

Biber der Informatik in einer App - Usability für mobiles Lernen

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Informatikdidaktik

eingereicht von

Ulrike Gerger

Matrikelnummer 0226074

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung
Betreuer: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Gerald Futschek

Wien, 07.04.2016

(Unterschrift Verfasserin)

(Unterschrift Betreuer)

Eidesstattliche Erklärung

Ulrike Gerger, Marktstraße 51, A-7532 Litzelsdorf

„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.“

Ort, Datum

Unterschrift

Kurzfassung

Durch die neuesten Entwicklungen im Bereich der Touchphones wird es möglich auch Lerninhalte sinnvoll auf mobile Geräte zu bringen. Für den Biber der Informatik, einen Wettbewerb zum Thema Informatik, gibt es zwar schon eine mobile Applikation in Form eines Lernspiels, das aber nur für den Einsatz auf Android-Betriebssystemen konzipiert ist. Die Idee hinter dieser Arbeit ist nun einen Prototypen einer mobilen Applikation für den Biber der Informatik zu entwickeln, der für IOS- aber auch für Android-Betriebssysteme einsetzbar ist. Der inhaltliche Unterschied zwischen den beiden Apps zeigt sich dadurch, dass die BiberApp (umgesetzter Prototyp) nicht nur ein reines Lernspiel ist, sondern auch ein Tool, das im Unterricht eingesetzt werden kann. Dazu wurde speziell eine Suchfunktion für Fragen der App implementiert.

Daher haben sich für diese Arbeit folgende Forschungsfragen ergeben:

- Wie und wie gut können vorhandene Fragen des Biber Wettbewerbs in einer Lern-App abgebildet werden?
- Wie gut einsetzbar ist eine Volltextsuche in den Aufgaben des Biber der Informatik um bestimmte Themen für den Unterricht zu suchen?

Für die Umsetzung der Fragestellungen kommen die drei wissenschaftlichen Methoden Literaturrecherche, deren Ergebnis ein Richtlinienkatalog für die Gestaltung von Touchphone-Apps ist, der in Folge bei der Evaluation verwendet wird, die Entwicklung eines Prototypen und eine Evaluation des implementierten Prototypen zum Einsatz.

Die Evaluation der Umsetzung der Biber-Fragen in einer App hat ergeben, dass sie zwar in der App verwendet werden können, aber ihre Struktur an die ganz eigenen Bedürfnisse von Touchphones (sehr kleine Bildschirme) angepasst werden müssen.

Die speziell für die Suche von Fragen zu einem Themengebiet oder Schlagwort entwickelte Suchfunktion funktioniert sehr gut. Fragen zu einem bestimmten Thema werden gefunden, weil sie einerseits als Volltextsuche umgesetzt wurde und andererseits der Aufbau der Biberfragen mit ihren Abschnitten - Das ist Informatik und Keywords - eine Verbindung zu Fachbegriffen und Konzepten der Informatik herstellen.

Abstract

Based on the newest developments in the field of smartphones, content for learning can be used on mobile phones in a sensible way. There is already an App for the Bebras Challenge a contest based on the subject of computer science that is premised on gamefication but is only implemented for Android-Systems.

The idea of this thesis is the development of a prototype for the Bebras Challenge that can be run on IOS- and Android-Systems. In difference to the the existing App the implemented prototype is not only a game but a tool that could be used in teaching. For this purpose teachers should be able to search for a specific subject or a special question. So a search-function was implemented.

Based on this idea two reasearch questions were selected:

- How and in which quality can the questions of the Bebras Challenge be implemented in a learning app?
- How useful is the implemented full text search-function in the questions of the Bebras Challenge to find a specific topic for teaching lessons?

To deal with the research questions scientific methods were used. A literature re-search on existing findings in the area of app design was made which was the basis of a guidelines-catalog for the design of touchphone-apps that was later used for the evaluation of the implementation of the questions from the Bebras Challenge. Next a prototype was implemented and was followed by the evaluation of the implemented questions from the Bebras Challenge and the evaluation of the implemented search-function.

The results of the evaluation of the implemented questions are, that they can be used in the mobile app but have to be adapted in structure to meet the special requirements of touchphones (very small displaysizes).

The specially designed search-function for searching for a subject or for a keyword is working just fine. Even if you search for a special topic bebras tasks are found because the search was implemented as full text search and because of the strukture of the bebras tasks with the sections - it's informatics and keywords - that establish a connection to technical terms and concepts of computer science.

Danksagung

An dieser Stelle möchte ich mich bei all jenen Menschen bedanken, die mich während meines Studiums und dem Verfassen dieser Diplomarbeit unterstützt haben:

Danken möchte ich in erster Linie Herrn Professor Gerald Futschek, der diese Arbeit betreut hat.

Ein besonderer Dank gilt meiner Familie und speziell meinen Eltern, die mir diese Ausbildung ermöglicht haben und ohne deren Unterstützung dieses Studium nicht möglich gewesen wäre, sowie meinem Freund, der mich immer wieder motiviert hat und mir unterstützend zur Seite gestanden ist.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Forschungsfragen	2
1.3	Ziel der Arbeit	2
1.4	Methodisches Vorgehen	2
1.5	Aufbau der Arbeit	4
2	Der Biber Wettbewerb	5
2.1	Geschichte	6
2.2	Daten und Fakten	7
2.3	Computational Thinking	8
2.4	Aufbau der Aufgaben für den Biber Wettbewerb	10
3	Mobile Geräte	13
3.1	Kategorien mobiler Telefone	16
4	Apps	19
4.1	Definition von Apps	19
4.2	Native Apps	20
4.3	WebApps	21
4.4	Hybride Apps	22
4.5	WebApps, native oder hybride Apps?	23
5	Usability, User Experience und Mobile Learning	25
5.1	Usability	25
5.2	User Experience	27
5.3	Mobile Learning	28
5.4	Entwicklung des Mobile Learnings	30
5.5	Microlearning	33
6	Gestaltungsrichtlinien	35
6.1	Allgemeines Konzept	35

6.2	Layout	37
6.3	Navigation	39
6.4	Design	42
6.5	Inhalt	48
6.6	Performance	53
7	Verwendete Technologien	55
7.1	Webtechnologien	55
7.2	Crossplattformtool	56
7.3	Speicher	57
8	Der Prototyp	61
8.1	Startseite	61
8.2	Navigationsseite	62
8.3	Contest	63
8.4	Fragen üben	64
8.5	Falsch beantwortete Fragen üben	65
8.6	Frage suchen	66
8.7	Resultate	68
8.8	Logout	68
8.9	Übersicht über die Funktionsweise des Prototypen	68
9	Evaluation	71
9.1	Umsetzung der Biberfragen	71
9.2	Evaluation Suchfunktion der BiberApp	93
10	Ergebnisse und Zusammenfassung	101
10.1	Ergebnisse Umsetzung der Biberfragen	101
10.2	Ergebnisse Suchfunktion	102
A	Unveröffentlichter Lehrplan Informatik AHS 5. Schulstufe	103
B	Unveröffentlichter Lehrplan Informatik Wahlpflichtfach AHS 6.-8. Schulstufe	107
	Glossar	113
	Literaturverzeichnis	117

Abbildungsverzeichnis

2.1	Biber	7
3.1	Klassifikationsbaum für mobile Geräte nach Schiefer und Decker [40]	14
3.2	Entwicklung des Handys von 1992 bis 2014	17
5.1	Unterschied zwischen Usability und User Experience nach Geis [16]	28
5.2	Entwicklung vom Distant zum Mobile Learning	30
8.1	Startseite	61
8.2	Navigationsseite	62
8.3	Startseite	63
8.4	Fragen üben	64
8.5	Richtig und falsch beantwortete Fragen	65
8.6	Falsch beantwortete Fragen üben	66
8.7	Möglichkeiten der Suche	67
8.8	Beispiel für Suche	67
8.9	Anzeige der Frage in Bereich der Suche	68
8.10	Übersicht des Prototypen	69
9.1	Frage und Antwortmöglichkeiten	74
9.2	Antwortmöglichkeiten	74
9.3	Login und Results	75
9.4	Dialog	76
9.5	Navigationsseite	77
9.6	Header des Prototypen	78
9.7	Frage und richtige Lösung	80
9.8	Biber der Informatik	81
9.9	verwendete Icons für Funktionen der BiberApp	82
9.10	verwendeten Icons für Steuerelemente	82
9.11	Textfelder in der App	83
9.12	Tastatur	84
9.13	Dialog	85
9.14	Feedback	86

9.15	Text in der BiberApp	88
9.16	Hervorhebung	89
9.17	Bild und Text in BiberApp 1	91
9.18	Bild und Text in BiberApp 2	92
9.19	Originale Frage	92
9.20	Fundamentale Ideen der Informatik nach Schwill [42]	95
9.21	Suchbeispiel 1: Schlagwort und Ergebnismenge	96
9.22	Suchbeispiel 1: Vorkommnisse des Schlagwortes	96
9.23	Suchbeispiel 2: Schlagwort und Ergebnismenge	97
9.24	Suchbeispiel 2: Vorkommnisse des Schlagwortes	97
9.25	Suchbeispiel 3: Schlagwort und Ergebnismenge	98
9.26	Suchbeispiel 3: Vorkommnisse des Schlagwortes	98
9.27	Suchbeispiel 4: Schlagwort und Ergebnismenge	99
9.28	Suchbeispiel 5: Schlagwort und Ergebnismenge	99
9.29	Suchbeispiel 6: Schlagwort und Ergebnismenge	100
9.30	Suchbeispiel 6: Vorkommnisse des Schlagwortes	100

Tabellenverzeichnis

6.1	fingerfreundliche UI-Komponenten	44
-----	--	----

Kapitel 1

Einleitung

1.1 Motivation

Für den Bereich der Informatik wurde 2004 die Bebras Challenge (international) in Österreich bekannt als Biber der Informatik ins Leben gerufen. Bei diesem handelt es sich um einen Wettbewerb zum Thema Informatik. Dabei liegt der Schwerpunkt der Fragen auf Konzepten der Informatik. Die Fragen des Contests sind dabei so formuliert, dass man kein explizites Informatikvorwissen besitzen muss, um sie beantworten zu können.

Seit 2004 finden jährlich Biber-Wettbewerbe in verschiedenen Ländern und auch Österreich statt, worauf in einem späteren Kapitel genauer eingegangen wird. Die Fragen zu vergangenen Wettbewerben sind online auf der Homepage des Biber der Informatik zu finden. Die internationale Adresse lautet: <http://bebras.org/> Die österreichische Adresse dazu ist: <http://www.ocg.at/de/biber>.

Durch die jüngsten Entwicklungen im Bereich der Smartphones, wie große Displays, größerer Datenspeicher und einfachere Bedienung können nun auch viele Inhalte sinnvoll auf die mobilen Geräte gebracht werden. So haben in den letzten Jahren unzählige Quiz-Apps und Lernspiele den Weg in den AppStore und in den Google Play Store gefunden. Daher erscheint es sinnvoll, den Inhalt des Biber Challenge anhand einer App für mobile Geräte, im besonderen iPhones und Android-Geräten, verwendbar zu machen. Der Fokus liegt auf iPhones und Android-Geräten, weil diese mit Abstand die am weitesten verbreiteten Smartphones sind und wahrscheinlich der Großteil der Zielgruppe der App ein Smartphone mit einem IOS- oder Android-Betriebssystem besitzt.

Wie schon zuvor erwähnt, existiert bereits eine Homepage mit Informationen zum Biber der Informatik, wo auch die Fragen früherer Wettbewerbe zum Download zu finden sind. Neben diesem Wettbewerb gibt es auch einen mit dem Schwerpunkt Mathematik [49] genannt der Känguru der Mathematik. Für diese Initiative wurde schon eine App - Mathe mit dem Känguru - entwickelt mit der Kinder für den nächsten Wett-

bewerb trainieren können. Es gibt bereits eine App für den Biber der Informatik die Bebras-App, die aber nur für Android-Systeme im Google Play Store zum Download zur Verfügung steht.

Der Unterschied zwischen der Bebras-App und der als Prototypen in dieser Arbeit entwickelten App liegt darin, dass die BiberApp nicht nur auf Android-Betriebssystemen verwendbar sein soll, sondern auch für das iPhone konzipiert ist. Weiters ist sie nicht hauptsächlich als Lernspiel angelegt, sondern enthält eine Suchfunktion für die gezielte Suche einer Frage oder Fragen einer Altersgruppe oder eines Schwierigkeitsgrades.

1.2 Forschungsfragen

Dem Trend der Lern-Apps und deren Einsatz im Unterricht folgend entstand die Idee den Prototypen einer App zu entwickeln, mit der man die Fragen des Biber Wettbewerbs üben kann. Diese App soll aber auch im Unterricht sinnbringend einsetzbar sein, was mit Hilfe einer Suchfunktion umgesetzt wurde, auf die in späteren Kapiteln noch genauer eingegangen wird. Basierend auf diesen Überlegungen sind folgende Forschungsfragen für die Arbeit entstanden:

1. Wie und wie gut können vorhandene Fragen des Biber Wettbewerbs in einer Lern-App abgebildet werden?
2. Wie gut einsetzbar ist eine Volltextsuche in den Aufgaben des Biber der Informatik um bestimmte Themen für den Unterricht zu suchen?

1.3 Ziel der Arbeit

Mit dieser Arbeit werden im Wesentlichen zwei Ziele verfolgt:

- Ein Ziel dieser Arbeit ist es, herauszufinden und zu evaluieren, wie gut die Fragen des Biber Wettbewerbs in einer Lern-App auf einem Smartphone umgesetzt werden können.
- Ein Zweites ist es, eine Aussage darüber machen zu können, wie gut die in dem Prototypen umgesetzte Suchfunktion im Unterricht einsetzbar ist.

Um diese Ziele der Arbeit zu erforschen, werden wissenschaftliche Methoden eingesetzt, auf die im nächsten Abschnitt noch genauer eingegangen wird.

1.4 Methodisches Vorgehen

Um die oben genannten Forschungsfragen zu beantworten, stützt sich diese Arbeit auf die folgenden drei wissenschaftliche Methoden:

- Literaturrecherche
- Prototyping und
- Evaluation

Literaturrecherche

Der Ausgangspunkt der Arbeit bildet eine Literaturrecherche schon bestehender Fachliteratur und Untersuchungen zu ähnlichen Themen, die sich speziell auf folgende Aspekte konzentriert:

- **mobile Geräte:** Dieser Abschnitt beschäftigt sich mit den verschiedenen mobilen Geräten, ihrer Verwendung und Kategorisierung.
- **mobile Applikationen (Apps):** Es wird behandelt welche Arten von Apps es gibt und welche Vor- und Nachteile sie jeweils mit sich bringen.
- **Mobile Learning:** In diesem Kapitel wird versucht ein Überblick über die unterschiedlichen Definitionen und Ausprägungen des mobilen Lernens zu geben und eine für den Kontext dieser Arbeit passende Definition von Mobile Learning zu finden.
- **Gestaltungsrichtlinien:** Dieser Teil der Arbeit befasst sich mit Richtlinien für das Gestalten einer App für Smartphones.

Aus dieser Literaturrecherche geht ein Richtlinienkatalog hervor, der in weiterer Folge als Basis für die Evaluation in der Arbeit genutzt wird.

Prototyp

Die zweite wissenschaftliche Methode der Arbeit ist die Entwicklung eines Prototypen, der die Fragen des Biber der Informatik in einer mobilen Applikation umsetzt. Die Entwicklung des Prototypen basiert vor allem auf den in der Literaturrecherche beleuchteten Themen wie Mobile Learning, mobile Applikationen, mobile Geräte und den Gestaltungsrichtlinien.

Evaluation

Bei der Evaluation werden zwei große Teilbereiche beleuchtet. Im ersten Teil werden die in der Literaturrecherche gefundenen Gestaltungsrichtlinien auf den entstandenen Prototypen angewendet und etwaiges Abweichen und Einhalten von Richtlinien diskutiert.

Der zweite Bereich der Evaluation konzentriert sich auf die im Prototypen umgesetzte Suchfunktion. Gestützt durch Erkenntnisse aus der Literatur den Biber der Informatik betreffend, wird die implementierte Volltextsuche argumentativ evaluiert.

1.5 Aufbau der Arbeit

Der erste Teil der Arbeit beinhaltet eine Einführung zum Biber der Informatik, dem Begriff des Computational Thinkings und dem Aufbau einer Aufgabe für den Biber Wettbewerb. Folgende Kapitel ermitteln den derzeitigen Stand im Bereich der mobilen Geräte, der Apps und des Mobile Learnings, die schon vorbereitend auf den eigentlichen Kernbereich der Arbeit auf das nachfolgende Kapitel Gestaltungsrichtlinien, den Prototypen und die Evaluation hinführen.

Der Abschnitt - Gestaltungsrichtlinien - beinhaltet eine Sammlung von Richtlinien aus dem Bereich des Mobile Learnings, der Usability und der User Experience aus verschiedensten wissenschaftlichen Quellen unterteilt in die Bereiche: Allgemeines Konzept, Layout, Navigation, Design, Inhalt und Performance, die beim Erstellen einer Touchphone-Apps beachtet werden sollten.

Die Kapitel Verwendete Technologien und Prototyp widmen sich ganz der Erstellung des Prototypen. Erstgenannter Abschnitt beschreibt kurz, die für den Prototypen verwendeten Technologien. Das Kapitel Prototyp beschreibt den entstandenen Prototypen in Bild und Text mit all seinen bereitgestellten Funktionen.

Den Abschluss der Arbeit bilden die Kapitel Evaluation und Ergebnisse und Zusammenfassung. Der Abschnitt Evaluation unterteilt sich in den Bereich, der sich mit der Umsetzung der Biber-Fragen in der App auseinandersetzt und der zweite Teil mit der Evaluation der Suchfunktion.

Kapitel 2

Der Biber Wettbewerb

Der Biber der Informatik hat, wie es die Initiatoren des Wettbewerbs auf ihrer eigenen Homepage [37] kundmachen, die Aufgabe Lehrerinnen und Lehrer und Schülerinnen und Schüler für die Informatik zu interessieren, sie ihnen näher zu bringen und sie bekannt zu machen. Den Biber der Informatik gibt es nicht nur in Österreich, er ist international als Bebras Challenge bekannt. Die Bebras Challenge wird in jedem Mitgliedsland abgehalten. Dabei werden Fragen an die Teilnehmerinnen und Teilnehmer gestellt, die ohne direktes Informatikwissen beantwortet werden können. Die Fragen beinhalten aber Themen wie:

- Information
- diskrete Strukturen
- Kalkulation
- Daten Verarbeitung und auch
- algorithmische Konzepte

Damit kann aber nicht nur ein Interesse für Informatik geweckt werden, sondern anhand der Fragen eine Begabung für Informatik getestet werden. Um den Biber der Informatik möglich zu machen gibt es Bebras International Task Workshops. Jedes Mitgliedsland reicht seine eigenen neuen Aufgaben ein und diese werden dort von allen diskutiert. Für die jeweiligen nationalen Wettbewerbe der Bebras Challenge gibt es einen Fragenpool, der für alle unterschiedlichen Länder gleich bleibt und verpflichtend ist und den Rest wählen die Verantwortlichen des Wettbewerbs aus dem vorher neu entstandenen Fragenpool aus.

Mittlerweile sind auch interaktive Aufgaben für die Bebras Challenge ein Teil des jährlichen Workshops. Das Internationale Bebras Board übernimmt die Koordination aller internationalen Bebras-Aktivitäten.

2.1 Geschichte

Die Begründerin der Bebras Challenge ist Professor Valentina Dagiene. 2004 wurde in Litauen die erste Bebras Challenge von ihr initiiert. Auch damals hatte sie schon das Ziel den Wettbewerb international zu machen. Nach diesem ersten Schritt traten bald darauf 2006 Estland, Deutschland, die Niederlande und Polen bei. 2007 kamen Österreich, Lettland und die Slowakei dazu. 2008 folgten Tschechien und die Ukraine. Italien trat 2009 bei und 2010 Finnland und die Schweiz. 2011 gab es erstmals eine Bebras Challenge in Frankreich, Ungarn und Slowenien.

In diesem Jahr gab es noch eine besondere Ausnahme zu verzeichnen, weil Japan die erste nicht europäische Bebras Challenge organisierte. Inzwischen ist der Wettbewerb noch internationaler geworden und weitere Länder haben sich angeschlossen.

Folgende Länder organisieren einen jährlichen Wettbewerb (Stand 2016):

- Australien
- Aserbaidshan
- Belgien
- Bulgarien
- Canada
- Deutschland
- England
- Estland
- Finnland
- Frankreich
- Iran
- Irland
- Island
- Israel
- Italien
- Japan
- Lettland
- Litauen
- Malaysia
- Macedonien
- Niederlande
- Neuseeland
- Österreich
- Pakistan
- Polen
- Russland
- Schweden
- Schweiz
- Serbien
- Slowakei
- Slowenien
- Spanien
- Südafrika
- Taiwan
- Türkei
- Tschechien
- Ukraine
- Ungarn
- USA
- Weißrussland
- Zypern

Die unterhalb aufgelisteten Länder planen eine Bebras Challenge zu organisieren:

- Ägypten
- Brasilien
- Indonesien
- Kasachstan
- Mexiko
- Singapur
- Südkorea
- Bosnien und Herzegowina
- Vietnam
- Rumänien
- Kroatien
- Mongolei

2.2 Daten und Fakten

Den Biber der Informatik, wie die Bebras Challenge in Österreich genannt wird, gibt es seit 2008. Das Markenzeichen für diesen Wettbewerb ist der Biber, wie er in Abbildung 2.1 angezeigt wird. Alle weiteren hier zusammengefassten Informationen sind genauer auf der Homepage des österreichischen Biber der Informatik nachzulesen [9].



Abbildung 2.1: Biber der Informatik

Zielgruppe

Die Zielgruppe des Biber der Informatik sind Schülerinnen und Schüler im Alter von ca. 8 bis 20 Jahren. Die Teilnahme beginnt für Schülerinnen und Schüler in der dritten Klasse Volksschule und ist bis zur 13. Schulstufe möglich, das entspricht der fünften Klasse einer höheren berufsbildenden Schule. Da es einen sehr großen Altersunterschied und damit auch unterschiedlichen Wissensstand gibt, wurden fünf verschiedene Alterskategorien angelegt.

- 3. - 4. Schulstufe (Volksschule)
- 5. - 6. Schulstufe (Unterstufe)
- 7. - 8. Schulstufe (Unterstufe)
- 9. - 10. Schulstufe (Oberstufe)
- 11. - 13. Schulstufe (Oberstufe)

Ablauf des Wettbewerbs

Der Wettbewerb wird im Rahmen der Schulen abgehalten, daher ist es notwendig, dass sich eine Lehrperson als Koordinator anmeldet. Erst danach kann diese Schülerinnen und Schüler für den Wettbewerb registrieren. Diese absolvieren den Wettbewerb dann meist online über PCs oder Laptops während des Unterrichts.

