed by Peter Merholz*. http://uxdesign.com/about-user-experience-design/article/don-norman-interview-ux-video/49. UX Week. 2008.

[86] Donald A. Norman. *The Design of Everyday Things*. 1st ed. Philadelphia: Basic Books, 2013.

[88] Bernd Oestereich. *Analyse und Design mit UML 2.1 - Objektorientierte Softwareentwicklung*. 8th ed. München; Germany: Oldenbourg Wissenschaftsverlag GmbH, 2006.

[90] Statistik Austria Bundesanstalt Statistik Österreich. *Familien 1985–2014*. Tech. rep. STATISTIK AUSTRIA, 2015.

[91] Pekka Parhi, Amy K. Karlson, and Benjamin B. Bederson. “Target Size Study for One-handed Thumb Use on Small Touchscreen Devices”. In: *Proceedings of the 8th Conference on Human-computer Interaction with Mobile Devices and Services*. MobileHCI ’06. Helsinki, Finland: ACM, 2006, pp. 203–210. ISBN: 1-59593-390-5.

[95] Michael Richter and Markus Flückiger. *Usability Engineering kompakt*. 2nd ed. Spektrum Akademischer Verlag, 2010. ISBN: 978-3-8274-2328-3.

[97] S. Rosenbaum, J. Rohn, and J. Humberg. “A toolkit for strategic usability: Results from workshops, panels, and surveys”. In: *Proceedings of Conference on Human-computer Interaction 2000*. New York, USA: ACM, 2000, pp. 337–344.

[98] Sharmistha Roy, Pasant Kumar Pattnaik, and Rajib Mall. “A quantitative approach to evaluate usability of academic websites based on human perception”. In: *Egyptian Information Journal* (2014), pp. 159–167.

[99] Chris Rupp and Sophist Group. *Requirements-Engineering und -Management*. 5th ed. Munich, Germany: Carl Hanser Verlag, 2009.

[100] *Technische Daten - Google Nexus 6P*. Google Inc. 1600 Amphitheatre Parkway; Mountain View, CA 94043; USA, 2016.

[102] Ralph Tragale. *Airport Traffic Report 2013*. Tech. rep. The Port Authority of NY & NJ, 2014.

[104] Sally Wehmeier. *Oxford Advanced Learner's Dictionary*. 6th ed. Berlin: Cornelsen & Oxford, 2000.

[106] *What is User-Centered Design?* http://www.usabilityprofessionals.org/usability_resources/about_usability/what_is_ucd.html. [Online; accessed 22-march-2014]. 2014.

[108] Chauncey Wilson. *User Interface Inspection Methods – A User-Centered Design Method*. 1st ed. Waltham, Massachusetts: Morgan Kaufmann, 2013.

[109] Christoph Wimmer. *Usability Engineering - Lecture 5*. Tech. rep. Industial Software - TU Wien, 2017.

[110] Christoph Wimmer. *Usability Engineering - Lecture 6*. Tech. rep. Industial Software - TU Wien, 2017.

Online References

[1] GROUPE ADP. *Paris - Charles de Gaulle*. 2015. URL: http://www.parisaeroport.fr/en/group/group-strategy/in-brief/platforms/paris-charles-de-gaulle.

[2] Flughafem Wien AG. *Annual Report 2015*. 2015. URL: http://www.viennaairport.com/jart/prj3/va/uploads/data-uploads/Konzern/Investor%20Relations/Geschaeftsberichte/GB_2015_en.pdf.

[3] Flughafem Zürich AG. *Facts and Figures 2015*. 2016. URL: https://www.zurich-airport.com/the-company/zurich-airport-ag/facts-and-figures.

[4] Flughafen Wien Ag. *Vienna Airport*. 2014. URL: https://itunes.apple.com/at/app/viennaairport/id541132906?mt=8.

[5] Flughafen Zuerich AG. *Flughafen Zürich / ZRH*. 2015. URL: https://itunes.apple.com/at/app/flughafen-zurich-zrh/id484076423?mt=8.

[6] Fraport AG. *Zahlen, Daten, Fakten 2015 zum Flughafen Frankfurt*. 2015. URL: http://static.fraport.de/ONLINE/e-paper/2015/zdf2015de/mobile/index.html#p=1.

[10] Industrial Designers Society of America. *What is Industrial Design?* 2015. URL: http://www.idsa.org/education/what-is-industrial-design.

[13] Francesca Bacard. *Size-22 Model Tess Holliday Talks Relationships and Her Career, Says "Black Men Love Me"*. 2015. URL: https://www.eonline.com/news/767973/tess-holliday-gives-birth-to-a-baby-boy-find-out-his-name.