Der Wettbewerb umfasst normalerweise achtzehn Fragen, nur in den Volksschulen wird die Zahl der Fragen auf neun reduziert. Davon haben je sechs einen Schwierigkeitsgrad leicht, mittel oder schwer. Die Schülerinnen und Schüler haben dann 40 Minuten Zeit um die Fragen zu lösen.

Punktesystem

Man beginnt mit einem Punktepolster von 54 Punkten. Wird eine Aufgabe richtig beantwortet, werden Punkte hinzugefügt. Bei einer falschen Antwort gibt es einen Punktabzug. Beantwortet man eine Aufgabe nicht, bleibt der Punktestand gleich. Die Anzahl der Punkte, die man gewinnen oder verlieren kann, hängt vom Schwierigkeitsgrad der Frage ab. Im schlimmsten Fall kann man also null Punkte erreichen im besten Fall 216.

	<i>leicht</i>	<i>mittel</i>	<i>schwer</i>
richtig	6 Punkte	9 Punkte	12 Punkte
falsch	-2 Punkte	-3 Punkte	-4 Punkte

2.3 Computational Thinking

Der vollständige Titel der Bebras Challenge lautet: Bebras - International Challenge on Informatics and Computational Thinking. Im Vordergrund stehen das Problemlösen und wie im Titel enthalten ist, das Computational Thinking. Dieser Begriff wurde 2006 von Jeannette Wing [51] eingeführt.

In ihrem Artikel beschreibt sie das Computational Thinking als etwas sehr Grundlegendes, das im 21. Jahrhundert zum Skillset jedes Menschen gehören wird und nicht

nur auf die Informatik beschränkt ist, sondern auch in alle anderen Disziplinen übergreift. Daher sollten laut der Autorin Kinder nicht nur im Lesen, Schreiben und Rechnen unterrichtet werden, sondern auch im Computational Thinking. In einem weiteren Artikel zu diesem Thema [52] gibt Wing folgende Definition für Computational Thinking an:

„Computational thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent.“ [52]

Sie beschreibt hier das Computational Thinking als die mentale Aktivität, die passiert, wenn ein Problem formuliert wird, um eine berechenbare Lösung zu finden. Die Lösung kann entweder vom Menschen, der Maschine oder einer Kombination von beiden ausgeführt werden.

Ein Problem ist in diesem Sinne nicht nur ein mathematisches, klar definiertes, sondern auch eines aus dem Alltagsleben. Laut der Autorin überschneidet sich das Computational Thinking mit dem logischen Denken, dem System-Denken, dem algorithmischen Denken und dem parallelen Denken. Für Wing ist der wichtigste Denkprozess beim Computational Thinking die Abstraktion. Sie wird dafür verwendet um:

- Muster zu definieren,
- spezifischen Instanzen zu generalisieren,
- zu parametrisieren,
- ein Objekt für mehrere stehen zu lassen,
- um wichtige Eigenschaften von Objekten zusammenzufassen und dabei Unwichtiges auszublenden und
- mit Komplexität umzugehen.

Außerdem bedient sich das Computational Thinking am mathematischen Denken und am technischen Denken. Zum Computational Thinking gehört das Lösen von Problemen, das Designen von Systemen, aber auch das Verstehen des menschlichen Verhaltens. Hilfsmittel dafür sind fundamentale Konzepte der Informatik. Mit Hilfe des Computational Thinking kann ein schwieriges Problem in eines umgewandelt werden, das man schon kennt und bearbeiten kann.

Die Autorin grenzt das Computational Thinking von der Computer Literacy und dem Programmieren in einer bestimmten Sprache ab, denn sie ist der Meinung, dass es neben Soft- und Hardware auch die fundamentalen Konzepte der Informatik sein werden, die uns im Alltag begleiten um Probleme zu lösen, mit anderen zu kommunizieren und zu interagieren. Diese Konzepte zeichnen sich laut Futschek [15] durch ihre langfristige Bedeutung für das Alltagsleben und alle wissenschaftlichen Disziplinen aus.

2.4 Aufbau der Aufgaben für den Biber Wettbewerb

Die Aufgaben des Biber der Informatik sind, wie schon vorhin erwähnt, so gestaltet, dass sie durch reines Nachdenken gelöst werden können. Für Dagiene und Futschek [10] ist der beste Weg das Nachdenken zu fördern Probleme zu lösen. Für das Problemlösen sind aber Interesse und Einsatz auf der Seite der Schülerinnen und Schüler gefragt, daher sollten interessante Aufgaben aus möglichst vielen Bereichen der Wissenschaft und des Lebens gewählt werden.

Die Aufgaben des Biber Wettbewerbs wurden in folgende Kategorien unterteilt:

- Informationsverständnis
- Algorithmisches Denken
- Verwendung von Computersystemen
- Strukturen, Muster und Anordnung
- Puzzles
- IKT und Gesellschaft

Weiters geben sie Kriterien für gute Aufgaben für den Biber der Informatik an, die erfüllt sein müssen, damit die Aufgabe in den Aufgabenpool aufgenommen werden kann:

- Die Aufgabe soll in drei Minuten lösbar sein.
- Die Problemstellung soll leicht verständlich sein.
- Die Aufgabe kann auf einer Seite präsentiert werden.
- Es werden keine zusätzlichen Mittel zur Lösung gebraucht.
- Die Aufgabe ist unabhängig von speziellen Systemen.

Die Autorin und der Autor geben an, dass es für jüngere Teilnehmerinnen und Teilnehmer in manchen Fällen einfacher ist, wenn die Aufgabenstellung in eine passende Geschichte verpackt wird.

Pohl und Hein [38] beschäftigen sich ebenfalls damit, wie gute Aufgaben für den Biber der Informatik gestaltet sein sollten. Sie fassen die oben genannten fünf Kriterien in eine Richtlinie zusammen:

„A task needs to be understood quickly, correctly and without prior knowledge.“ [38]

Die Autoren geben in ihrer Arbeit auch den grundlegenden Aufbau von Aufgaben für den Biber der Informatik an. Laut ihnen besteht ein Bebras task aus zwei großen Abschnitten. Einerseits dem in-contest - Teil, der während des Wettbewerbs angezeigt wird und dem after-contest - Teil, der nach der Beantwortung der Aufgabe angezeigt wird. Der in-contest-Bereich besteht aus folgenden Abschnitten:

- einleitender Text
- zentrales Bild
- Frage
- Antwortmöglichkeiten

Zum after-contest-Bereich gehören die Abschnitte:

- Text zur Antwort mit Erklärung welche Antwort korrekt ist und warum. Ebenso wird erklärt, warum die anderen Antworten nicht stimmen.
- Text der eine Erklärung liefert, wie die Aufgabe mit der Informatik zusammenhängt und welches Thema/Konzept behandelt wird.

Das auf der Seite der Bebras Challenge [37] zur Verfügung gestellte Template für die Entwicklung von Aufgaben enthält noch weitere Abschnitte, wie zum Beispiel die Keywords mit denen eine Verbindung zwischen der Aufgabe und Konzepten und Fachbegriffen der Informatik hergestellt werden können.

Kapitel 3

Mobile Geräte

Claudia De Witt [53] spricht in ihrer Definition vom Mobile Learning von mobilen Geräten, auf denen das M-Learning basiert. In ihrer Beschreibung von diesen nennt sie insbesondere

- Netbooks
- Tablets und
- multimediafähige Smartphones

Eine genaue Definition eines mobilen Gerätes haben Inostroza, Rusu, Roncagliolo und Rusu in ihrem Artikel so beschrieben:

„A mobile device is a small electronic appliance, with some processing capabilities, with permanent or intermittent connection to a network.“ [23]

Ihrer Definition entsprechend sind neben Handys und Laptop/Netbooks auch MP3-Player, und GPS-Geräte mobile Geräte. Notebooks und Netbooks wurden schon im Bereich des E-Learnings verwendet und zählen zu den klassischen mobilen Geräten. Ein besonderer Meilenstein in der Weiterentwicklung der mobilen Geräte war sicher die Entwicklung von Tablets und den heute nicht mehr wegzudenkenden Smartphones.

Schiefer und Decker [40] geben in ihrem Artikel „Taxonomy for mobile Terminals“ folgende Kriterien an um mobile Geräte zu kategorisieren:

- Größe und Gewicht
- Eingabemöglichkeiten
- Ausgabemöglichkeiten
- Performance
- Art der Nutzung

- Kommunikationsmöglichkeiten
- Art des Betriebssystems
- Erweiterbarkeit

Die Autoren unterscheiden aufgrund dieser Kriterien folgende Klassen mobiler Geräte, die sie auch in einem Klassifikationsbaum darstellen.

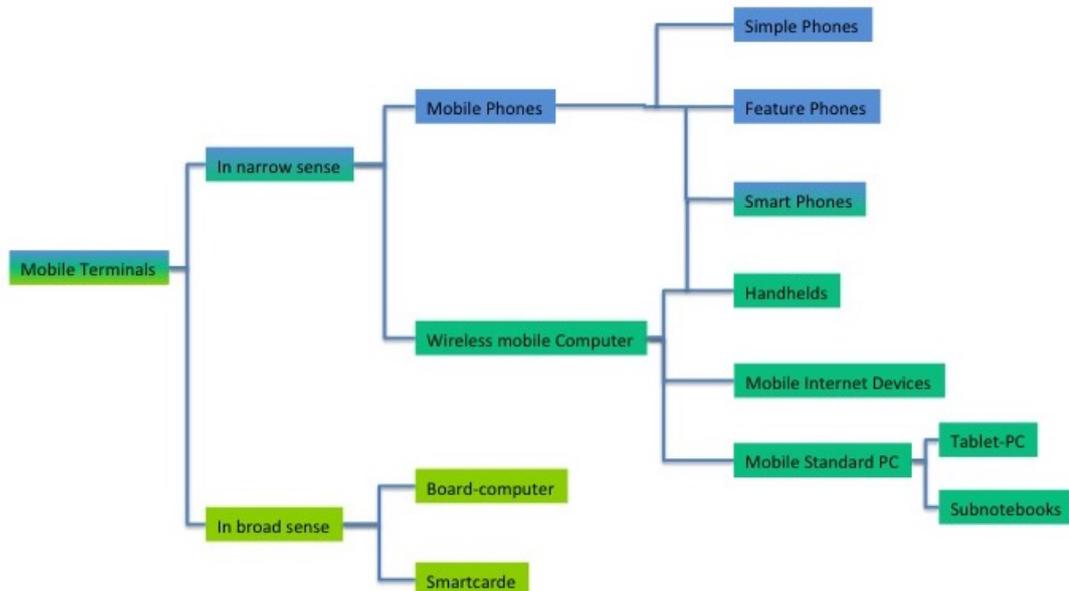


Abbildung 3.1: Klassifikationsbaum für mobile Geräte nach Schiefer und Decker

Der Farbverlauf in Abbildung 3.1 lässt darauf schließen, dass die Grenzen zwischen den Kategorien nicht immer klar definiert sind. Smartphones sind einerseits Handys aber andererseits gehören sie auch zur Kategorie „Wireless Mobile Computers“. Die einzelnen Geräte werden laut den oben angeführten Kriterien von den Autoren wie folgt beschrieben.

1. **Mobile Standard PCs:** Zu dieser Kategorie gehören Laptops und Notebooks, Netbooks, die von Schiefer und Decker [40] als Subnotebooks bezeichnet werden, Tablet-PCs und Ultra Mobile PCs.
 - Das *Netbook* ist eine kleinere, leichtere Version des Notebooks. Es fehlen aber meiste CD/DVD Laufwerk und sie haben auch eine kleinere Tastatur.
 - Unter einem *Ultra Mobile PC* verstehen die Autoren einen Tablet-PC, der nur 8 Inches groß ist, maximal 900 Gramm wiegt und eine Bildschirmauflösung von mindestens 800x480 Pixel hat.

Bei Bedarf kann eine Tastatur angeschlossen werden. Diese UMPCs können meist mit einem Linux- oder Windows-Betriebssystem verwendet werden und auch mit kompatiblen Softwarepaketen. Die unterscheidenden Kriterien zwischen einem UMPC und einem herkömmlichen Tablet-PC sind die Größe und das Gewicht, periphere Geräte wie CD/DVD Laufwerk und wie das Gerät eingesetzt wird.

- *Notebooks* gehören ebenfalls zu den mobilen Geräten brauchen aber immer einen Platz, wo sie aufgestellt werden können. Die Benutzung im Stehen und Gehen ist sehr umständlich.
- *Tablet PCs* gehören zu den mobilen Standard Computern unterscheiden sich aber wesentlich von ihnen durch ihren Touchscreen. Sie können daher auch im Gehen und Stehen, sogar mit nur einer Hand bedient werden.

2. **Mobile Internet Geräte:** Die mobilen Internet Geräte (Web-Tablet, Thin Client) sind den mobilen Standard PCs sehr ähnlich, haben aber einen eingeschränkten Funktionsumfang. Sie sind meist wie Handhelds ausgestattet und sind nach dem Einschalten sofort verwendbar. Mobile Internet Geräte werden hauptsächlich für internetbasierte Aufgaben verwendet und brauchen daher ständig eine Internet-Verbindung.

- *Web-Tablets* haben so wie Tablet-PCs einen Touchscreen.
- *Thin Clients* werden üblicherweise mit einer Tastatur bedient.

3. **Handhelds (auch Person Digital Assistent PDA):** sind mit einer Hand verwendbare kleine Computer. Diese Geräte haben sich aus den Kalendern, Adressbüchern und Organizern entwickelt und haben daher auch die Funktion persönliche Daten zu organisieren. Mit der Zeit haben die Handhelds mehr Funktionalität bekommen und enthalten nun meist auch Multimedia-Applikationen.

Als mobiles Gerät unter den Handhelds - laut Kriterien oben - gelten aber nur jene, die eine Internetverbindungsmöglichkeit besitzen. Sowie auch die Web-Tablets sind sie nach dem Start sofort einsetzbar und besitzen meistens einen Touchscreen oder sonst eine Text-Tastatur und Navigationsbuttons. Der Unterschied zu Handys ergibt sich darin, dass das Handheld keine Verbindung über GSM oder UMTS herstellen kann.

4. **Smartphones:** Die Autoren beschreiben das Smartphone als ein Kombination aus Feature Phone und einem Handheld. Der Unterschied zum Handheld ist eine kleinere Größe und ein eingeschränkterer Funktionsumfang.

Schiefer und Decker schreiben dieser Kategorie alle mobilen Computer zu, die einem Handheld ähneln, aber über die Telefonnetze ebenfalls eine Internet-Verbindung herstellen können. Andererseits gehören laut ihnen auch Feature Phone

ähnliche Geräte dazu, die mehr Eingabemöglichkeiten bereitstellen (zB.: Touchscreen oder Tastatur).

5. **Feature Phones** sind mobile Telefone, die ein größeres Display und größeren Funktionsumfang haben. Der Hauptzweck eines Feature Phones ist die Kommunikation, sowohl die über die Sprache, als auch die über Text (SMS, MMS, Email, usw.).

Der Datenaustausch dieser Geräte läuft meist über GPRS oder UMTS ab und sie haben mehr Eingabemöglichkeiten, die aus einer Nummerntastatur und wenigen zusätzlichen Buttons bestehen. Manche der Geräte stellen einen Webbrowser bereit und ermöglichen es Multimedia-Inhalte darzustellen. Zusätzliche Software kann meist über die Java ME, die fast von jedem Gerät unterstützt wird, auf das Gerät geladen werden.

6. **Einfache mobile Telefone:** Die üblichen mobilen Telefone haben als Hauptverwendungszweck die klassische Kommunikation über gesprochene Sprache. Auch die textbasierte Kommunikation über SMS, die sich im Laufe der Zeit entwickelt hat, ist nicht mehr wegzudenken. Sie bieten keine Möglichkeit im Internet zu surfen oder Emails zu empfangen. Die Displays dieser Geräte haben sich zunehmend verbessert, sie sind aber im Vergleich zu den Feature Phones immer noch kleiner. Es gibt auch nur sehr wenig Möglichkeiten das Gerät mit Hilfe von Software zu erweitern, was sich aus der geringen Prozessorleistung und Speicherkapazität ergibt.

Mobile Geräte, die die Autoren mit ihren Kriterien nicht kategorisieren können sind Geräte wie E-bookreader, Navigationsgeräte, BarcodeScanner, GPRS-Geräte usw.

3.1 Kategorien mobiler Telefone

Da diese Arbeit besonders auf den Einsatz einer App auf einem Smartphone abzielt möchte ich auch auf die von Budiu und Nielsen beschriebenen drei Kategorien von derzeit verwendeten Handys eingehen [32]. Die Autorin und der Autor betonen die Wichtigkeit einer solchen Kategorisierung im Zusammenhang mit der „Mobile User Experience“. Nielsen und Budiu unterscheiden also sogenannte Barren-Handys, Smartphones und Touchphones, wobei sich ihre Definitionen mit den oben vorgestellten teilweise überschneiden, aber auch unterscheiden.



Abbildung 3.2: Entwicklung des Handys von 1992 bis 2014

Barren-Handys

Budiu und Nielsen nennen die schon oben erwähnten einfachen mobilen Telefone - Barren-Handys. Diese Art der Mobiltelefone sind laut der Autorin und dem Autor sehr einfach aufbereitet und verstehen sich als Handy im klassischen Stil mit einem sehr kleinen Display und einer üblichen Handytastatur, bei der die Zahlentasten mit mehreren Buchstaben belegt sind. Diese beiden Merkmale, die für Barren-Handys bezeichnend sind, sind aber auch ihre Schwäche. Durch die kleine Tastatur und dem winzigen Bildschirm ist eine Verwendung zur Kommunikation mit Webseiten sehr beschwerlich. Budiu und Nielsen beschreiben die Usability als „grauenhaft“. Diese Art der Handys sind derzeit am weitesten verbreitet und werden vor allem in Entwicklungsländern eingesetzt.

Smartphones

Budiu und Nielsen [32] teilen in ihrer Definition von mobilen Telefonen die Kategorie Smartphones von Schiefer und Decker in Smartphones und Touchphones auf. Die vorher gegebene Definition von Schiefer und Decker [40] beinhaltet aufgrund des Entstehungszeitraumes noch nicht die neuen Entwicklungen im Bereich der mobilen Telefone (z.B.: Touchscreens). Sie beschreiben daher ein Smartphone als ein Handy, dass

ein etwa mittelgroßes Display besitzt und über eine Tastatur verfügt, die von A-Z alle Buchstaben beinhaltet.

Budiu und Nielsen weisen darauf hin, dass frühe BlackBerry-Modelle ein gutes Beispiel liefern. Sie geben an, dass die meisten dieser Modelle über 3G- Internetkonnektivität oder WLAN verfügen. Auch bei diesen Geräten beurteilen sie die Usability als nicht gut [32].

Touchphones

Unter Touchphones verstehen Budiu und Nielsen, beispielsweise das iPhone, verschiedene Android-Handys und viele weitere Modelle. Sie nennen mehrere Merkmale, die für diese Kategorie der Handys bezeichnend sind:

- **Display:** Das Display nimmt fast den gesamten Teil des Handys ein. Es ist ein sogenannter Touchscreen und über Gesten und Hardwarebuttons kann direkt interagiert werden. Auch die Tastatur wird direkt auf dem Touchscreen angezeigt.
- **Internetkonnektivität:** Bei der Internetverbindung greifen solche Geräte nicht nur auf 3G zurück, sondern schon auf bessere, schnellere Technologien. Am schnellsten agieren diese Mobiltelefone über die WLAN-Verbindungen.

Touchscreens haben im Bereich der Mobilen User Experience am besten abgeschnitten, aber laut Nielsen ist die Usability bei weitem noch nicht zufriedenstellend. Eine Abhilfe kann hier nur durch die Verwendung von Suchmaschinen oder einer sehr guten speziell für mobile Zwecke designte Webseite oder App geschaffen werden.

Kapitel 4

Apps

Apps sind heutzutage in aller Munde und werden auf den jeweiligen Mobiltelefonen häufig verwendet. Nun stellt sich die Frage, was eine App wirklich ist. Außerdem ist App nicht gleich App und in Fachkreisen wird zwischen nativen Apps, Web-Apps und hybriden Apps unterschieden. Budiu und Nielsen [32] haben auch hier die Merkmale der verschiedenen Arten zusammengefasst.

4.1 Definition von Apps

Dass Apps unseren Alltag mittlerweile erobert haben, lässt sich nicht verleugnen. Fast jeder kennt Apps durch die Verwendung auf dem eigenen Telefon und verwendet täglich mindestens eine davon. Die feste Verankerung dieses Begriffs in unserer Gesellschaft zeigt sich auch dadurch, dass er schon Einzug in unsere Lexika und Wörterbücher gefunden hat, wie das nachfolgende Zitat aus dem Oxford Dictionary zeigt. Eine App ist laut diesem Lexikon:

„[...] a self-contained program or piece of software designed to fulfill a particular purpose; an application, especially as downloaded by a user to a mobile device.“ [1]

Das Online-Magazin für die Digitalwirtschaft Gründerszene [18] stellt eine detaillierte Definition des Begriffs Apps zur Verfügung. Sie führen den Begriff auf das englische Wort „application“, im deutschen „Anwendung“, zurück, das für alle Arten von Computerprogrammen steht.

Im Deutschen hat sich der Begriff App als Beschreibung einer Software etabliert, die auf mobile Endgeräte heruntergeladen wird. Laut den Autoren unterscheiden sich Apps von herkömmlichen Applikationen dadurch, dass Benutzerinnen und Benutzer unmittelbar einen Mehrwert für sich erhalten, wenn sie die App einsetzen. Ein weiterer Unterschied zu normalen Anwendungen liegt in der eingeschränkten Funktiona-

lität. Welche verschiedenen Arten der Apps basierend auf der Unterscheidung durch Technologie es gibt, werden in den nächsten Kapiteln behandelt.

4.2 Native Apps

Native Apps werden immer für eine bestimmte Plattform entwickelt. Eine native App für das iPhone wird also beispielsweise für iOS entwickelt und hält sich an deren Konventionen. Die Installation einer solchen App auf dem mobilen Gerät erfolgt immer über einen speziellen App Store (z.B. Apple Store). Zum Starten der App wird bei der Installation ein Symbol auf dem Homescreen des Handy platziert. [32]

Vorteile nativer Apps

Laut Büttner und Simmons [8] sowie Franke und Ippen [13] brillieren native Apps im Gegensatz zu den beiden anderen Arten von Apps vor allem im Bereich von Performance, Schnelligkeit und Effizienz. Ein weiterer Vorteil wie Büttner und Simmons beschreiben, ist die Nutzung von Inhalten ohne dass eine Verbindung zu einem Telefonnetz oder einem WLAN-Netz besteht.

Was native Apps im Vergleich zu WebApps hervorhebt, ist die Verfügbarkeit in einem App Store. Dort ist die App für alle sichtbar und kann leicht gefunden werden. Die Vermarktung übernimmt sozusagen der App Store durch eine Kategorisierung und Hervorhebung von für die Userinnen und User interessanten Apps. Für die Entwickler liegt der Vorteil des App Stores daran, dass es einen bekannten Bezahlsmechanismus gibt, dem die Userinnen und User vertrauen. Franke und Ippen fügen noch hinzu, dass der Entwickler keinen eigenen Download- und Installationsprozess implementieren muss, da auch das durch den App Store bereitgestellt wird.

Mit nativen Apps ist es auch möglich, die auf dem Gerät vorhandene Hardware anzusprechen, wie zum Beispiel die Kamera, GPS aber auch auf dem Gerät gespeicherte Daten wie Foto-Alben.

Nachteile nativer Apps

Büttner und Simmons [8] zeigen aber auch die Nachteile von nativen Apps auf. Durch den App Store wird die App in direkter Konkurrenz zu anderen Produkten angeboten, was einen direkten Vergleich einfacher macht. Außerdem kann es im App Store negative Bewertungen durch die User geben, was den Verkauf der App ins Stocken bringen kann. Der App Store hat gewisse Regulationen, die sich auf Kosten, Zeitmanagement und den Inhalt auswirken können. Der App Store muss seine Zustimmung zum Veröffentlichlichen der App geben und das kann auch einige Zeit dauern. Außerdem verlangt der App Store einen Teil des Gewinns pro Verkauf der App.

Der größte Nachteil im Vergleich zu mobilen Apps liegt in der Erreichbarkeit der Userinnen und User. Eine native App muss für jedes Betriebssystem eigens implementiert werden.

tiert werden, was die Kosten für die Entwicklung erheblich erhöht. Franke und Ippen [13] fügen noch zwei weitere Nachteile nativer Apps hinzu. Sie führen die Einstiegs-kosten an, die meist erst anfallen, wenn man die App testen oder verkaufen möchte. Es ist bei nativen Apps auch nicht möglich, das Userverhalten nachzuvollziehen. Google Analytics bietet für WebApps diese Möglichkeit.

4.3 WebApps

Laut Franke und Ippen [13] besteht das Wort WebApp aus zwei Wörtern „Web“ und dem Wort „App“, wobei „Web“ für das World Wide Web steht und „App“ für „application“, was im Deutschen „Anwendung“ bedeutet. Sie heben hervor, dass sich eine solche WebApp von einer herkömmlichen Internetseite durch ihren Aufbau unterscheidet. Die WebApps basieren eher auf dem Softwareschema (Dateneingabe → Verarbeitung → Datenausgabe).

Auch die Verwendung softwaretypischer Bedienelemente wie Symbolleisten, Buttons, Dialogfelder, Eingabefelder und Formulare unterscheidet WebApps von Webseiten. WebApps funktionieren, wie Internetseiten auf jedem mobilen Gerät. Sie werden in der gängigen Programmiersprache HTML5 entwickelt, sind aber trotzdem plattform-spezifisch, weil unterschiedliche Browser verschiedene Versionen von HTML5 verwenden. Die App kann durch das Eingeben einer URL, wie beim Öffnen einer Webseite, gestartet werden.

Vorteile von WebApps

Büttner und Simmons [8] sehen die Vorteile einer WebApp in ihrer Unabhängigkeit von Betriebssystemen. Da sie auf bekannten Internet-Browser-Technologien basiert, kann sie von jedem internetfähigen Smartphone aufgerufen werden. Wenn Benutzerinnen und Benutzer die Webseite schon von der Benutzung ihrer Computer kennen, werden sie sie wahrscheinlich auch über ihr Smartphone aufrufen.

Die WebApp ist nicht auf einen App Store angewiesen und muss sich auch nicht um Release Zyklen kümmern. Updates können sofort gemacht werden und man muss nicht erst auf die Zustimmung des App Stores warten. Da die App nicht über den Store vertrieben wird, fallen auch die Kosten dafür weg. Franke und Ippen [13] heben drei Vorteile für die Benutzer von WebApps hervor.

- **Installation:** Die WebApp muss nicht wie andere Apps installiert werden und kann über den Browser gestartet werden. Budiu und Nielsen sehen diesen von Franke und Ippen genannten Vorteil eher als etwas Nachteiliges. Das Anlegen eines Lesezeichens für diese App auf dem Homescreen ist das was für Budiu und Nielsen einer Installation am Nächsten kommt.

- **Updates:** Benutzerinnen und Benutzer müssen keine manuellen Updates machen, weil bei jedem Aufruf der WebApp die neueste Version gestartet wird.
- **Plattformunabhängigkeit:** Hier stimmten Franke und Ippen [13] mit Büttner und Simmons [8] überein, dass die Plattformunabhängigkeit einen großen Nutzen bringt. Erstere betonen hier aber speziell den Nutzen für die Benutzerinnen und Benutzer der App.

Nachteile von WebApps

Ein großer Nachteil von WebApps ist laut Büttner und Simmons [8], dass sie nicht offline funktionieren. Auch spezielle Design-Konventionen, die für jedes Betriebssystem eigen sind, können nicht berücksichtigt werden, weil die WebApps auf allen Betriebssystemen verwendet werden sollen. Budiu und Nielsen nennen noch einen zusätzlichen Nachteil von WebApps. Laut ihnen können sie die Funktionalität des Mobiltelefons nur eingeschränkt ausnützen. [32]

4.4 Hybride Apps

Wie man schon anhand des Names vermuten kann, sind hybride Apps eine Verschmelzung von nativen Apps und Web-Apps. Sie sind grundsätzlich native Apps und werden auch direkt über einen App Store heruntergeladen. Marco Bergmann von der Agentur Flyacts [7] beschreibt hybride Apps als Anwendungen, die auf HTML5, CSS3 und JavaScript basieren. Ein Container verschmilzt dann WebApp mit nativer App. Es gibt einige verschiedene Container wie zum Beispiel Phonegap [7].

Wie Budiu und Nielsen beschreiben, funktioniert das Starten ebenfalls über ein Symbol auf dem Homescreen des mobilen Geräts. Der Unterschied ist, dass sie einen Browser einer anderen App nutzen. Diese hybriden Apps können auch auf schon bestehende Webseiten angewendet werden und somit der Arbeitsaufwand für die Entwicklung einer nativen App gespart werden [32]. Eine hybride App kann laut Marc Bergmann [7] auch auf die Hardware des Smartphones zugreifen. Das wird dadurch möglich, dass zum Beispiel Adobe Phonegap die Kommunikation über eine Bibliothek zwischen JavaScript und der Sprache des Betriebssystems ermöglicht.

Vorteile hybrider Apps

Im Großen und Ganzen könnte man sagen, dass hybride Apps die Vorteile von nativen und WebApps vereinen. Der Vollständigkeit halber möchte ich sie hier aber noch einmal anführen, wie sie von Marco Bergmann [7] gesammelt wurden.

- **Plattformunabhängigkeit:** Dadurch, dass die App nicht an eine bestimmte Plattform gebunden ist, können Kosten bei der Entwicklung eingespart werden.

- **Verwendbarkeit der Hardwarefunktionen:** Durch die vom Container bereitgestellte Bibliothek kann auf die Hardware des Gerätes (z. B.: Kamera) zugegriffen werden.
- **Bedienbarkeit:** Durch viele Verbesserungen im User Interface Design sind diese Apps sehr gut bedienbar.
- **Vertrieb im App Store:** Durch ihren besonderen Aufbau kann eine hybride App wie eine native App im App Store vertrieben werden.
- **Aktualisierungen:** Aktualisierungen der App können in Echtzeit passieren.
- **Updates:** Auch Updates können schnell umgesetzt werden.

Nachteile hybrider Apps

Marco Bergmann [7] listet in seinem Artikel vier Nachteile von hybriden Apps auf:

- **Arbeitsspeicher:** Leider belegen hybride Apps einen großen Teil des Arbeitsspeichers.
- **Apps mit viel Arbeitsspeicherverbrauch:** Bei solchen Apps ist die Reaktion von hybriden Apps verlangsamt.
- **User Interface:** Betriebssystemspezifisches User Interface kann nicht verwendet werden.
- **JavaScript:** JavaScript ist keine höhere Programmiersprache.

4.5 WebApps, native oder hybride Apps?

Nun stellt sich die Frage, welche Art von App man verwenden soll. Büttner und Simmons [8] empfehlen WebApps zu verwenden, wenn man möglichst viele Kunden damit erreichen möchte. Es existieren viele verschiedene Betriebssysteme für Smartphones und mit der plattform unabhängigen Art der Apps kann man die größte Zahl an Smartphone-Userinnen und -Usern erreichen. Ein besonderer Anreiz ist auch, dass damit bei der Entwicklung der App Kosten gespart werden können. Die Benutzerinnen und Benutzer haben außerdem die Möglichkeit, die Web-Suche wie gewohnt zu verwenden.

Bei nativen Apps sind sich Büttner und Simmons [8] sowie Franke und Ippen [13] in ihrer Meinung zur Verwendung einig. Bei sehr komplizierten Interaktionsvorgängen und einem sehr großen Arbeitsspeicherverbrauch sollte man eine native App entwickeln. Auch wenn man die Hardware des Gerätes in größerem Ausmaß nutzen will, empfiehlt es sich diese Art der App zu implementieren. Ein Beispiel für eine App, die als native App umgesetzt werden sollte, ist das Spiel Angry Birds.

Hybride Apps werden speziell für Shop Systeme, Mobile Business und Mobile Marketing sowie Geo Location und Controlling eingesetzt, wie Marco Bergmann [7] in seinem Artikel schreibt. Diese Art der Apps kommen vor allem dann zum Einsatz, wenn eine ständige Aktualität bei Individualisierung der App wichtig ist. Man kann damit auch eine breite Masse erreichen, aber nicht in dem Ausmaß, wie es WebApps können.

Kapitel 5

Usability, User Experience und Mobile Learning

Neue mobile Technologien haben unser Leben erobert, speziell die Entwicklung von leistungsfähigen mobilen Geräten, wie zum Beispiel Smartphones, hat dazu beigetragen, dass wir wie Lam, Yau und Cheung [56] es beschreiben immer und überall mit anderen kommunizieren können, Zugriff auf Information haben, uns Wissen aneignen können, auch eigene Gedanken und Ideen aufzeichnen und sie danach mit anderen teilen können. Parallel dazu haben sich Begriffe wie Usability und User Experience entwickelt, die in den nachfolgenden Abschnitten kurz umrissen werden sollen.

5.1 Usability

Der Begriff der Usability existiert mittlerweile schon sehr lange. In den Abschnitten Definition des Usability Begriffs und Usability und Smartphones wird näher auf die für diese Arbeit wichtigen Aspekte der Usability eingegangen.

Definition des Usability Begriffs

Es existieren zu diesem Begriff jede Menge unterschiedliche Definitionen. Hier soll eine der anerkanntesten aus der ISO 9241-11 Norm [25] genannt werden. Usability wird darin definiert als:

„[...] 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.“ [25]

Nielsen [31] beschreibt in seinem Artikel im Buch „User Experience Re-Mastered“ die Usability als einen Komplex aus den folgenden fünf Dimensionen:

- **Erlernbarkeit:** Eine Applikation oder ein System sollte so gestaltet sein, dass es intuitiv nutzbar und leicht erlernbar ist.
- **Effizienz:** Die Effizienz in der Verwendung sollte bei einer Applikation im Vordergrund stehen, damit Benutzer und Benutzerinnen effizient und produktiv arbeiten können.
- **Memorierbarkeit:** Eine Applikation oder ein System sollte so gestaltet sein, dass Benutzerinnen und Benutzer, die es nur sporadisch verwenden auch nach längerem nicht benutzen noch wissen, wie man damit arbeitet.
- **Fehler:** Eine geringe Fehleranfälligkeit ist das Ziel einer Applikation oder eines Systems. Es sollte nur wenig Möglichkeiten geben um Fehler machen zu können. Wenn sie aber trotzdem passieren, sollten sie leicht wieder rückgängig gemacht werden können. Schwerwiegende Fehler dürfen überhaupt nicht passieren.
- **Zufriedenheit:** Benutzerinnen und Benutzer sollten die Applikation/das System gerne verwenden wollen und die Verwendung sollte Zufriedenheit auslösen.

Usability und Smartphones

Budiu und Nielsen [32] nennen in ihrem Buch „Mobile Usability“ folgende Einschränkungen des Smartphones, die sich schwerwiegend auf die Usability auswirken:

- **Display:** Um die Mobilität zu gewährleisten sind Smartphones so gebaut, dass sie einfach in eine Hosen- oder Jackentasche passen. Daraus resultiert ein sehr kleines Display. Auf diesem kann natürlich nur Inhalt in geringstem Ausmaß auf einmal angezeigt werden. Es ist daher schwierig für die Benutzerinnen und Benutzer die Information in ihrer Gesamtheit zu erfassen. Auch die Interaktion von Userinnen und Usern mit der Applikation ist auf kleinen Displays sehr eingeschränkt.
- **Dateneingabe:** Nielsen und Budiu [32] sehen in der Texteingabe auf dem Smartphone die größte Hürde für die Usability. Aus ihren Untersuchungen geht hervor, dass das Tippen und das Klicken von Buttons und Links auf dem Smartphone sehr fehleranfällig ist.
- **Konnektivität/Downloadgeschwindigkeiten:** Für die Usability ein großer negativer Faktor sind eine eingeschränkte Konnektivität und dadurch entstehende lange Downloadzeiten. Das Anzeigen des nächsten Bildschirms kann dann unter Umständen sehr lange dauern.

Zhang und Adipat [55] nennen in ihrem Artikel „Challenges, methodologies, and issues in the usability testing of mobile applications“, der schon aus dem Jahr 2005 stammt, ebenfalls die oben beschriebenen Aspekte und zusätzlich noch drei weitere, die aber auch heute für Smartphones gültig und aktuell sind:

- **mobiler Kontext:** Einfluss auf die Usability hat ebenfalls der mobile Kontext. Wenn die Userinnen und User ihr Smartphone benutzen, sind sie meist unterwegs. Im Großteil der Fälle sprechen sie gleichzeitig mit anderen Personen oder sind Umwelteinflüssen ausgesetzt, die die Userinnen und User ablenken und die Konzentrationsfähigkeit einschränken.
- **unterschiedliche Display-Auflösung:** Da der Bildschirm kleiner ist als auf dem PC, wirkt sich das natürlich auch auf die Bilder aus. Sie sind in der Qualität daher nicht so gut.
- **eingeschränkte Prozessorleistung:** Apps müssen sich an die durch das Smartphone gegebenen Einschränkungen in der Prozessorleistung anpassen.

5.2 User Experience

Laut Schneidermayer, Maier und Schricker [41] hat sich der Begriff der User Experience erst später entwickelt und ist eine Erweiterung des Usability-Begriffs. Die Usability hat ihren Hauptfokus auf den Zeitraum der Benutzung einer Applikation, die User Experience erweitert diesen Fokus auf die Zeit vor und nach der Benutzung. In der ISO 9241-210 Norm ist die User Experience:

„[...] a person's perceptions and responses that result from the use and/or anticipated use of a product, system or service.“ [22]

Nielsen und Wilson [31] geben an, dass die User Experience die oben genannten fünf Dimensionen noch um die nachfolgend genannten erweitert:

- **Ästhetik:** eine subjektiv optisch ansprechende App wird man eher verwenden.
- **Freude:** eine Applikation oder ein System, bei der/dem man bei der Benutzung Freude hat, wird man gerne verwenden.
- **Vereinbarkeit mit moralischen Werten:** eine Applikation oder ein System, das Inhalt verwendet, der nicht mit den eigenen moralischen Werten übereinstimmt, wird man wahrscheinlich nicht benutzen.

Die Unterscheidung zwischen Usability und User Experience wird in Abbildung 5.1 grafisch dargestellt.

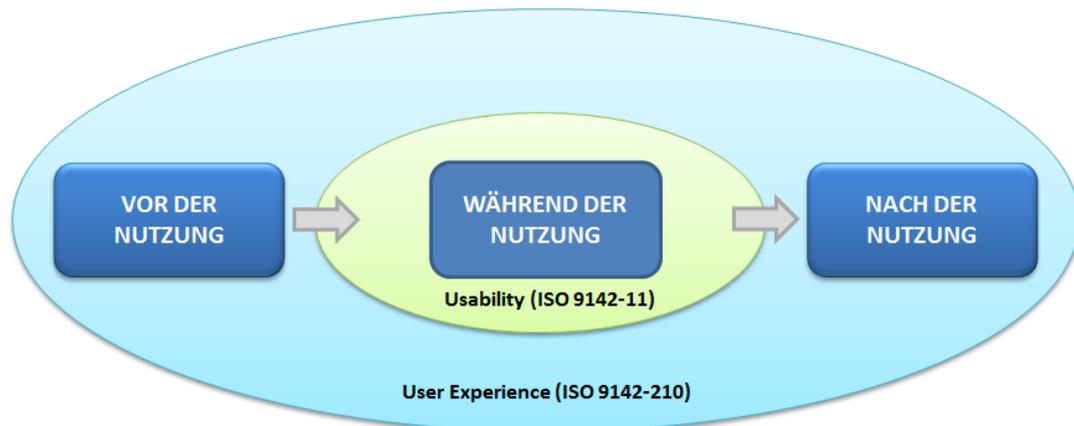


Abbildung 5.1: Unterschied zwischen Usability und User Experience nach Geis [16]

Nielsen und Wilson [31] betonen, dass diese zusätzlichen Dimensionen beim Gestalten sehr wichtig sind, eine erfolgreiche Applikation kann aber laut ihnen nur mit Hilfe einer fundierten Usability entstehen.

5.3 Mobile Learning

Für den Begriff Mobile Learning bestehen eine Unmenge an verschiedenen Definitionen. Einen kleinen Einblick in diese Diversität des Begriffes geben die verschiedenen Synonyme, die für den Begriff des M-Learnings verwendet werden. De Witt [53] listet folgende gleich verwendete Namen für Mobile Learning auf: M-Learning, Mobile Education, Persuasive oder Ubiquitous Learning. Viele der verschiedenen Definitionen konzentrieren sich nur auf einen Teilbereich, den Mobile Learning umfasst und keine ist umfassend und abgeschlossen. Schon die Bezeichnung Mobile Learning zeigt zwei unterschiedliche Aspekte des Begriffes:

- „**Mobile**“, das eindeutig auf die verwendeten Technologien hinweist und
- „**Learning**“, die eine Betonung auf die Didaktik legt.

Es gibt mehrere Standpunkte von denen aus Mobile Learning derzeit definiert wird, wie Traxler [48] in "Mobile Learning: Transforming the Delivery of Education and Training" argumentiert. Es gibt die Sicht der

- **Technologie** basierend auf mobilen Geräten
- **Mobilität** der Lernenden und des Lernens

- des **Lernerempfindens** beim Arbeiten mit mobilen Geräten

Im Zusammenhang mit einem Versuch einer Definition von Mobile Learning basierend auf mobilen Geräten spricht Traxler davon, dass diese neuen Technologien nicht nur ändern, wie man etwas lernt, sondern auch wie Lernen präsentiert werden kann. Genauer gesagt:

„Learning that used to be "just-in-case," can now be delivered "just-in-time, just enough, and just-for-me.“ [48]

Daher meint er, dass mobiles Lernen etwas ganz Neues, für sich selbst Stehendes ist und nicht als etwas aus dem E-Learning Entstandenes zu sehen ist.

Im Bezug auf die Mobilität geht Traxler davon aus, dass der spezielle Umgang der Userinnen und User mit Lerninhalten auf mobilen Geräten (Learning Objects) einen Einfluss auf die Art zu lernen haben wird. Weiters hält er eine Definition von mobilem Lernen für schwierig, weil es persönlich, auf den Kontext bezogen und situiert ist. Unter dem Begriff „personalized learning“ versteht Traxler Lernen, dass sich auf Unterschiede, Vielfalt und Individualität einstellt. Das Lernen wird sozusagen auf die Userinnen und User zugeschnitten. Das mobile Lernen gibt den Benutzerinnen und Benutzern durch seine besonderen Eigenschaften die Möglichkeit in einem bedeutungsvollen Zusammenhang zu lernen, was ein Zeichen für das situierte Lernen ist. Laut Traxler fördert es zusätzlich noch das Kontext spezifische und direktes Lernen. Authentisches Lernen, Lernen anhand von realen Problemen und Projekten kann ebenso realisiert werden. Traxler weist darauf hin, dass das Mobile Learning Reflektion und Selbstevaluation fördert.

Traxler ist auch überzeugt, dass die soziale und kulturelle Entwicklung der Gesellschaft eindeutig Auswirkung auf die Entwicklung von Mobilem Lernen zeigt. Er vergleicht dazu die Entwicklung dieser Art des Lernens in Entwicklungsländern mit unserer fortgeschrittenen Gesellschaft und hebt Unterschiede hervor.

Pachler, Bachmeier und Cook [34] gehen sogar soweit zu sagen, dass diese Art des Lernens nur dann richtig verstanden werden kann, wenn man die derzeitigen sozialen und kulturellen Veränderungen der Gesellschaft miteinbezieht. Nach Betrachtung all dieser möglichen Aspekte des Mobile Learning erscheint mir die nachfolgende Definition von Claudia DeWitt die aktuellste und treffendste Definition:

„[...] bezeichnet Mobile Learning das Lernen und Informieren unterwegs mit portablen, mobilen Endgeräten, die einen sofortigen und direkten Zugriff auf Informationen und Wissen ermöglichen und zumeist vernetzt sind. Vor allem ist Mobile Learning dabei zeit- und ortsunabhängig. Durch den Faktor der Mobilität, durch die Unabhängigkeit von Stromquellen, permanentem Netzzugang und damit durch den ubiquitären, allgegenwärtigen Zugang zu Wissen werden neue Situationen für Kontext bezogenes Lernen und Wissensmanagement geschaffen.“ [53]

Diese Definition berücksichtigt den vorher schon erwähnten Standpunkt, dass das Lernen sich durch das Mobile Learning vom formellen zum informellen Lernen wandelt. Die Betonung liegt vor allem auch auf dem Kontext bezogenen Lernen, auf das später noch eingegangen wird.

Auch die Technologien finden durch die Formulierung - unterwegs mit portablen, mobilen Endgeräten - in ihrer Definition einen Platz. Auch wird die Vernetzung in die Definition miteinbezogen, jedoch wird nicht davon ausgegangen, dass hauptsächlich online gearbeitet wird. Auch ein Offline-Arbeiten auf den portablen Geräten ist möglich. Unter solchen portablen Geräten versteht de Witt multimediafähige Smartphones, aber auch Netbooks und Tablets in DIN A4 Größe oder kleiner. Wie viele andere Autoren legt auch sie Wert auf einen der Hauptaspekte des Mobilen Lernens, die Zeit- und Ortsunabhängigkeit. So vereint diese Definition die schon vorher kurz angeschnittene Zweidimensionalität des Begriffs Mobile Learning und geht auf den technologischen und didaktischen Aspekt ein.