[20] Schiphol Nederland B.V. *Schiphol Amsterdam Airport*. 2015. URL: https://itunes.apple.com/at/app/schiphol-amsterdam-airport/id409161665?mt=8.

[22] Paras Chopra. *The Ultimate Guide To A/B Testing*. 2010. URL: https://www.smashingmagazine.com/2010/06/the-ultimate-guide-to-a-b-testing/.

[25] Danielle Dr. Sheypuk. *NY FASHION WEEK – PUTS WHEELCHAIR ON RUNWAY*. 2014. URL: http://www.workhousepr.net/social/category/wheelchair.

[27] futurezone.at. *61 Prozent der Österreicher nutzen ein Smartphone*. 2013. URL: http://futurezone.at/digital-life/61-prozent-der-oesterreicher-nutzen-ein-smartphone/32.229.735.

[28] Stefan von Gagern. *Native vs. Web App vs. Hybrid: Was ist die perfekte Entwicklerstrategie?* 2013. URL: https://www.developergarden.com/de/blog/artikel/article/native-vs-web-app-vs-hybrid-was-ist-die-perfekte-entwicklerstrategie/.

[29] Gartner. *Marktanteile der Betriebssysteme am Endkundenabsatz von Smartphones weltweit von 2009 bis 2014*. 2014. URL: http://de.statista.com/statistik/daten/studie/12885/umfrage/marktanteil-bei-smartphones-nach-betriebssystem-weltweit-seit-2009/.

[30] Stephanie Glen Ducasse. *Latin Square Design: Definition and Balanced Latin Square Algorithm*. 2017. URL: https://www.statisticshowto.datasciencecentral.com/latin-square-design/.

[31] Changi Airport Group. *Traffic Statistics*. 2016. URL: http://www.changiairport.com/corporate/about-us/traffic-statistics.html.

[32] Schiphol Group. *Traffic Review 2015*. 2016. URL: http://trafficreview2015.schipholmagazines.nl/definitions.html.

[36] Alexandra Hayman. *The Debate: HTML5 vs. Native Apps*. 2013. URL: http://ideas.dnsee.com/2013/03/27/the-debate-html5-vs-native-apps/.

[37] U.S. Department of Health and Human Services. *Planning a Usability Test*. 2015. URL: https://www.usability.gov/how-to-and-tools/methods/usability-testing.html.

[38] U.S. Department of Health and Human Services. *Running a Usability Test*. 2015. URL: https://www.usability.gov/how-to-and-tools/methods/running-usability-tests.html.

[39] U.S. Department of Health and Human Services. *System Usability Scale (SUS)*. 2018. URL: https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html.

[40] U.S. Department of Health and Human Services. *Usability Test Plan Template*. 2018. URL: https://www.usability.gov/how-to-and-tools/resources/templates/usability-test-plan-template.html.

[41] U.S. Department of Health and Human Services. *Usability Testing*. 2016. URL: https://www.usability.gov/how-to-and-tools/methods/planning-usability-testing.html.

[42] U.S. Department of Health and Human Services. *User Interface Design Basics*. 2015. URL: http://www.usability.gov/what-and-why/user-interface-design.html.

[44] Gustav Hoiland. *The Grandfather – Christmas Portraits*. 2011. URL: https://gustavhoiland.com/2011/12/25/the-grandfather/.

[45] Steven Hoober. *How Do Users Really Hold Mobile Devices?* 2013. URL: http://www.uxmatters.com/mt/archives/2013/02/how-do-users-really-hold-mobile-devices.php.

[49] Apple Inc. *iOS Human Interface Guidelines*. 2014. URL: https://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/index.html#//apple_ref/doc/uid/TP40006556-CH66-SW1.

[50] Apple Inc. *iPhone 6s - Technical Data*. 2015. URL: https://www.apple.com/at/iphone-6s/specs/.

[51] Apple Inc. *iPod touch (4th generation) - Technical Specifications*. 2011. URL: https://support.apple.com/kb/SP594.

[52] Apple Inc. *Open Source*. 2014. URL: https://developer.apple.com/opensource/.

[53] Google Inc. *Android developers*. 2018. URL: http://developer.android.com.

[54] Google Inc. *Patterns - Android developers*. 2018. URL: http://developer.android.com/design/patterns/index.html.

[55] insivia. *What is the difference between a native mobile app vs. web app?* 2014. URL: http://www.insivia.com/what-is-the-difference-between-a-native-mobile-app-vs-web-app/.

[64] Emil Lamprecht. *The Difference Between UX and UI Design- A Layman's Guide*. 2015. URL: http://blog.careerfoundry.com/the-difference-between-ux-and-ui-design-a-laymans-guide/.

[66] Heathrow Airport Limited. *Heathrow Airport Guide*. 2015. URL: https://itunes.apple.com/gb/app/heathrow-airport-guide/id427951859?mt=8.

[67] LHR Airports Limited. *Heathrow - Facts and figures*. 2015. URL: http://www.heathrow.com/company/company-news-and-information/company-information/facts-and-figures.

[68] Changi Airport Group (Singapore) Pte Ltd. *iChangi*. 2015. URL: https://itunes.apple.com/at/app/ichangi/id391730848?mt=8.

[69] I. Scott MacKenzie. *Within-subjects vs. Between-subjects Designs: Which to Use?* 2013. URL: https://www.yorku.ca/mack/RN-Counterbalancing.html.

[72] Florian Matthey. *iOS 7: Wieder einmal mit Sicherheitslücke im Lock Screen*. 2014. URL: http://www.giga.de/downloads/ios-7/news/ios-7-wieder-einmal-mit-sicherheitsluecke-im-lock-screen/.

[75] MeasuringU. *The Methods UX Professionals Use (2016)*. 2017. URL: https://measuringu.com/ux-methods-2016/.

[76] Ursula Meseberg. *Use Case 2.0: Agile Projektplanung mit Use Case Slices*. 2015. URL: http://www.heise.de/developer/artikel/Use-Case-2-0-Agile-Projektplanung-mit-Use-Case-Slices-2535642.html.

[78] nachrichten.at. *1,6 Millionen Behinderte in Österreich*. 2009. URL: http://www.nachrichten.at/nachrichten/ticker/Soziales-Behinderte-Statistik-Oesterreich-Hintergrund-Grafik;art449,303448.

[80] Jakob Nielsen. *10 Usability Heuristics for User Interface Design*. 1995. URL: https://www.nngroup.com/articles/ten-usability-heuristics/.

[81] Jakob Nielsen. *How Many Test Users in a Usability Study?* 2012. URL: https://www.nngroup.com/articles/how-many-test-users/.

[82] Jakob Nielsen. *Return on Investment for Usability*. 2003. URL: https://www.nngroup.com/articles/return-on-investment-for-usability/.

[84] Jakob Nielsen and Don Norman. *The Definition of User Experience*. 2016. URL: https://www.nngroup.com/articles/definition-user-experience/.

[87] *Number of apps available in leading app stores as of 1st quarter 2018*. 2018. URL: https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/.

[89] Optimizely. *Multivariate Testing*. 2017. URL: https://www.optimizely.com/optimization-glossary/multivariate-testing/.

[92] Aeroports de Paris. *My Airport – Offizieller Service von Aéroports de Paris*. 2015. URL: https://itunes.apple.com/at/app/my-airport-offizieller-service/id322543617?mt=8.

[93] Redaktion. *Statistik: 3,1 Millionen Brillenträger*. 2004. URL: http://derstandard.at/1856190/Statistik-31-Millionen-Brillentraeger.

[94] David Reuter. *David Reuter - Bewerbungsfoto*. 2019. URL: https://www.pinterest.com/davidr2564/bewerbungsfotos/.

[96] Jennifer Romano Bergstrom. *Moderating Usability Tests*. 2013. URL: https://www.usability.gov/get-involved/blog/2013/04/moderating-usability-tests.html.

[101] Anders Toxbue. *UI Patterns*. 2016. URL: http://ui-patterns.com.

[103] W3C. *Mobile Web Best Practices 1.0*. 2008. URL: https://www.cs.umd.edu/users/ben/goldenrules.html.

[105] *What is usability?* 2016. URL: http://www.usabilitynet.org/management/b_what.htm.

[107] Stephan Wiesend. *Die Evolution von iOS*. 2014. URL: http://www.macwelt.de/news/Die-Evolution-von-iOS-8743483.html.

[111] Fraport AG Frankfurt Airport Services Worldwide. *Frankfurt Airport (FRA Airport)*. 2015. URL: https://itunes.apple.com/at/app/frankfurt-airport-fra-airport/id453191399?mt=8.

[112] Daniel Würstl. *Native Apps vs. Web Apps - Unterschiede und Vorteile*. 2017. URL: https://app-entwickler-verzeichnis.de/faq-app-entwicklung/11-definitionen/586-unterschiede-und-vergleich-native-apps-vs-web-apps-2.