5.4 Entwicklung des Mobile Learnings

Laut Lam, Yau und Cheung [56] entwickelte sich das Mobile Learning (M-Learning) in den 1970er Jahren und hat sich von da an rapide verbreitet. Sie stützen sich auf eine Arbeit von Keegan, 2002 „From e-learning to m-learning“, die beschreibt, dass sich das Mobile Lernen aus dem Distant und dem E-Learning entwickelt hat.

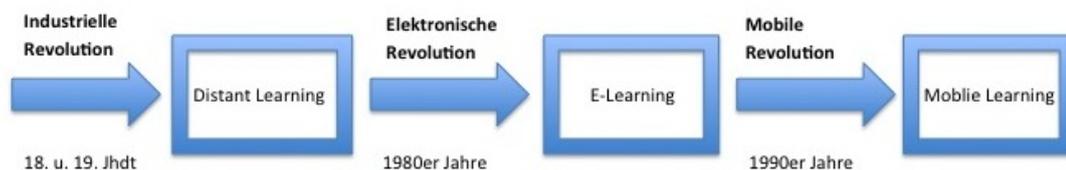


Abbildung 5.2: Entwicklung vom Distant zum Mobile Learning

Keegan [26] beschreibt, dass sich mit der industriellen Revolution und deren technologischem Fortschritt neue Möglichkeiten eröffnet haben. Die Kommunikation über die Post und die neuen Transportmittel machten somit ein Lernen über weite Strecken hinweg möglich (Distant Learning).

Der nächste revolutionäre Schritt in der Weiterentwicklung des Lernens wurde mit der sogenannten elektronischen Revolution gelegt. Eine entscheidende Rolle spielte dabei die Entwicklung der Telekommunikationsindustrie.

Keegan nennt drei Schlüsselfaktoren, die die elektronische Revolution eingeleitet haben:

- die Deregulierung der Telekommunikationsindustrie

- Chips wurden schneller und
- Breitband wurde entwickelt.

Daraus konnte sich das E-Learning entwickeln und die technologischen Errungenschaften führten später zum World Wide Web und dem Internet. Ab diesem Zeitpunkt war ein Lernen mit direktem Kontakt über Medien möglich. Die Entwicklung des Mobile Learnings hat seinen Anfang mit der Mobile und Wireless Revolution genommen, die durch die rapide Verbreitung von Handys angestoßen wurde. Ab diesem Zeitpunkt werden mobile Technologien zum Lernen eingesetzt.

Didaktik für Mobile Learning

Bachmair, Risch, Friedrich und Mayer [5] sind der Überzeugung, dass eine Didaktik für mobiles Lernen nicht nur aus dem intelligenten Einsatz von Smartphones im Unterricht bestehen kann. Sie zeigen auf, dass sich wie die Geschichte der Mediendidaktik zeigt, immer zuerst ein gesellschaftlich technologischer Komplex außerhalb der Schule entwickelt, der Basis für das Entstehen neuer Medien ist und ebenso bedingt, wie diese eingesetzt werden. Um ein detailliertes Bild von der derzeitigen Situation im Bezug auf den Handy-Einsatz im Unterricht zu geben, machen sie einen kleinen Rückblick in die Entwicklung der Didaktik basierend auf den entstandenen Medien.

So war 1920 das Einbringen von Filmmaterial in den Unterricht ebenso kontrovers, wie heute die Verwendung von Handys zum Lehren und Lernen. Andererseits hat sich der Einsatz von Computern und vom Internet sehr schnell in Schulen verbreitet. Bachmair, Risch, Friedrich und Mayer sehen den Grund dafür darin, dass Wirtschaft, Politik, Bildungsverwaltung und andere gesellschaftliche Gruppen darin eine wichtige Modernisierung und Kulturtechnik sahen. Die Schule sollte daher den Erwerb dieser Kulturtechnik unterstützen. Das damalige dazugehörige Leitmotiv war „Zukunft gewinnen - Bildung erneuern“. Im Fall von Film und Handy soll eine Ressource des Alltagslebens ebenso als didaktische Ressource dienen.

Für Pädagogik und Schule ist der Einsatz von Medieninnovationen der Gesellschaft als Kulturressource in der Schule gleichbedeutend mit einer Entfremdung des Lernens, der direkten Erfahrung und vom bisher eingesetzten Medium Buch. Darüber hinaus vermuten Bachmair, Risch, Friedrich und Mayer die Ablehnung vom Handy sowie 1920 vom Film darin, dass sie Unterhaltungsmedien und Unterhaltungsinhalte repräsentieren. Sie begründen ihre Idee Eckpunkten einer Didaktik für mobiles Lernen so:

„Das Handy als sichtbares Instrument eines globalen Mobilitätskomplex und dessen Innovationspotential lässt es deshalb geraten erscheinen, pädagogisch nachzudenken, wie sich denn das Handy als Kulturressource didaktisch nutzen lässt.“ [5]

Aus dem Internetkomplex hat sich durch die Entwicklung von mobilen Geräten und deren Möglichkeit das Internet, Medien- und Informationssysteme immer und allgegenwärtig zur Verfügung zu haben, der Mobilitätskomplex entwickelt. Der Mobilitätskomplex ist aber nicht nur ein rein technisches Thema. Die Technologie ist laut den Autoren in sozialkulturelle Veränderungen eingebunden. Es spielen also auch der Lebensstil und das sozialkulturelle Milieu der Personen eine Rolle. Einen Hauptaspekt sehen sie beim Mobilitätskomplex in der Medienkonvergenz - dem vernetzten und globalen Medienangebot, mit vernetztem Zugang zu den Medien und einer Mediennutzung in Netzen. Als ein Beispiel für Lernen basierend auf dieser oben genannten Medienkonvergenz werden Hausaufgabenforen auf YouTube genannt.

Die Schule steht dem Handy sehr skeptisch gegenüber und versucht es durch Verbote fernzuhalten und Schülerinnen und Schüler den kritischen, bewussten Umgang mit dem Handy zu lehren. Die Autorinnen und der Autor sehen in der Nutzung des Handys im Unterricht Potenzial für die Schule und haben sechs folgende Eckpunkte für eine Didaktik des mobilen Lernens definiert.

Mit dem Handy soll/sollen:

- **informelles Lernen** in der Schule integriert werden. Besonders durch das Handy, das als Schnittstelle zwischen der Kultur von Kindern und Jugendlichen verwendet werden kann, können das informelle Lernen und Wissen, welches man sich im Alltag aneignet in die schulische Umgebung integriert werden.
- **Episoden für situiertes Lernen** geschaffen werden. Durch den Einsatz des Handys ergeben sich neue Formen des situierten Lernens. Ein Vorschlag der Autorinnen und des Autors ist, den von Lehrerinnen und Lehrern geleiteten Unterricht mit Aktivitäten der Schülerinnen und Schüler, die sie am Handy ausführen, zu erweitern. Sie beschreiben die Vorgangsweise wie folgt: Lehrerinnen und Lehrer führen die Schülerinnen und Schüler entlang von Lernstraßen, die ein bestimmtes Lernziel verfolgen. Die Lernstraßen werden durch die Schüleraktivitäten, die von Lehrerinnen und Lehrern angeboten werden, erweitert. Diese Aktivitäten, Episoden situierten Lernens, bezeichnen sie als Lernplätze.
- **Lern- und Medienkontexte** generiert werden. Das Handy gibt Schülerinnen und Schülern die Möglichkeit, eigene Lernkontexte zu erstellen. Die Basis für diese bilden die Medienkonvergenz des Internets, die Unterhaltungsmedien und die Schule. Diese entstandenen Lernkontexte gehen über die Schule hinaus und Schülerinnen und Schüler schaffen sich ihr Wissen selbst, statt es zu übernehmen. Das Handy ist das Bindeglied zwischen Lernplätzen der Schülerinnen und Schüler und der vernetzten Medienwelt, dem Internet.
- **Kommunikationsbrücken und Kommunikationsketten** geschaffen werden Bachmair, Risch, Friedrich und Mayer definieren Kommunikationsbrücken und Kom-

munikationsketten als Verbindungslinien zwischen dem Alltag der Schülerinnen und Schüler und dem Lernen in der Schule.

- die **Schülerinnen und Schüler als Experten ihres Alltagslebens** in der Schule individuell aktiv gemacht werden. Durch die Verwendung des Handys auf den Lernstraßen und den Lernplätzen können die unterschiedlichen Handlungs- und Lernmuster der Schüler auch für den Lernerfolg in der Schule genützt werden. Es können auf diesem Weg Kompetenzen der Schülerinnen und Schüler sichtbar gemacht werden.
- **sensible Entwicklungs- und Lernkontexte** geschaffen werden. Die aktuelle Entwicklung der Kinder spiegelt sich in den von ihnen gestalteten Lernkontexten wieder. Diese kann man auch wieder in die Lernsituation einfließen lassen. Das Aufgreifen dieser entstandenen Themen kann für die Kinder wieder nutzbringend sein.

5.5 Microlearning

Durch die Nutzung von Smartphones zum Wissenserwerb durch Lernapplikationen und -spiele hat sich auch der Begriff des Microlearnings entwickelt. Peter Baumgartner beschreibt diese neue Art des Lernens wie folgt:

„Unter Microlearning verstehe ich sehr kleine und damit kurze Lerneinheiten, die mit einem raschen, d.h. unmittelbaren Feedback für die Lernenden versehen sind.“ [6]

Durch die besonderen Eigenschaften und aber auch Einschränkungen von mobilen Endgeräten sieht er für eine Didaktik des Microlearnings die unten angeführten Herausforderungen:

- **Robuste Lernumgebung:** Bedingt durch unsere schnelllebige Gesellschaft sollen Leerlaufzeiten, wie Reise- und Wartezeiten effizient genutzt werden. Hier kann speziell das Microlearning eingesetzt werden, das auch mit schnell wechselnden und herausfordernden Umgebungen zurechtkommt. Baumgartner sieht die Herausforderung für die Didaktik hier darin, eine Lernumgebung zu schaffen, die sich gegen Ablenkungen und Störungen durchsetzen kann und für den Lernenden auch von Nutzen ist.
- **Motivierende und interessante Lernherausforderung mit Feedback:** Baumgartner beschreibt die zweite Herausforderung als die Aufgabe für den Lernenden motivierende und interessante Aufgaben mit Feedback zu gestalten. Dieser Aspekt kann laut dem oben genannten Autor nur mit Hilfe von Abwechslung sichergestellt werden. Dazu müssen spezielle Aufgaben und das dazugehörige Feedback

erstellt werden. Außerdem ist er der Meinung, dass eine Interaktion das Lernen betreffend nicht länger als ein paar Minuten dauern darf.

- Ausarbeitung einer **Kommunikationstheorie für das Microlearning**: Baumgartner sieht die dritte Herausforderung der Didaktik für das Microlearning in der Entwicklung einer Kommunikationstheorie für Interaktionen.
- **Microcontent statt Memorieren**: Bisherige Microlearningansätze beschränkten sich häufig auf das individuelle Erlernen von Inhalten durch Memorieren. Die vierte Herausforderung der Didaktik für das Microlearning ist Microcontent einzusetzen und Aspekte der Gamification in das Microlearning mit einzubringen.

Kapitel 6

Gestaltungsrichtlinien

In den folgenden Abschnitten wurden Kriterien zusammengefasst, die im Zuge der Literaturrecherche zusammengestellt worden sind. Einerseits werden die neu gewonnene Information im Bezug auf Apps, Mobile Learning und Microlearning und allgemeine wissenschaftliche belegte und untersuchte Fragen und Richtlinien, die beim Erstellen von Apps in Bezug auf Usability und User Experience wichtig sind behandelt, aber auch der Bereich der Darstellung von Text und Bild werden diskutiert. Dieser hier zusammengefasste Kriterienkatalog ist die Basis für die spätere Evaluation des Prototypen in Bezug auf die Umsetzung der Fragen des Biber der Informatik in einer Smartphone-App.

Zu Beginn werden Richtlinien zum allgemeinen Konzept einer Mobile Learning Applikation angeführt, die aus den Erkenntnissen vorheriger Kapitel entstanden sind.

Shitkova, Holler, Heide, Clever und Becker [45] unterteilen in ihrem Artikel „Towards Usability Guidelines for Mobile Websites and Applications“, die von ihnen identifizierten Richtlinien in die Bereiche Layout, Navigation, Design, Content und Performance auf. Da diese Aufteilung auch für die Einteilung der während der Literaturrecherche gefundenen Richtlinien für das Erstellen von Touchphone-Apps im Hinblick auf die Übersichtlichkeit und Aufteilung sinnvoll ist, wird diese grundlegende Unterteilung hier übernommen.

6.1 Allgemeines Konzept

In diesem Abschnitt werden allgemeine Richtlinien zusammengefasst, die aus den vorherigen Kapiteln Apps und Usability, User Experience und Mobile Learning hervorgegangen sind.

Laut Claudia De Witts [53] Definition vom mobilen Lernen, die sowohl auf die Zweidimensionalität des Begriffs (Lernen, mobil) an sich, aber auch die technologische Dimension (portable Geräte, Vernetzung) beinhaltet, soll man immer und überall ler-

nen können, dabei soll man auch nicht durch die Notwendigkeit einer Verbindung zum Stromnetz oder zum Internet eingeschränkt werden. Daher hat sich folgende Richtlinie RI ergeben:

RI: Eine mobile Lernapplikation sollte orts- und zeitunabhängig und auch unabhängig von Stromquelle und Netzzugang nutzbar sein.

Baumgartner [6] beschreibt im Zusammenhang mit dem Mobilien Lernen eine Weiterentwicklung, das Microlearning. Basierend auf seiner Definition ist das Microlearning geprägt durch kleine, kurze Lerneinheiten mit direktem Feedback. Bei dieser Art des Lernens hebt er einige Kriterien hervor, die im Vorhinein bedacht werden sollten (siehe Kapitel 5 Abschnitt Microlearning). Aus dieser Definition und den genannten Kriterien ist folgende Richtlinie entstanden.

RII: Lerneinheiten sollten nicht länger als ein paar Minuten dauern, mit direktem Feedback versehen und robust gegen Ablenkungen aus der Umgebung sein.

Büttner und Simmons [8] sowie Bergmann [7] und Budiu und Nielsen [32] geben die Empfehlung die Entscheidung welche Art der App man umsetzen sollte nach den Kriterien zu treffen, die für den späteren Einsatz der App wichtig sind. Laut ihnen und Bergmann sollte man:

- eine **WebApp** umsetzen, wenn viele User auf einmal und möglichst plattformunabhängig erreicht werden sollen, möglichst geringe Kosten wichtig sind und der Aspekt, dass sie meist nur online verwendbar sind, sowie die Funktionen des Smartphones nur begrenzt ausnützen, nicht wichtig sind.
- eine **native App** umsetzen, wenn man die Hardware des Gerätes nutzen, komplizierte Interaktionsvorgänge umsetzen will und einem sehr großen Arbeitsspeicherverbrauch hat.
- eine **hybride App** umsetzen, wenn viele Userinnen und User auf einmal erreicht werden sollen und ständige Aktualität bei Individualisierung der App wichtig ist. Ein weiterer Faktor für hybride Apps ist eine Zeit- und Kostenersparnis bei der Entwicklung, weil die App nur einmal entwickelt werden muss.

Basierend auf diesen Überlegungen ist die unten angeführte Richtlinie hervorgegangen:

RIII: In welcher Form - WebApp, native oder hybride App - eine App umgesetzt werden soll, sollte entschieden werden durch die Kriterien der späteren Nutzung.

6.2 Layout

Die folgenden Richtlinien beziehen sich auf den allgemeinen Inhalt einer App zum Beispiel Menüs, Buttons, Login und alles was das Layout einer App betrifft.

Fuglerud und Røssvoll [14] fanden in einer ihrer Studien heraus, dass das Anzeigen von zu vielen Optionen sehr schnell zu zu breiten Seiten führen kann (z.B.: Navigationsmenü auf der linken Seite). Bei diesen zu breiten Seiten kann es passieren, dass Userinnen und User durch die Verwendung der Vergrößerungsfunktion Inhalte und Buttons übersehen. Daher sollte die unten angeführte Richtlinie beim Design einer App berücksichtigt werden.

R1: Zeige den gesamten Inhalt, wenn möglich auf einer Seite an

Wenn der Inhalt nicht vollständig auf eine Seite passt, muss wohl oder übel gescrollt werden. Alonso-Ríos, Raneburger, Popp, Kaindl und Falb [4] fanden in ihrer Studie heraus, dass Benutzerinnen und Benutzer durch den schon bekannten Umgang mit Desktop-Computern vertikales Scrollen als intuitiver empfinden. Einigen Userinnen und Usern war nicht klar, dass auch horizontal gescrollt werden könnte. Auch Budiu und Nielson [32] und Holzinger und Errath [20] sprechen sich gegen ein horizontales Scrollen aus.

R2: Vermeide horizontales Scrollen.

Weiters fanden die Autoren [4] in ihrer Studie, dass das vertikale Scrollen bereits so im Alltag verankert ist, dass die Userinnen und User es bei der Benutzung von mobilen Geräten voraussetzen. Daher empfehlen sie Inhalte prinzipiell untereinander anzuordnen. Fuglerud und Røssvoll [14] weisen darauf hin, nicht zu viele Optionen und Ebenen zu verwenden, da dies die Benutzerinnen und Benutzer verwirren kann. Laut Budiu und Nielsen [32] sollte man es vermeiden einen Text auf mehrere Seiten aufzuteilen (lineare Paginierung), die man dann nur mit einem „Next“-Button durchblättern kann. Für diesen Fall soll der Inhalt einfach scrollbar untereinander angezeigt werden.

R3: Zeige Inhalt vertikal untereinander an und vermeide zu viele Optionen und Ebenen.

Budiu und Nielsen [32] appellieren in ihrem Buch Logins nicht direkt auf die Startseite einer Applikation zu platzieren. Sie kritisieren daran, dass die Userinnen und User in den meisten Fällen noch nicht einmal wissen, welcher Inhalt auf der Webseite/in der App präsentiert werden soll, aber sie werden schon um einen Login gebeten. Um herauszufinden worum es geht, müssen die Userinnen und User den Registrationsprozess durchlaufen und kommen danach unter Umständen zum Schluss, dass sie der Inhalt

nicht interessiert und die Zeit und Mühe für die Registrierung vergebens waren.

R4: Logins sollten nicht direkt zu Beginn der Webseite/der App platziert sein.

Benutzerinnen und Benutzer verwenden sehr schnell die Suchfunktion, wenn sie zum Beispiel nicht wissen, wo sie etwas auf einer Seite finden können. Budiu und Nielsen [32] empfehlen die Suchfunktion immer gut sichtbar zu platzieren. Mehrere Suchfelder, die sich auf verschiedene Inhaltsgruppen beziehen, sollten vermieden werden, weil die Benutzerinnen und Benutzer in den meisten Fällen diesen Unterschied nicht wahrnehmen.

R5: Zeige das Suchfeld so an, dass es sichtbar und benutzbar ist.

Lobo, Kaskaloglu, Kim und Herbert [28] sprechen sich dafür aus, das Layout einer App sehr klar und einfach zu halten. Auch Budiu und Nielsen [32] empfehlen basierend auf den Ergebnissen ihrer Usertestings verschiedenster Apps ein übersichtliches, klares und einfaches Layout. Auch Fuglerud und Røssvoll [14] haben Ähnliches in ihrer Studie beobachtet. Die Userinnen und User hatten die App am besten beurteilt, die den einfachsten und klarsten Aufbau hatte.

R6: Verwende eine klares, einfaches Layout für die App.

Laut den Designrichtlinien von Google für Android sollen Dialoge nicht mehr als zwei Buttons enthalten. Ein dritter Button zum Beispiel „learn more“ würde zu einem neuen Inhalt verlinken und die vorhergehende Aktion würde nicht abgeschlossen werden. Daher werden zwei Buttons für Dialoge empfohlen - einer zum Abbrechen der Aktion und einer um die Aktion abzuschließen. Auch Shitkova, Holler, Heide, Clever und Becker [45] haben in ihren gefundenen Guidelines eine Beschränkung der Buttons angegeben, die die Zahl drei nicht überschreiten sollte.

R7: In einem Dialog sollten nicht mehr als zwei Buttons verwendet werden.

Seipp und Devlin [43] schlagen basierend auf ihren Untersuchungen vor, die Buttons in Dialogen immer in der Nähe des Textes zu platzieren. Dadurch erkennen die Benutzerinnen und Benutzer sie als zum Text gehörig. Außerdem wird die Interaktion erleichtert, weil die Userinnen und User schon erwarten, den Button dort zu finden.

R8: Buttons in einem Dialog findet man immer nach dem Dialogtext, aber in unmittelbarer Nähe dazu.

Seipp und Devlin [43] haben auch untersucht, welche Anordnung der Buttons auf einem Touchphone-Bildschirm sinnvoll und für die Usability fördernd wirken. Sie ha-

ben herausgefunden, dass der Button, mit der höchsten Auswahlwahrscheinlichkeit in der Mitte platziert werden sollte, weil dort die Aufmerksamkeit der Benutzerinnen und Benutzer liegt; den Button mit der zweithöchsten Wahrscheinlichkeit entsprechend dem Design darunter oder rechts daneben; den möglichen, wie schon oben erwähnten, dritten Button darüber oder links davon. Diese Strukturierung ergibt sich aus der Erreichbarkeit der Buttons.

R9: Für Buttons in Dialogen gilt: Der Button, der mit höchster Wahrscheinlichkeit gewählt werden soll, steht dabei in der Mitte; je nach Design den zweitwichtigsten Button rechts daneben oder darunter; ein weiterer unwichtigerer Button links oder darüber.

6.3 Navigation

Die Richtlinie „Keep it simple“ von Lobo, Kaskaloglu, Kim und Herbert [28] bezieht sich nicht nur auf das Layout einer App, sondern natürlich auch auf die Navigation. Eine einfach nachzuvollziehende Navigation fördert die Usability. Auch Han und Park [19] sprechen sich in ihrer Studie für ein einfaches Menü aus.

R10: Die Navigationsmenüs sollten so einfach wie möglich gestaltet sein.

Han und Park [19] fanden in ihrer Studie heraus, dass wichtige Seiten direkt von der Hauptseite zu erreichen sein sollten. In ihrer Arbeit waren wichtige Einstellungen für die gesamte Applikation in einem Untermenü verborgen, was dazu führte, dass die Userinnen und User lange suchen mussten. Auch Neto und Pimentel [29] sprechen sich dafür aus. Laut diesen Autoren, darf kein Feature der App schwer zu finden sein.

Für Holzinger und Errath [20] ist es ein Muss, dass alle Hauptfeatures von der Startseite aus erreichbar sind. Sie sind der Meinung, dass die Navigation der wichtigste Aspekt für die Usability darstellt. Auch Inostroza und Rusu [24] bieten in ihrer Heuristik - Physische Interaktion und Ergonomie - eine Richtlinie für Android-Systeme an, die besagt, dass wichtige Funktionen schnell und leicht zu finden sein sollten.

R11: Wichtige Seiten sollten direkt von der Startseite erreichbar sein.

Befolgt man oben angegebene Richtlinie basierend auf den Untersuchungsergebnissen von Han und Park [19] sowie Holzinger und Errath [20] folgt daraus, dass Navigationsmenüs nur wenige Ebenen haben sollten.

Ein auf den ersten Blick kontroverses Ergebnis haben Geven, Sefelin und Tscheligi [17] in ihrer Studie gefunden. Sie empfehlen eine tiefe Hierarchie einer breiten vorzuziehen. Hier muss aber mit in Betracht gezogen werden, dass die breite Hierarchie unter Umständen viele Optionen beinhaltet, was sich negativ auf die Navigation auswirken

würde. Daher kann man schließen, dass eine Hierarchie mit nur wenigen Ebenen sinnvoll ist, wenn in jeder Ebene nur wenige Optionen angezeigt werden.

R12: Navigationsmenüs sollten nur wenige Ebenen haben.

In der Studie von Geven, Sefelin und Tscheligi [17] hat sich aufgrund der kleinen zur Verfügung stehenden Anzeigefläche ergeben, dass die Userinnen und User eine schmale Navigationshierarchie mit nur wenigen Optionen bevorzugen. Manche Benutzerinnen und Benutzer haben angegeben, keine Motivation zu haben, die lange Liste zu lesen und lieber mehrere Ebenen mit kurzen Listen zur Auswahl zu durchlaufen.

Zusätzlich kommt der Aspekt des Scrollens bei langen Listen hinzu. Die Autoren sprechen sich hier klar gegen ein Scrollen aus, weil es auf mobilen Geräten nicht einfach auszuführen ist. Noch dazu kommt laut Budiu und Nielsen [32] eine zusätzliche Belastung für das Gedächtnis, weil man sich die schon gesehenen Optionen beim Scrollen merken muss. Geven, Sefelin und Tscheligi [17] kommen zu dem Schluss, dass bei einer Auswahlliste 4-8 Einträge optimal sind.

R13: Vermeide zu viele Optionen und Scrollen in einem Navigationsmenü (nur 4-8 Einträge)

Holzinger und Errath [20] sind der Meinung, dass man den Userinnen und Usern die Möglichkeit geben sollte, die App schnell zu verlassen. Dabei sollten keine Informationen verloren gehen. Ein weiteres Feature sollte sein, wenn es sich technisch umsetzen lässt, dass die App beim Beenden und einem nachfolgenden Neustart wieder an der Abbruchstelle beginnt.

R14: Gib den Userinnen und Usern die Möglichkeit die App schnell zu verlassen. Bei einem Restart beginne wieder an der Abbruchstelle.

Lobo, Kaskaloglu, Kim und Herbert [28] geben an, dass es für die Userinnen und User eine Erleichterung bringt, wenn man ihnen die Möglichkeit gibt schnell wieder auf die Startseite zurückzuwechseln. Die Startseite ist jene, von welcher man alle wichtigen Features erreichen kann. Außerdem ist es laut den Autoren, laut Budiu und Nielsen [32] und auch Dias, De Pontin, Fortes, Masiero, Watanabe und Ramos [11] empfehlenswert, die aktuelle Position via Breadcrumbs zu kennzeichnen.

R15: Gib den Benutzerinnen und Benutzern die Möglichkeit zum schnellen Erreichen der Startseite und verwende zur Darstellung der aktuellen Position Breadcrumbs.

Ein wichtiger Teil der Navigation ist die Strukturierung des Menüs und die Benennung der einzelnen Abschnitte. Fang und Holsapple [12] definieren zwei Arten von

Navigation: eine themenorientierte und eine nutzungsorientierte Navigation. Auch Budiu und Nielsen [32] empfehlen eine Strukturierung des Inhalts auf diese Weise, wobei sie aber von den besten Ergebnissen berichten, wenn eine nutzungsorientierte Struktur verwendet wird.

Dabei wird darauf eingegangen, wie Benutzerinnen und Benutzer bei der Informationsbeschaffung vorgehen. Laut den Autoren ist das Erlangen von Information in einer basierend auf einem Organigramm der Organisation strukturierten Seite schwerer als nach einer basierend auf den Aufgaben, die von den Userinnen und Usern erledigt werden müssen.

R16: Um vieles Herumsuchen zu vermeiden, strukturiere Menüs nach Themen und Art der Verwendung. [32]

Tirapat und Achalakul [47] haben verschiedene Arten von Links untersucht. Sie nennen zwei Arten von Links. Externe Links, die auf andere Webseiten führen und interne Links, mit denen man an einen anderen Punkt der Webseite gelangen kann.

Eine weitere Einteilung über die Art, wie Links angezeigt werden, ist laut Tirapat und Achalakul ebenfalls möglich. Ein Link kann entweder im gleichen Fenster angezeigt werden und ersetzt den vorhandenen Inhalt oder es wird ein neues geöffnet.

Laut Leavitt und Shneiderman [27] sollen Links in einem neuen Fenster geöffnet werden und dieses neue Fenster sollte Bedienelemente für das Zurückkehren zum Ursprungsfenster haben. Sie beschreiben ein Szenario, das die Userinnen und User unter Umständen sehr verwirren kann. Bei diesem Beispiel wird der Link als Vollbild geöffnet. Die Benutzerinnen und Benutzer wissen daher nicht, dass sie sich auf einer anderen Webseite befinden. Wenn dann noch keine Steuerungselemente vorhanden sind um zum Ausgangspunkt zurückzukehren, sind die Userinnen und User schnell frustriert.

R17: Öffne Links in einem neuen Fenster, das aktuelle Fenster bleibt unverändert und das neue Fenster hat Bedienelemente das Fenster zu schließen oder zur Seite zurückzukehren.

Han und Park [19] sowie Fuglerud und Røssvoll [14] haben in ihren Untersuchungen herausgefunden, dass Userinnen und User bei der Navigation speziell dann Fehler machen, wenn die Beschriftung der Links nicht eindeutig sind. Laut ihnen verwirren unklare, den Inhalt nicht beschreibende Labels die Benutzerinnen und Benutzer.

Auch Budiu und Nielsen [32] schließen sich dieser Meinung an. Sie präsentieren die Theorie des „Information Foraging“ als eine der wichtigsten Richtlinien der Mensch-Computer-Interaktion. Die Theorie beschäftigt sich damit, wie der Mensch sich bei der Informationssuche im Internet verhält. Er geht, wenn es möglich ist den Weg des geringsten Widerstandes - den bequemsten Weg. Mit minimalem Aufwand soll das beste Ergebnis erzielt werden.

Budiu und Nielsen [32] geben die „Informationsfährte“ als das wichtigste Prinzip der Theorie des Information Foragings an. Eine gute Informationsfährte legen bedeutet, dass die Benennung von Links, Elementen, Schaltflächen usw. so gestaltet ist, dass die Userinnen und User genau wissen, was damit gemeint ist und was sich dahinter verbirgt.

Auch Shneiderman, Plaisant, Cohen und Jakobs [46] sind der Meinung, dass vor allem Links, die auf anderen Inhalt auf der selben Seite verweisen, beschreibend sein müssen. Überschriften sollen laut ihnen darüber hinaus auch noch einzigartig und eindeutig sein. Für Neto und Pimentel [29] müssen alle Metaphern für Elemente unmissverständlich, leicht erkennbar sein.

R18: Titel, Links und Navigationselemente sollten unmissverständlich, selbst-erklärend und beschreibend sein.

6.4 Design

Das schon oben erwähnte Prinzip „Keep it simple“ aus dem Artikel von Lobo, Kaskaloglu, Kim und Herbert [28] bezieht sich natürlich auch auf das Design. Interaktion über die Touchfunktion ist für die Userinnen und User schwierig, daher sollten komplizierte Formulare als auch unübersichtliche, lange Prozesse (z.B.: Bestellungsprozess) vermieden werden. Sie empfehlen auch noch aufgrund der sehr geringen Anzeigemöglichkeiten auf große Bilder und Flashinhalt zu verzichten.

Neto und Pimentel [29] warnen davor, dass zu viele Elemente auf einem Bildschirm die Userinnen und User überfordern können. Inostroza und Rusu [24] und auch Neto und Pimentel [29] plädieren für die Einhaltung von Konsistenz um Nutzerinnen und Nutzern die Möglichkeit zu geben, schon von der Verwendung einer anderen App Erlerntes in der neuen App einzusetzen. Sie raten davon ab, Inhalte so zu designen, dass bekannte Designelemente ein anderes Verhalten aufweisen. Eine Design-Richtlinie für IOS-Betriebssysteme greift genau diese Form der Konsistenz auf, wie Inostroza und Rusu [24] betonen.

Inostroza und Rusu [24] haben in ihrer Studie ebenfalls eine Heuristik „Consistency and Standards“ denen sie Guidelines für die bekanntesten Betriebssysteme für mobile Geräte zuordnen. Sie geben eine Designrichtlinie für Android an, die besagt, dass ähnlich aussehende Elemente sich auch ähnlich verhalten sollten.

Budiu und Nielsen [32] sprechen sich eindeutig gegen die Verwendung von mehrdeutigen Befehlen aus. In diesem Fall wird der selbe Befehl verwendet, um unterschiedliche Ergebnisse zu erzielen. Der Unterschied ist oft nur durch den Kontext gegeben, was die Nutzerinnen und Nutzer meist übersehen (zwei Suchfelder auf einem Anzeigefeld, wobei jedes zu einem bestimmten Kontext gehört).

Budiu und Nielsen empfehlen die Verwendung von generischen Befehlen. Ein Befehl wird genutzt, um vom Konzept her das selbe Ergebnis zu erzielen (z.B.: Geste für das Aufziehen/Vergrößern vom Inhalt).

Neto und Pimentel [29] geben noch darüber hinaus an, dass eine Metapher für ein Element einzigartig sein muss. Shneiderman, Plaisant, Cohen und Jakobs [46] geben als ein Mittel zur Steigerung der Konsistenz das Erstellen eines Richtlinien-Dokuments für die Anzeige an. In diesem sollten die Farben, Schriften, Formate usw. festgehalten werden. Inostroza und Rusu geben zwei Richtlinien für IOS- und Android-Systeme an, in denen empfohlen wird, dass ein durchdachtes Design den Nutzerinnen und Nutzern ein Gefühl der Leichtigkeit in der Benutzung vermittelt.

R19: Das Design sollte einfach, konsistent, gleichbleibend und klar sein.

Bei der Entwicklung einer mobilen Applikation, die als Basis eine schon existierende Desktopversion hat, empfehlen Nielsen und Budiu [32] in ihrem Buch das selbe Branding zu verwenden. Anders gesagt bedeutet das, die App so zu gestalten, dass sich ein Wiedererkennungswert für die Benutzerinnen und Benutzer ergibt. Dabei helfen die Verwendung von den selben Logos und ähnlichen Farben usw..

R20: Verwende ähnliches, wiedererkennbares Design für verschiedene Versionen eines Produktes.

Laut Holzinger und Errath [20] sind Icons Repräsentation von Text in Form eines Bildes. Sie betonen darauf zu achten, dass jedes Bild eine bestimmte Botschaft vermittelt. Wesentlicher Aspekt für das Umsetzen der Bedeutung ist, dass das Icon gut erkennbar ist. Das setzt voraus, dass sich das Icon sehr gut vom Hintergrund abhebt.

Bei der Interpretation von Icons spielen auch kulturelle Begebenheiten eine Rolle, denn je nach gesellschaftlichem Hintergrund können Bilder anders interpretiert werden.

Die Aussagekraft eines Icons alleine ist sehr gering. Laut Holzinger und Errath sind sie immer mit einem bestimmten Kontext verbunden und bedürfen der Interpretation der Userinnen und User. Daher sollte man laut den Autoren darauf achten, dass bekannte Icons mit einem dazugehörigen Kontext auch wieder so eingesetzt werden, damit keine Verwirrung entsteht.

R21: Verwende Icons, die einfach zu erkennen und zu sehen sind

Park, Han, Park und Cho [35] haben in ihrer Studie einen Zusammenhang von der Größe und dem Ort, wo sich eine Schaltfläche befindet und der Fehlerhäufigkeit gefunden. Es hat sich herausgestellt, dass, wenn ein Button in einer für Userinnen und User leicht erreichbaren Region liegt und er groß genug ist, die Fehlerquote beim Klicken sinkt. Budiu und Nielsen [32] sowie Holzinger und Errath [20] geben als optima-

le Größe für klickbare Elemente 7-10mm an. Sie geben auch folgende User-Interface-Komponenten als fingerfreundlich an:

Tabelle 6.1: **fingerfreundliche UI-Komponenten**

UI-Komponente	Fingerfreundlichkeit
Button	Wenn die Buttons der schon vorher angegebenen Größe entsprechen, sind sie für die Touchsteuerung sehr gut einsetzbar.
TextBox	Wie schon bei den Buttons sind größere Textfelder leichter anzuklicken als kleine. Für das leichtere Editieren von in einer Textbox eingegebenem Text muss die Schriftgröße erhöht werden.
Checkbox	Checkboxes sind sehr fingerfreundlich, wenn nicht nur die Box geklickt werden muss, sondern die ganze Komponente und wenn genügend Abstand zu anderen Elementen vorhanden ist.
Radiobutton	Für Radiobuttons gelten die selben Kriterien, wie für die Checkboxes. Speziell bei der Gruppierung der Radiobuttons ist der Abstand zueinander wichtig, um komfortables Klicken zu ermöglichen.
Listbox/Combobox	Listboxen/Comboboxen sind nicht besonders fingerfreundlich, weil meist gescrollt werden muss und einzelnen Elemente schwer klickbar sind. Eine Vergrößerung der Schriftart führt zu mehr Scrollarbeit.
Listview	Listviews mit Icons oder großen Icons sind sehr gut für die Finger-Bedienung geeignet.
Treeview	Eine Treeview ist mit Fingern sehr schwer zu bedienen und es gibt kaum Anpassungsmöglichkeiten.
Tabs	Alle Tabs sollten immer auf dem Bildschirm sichtbar sein um ein Scrollen zu verhindern. Größere Tabs sind leichter klickbar. Die Größe jedoch hängt von der Länge des Wortes und der Schriftart ab.
Scrollbar	Scrollen mit einem Scrollbalken ist sehr fingerunfreundlich. Die Balken sind meist zu schmal. Es ist daher vorteilhaft andere Arten des Scrollens zu verwenden.
UpDown	Updowns sind sehr fingerunfreundlich zu bedienen, weil die Buttons für „up“ und „down“ viel zu klein sind.
TrackBar/Slider	TrackBars/Sliders sind mit Fingern nur sehr ungenau zu bedienen.
<i>Fortsetzung auf der nächsten Seite</i>	

UI-Komponenten – Fortsetzung

UI-Komponente	Fingerfreundlichkeit
Menüelemente	Menüelemente eignen sich sehr gut bei der Verwendung von Touchsteuerung
ContextMenus	Kontextmenüs sind ähnlich zu bedienen wie Menüelemente. Die Schwierigkeit bei ihnen liegt darin das Element zu klicken, zu dem es gehört.

Quelle: Nilsson [33]

R22: Gestalte das Interface so, dass die ungenaue Touchfunktion nicht ins Gewicht fällt. Verwende eine Schaltflächen-Größe von einer Größe zwischen 7-10 mm.

Für Neto und Pimentel [29] müssen Eingabeelemente immer so gestaltet sein, dass Benutzerinnen und Benutzer unmissverständlich wissen, was eingetragen werden soll. Die eingegebenen Daten sollten immer so dargestellt werden, dass sie für Userinnen und User gut lesbar sind und die Benutzerinnen und Benutzer immer Herr über die Situation bleiben.

Sowie Neto und Pimentel sprechen sich Budiu und Nielsen [32] auch für eine einfache Eingabe von Daten aus. Sie geben alternative Möglichkeiten an, um die Dateneingabe für Nutzerinnen und Nutzer zu vereinfachen. Sie nennen daüfür:

- Radiobuttons
- Comboboxen
- Checkboxes
- Spinners
- Sliders und
- Menüs

Holzinger und Errath [20] geben für die Verbesserung der Dateneingabe noch die Möglichkeit von Default-Werten an. Sie empfehlen immer wenn möglich die Eingabeoptionen schon mit Default-Werten vorzubelegen. Nilsson [33] gibt die Möglichkeit von vordefinierten Werten an, aus denen die Benutzerinnen und Benutzer wählen können. Insgesamt vermindern all die angegebenen alternativen Dateneingabeoptionen den Tippaufwand für die Nutzerinnen und Nutzer.

R23: Optimierte die Eingabe von Daten für mobile Geräte.

Laut Nilsson [33] gibt es mehrere Möglichkeiten mit dem Einblenden der Tastatur auf Smartphones umzugehen. Eine davon ist es das Design so anzupassen, dass wichtiger Inhalt nur in dem Teil der Anzeige verwendet wird, die nicht von der Tastatur verdeckt werden kann.

Eine zweite Möglichkeit ist es Scrollbars einzufügen; eine weitere den Inhalt einfach von der Tastatur überdecken zu lassen oder die Tastatur als Bestandteil des Screens von vornherein zu integrieren. Eine letzte Möglichkeit ist ein Element des Screen zu verkleinern, um Platz für die Tastatur zu schaffen.

R24: Passe den Inhalt für das mögliche Einblenden der Tastatur an.

Budiu und Nielsen [32] empfehlen bekannte Gesten, wie das Aufziehen von Inhalten zur Vergrößerung in der mobilen App zu ermöglichen. Speziell diese Geste ist so bekannt, dass Userinnen und User erwarten, dass sie immer und überall funktioniert. Inostroza und Rusu [23] haben zu ihrer Heuristik eine Design-Guideline für Android zugeordnet, die besagt, dass bekannte Gesten/Tricks überall funktionieren sollten.

R25: Unterstütze bekannte Gesten.

Holzinger und Errath [20], Leavitt und Shneiderman [27], Budiu und Nielsen [32] und Nilsson [33] sprechen sich für das zur Verfügung stellen einer automatische Vervollständigung von Text aus. Diese Funktion erleichtert es den Benutzerinnen und Benutzern Text einzugeben. Statt das ganze Wort einzutippen werden Vorschläge zur Auswahl vorgegeben.

R26: Stelle automatische Vervollständigung zur Verfügung.

Holzinger und Errath [20] haben in ihrer Studie herausgefunden, dass die Benutzbarkeit einer App steigt, wenn Userinnen und Usern eine automatische Tippfehlerkorrektur zur Verfügung steht, da jegliches Nachbearbeiten von eingegebenem Text sehr mühsam ist.

R27: Stelle automatische Tippfehler Korrektur zur Verfügung.

Laut Holzinger und Errath [20] muss es für die Benutzerinnen und Benutzer möglich sein ihre Aktionen rückgängig zu machen, weil die Interaktion über die Touchsteuerung wesentlich fehleranfälliger und ungenauer ist. Ein Finger kann keine so pixelgenaue Interaktion ausführen, wie eine Maus. Eben solche Guidelines geben Shneiderman, Plaisant, Cohen und Jakobs [46] und Budiu und Nielsen [32] an.

R28: Aktionen von Userinnen und Usern sollten rückgängig-machbar sein.

Neto und Pimentel [29] empfehlen Userinnen und User über Fehler zu informieren und eine gewisse Fehlerstabilität zu implementieren. Falls also ein Fehler passiert, soll die App zum letzten stabilen Status zurückkehren. Ist dies nicht einfach möglich, sollen laut den Autoren Userinnen und User entscheiden, wie weiter vorgegangen wird.

R29: Bei Fehlern sollen die Nutzerinnen und Nutzer gewarnt werden und die App in den letzten stabilen Zustand zurückkehren. Wenn der stabile Zustand nicht einfach zu erreichen ist, soll die Kontrolle an die Benutzerinnen und Benutzer übergeben werden.

Holzinger und Errath [20] geben an, dass man Userinnen und Usern alle Änderungen und Editierungen, die sie gemacht haben auch bestätigen lassen sollte, um Fehler zu vermeiden.

R30: Verlange eine Bestätigung für Änderungs- und Editierungsaufgaben.

Laut Neto und Pimentel [29] sollte das System auf Situationen achten, aus denen ein Fehler von Userinnen und Usern entstehen kann. So sollte eine Warnung an die Benutzerinnen und Benutzer ausgegeben werden, wenn Daten verloren gehen könnten.

R31: Erinnere die Userinnen und User an ungespeicherte Veränderungen.

Neto und Pimentel [29] geben in ihren Richtlinien an, dass das Feedback für Userinnen und User leicht zu sehen und zu verstehen sein muss. Wenn Benutzerinnen und Benutzer kein Feedback bekommen, verlieren sie das Gefühl dafür, in welchem Status sich das System befindet.

Beim Feedback in Dialogform oder als Alert geben sie die Empfehlung, dass der Text keinesfalls offensichtliche und überflüssige Meldungen enthalten sollte. Er sollte in einer positiven Art und Weise formuliert sein.

Auch Nilsson [33] ist der Meinung, dass Userinnen und User darüber informiert werden sollen, was passiert. Ein mögliches Beispiel für die Umsetzung ist eine Fortschrittsanzeige.

R32: Implementiere Feedback für die Userinteraktion.

Inostroza und Rusu [24] geben in einer ihrer Design-Guidelines für Android-Systeme an, dass eine Anpassungsmöglichkeit an Userbedürfnisse die Usability steigert.

R33: Erlaube es den Userinnen und Usern die App an ihre Bedürfnisse anzupassen.

Inostroza und Rusu [24] geben eine Android-Design-Richtlinie an, die empfiehlt

Funktionen bereitzustellen, die mehrere Arbeitsschritte zusammenfassen und damit ungeübte Userinnen und User unterstützen.

R34: Fasse wichtige Arbeitsschritte als Feature zusammen, um ungeübte Userinnen und User zu unterstützen.

Inostroza und Rusu [24] haben ihrer Heuristik - Physische Interaktion und Ergonomie - zwei Richtlinien zugeordnet eine für IOS- und eine für Android-Applikationen, die empfehlen, Elemente direkt manipulieren zu lassen, anstatt sie über Menüs und Buttons zu beeinflussen, weil dabei die Belastung für das Gedächtnis der Userinnen und User verringert wird.

R35: Gib den Userinnen und Usern die Möglichkeit Elemente direkt zu manipulieren.

6.5 Inhalt

Bei der Gestaltung des Inhaltes für eine mobile Applikation, die auf einer Desktop-Applikation basiert. Sollte laut Budiu und Nielson [32] darauf geachtet werden, dass die App zwar ähnliche Funktionen enthält, aber nur so viele, wie für die Verwendung notwendig sind. Zu viele unnötige Funktionen verkomplizieren das Bedienen der App und führen bei Userinnen und Usern zu Verwirrung und Frustration. Jeglicher unnötiger, zusätzlicher Inhalt ist überflüssig.

R36: Verwende für die Desktop-Version und für die App ähnlichen Inhalt und ähnliche Funktionen.

Shneiderman, Plaisant, Cohen und Jakobs [46] nennen den guten Kontrast zwischen Hintergrund und dem Inhalt neben dem Einsatz von Farbe und Abständen als gutes Mittel den Inhalt unterscheidbar, erkennbar und leicht erfassbar zu machen.

Für Neto und Pimentel [29] dürfen einzelne Elemente weder zu dicht beieinander noch zu weit entfernt sein. Um eine Zusammengehörigkeit von einzelnen Bereichen zu visualisieren sollen sie nahe beieinander stehen. Außerdem muss für sie die auf einer Seite platzierte Information immer sichtbar und lesbar sein, auch wenn sich die Ansicht von Portrait auf Landscape ändern würde. Alle Elemente müssen genügend Kontrast haben und sinnvoll platziert werden.

R37: Inhalt sollte gut erkennbar und leicht zu überblicken sein.

Holzinger und Errath [20] sind der Meinung, dass die Darstellung von Text in einer mobilen Applikation überhaupt nur minimal verwendet werden soll. Sie untermauern

ihre Meinung mit dem Fakt, dass die Benutzerinnen und Benutzer, wenn sie die App verwenden unterwegs sind und Einflüssen ausgesetzt sind, die störend wirken und das Lesen von Texten damit erheblich eingeschränkt ist.

Auch Budiu und Nielsen [32] sprechen sich dafür aus, Texte insbesondere auf den eingeschränkten Anzeigebereich von Smartphones anzupassen. Beide Autoren sehen den Grund für das erschwerte Lesen von Texten auf kleinen Displays darin, dass immer nur ein kleiner Teil des Bildschirms sichtbar ist. Daher ist es extrem schwierig, den Gesamthalt eines Textes zu erfassen. Die Benutzerinnen und Benutzer sind gezwungen, sich viele Teile zu merken, was die Belastung des Kurzzeitgedächtnisses steigert.

Wenn lange Texte nicht vermeidbar sind, schlagen die Autorinnen und Autoren vor den Text einfach vertikal untereinander anzuzeigen. Eine lineare Paginierung wie bei Büchern ist aber nicht sinnvoll, weil das Suchen nach einer bestimmten Information in einer solchen Anzeige noch schwieriger ist und viel länger dauert, als in einem scrollbaren Text. Eine weitere Empfehlung der Autorinnen und Autoren ist, es die Sprache an die Userinnen und User anzupassen, um eine größere Akzeptanz und leichtere Informationsaufnahme zu gewährleisten.

Auch Neto und Pimentel [29] sprechen sich für eine user-orientierte Sprache aus. Nutzerinnen und Nutzer sollen nicht das Gefühl bekommen unter Druck zu stehen. Alle Anweisungen müssen klar, unmissverständlich und objektiv sein.

Inostroza und Rusu [24] ordnen ihrer Heuristik - ästhetisches und minimalistisches Design - eine Android-Design-Guideline zu, die besagt, dass Sätze möglichst einfach und kurz sein sollten, weil die Benutzerinnen und Benutzer dazu neigen, zu lange Sätze nicht zu lesen.

R38: Vermeide lange Texte und verwende einfache Sätze.

Adipat, Zhang und Zhou [2] beschreiben in ihrem Artikel die Wichtigkeit des Hervorhebens von Schlüsselwörtern und Textpassagen. Eine Akzentuierung von bestimmten Teilen eines Dokumentes zieht laut der Autorin und den Autoren die Aufmerksamkeit der Benutzerinnen und Benutzer auf sich. Statt den ganzen Text zu lesen suchen die meisten Userinnen und User nach bestimmten Schlüsselwörtern im Dokument oder auf einer Webseite.

Shneiderman, Plaisant, Cohen und Jakobs [46] nennen noch weitere Möglichkeiten der Hervorhebung um die Aufmerksamkeit der Benutzerinnen und Benutzer zu erhalten:

- **Intensität:** Es sollten nur zwei unterschiedliche Kategorien verwendet werden, wobei die intensivere nur spärlich eingesetzt werden sollte.
- **Markierung:** Unterstreichen, einschließen in Box, mit einem Pfeil markieren oder das verwenden von Indikatoren sind Markierungsmöglichkeiten.

- **Größe:** Bis zu vier unterschiedliche Schriftgrößen sind verwendbar. Je größer die Schrift, desto größer die Aufmerksamkeit.
- **Schriftart:** Es sollten nicht mehr als drei unterschiedliche Schriftarten verwendet werden.
- **Farben umkehren**
- **Blinken:** Diese Art der Hervorhebung sollt nur ganz selten eingesetzt werden.
- **Farbe:** Vier verschiedene Farben sollten als Standard festgelegt werden, zusätzliche Farben können hin und wieder verwendet werden.
- **Audio:** Wenn Audio verwendet wird, sollten nur angenehme Sounds zum Zweck des Feedbacks eingesetzt werden. Schrille, unangenehme Sounds sind für Warnungen vorbehalten.

R39: Hebe wichtige Information hervor.

Budiu und Nielsen [32] sehen die Startseite als einen wichtigen Einstiegspunkt für die App. Dort sollte genug Information vorhanden sein, damit die Benutzerinnen und Benutzer einen klaren Überblick darüber haben, worum es in der App geht. Außerdem muss darauf geachtet werden, dass nur klare Inhalte vermittelt werden, weil sich schon hier entscheiden kann, ob die Userinnen und User die App weiter benutzen werden. Eine schlechte, verwirrende Startseite kann sich laut den Autoren auf das Image einer Firma auswirken.

R40: Auf der Startseite sollte nicht zu viel Information stehen.

Besonderen Wert legen Budiu und Nielsen [32] sowie auch Shneiderman, Plaisant, Cohen und Jakobs [46] auf einen aufgabenorientieren Workflow, der die Benutzerinnen und Benutzer bei der Erledigung ihrer Tasks unterstützt. Jeder Schritt sollte dabei zielgerichtet und selbsterklärend sein.

Inostroza und Rusu [24] geben in ihrer Arbeit eine Richtlinie für Android-Applikationen an, die besagt, dass die Arbeit der Benutzerinnen und Benutzer - der Workflow - nur dann unterbrochen werden soll, wenn es wirklich notwendig ist, sonst werden sie unnötig aus der Konzentration gerissen.

R41: Überlege einen klaren, zielgerichteten Workflow.

Laut Holzinger und Errath [20] sowie Budiu und Nielsen [32] muss Inhalt in einer mobilen App besonders sorgfältig strukturiert werden. Wie schon weiter oben erwähnt ist die Startseite der Punkt, von wo aus man die wichtigsten Features und Inhalte erreichen können muss.

Inhalte, die komplexer sind, sollten nach Wichtigkeit strukturiert werden. Budiu und Nielsen [32] schlagen dazu vor den Inhalt in einen Teil aufzubereiten, der die wichtigsten Informationen enthält und in einen zweiten Teil mit sekundärer Information, die erst auf einer weiteren Seite zur Ansicht bereitgestellt wird. Dazu muss man sich genau überlegen, was die Benutzerinnen und Benutzer zuerst sehen wollen. Die wichtigste Information muss auf einen Blick sichtbar sein, weil Userinnen und User von mobilen Anwendungen laut Budiu und Nielsen noch weniger Zeit haben als Desktopuserinnen und -user. Die Autorin und der Autor geben mehrere Möglichkeiten an, Inhalte zu strukturieren.

1. **Sekundäre Seiten:** Wie schon oben erwähnt wird die Kerninformation auf einer Seite präsentiert und Weiterführendes auf sekundären Seiten.
2. **Mini-IA (Mini-Informationsarchitektur):** Ist der Begriff für die Strukturierung von Inhalt zu einem einzelnen Thema. Eine Mini-IA kann nach mehreren Kriterien geordnet werden:
 - *lineare Paginierung:* Die ist wie schon oben kurz angesprochen außer für das Lesen von Büchern meist ungeeignet.
 - *alphabetische Sortierung:* Sie ist nur sinnvoll, wenn die Benutzerinnen und Benutzer nach einem bestimmten schon bekannten Schlagwort suchen.
 - *vom Generellen, Wichtigen zum Detail:* Wie schon im Absatz davor genannt, kann das eine sehr effektive Weise sein, um den Inhalt für Benutzerinnen und Benutzer schneller zugänglich zu machen.

Inostroza und Rusu [24] erwähnen basierend auf ihrer Heuristik - ästhetisches und minimalistisches Design - eine Design-Richtlinie von Google für Android-Systeme, die ebenfalls besagt, dass Information in kleine, verarbeitbare Teile aufgliedert werden sollte, um die Benutzerinnen und Benutzer nicht zu überfordern und Information nur dann angezeigt werden soll, wenn sie gebraucht wird.

R42: Strukturiere Inhalt nach seiner Wichtigkeit.

Text und Bild

Die Biberfragen werden im Allgemeinen mit Bildern kombiniert. Bei vielen der Fragen dient das Bild zur Veranschaulichung des im Textes erklärten Inhaltes. Daher wurde auch hier nach Richtlinien gesucht, die die Darstellung von Text und Bild in Kombination behandeln.

Niegemann, Domagk, Hessel, Hein, Hupfer und Zobel [30] fassen im „Kompendium Multimedialen Lernen“ für die Kombination von Text und Bild Prinzipien aus der

kognitiven Theorie des multimedialen Lernens und dem integrierten Modell des Text- und Bildverstehens zusammen.

Im Besonderen sollte bei der Verwendung von Text und Bild das Prinzip der Kontiguität aus der kognitiven Theorie des multimedialen Lernens und das Prinzip der Reihenfolge aus dem integrierten Modell des Text- und Bildverstehens beachtet werden. Sie beschreiben zwei Arten dieses Kontiguitätsprinzips - die räumliche und die zeitliche Kontiguität. Bei der ersten Variante sollen Text und Bilder, die miteinander in Verbindung stehen, nahe zueinander dargestellt werden. Die zeitliche Kontiguität bezieht sich auf Audio und Bilder. Hier sollen Bilder zur selben Zeit wie der dazu gehörende audielle Inhalt dargestellt werden. Im Fall der Biberfragen wird aber nur die räumliche Kontiguität tragend. Niegemann führt weiter aus, dass das Kontiguitätsprinzip ein Teilbereich des Split-Attention-Prinzips ist. Laut diesem soll es vermieden werden, Inhalte so aufzuteilen, dass die Aufmerksamkeit auf mehrere Bereiche aufgeteilt werden muss. Die vier folgenden Punkte sollen daher beachtet werden:

- **Erklärender Text** sollte, wenn möglich direkt in eine Grafik miteingefügt werden, sonst nahe beieinander.
- Um eine **Zusammengehörigkeit** von Grafikteilen und erklärendem Text darzustellen können Linien verwendet werden .
- **Erklärungen zu Zuständen oder Ereignissen** sollten mit Hilfe von Pop-ups beim „mouse-over“ dargestellt werden.
- **Grafik und Text** sollten immer gleichzeitig sichtbar sein.

Das Prinzip der Reihenfolge von Text und Bild besagt, dass wenn eine gleichzeitige Darstellung von Text und Bild nicht möglich ist, das Bild zuerst angezeigt werden soll. Dieser Fall tritt meist ein, wenn das Bild im Verhältnis zur Anzeigefläche zu groß oder zu komplex ist.

Das integrierte Modell des Text- und Bildverstehens empfiehlt das Bild zuerst anzuzeigen, weil es mit Hilfe eines Textes nicht möglich ist, Inhalt so zu beschreiben, dass eindeutig nur ein Bild dazu gehören kann. Der Lernende erzeugt beim Lesen des Textes ein mentales Bild, das dann vom folgenden Bild abweichen kann. In diesem Fall würden sich das mentale Bild und das angezeigte Bild stören. Daher ergeben sich für die Kombination von Text und Bild folgende beiden Richtlinien:

R43: Zeige ein Bild und den dazugehörigen Text nahe beieinander an.

R44: Wenn eine gleichzeitige Anzeige von Text und Bild nicht möglich ist, zeige zuerst das Bild an.

6.6 Performance

Wie schon im Abschnitt - Usability und Smartphones - beschrieben, ist die Usability abhängig von der Konnektivität. Die mobile Datenübertragung hat sich in den letzten Jahren sehr verbessert, aber es gibt immer noch kein Netz, das vollkommen flächendeckend ist. Es gibt daher noch immer Regionen, wo nur sehr geringe in manchen Fällen sogar keine Konnektivität vorhanden ist. Außerdem gibt es das Problem von Ballungsräumen: je mehr Userinnen und User das Netz an einem Punkt benutzen, desto größer die Übertragungszeiten. Daher sollte man die Ladezeiten sehr gering halten. Benutzerinnen und Benutzer werden schnell ungeduldig, wenn der nächste Bildschirm nicht nach kurzer Zeit erscheint und werden die App voraussichtlich schließen.

R45: Verringere Ladezeiten für Benutzerinnen und Benutzer. Die, die Applikation öfter verwenden sind bei Ladezeiten toleranter.

Kapitel 7

Verwendete Technologien

Die Umsetzung des Prototypen basiert auf bekannten Webtechnologien, einem Cross-plattformtool und einer Javascript Library für die Gestaltung der Oberfläche. Die Datenspeicherung innerhalb der App basiert auf einer WEB-Database. Welche der aktuellen Technologien für den Prototyp verwendet worden sind, wird unten folgend angeführt und deren Eigenschaften werden kurz zusammengefasst.

7.1 Webtechnologien

Für die grafische und funktionelle Umsetzung der App wurden bekannte Webtechnologien eingesetzt, die in den folgenden Abschnitten kurz beschrieben werden.

Html5

Hypertext Markup Language (HTML) ist eine Beschreibungssprache, durch die der Web Browser weiß, wie Inhalt (zum Beispiel: Text und Bilder) anzuzeigen ist. Die derzeit aktuelle Version ist HTML5, die HTML4, XHTML und HTML DOM Lever 2 ablösen soll.

HTML5 wurde so entwickelt, dass bei der Anzeige von komplexen Inhalten auf zusätzliche Plugins verzichtet werden kann. Die Plattformunabhängigkeit ist einer der größten Vorteile von HTML5 [44].

CSS

Cascading Style Sheets ist eine Technologie, mit der die Anzeige von Inhalten auf einer Seite gesteuert werden kann. Zu einer HTML-Seite kann ein Style Sheet File erstellt werden, indem die Anzeigoptionen, Strukturierung und Positionierung für Text, Bilder und viele weitere Elemente definiert werden können. Außerdem können in diesem File die allgemeinen Seitenanzeigoptionen gesetzt werden. Diese Anzeigedefinition

muss nur einmal gemacht werden und kann dann in beliebig vielen HTML-Seiten eingebunden werden um eine persistente Anzeige von Elementen zu garantieren und Codezeilen einzusparen [44].

Javascript

Um Inhalt von Webseiten interaktiver und dynamischer zu gestalten, wird die Skriptsprache Javascript verwendet. Mit ihr können andere Applikationen, wie ein Web-Browser kontrolliert werden. Javascript ist eine clientseitige Skriptsprache, die im Web Browser des Clients und nicht auf einem Server ausgeführt wird. Diese Programmiersprache ist sehr populär und wird ständig weiterentwickelt und ist deshalb performanter [44].

Polymer

Die von HTML zur Verfügung gestellten Elemente wie zum Beispiel `<button>` haben Attribute, Eigenschaften, Methoden und Events, sowie Styling, sind aber in ihrer Funktion beschränkt. Um komplexere Elemente zu erstellen, müssen mehrere Komponenten zusammenarbeiten (HTML + CSS + Scripts). Web Components stellen die Basis zur Verfügung, um die Inhalte von HTML mit eigenen Custom-Elements zu erweitern. Polymer baut auf den Web-Components auf und erleichtert es eigene Custom-Elements zu erstellen [39].

7.2 Crossplattformtool

Um eine hybride App zu erstellen, die sowohl auf IOS- als auch Android-Geräten verwendbar ist, war es nötig ein Crossplattformtool zu verwenden. Diese Tools ersparen es dem Entwickler je eine separate native App für das jeweilige Betriebssystem zu entwickeln. Es werden im Allgemeinen, wie schon erwähnt, nicht nur Zeit, sondern auch Kosten gespart. Im Fall der BiberApp wurde Apache Cordova als Crossplattformtool ausgewählt.

Apache Cordova

Apache Cordova ist eine Sammlung von Geräte APIs, die das Programmieren von hybriden Apps und die Verwendung der nativen Gerätefunktionen erlaubt. Die App wird nur mit bekannten Technologien wie HTML, CSS und Javascript erstellt. Mit Apache Cordova ist es möglich lokal am eigenen Computer Apps zu entwickeln. Nach der Installation und dem Erstellen eines Projektes können die verschiedenen Plattformen hinzugefügt werden, für die die App entwickelt werden soll. Es ist dann möglich, die App für eine bestimmte Plattform zu generieren und sie entweder auf einem Gerät oder mittels Emulator zu testen [36].

Bestandteile von Apache Cordova

Apache Cordova besteht aus den folgenden Komponenten:

- **Source Code** für einen nativen Container pro unterstützter Plattform, der den Web Application Content auf dem Gerät anzeigt. Bei Plattformen, deren native Applikation schon eine Web-Applikation ist, gibt es keinen Container.
- **Mehrere APIs**, die als Plugins umgesetzt wurden. Über diese APIs kann die Web-Applikation, die in dem Container läuft, auf native Funktionen zugreifen.
- **Tools** zum Erstellen von Projekten, Einfügen und Arbeiten mit Plugins, zum Erstellen von nativen Apps und zum Testen von Apps auf einem Emulator oder auf dem Gerät selbst.
- **Dokumentation** für alle APIs und Tools.

Apache Cordova oder Adobe Phonegap

Apache Cordova oder Adobe Phonegap? Was ist der Unterschied? Wargo findet in seinem Buch „Apache Cordova 4 Programming“ eine deutliche Unterscheidung. Adobe Phonegap ist eine Distribution von Apache Cordova, bei der zum Basisinhalt von Apache Cordova - den oben genannten Komponenten - noch extra Funktionen hinzugefügt wurden. Beide unterscheiden sich laut Autor besonders durch die Command-Line-Tools, Phonegap-Build und die PhoneGap-Developer-App. Die Gemeinsamkeit, die Apache Cordova und Phonegap teilen ist, dass beide gratis zu nutzen sind [50].

7.3 Speicher

Der Speicher ist ein zentraler Teil der App. In ihm sollen alle Biberfragen samt zugehörigen Bildern hinterlegt und die während der Benutzung der App entstehenden Daten sollen dorthin geschrieben und auch von dort wieder ausgelesen werden.

SQLite-Datenbank - SQLite-Plugin

Der Speicher der App wurde anhand einer SQLite-Datenbank umgesetzt. Speziell für Apache Cordova und Adobe Phonegap existieren mehrere Plugins, die einen Speicher in Form einer SQLite-Datenbank zur Verfügung stellen. Sie ist, wie Owens und Grant [3] in ihrem Buch beschreiben, eine eingebettete Datenbank und sie läuft als eigenständiger Prozess innerhalb einer Applikation, der sie Daten liefert. Der Code für die Datenbank ist in das Programm für die Applikation eingebettet. Von außen ist die relationale Datenbank nicht sichtbar. SQLite-Datenbanken haben mehrere Features, die sie im Besonderen für mobile Applikationen sehr attraktiv machen:

- **Keine Konfiguration:** Durch das besondere Design von SQLite brauchen diese Datenbanken keine Administration. Es gibt keinen Server, der spezielle Konfigurationen nötig machen würde und eine Netzwerkkonfiguration wird daher auch nicht benötigt [21].
- **Portabilität:** Ein Aspekt bei der Entwicklung von SQLite war die Portabilität. SQLite ist mit den meisten bekannten Betriebssystemen und ihren mobilen Versionen verwendbar. Es unterstützt sowohl 32- als auch 64-bit Systeme. Auch die SQLite-Datenbanken selbst sind höchst portabel. Da es einfache Files sind, können sie mit einem Betriebssystem erstellt werden und von einem anderen nahtlos weiterverwendet werden. SQLite-Datenbanken können bis zu 2 Terabyte groß sein. Eingeschränkt wird die Größe nur durch die maximal zulässige Filegröße eines Betriebssystems [3].
- **Kompaktheit:** SQLite wurde mit Absicht sehr kompakt gehalten. Es gibt nur ein Header File und eine Library. Es wird kein Server benötigt und daher ist auch keine spezielle Konfiguration nötig. SQLite ist insgesamt weniger als ein halbes Megabyte groß. SQLite-Datenbanken sind ganz normale Files. In diesem File befinden sich alle Objekte wie Tabellen, Trigger, Schemas, Indizes und Views. Hauptsächlich werden Records mit variabler Länge verwendet, um so wenig Speicher wie möglich für ein einzelnes Feld zu verwenden [3].
- **Einfachheit:** Die API von SQLite ist laut Autoren sehr einfach und intuitiv. Sie wurde dafür entwickelt um sie mit eigenen Funktionen zu erweitern. Daher wurden viele Language- und Library-Interfaces erstellt, die mit SQLite nutzbar sind. Die Modularität hilft SQLite dabei seine Einfachheit trotz Weiterentwicklung zu behalten. Jedes Modul hat nur eine spezielle Aufgabe und ist ein unabhängiges System. Diese Grundlage macht es möglich jedes einzelne Modul unabhängig von den anderen zu entwickeln und Queries zu debuggen, wenn sie von einem zum anderen Modul wechseln. Das Frontend und das Backend sind bei SQLite strikt getrennt, so können auch diese unabhängig voneinander entwickelt werden [3].
- **Flexibilität:** Durch die spezielle Beschaffenheit der SQLite-Datenbanken kommt auch ihre Flexibilität zustande. Sie vereint das Frontend einer relationalen Datenbank mit dem Backend eines B-tree [3].
- **Liberale Lizenzierung:** Für SQLite gibt es keine Lizenz und kein Copyright. Der gesamte Code ist in der Public Domain [21].
- **Verlässlichkeit:** Die Verlässlichkeit begründet sich laut den Autoren Owens and Grant durch die sorgfältige Programmierung mit Schwerpunkt auf Modularität. Der Code ist durch umfassende Dokumentation gut nachvollziehbar. Es gibt auch eine API um das Anpassen und Erweitern von SQLite einfacher zu machen. Um

die Verlässlichkeit gewährleisten zu können, wurde großes Augenmerk auf das Testen gelegt [3].

- **Bequemlichkeit:** Die Bequemlichkeit von SQLite entsteht durch die enthaltenen Funktionen: dynamisches Typisieren (Typ einer Variable wird durch ihren Wert bestimmt nicht durch die Deklaration), Konflikt Auflösung (statt beim INSERT mit einem SELECT zu prüfen, ob es den Datensatz schon gibt und ein UPDATE zu machen, kann einfach INSERT OR REPLACE verwendet werden) und mehrere Datenbanken einer Session zuordnen zu können (ATTACH database) [3].

Kapitel 8

Der Prototyp

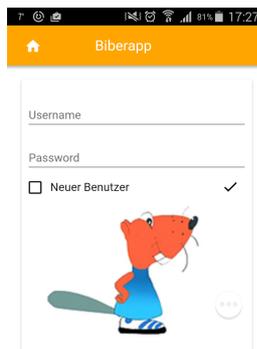
Die folgenden Abschnitte beschreiben den entstandenen Prototypen „BiberApp“ und richten sich dabei nach dem grundsätzliche Aufbau der App. Die BiberApp besteht aus einer Oberfläche für die Interaktion von Userinnen und Usern mit der App und einem darunterliegenden Speicher, der durch eine SQLite-Datenbank umgesetzt wurde.

Dieser Speicher wird lokal auf dem Endgerät erzeugt, wenn die App installiert wird. Alle Daten aus der Oberfläche, die gespeichert werden müssen, werden in Tabellen in der dahinterliegenden SQLite-Datenbank abgelegt. Das Kernstück dieser Datenbank ist die Tabelle, die die Information für die Anzeige der Biber Fragen enthält.

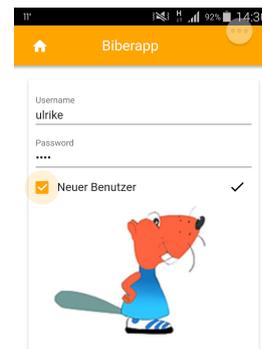
8.1 Startseite



(a) Starticon der BiberApp



(b) Leere Startseite



(c) Startseite mit Logindaten

Abbildung 8.1: Startseite

Der Einstiegspunkt der BiberApp ist eine Startseite, die nur aus wenigen Komponenten - einem Bild des Bibers und einem Login-Panel besteht. Das Login ist sehr übersichtlich gehalten und beinhaltet nur zwei Textinput-Felder und eine Checkbox. Eines der Eingabefelder ist vorbehalten für die Eingabe eines Benutzernamens und ein zweites für das Passwort. Die Checkbox „neuer Benutzer“ muss aktiviert werden, wenn eine neue Benutzerin oder ein neuer Benutzer angelegt werden soll.

8.2 Navigationsseite

Ist das Login abgeschlossen, wird man auf die zentrale Seite für die Navigation weitergeleitet. Von diesem Herzstück aus haben Userinnen und User die Möglichkeit ihr Wissen mit dem Feature „Contest“ zu testen, Fragen aus allen Alters- und Schwierigkeitskategorien zu üben, intensiv zu Üben, indem sie versuchen ihre falsch beantworteten Fragen richtig zu lösen, nach einer bestimmten Frage oder mehreren Fragen einer bestimmten Kategorie suchen, oder aber sie können sich ihre bisherigen Resultate ansehen.

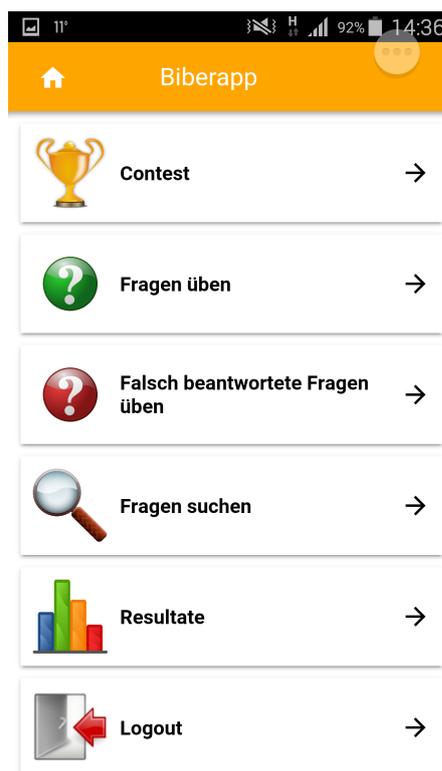


Abbildung 8.2: Navigationsseite

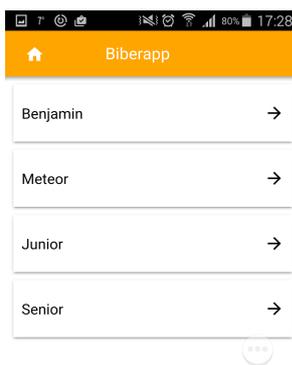
8.3 Contest

Dieses Feature der BiberApp ist dem Wettbewerb Biber der Informatik nachempfunden. Es soll den Benutzerinnen und Benutzern die Möglichkeit geben, sich selbst zu testen, aber auch ihre Erfolge mit anderen vergleichen zu können.

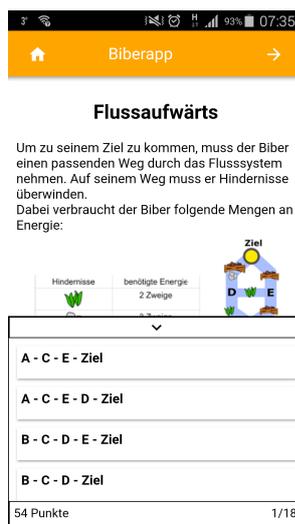
Nach Klick auf Contest in der Navigationsseite wählen die Benutzerinnen und Benutzer die entsprechende Alterskategorie der Auswahlseite, die in der Abbildung 8.3a angezeigt wird. Danach werden entsprechend dieser Auswahl achtzehn zufällig gewählte Fragen nacheinander zur Beantwortung angezeigt.

In der linken unteren Ecke der Frage Abbildung 8.3b findet man die aktuelle Punktezahl im Verlauf des Contests.

Man startet angelehnt an den Biber Wettbewerb das Quiz mit einem Polster von 54 Punkten, damit man nicht in den negativen Punktebereich fallen kann. Bei falscher Beantwortung werden je nach Schwierigkeitsgrad - leicht -2, mittel -3, schwer -4 - Punkte abgezogen, bei der richtigen Lösung 6, 9 oder 12 zum Punktestand addiert. Rechts unten wird angezeigt, bei welcher der achtzehn zufälligen Fragen man sich derzeit befindet und kann daraus schließen, wieviele Fragen noch verbleiben.



(a) Altersgruppenauswahl



(b) Frage aus dem Contest



(c) Resultat des Contests

Abbildung 8.3: Startseite

Hat man alle Fragen beantwortet, wird man zu einer Seite weitergeleitet, die alle Resultate geordnet nach der höchsten Punktezahl anzeigt und wer die Punkte erreicht hat.

Über den „Home-Button“ links oben kehrt man wieder auf die Navigationsseite zurück.

8.4 Fragen üben

Im Bereich - Fragen üben - können, wie der Name des Features schon verrät, alle Biber-Fragen geübt werden. Dem Klick auf die Schaltfläche „Fragen üben“ folgt wie schon oben beim Contest eine Seite mit Schaltflächen zur Auswahl der Fragen für eine bestimmte Altersgruppe.

Die Aufteilung der Fragen in die Altersgruppen basiert auf der schon bestehenden Kategorisierung der Fragen durch den Biber der Informatik. Man kann daher zwischen den in der Abbildung 8.4a angezeigten Möglichkeiten wählen.

Danach werden die der Auswahl entsprechenden Fragen nacheinander zum Üben angezeigt. Hat man alle Fragen durchlaufen erreicht man eine Endseite und kehrt wieder zur Navigationsseite zurück. Im Unterschied zum Contest werden nach der Aus-

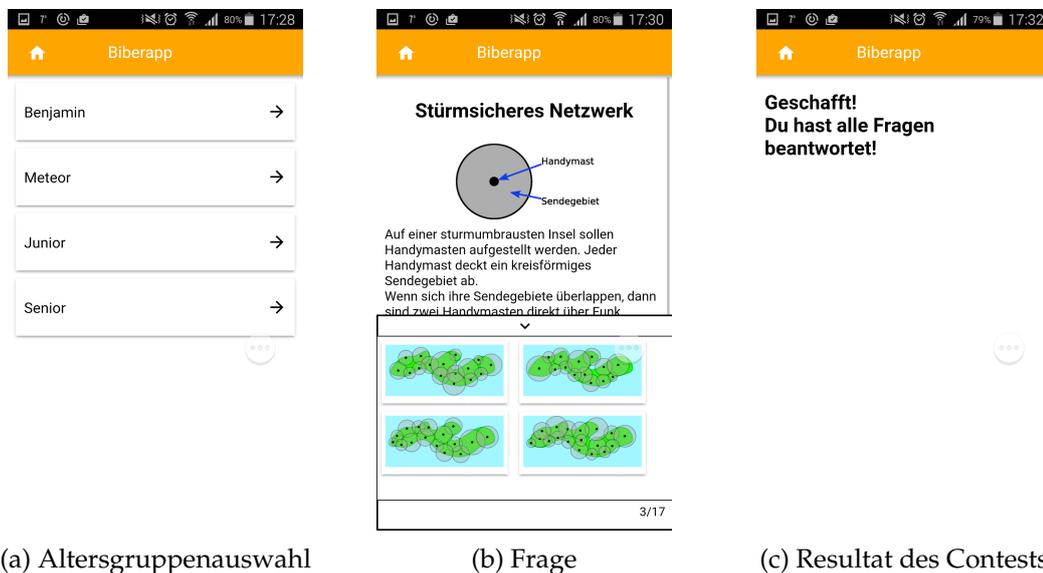
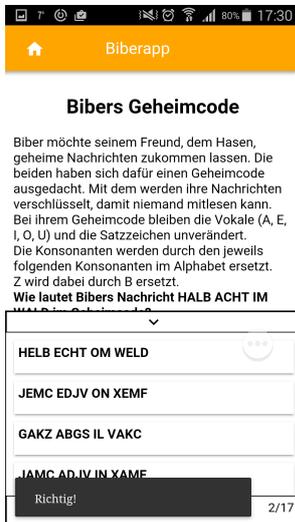


Abbildung 8.4: Fragen üben

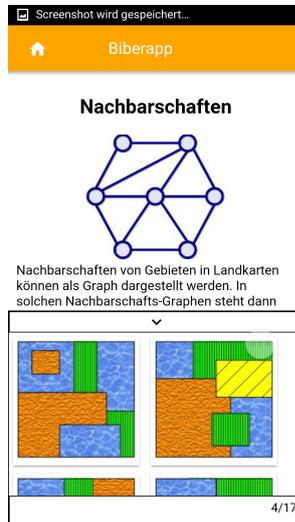
wahl der Alterskategorie alle Fragen aus dieser nacheinander angezeigt. Beim Quiz gibt es nur durch den Blick auf den Punktestand ein indirektes Feedback, ob eine Frage richtig beantwortet wurde. Beim Fragen üben bekommen die Benutzerinnen und Benutzer folgendes direktes Feedback:

- **Richtige Antwort:** Wurde die Frage richtig beantwortet, erscheint ein Alert mit dem Text „richtig“ wie in Abbildung 8.5a.
- **Falsche Antwort:** War die Antwort falsch wird man auf eine Seite weitergeleitet (Abbildung 8.5c), die die richtige Antwort und eine dazugehörige Erklärung enthält, aber es werden auch zusätzliche Informationen zur Frage angezeigt, die

dem Abschnitt „Das ist Informatik“ der Fragen aus dem Biber Wettbewerb entsprechen.



(a) Richtig beantwortete Frage



(b) Frage aus dem Contest



(c) Lösung der Frage

Abbildung 8.5: Richtig und falsch beantwortete Fragen

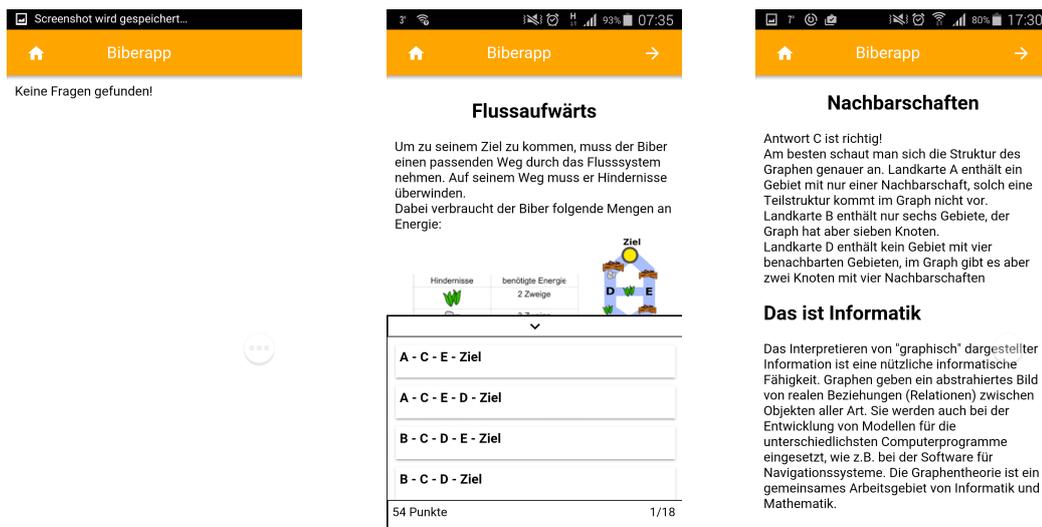
8.5 Falsch beantwortete Fragen üben

Die Kategorie - Falsch beantwortete Fragen üben - soll dazu dienen, speziell auf Fragen einzugehen, die einmal falsch beantwortet wurden.

Zu Beginn gibt es natürlich noch keine falsch beantworteten Fragen. Daher erhalten die Userinnen und User eine Meldung, in der sie über diesen Status informiert werden (Abbildung 8.6a).

Die von den Benutzerinnen und Benutzern während des „Contests“ oder beim „Fragen üben“ falsch beantworteten Fragen werden gesammelt und können über diese Funktion der BiberApp explizit geübt werden. So wie schon beim Fragen üben beschrieben, bekommen die Userinnen und User wieder direktes Feedback zur gewählten Lösung der Fragen.

Fragen werden nur dann wieder aus dieser Sammlung entfernt, wenn sie von Benutzerinnen und Benutzern drei Mal richtig beantwortet wurden.



(a) noch keine falsch beantwort. (b) Frage aus den falsch beantwort. Fragen (c) Abschlussseite dieses Features der BiberApp

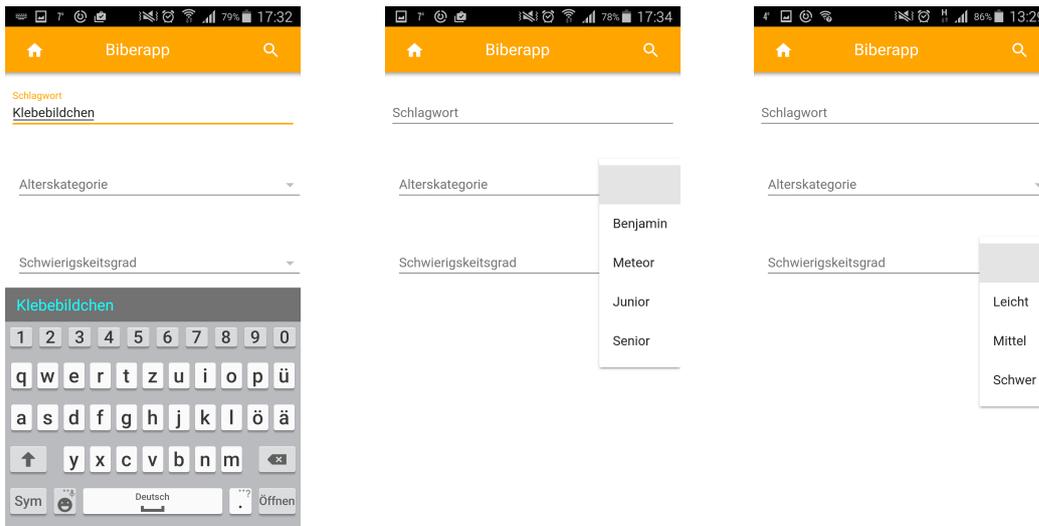
Abbildung 8.6: Falsch beantwortete Fragen üben

8.6 Frage suchen

Dieses Feature der App erlaubt es den Benutzerinnen und Benutzern nach Fragen zu suchen. Sie wurde im Prototypen als Volltextsuche umgesetzt. Die Suchmaske besteht aus einem Textfeld für die Schlagworteingabe und zwei Dropdowns, die die Wahl der Alters- und Schwierigkeitskategorie steuern. Demnach kann man über das Schlagwort nach einer bestimmten Frage oder aber auch nach Fragen zu einem Themengebiet suchen. Dazu wird das Schlagwort in dem Titel, dem Fragentext, der Info zur Frage aber auch in den zugeordneten Tags aller Fragen gesucht. Abbildung 8.7a zeigt die Eingabe eines Suchwortes in der Suchfunktion der BiberApp.

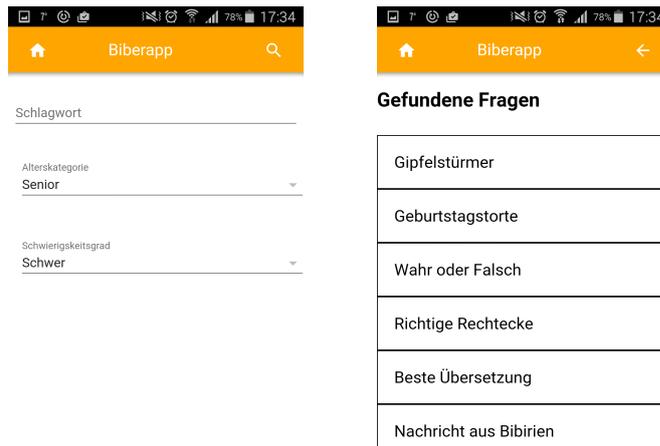
Ebenso ist es möglich nur Fragen einer bestimmten Alterskategorie anzuzeigen wie Abbildung 8.7b zeigt. Es ist auch möglich nur nach Fragen einer bestimmten Schwierigkeitskategorie zu suchen (Abbildung 8.7c).

Noch darüber hinaus kann man die schon genannten Möglichkeiten kombinieren um eine weitere Einschränkung der Ergebnismenge zu erzielen. Die Abbildungen 8.8a und 8.8b zeigen ein mögliches Suchbeispiel und die aus der Suche resultierende Fragenmenge. Die einzelnen Fragen sind als Listenelemente umgesetzt, die auf den Inhalt der Fragen verlinkt sind.



(a) Suche von Fragen über Schlagwort (b) Suche von Fragen aus best. Alterskategorie (c) Suche von Fragen aus best. Schwierigkeitskategorie

Abbildung 8.7: Möglichkeiten der Suche



(a) Parameter der Suche (b) Ergebnismenge der Suche

Abbildung 8.8: Beispiel für Suche

Die Abbildungen 8.9a 8.9b und 8.9c zeigen, wie Fragen im Kontext der Suche aufbereitet wurden. Die Frage selbst wurde mit Hilfe von Tabs in drei Bereiche unterteilt:

- die **Frage** mit ihren Antwortmöglichkeiten
- die **Lösung**

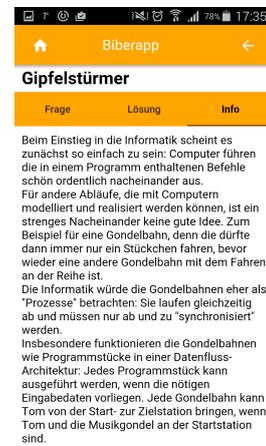
- **Zusatzinformation zur Frage**



(a) Frage



(b) Lösung der Frage



(c) Infotext zur Frage

Abbildung 8.9: Anzeige der Frage in Bereich der Suche

8.7 Resultate

Die Seite - Resultate - bietet einen Überblick über den von den Userinnen und Usern erreichten Punktestand. Die angezeigte Liste ist nach der Punktzahl absteigend geordnet. Damit die einzelnen Zeilen dieser Tabelle besser lesbar sind wurden sie jeweils abwechselnd in weiß und in orange gehalten.

8.8 Logout

Der Klick auf den Button - Logout - führt wieder zurück auf die schon zu Beginn beschriebene Startseite. Von dort aus hat man die Möglichkeit sich erneut anzumelden oder neue Benutzerinnen und Benutzer anzulegen.

8.9 Übersicht über die Funktionsweise des Prototypen

Die Abbildung 8.10 zeigt eine Übersicht über die Funktionen der BiberApp, sowie die in der App möglichen Workflows. Der Doppelpfeil zwischen zwei Knoten bedeutet immer, dass man einerseits von der Startseite zum Contest gelangt und andererseits von dort wieder zum Ausgangspunkt zurückkehrt.

Eine Ausnahme bildet der Knoten „Logout“. Man kehrt nicht zum „Startseite“-Knoten zurück, sondern man wird automatisch zum „Login“-Knoten geleitet.

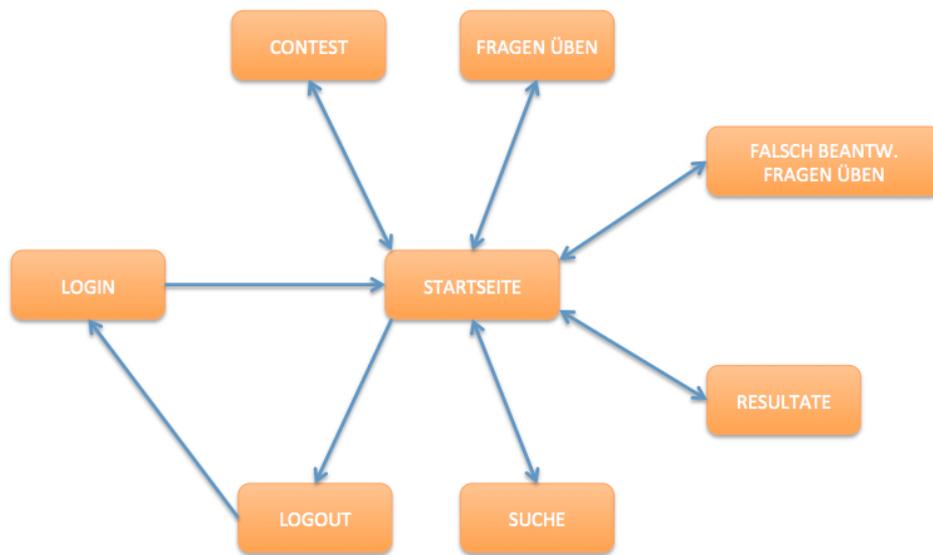


Abbildung 8.10: Übersicht des Prototypen

Kapitel 9

Evaluation

In diesem Kapitel wird einerseits die Evaluation der Umsetzung der Biber-Fragen in einer mobilen Applikation behandelt, aber andererseits wird auch darauf eingegangen, ob die im Prototypen implementierte Suchfunktion einsetzbar ist und wie gut sie für Lehrerinnen und Lehrer einsetzbar sein kann.

9.1 Umsetzung der Biberfragen

Dieses Kapitel beinhaltet die Evaluation zur Umsetzung der Biber-Fragen in der BiberApp. Zu Beginn soll hier noch einmal die zugehörige Forschungsfrage angeführt werden:

Wie und wie gut können vorhandene Fragen des Biber Wettbewerbs in einer Lern-App abgebildet werden?

Für die Evaluation wurden im Laufe der Literaturrecherche Richtlinien zum Erstellen von Smartphone-Apps ermittelt, die die Bereiche Layout, Navigation, Design, Inhalt und Performance betreffen. Dieser Kriterienkatalog wurde herangezogen um den implementierten Prototypen der BiberApp auf die Art und Qualität der Umsetzung der Fragen zum Biber Contest zu prüfen.

Folgende Abschnitte zeigen auf, wo der Prototyp nicht den gefundenen Kriterien aus der Literaturrecherche entspricht und ob auch bewusst und aus welchen Beweggründen auf die Einhaltung verzichtet wurde. Auch für das allgemeine Konzept des Prototypen wurden Richtlinien basierend auf der Literaturrecherche gefunden, deren Anwendung auf den Prototypen im folgendem Abschnitt besprochen werden.

RI: Eine mobile Lernapplikation sollte orts- und zeitunabhängig und auch unabhängig von Stromquelle und Netzzugang nutzbar sein.

Um die Richtlinie RI im Prototypen umzusetzen, wurde die App für Touchphones konzipiert. Dieses haben viele der Benutzerinnen und Benutzer immer mit dabei, weil es höchst portabel ist. Ein weiterer Vorteil der Konzeption für diese Geräte ist die Unabhängigkeit von einer Stromquelle (heutige Handyakkus halten bis zu einer Woche). Außerdem wurde der Prototyp so gestaltet, dass er ohne eine andauernde Verbindung zum Internet benutzt werden kann. Der für die App notwendige Speicher und damit jegliche Information, die für die App gebraucht wird, wird direkt am Gerät installiert. Somit wurde die allgemeine Richtlinie RI im Prototypen in vollem Maß umgesetzt.

RII: Lerneinheiten sollten nicht länger als ein paar Minuten dauern, mit direktem Feedback versehen sein und robust gegen Ablenkungen aus der Umgebung sein.

Diese Richtlinie ist durch das Konzept des Biber der Informatik schon teilweise beachtet. Der Biber Wettbewerb besteht aus nur achtzehn Fragen, die in wenigen Minuten beantwortet sind.

Der Prototyp enthält zu jeder wählbaren Kategorie in etwa zwanzig Fragen. Damit wird auch hier die vorgegebene Zeitbeschränkung eingehalten.

RIII: In welcher Form - WebApp, native oder hybride App - eine App umgesetzt werden soll, sollte entschieden werden durch die Kriterien der späteren Nutzung.

Mit der BiberApp sollen möglichst viele Userinnen und User erreicht werden können. Daher sollte sie, wie schon in der Einleitung genannt wurde, für IOS- und Android-Betriebssysteme verwendbar sein. Ebenso sollte sie sich möglichst wie eine native App verhalten. Daher wurde mit Hilfe von Webtechnologien, einem Crossplatform-Tool und der clientseitigen Datenbank eine hybride App entwickelt.

Layout

In diesem Abschnitt werden einzelnen Richtlinien behandelt, die sich mit der Platzierung von Elementen auf dem Bildschirm befassen. Die nachfolgenden Richtlinien

R1: Zeige den gesamten Inhalt, wenn möglich auf einer Seite an.

R2: Vermeide horizontales Scrollen.

R3: Zeige Inhalt vertikal untereinander an und vermeide zu viele Optionen und Ebenen.

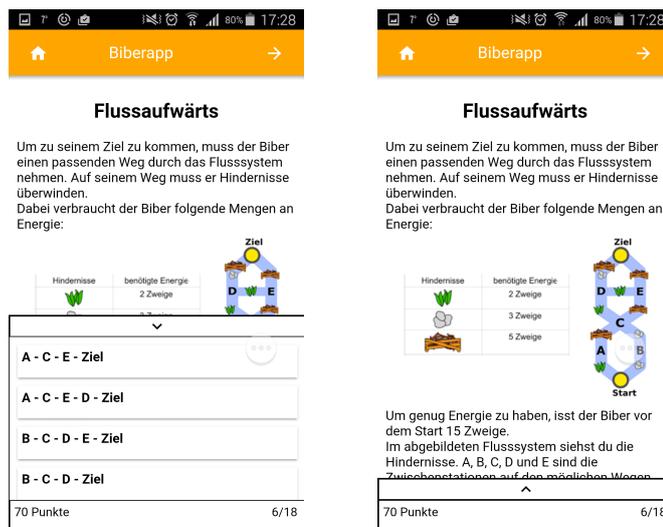
sind Guidelines, die in ihrer Einhaltung oder nicht Einhaltung großen Einfluss auf die zuvor gestellte Forschungsfrage haben können. Die beiden ersten Richtlinien waren in der BiberApp zum Großteil sehr eindeutig und einfach umzusetzen, da der Inhalt der BiberApp hauptsächlich aus den im Biber Wettbewerb verwendeten Fragen besteht. Beispiel dafür ist, dass in der gesamten App auf ein horizontales Scrollen verzichtet wurde.

Der Bereich der Darstellung der Biber-Fragen stellte auf dem Touchphone eine besondere Herausforderung dar: Der Fragen-Text der Biber-Fragen ist in den meisten Fällen eher umfangreich und nimmt auf einem Touchphone-Display schon sehr viel Platz ein. Inhalt und alle Interaktionsmöglichkeiten sollten aber laut R1 möglichst auf einem Bildschirm angezeigt werden. Die Frage allein füllte aber schon mehr als den Bildschirm aus, daher wurde basierend auf der Erkenntnis, dass das vertikale Scrollen mittlerweile in der Gesellschaft schon sehr verankert ist, die Möglichkeit des vertikalen Scrollens gegeben.

Was aber damit nicht gelöst war, war die Anzeige der möglichen Antworten zur Frage. Wenn eine Frage dargestellt wurde, war nicht zu erkennen, dass es sich um eine Quizfrage mit Antwortmöglichkeiten handelt, weil diese eben nicht sichtbar waren. Erst nach relativ langem Scrollen konnte man am Ende die angebotenen Antworten finden. Neben dem Problem des Erkennens war diese Version der Anzeige auch noch sehr unpraktikabel, denn bei jeder Frage musste erst zum Ende gescrollt werden um die Frage auch beantworten zu können.

Es musste daher eine Kompromisslösung gefunden werden. Die folgende Abbildung 9.1 zeigt die für den Prototypen entstandene Designvariante: Der Bildschirm wurde, wie in Abbildung 9.1a sichtbar, in zwei scrollbare Bereiche geteilt. Der obere Bereich wird nur für die Anzeige der Frage genützt, der untere Bereich enthält alle Antwortmöglichkeiten.

Da der Anzeigebereich für den Fragentext und dazugehörige Bilder dadurch noch kleiner geworden ist, wurde der Bereich mit den Antwortmöglichkeiten so gestaltet, dass er bei Bedarf eingeklappt werden kann (9.1b) um mehr von der Frage anzeigen zu können. Eine weitere Herausforderung bildeten die unterschiedliche Beschaffenheit der Antwortmöglichkeiten. Einerseits gibt es rein textuelle Antworten, aber einige der Fragen bieten als Lösungsmöglichkeiten Bilder an.



(a) Antworten sichtbar (b) Antworten eingeklappt

Abbildung 9.1: Frage und Antwortmöglichkeiten

Grundsätzlich wurden die Antworten untereinander als Listenelemente angezeigt. Bei den Bildern entstand das Problem, dass schon ein einziges von vier den ganzen vorbelegten Bereich für die Antwortmöglichkeiten ausgefüllt hat und meist gar nicht als Ganzes zu erkennen war.



(a) Antwortmöglichkeiten Text als Liste (b) Antwortmöglichkeiten Bilder im Raster

Abbildung 9.2: Antwortmöglichkeiten

Um den vorhandenen Bereich optimal zu nützen, wurden die Antwortmöglichkei-

ten, die aus Bildern bestehen, in einem 2 mal 2 Raster positioniert, wie man in der Abbildung 9.2 sehen kann.

R4: Logins sollten nicht direkt zu Beginn der Webseite/der App platziert sein.

Gegen die Regel R4 wird in der App eindeutig verstoßen, die erste Seite der App ist die Loginseite. Die BiberApp steht in direktem Zusammenhang mit dem Biber der Informatik, zu dem es schon eine Desktopapplikation gibt. Die Grundlage für die App waren der Biber der Informatik und ebendiese Desktopapplikation. Damit kann man schlussfolgern, dass schon im Vorhinein bekannt ist, worum es in dieser App geht.

Noch dazu ist es kein Login mit auf externen Servern gespeicherten Daten, sondern dient vorwiegend dazu, dass mehrere Userinnen und User die App gemeinsam auf dem selben Gerät benutzen können. Im Detail ist das Login dafür notwendig, dass Userinnen und User und ihre Spielstände differenziert werden können.



(a) Login für Userinnen und (b) Resultate der verschiedenen Userinnen und User

Abbildung 9.3: Login und Results

R5: Zeige das Suchfeld so an, dass es sichtbar und benutzbar ist.

Die Richtlinie R5 ist auf die BiberApp nicht anwendbar. In der App wurden nämlich auf Grund des sehr knapp gehaltenen Inhalts und dem Fakt, dass eine eigene Suchfunktion innerhalb der App implementiert wurde, keine zusätzlichen Suchfelder eingesetzt. Diese würden die Userinnen und User wahrscheinlich eher verwirren, als wirklich nützlich sein.

R6: Verwende ein klares, einfaches Layout für die App.

Beim Layout der BiberApp wurde besonders darauf geachtet, dass die Bildschirme weder überladen waren, noch wichtige Bereiche erst in Untermenüs oder auf anderen Seiten versteckt waren. Die Platzierung der einzelnen Elemente auf den Screens wurde so gewählt, dass sie möglichst intuitiv zu bedienen und übersichtlich sein sollten.

R7: In einem Dialog sollten nicht mehr als zwei Buttons verwendet werden.

R8: Buttons in einem Dialog findet man immer nach dem Dialogtext, aber in unmittelbarer Nähe dazu.

R9: Für Buttons in Dialogen gilt: Der Button, der mit höchster Wahrscheinlichkeit gewählt werden soll, steht dabei in der Mitte; je nach Design, der zweit wichtigste Button rechts daneben oder darunter; ein weiterer unwichtiger Button links oder darüber.

Alle drei dieser Guidelines wurden in der mobilen Applikation umgesetzt. Dialoge wurden in der BiberApp sehr, sehr sparsam eingesetzt. Es gibt genauer gesagt nur ein einziges Dialogfenster im Zusammenhang mit dem Feature Contest. Es wird dann angezeigt, wenn die Benutzerinnen und Benutzer den „Home“-Button während eines laufenden Contests betätigen. In der unten angezeigten Abbildung 9.4 sieht man, dass der angesprochene Dialog, laut Richtlinie R7 nicht mehr als zwei Buttons beinhaltet, die sich entsprechend R8 direkt unter dem Dialogtext befinden. Der wahrscheinlich am häufigsten geklickte Button befindet sich links und der zweite gleich rechts daneben.



Abbildung 9.4: Dialog

Navigation

Im Zusammenhang mit der Navigation sind die unten in der Box zusammengefassten Guidelines richtungsweisend.

R10: Die Navigationsmenüs sollten so einfach wie möglich gestaltet sein.

R11: Wichtige Seiten sollten direkt von der Startseite erreichbar sein.

R12: Navigationsmenüs sollten nur wenige Ebenen haben.

R13: Vermeide zu viele Optionen und Scrollen in einem Navigationsmenü (nur 4-8 Einträge)

Für die Gestaltung der Navigation wurden im Vorhinein alle möglichen Features, die der Prototyp haben sollte, ermittelt.

In der Navigation für die BiberApp ist es gelungen, diese vier genannten Guidelines strikt einzuhalten. Die Abbildung der Navigationsseite 9.5 zeigt, dass alle Features, die die App enthält, von dieser zentralen Seite erreichbar sind, damit ist die Richtlinie R11 erfüllt. Auch R10 dürfte ob der Kürze und einfachen Listengestalt des Menüs erfüllt worden sein. R12 ist definitiv beachtet worden, denn es gibt in der App gar nicht mehr als diese eine Navigationsebene. Mit den angebotenen sechs Listeneinträgen zur Navigation wurde auch die Richtlinie R13 in der BiberApp umgesetzt.

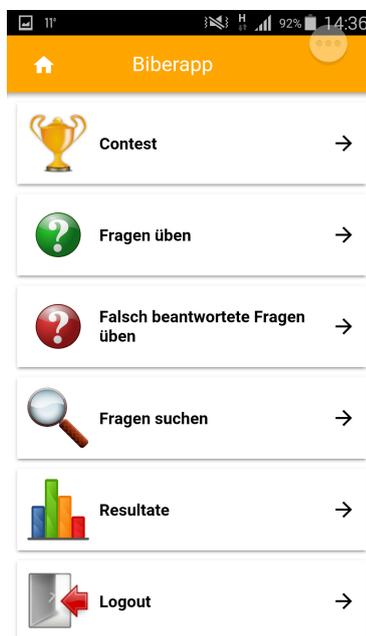


Abbildung 9.5: Navigationsseite

R14: Gib den Userinnen und Usern die Möglichkeit die App schnell zu verlassen. Bei einem Restart beginne wieder an der Abbruchstelle.

R15: Gib den Benutzerinnen und Benutzern die Möglichkeit zum schnellen Erreichen der Startseite und verwende zur Darstellung der aktuellen Position Breadcrumbs.

Die Richtlinien R14 und R15 sind ein weiteres Regelwerk zum Verkürzen der Navigationswege. Damit die Userinnen und User schnell zur Navigationsseite zurückkehren können und bei Bedarf die App in kurzer Zeit schließen können, wurde auf fast jeder Seite mit Ausnahme der Login- und der Navigationsseite des Prototyps ein „Home“-Button links oben im Header platziert (9.6). Dieser führt immer zur Navigationsseite zurück. Über diese Navigationsmöglichkeit kann ausgesprochen schnell zwischen den Features des Prototypen hin und her gewechselt werden.

Zum Verlassen der App geht man ebenfalls den Weg über den Home-Button zur Navigationsseite. Danach kann die App durch das Betätigen des Back-Buttons des Touchphones beendet werden.

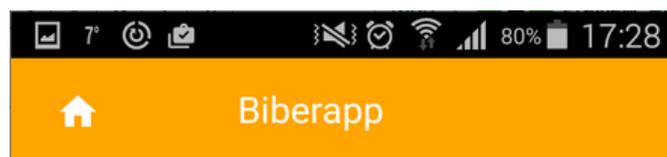


Abbildung 9.6: Header des Prototypen

R15 wurde nur teilweise umgesetzt, die derzeit aktuelle Position wurde nicht über Breadcrumbs sichtbar gemacht. Es sollte hier die Frage gestellt werden, ob es in Anbetracht der Übersichtlichkeit der App und deren geringem Funktionsumfang überhaupt notwendig ist, diese Funktion einzubauen.

R16: Um vieles Herumsuchen zu vermeiden, strukturiere Menüs nach Themen und Art der Verwendung.

Die Richtlinie R16 hat sich beim Prototypen im Besonderen in der Anordnung der verschiedenen Features der App ausgewirkt.

Zu Beginn stand die Frage - „Was würden Userinnen und User wahrscheinlich zuerst mit der App machen wollen?“ Ausgehend davon wurden daher drei Hauptfunktionen Contest, Fragen üben und Falsch beantwortete Fragen üben identifiziert. Diese sollten natürlich in der Reihenfolge im oberen Bereich platziert werden. Der Contest

wurde als erstes Listenelement ausgesucht, da die meisten Userinnen und User sich gerne selbst testen, sich gerne selbst herausfordern. Danach wurde als zweit wichtigste Funktion das Fragen üben platziert. Innerhalb dieser Gruppe bildete das Falsch beantwortete Fragen üben das Schlusslicht, weil man diese Funktion wahrscheinlich nur nutzen wird, wenn man sich intensiv auf einen Wettbewerb vorbereitet.

Bei den noch zur Verfügung stehenden Funktionen Suche, Resultate und Logout wurde die Feature Suche als das wichtigste angesehen und deshalb als viertes Listenelement eingefügt.

Die Funktion Logout hat noch eine besondere Position, denn es steht als letztes Element in der Liste und ist daher auch wieder ein wenig hervorgehoben.

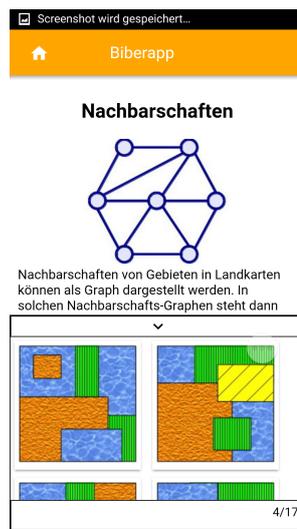
Folgende Anordnung der Funktionen der App sind aus diesen Überlegungen entstanden:

- Contest
- Fragen üben
- Falsch beantwortete Fragen üben
- Suche
- Resultate
- Logout

R17: Öffne Links in einem neuen Fenster, das aktuelle Fenster bleibt unverändert und das neue Fenster hat Bedienelemente, das Fenster zu schließen oder zur Seite zurückzukehren.

In der BiberApp wurden sehr wenige Links im ursprünglichen Sinne verwendet, wird aber ein Popup oder eine neuen Seite geöffnet, wurden Steuerelemente zum Zurückkehren zur Ausgangsseite eingebaut. Der Inhalt der Ursprungsseite wird nicht ersetzt, das neue Fenster wird nur darüber liegend angezeigt.

In BiberApp wurde dies beim Anzeigen der richtigen Lösung verwendet. Die Lösung und noch zusätzliche Information zum Thema werden nach dem Falsch-Beantworten der Frage in einem neuen Fenster angezeigt 9.7, das über einen „Weiter“-Button im Header wieder verlassen werden kann und man kehrt automatisch zu der Frage zurück, im Detail wird die nächste Frage angezeigt, um zusätzliches Klicken zu vermeiden.



(a) Frage



(b) Lösung

Abbildung 9.7: Frage und richtige Lösung

R18: Titel, Links und Navigationselemente sollten unmissverständlich, selbsterklärend und beschreibend sein.

Bei der Benennung von Buttons und Navigationselementen wurde in der BiberApp besonders darauf geachtet, dass sie eindeutig sind und keine Möglichkeit für eine Missinterpretation der Userinnen und User besteht.

Ein Beispiel dafür ist das Navigationselement Contest. Dahinter verbirgt sich genau das, was der Name des Elements schon impliziert.

Design

Die Einhaltung der Richtlinien zur Gestaltung der mobilen Applikation, wie das Aussehen der App, Verwendung von Icons und Steuerungselementen, die für eine Touchsteuerung geeignet sind, wird in diesem Abschnitt diskutiert.

R19: Das Design sollte einfach, konsistent, gleichbleibend und klar sein.

R20: Verwende ähnliches, wiedererkennbares Design für verschiedene Versionen eines Produktes.



Abbildung 9.8: Biber der Informatik

Für das Design der BiberApp wurde vor allem auf ein konsistentes Design geachtet. Ausschlaggebend dabei war auch die Farbgebung. Diese basiert auf dem offiziellen Logo des Biber der Informatik.

Die vorherrschenden Farben in der Abbildung 9.8 sind orange, blau, weiß und schwarz. Als Hauptfarbe wurde orange gewählt, so sind der Header und Hervorhebungen in dieser Farbe gehalten. Als Hintergrund wurde die Farbe weiß gewählt, da sie den besten Kontrast für Inhalt und Hintergrund verspricht. Text ist für den Header weiß, sonst immer schwarz.

Die App sollte im Großen und Ganzen angelehnt an die Designrichtlinien von Google und Apple eher schlicht sein, wichtig dabei waren auch keine überladenen Seiten und klar zu unterscheidende und abgegrenzte Bedienelemente. Auf zu viele Farben sollte möglichst verzichtet werden, da sie die Userinnen und User vom Wesentlichen ablenken können.

In einem frühen Designversuch sollte auch noch die Farbe blau verwendet werden, was aber die farbliche Balance erheblich gestört hat. Dieses 2-Farben Konzept wurde in der gesamten App umgesetzt. Zusätzliche Farbe wird nur über die im Prototypen verwendeten Icons und die in den Biber-Fragen enthaltenen Bilder miteingebracht.

R21: Verwende Icons, die einfach zu erkennen und zu sehen sind.

In der BiberApp sollten Icons nur sehr spärlich eingesetzt werden. Es gibt also nur zwei Bereiche, die in der App in Frage gekommen sind. Einerseits werden Icons für die unterschiedlichen Funktionen der App verwendet, andererseits für Steuerelemente. Insgesamt wurden nur zehn verschiedene Icons eingesetzt. Auch für die Icons für die Steuerelemente sollte ein guter Kontrast vorhanden sein.

Bei den Icons für die Funktionen der App wurde darauf geachtet, dass sie so aussagekräftig wie möglich sind. Abbildung 9.9 zeigt alle Icons, die für die im Prototypen enthaltenen Funktionen eingesetzt wurden. Ebenfalls wurde großes Augenmerk darauf gelegt, dass die Icons in gutem Kontrast zum Hintergrund stehen. Da die Hintergrundfarbe weiß gewählt wurde, konnte das sicher für die Icons der App-Funktionen gewährleistet werden.

Ein Problem entstand nur bei den Funktionen Fragen üben und Falsch beantwortete Fragen üben. Ein Fragezeichen wird meist mit einer Frage verbunden, aber die Schwierigkeit war, zwei unterschiedliche Icons für diese beiden Kategorien zu finden.

Im Prototypen wurde dann für das Fragen üben ein grünes Fragezeichen eingesetzt, für das Falsch beantwortete Fragen üben ein rotes Fragezeichen. Alleinstehend würden die Icons wahrscheinlich nicht richtig interpretiert werden können oder einfach nur zu Verwirrung führen, daher wurden sie mit dem jeweiligen Namen der Funktion in Kombination angezeigt. Für die Steuerungselemente sollte kein zusätzlicher Text für die



Abbildung 9.9: verwendete Icons für Funktionen der BiberApp

Icons notwendig sein, daher wurden Icons verwendet, die den schon bekannten üblichen Konventionen von Desktop- und mobilen Applikationen entsprechen. Abbildung 9.10 zeigt alle verwendeten Icons für Steuerelemente.



Abbildung 9.10: verwendeten Icons für Steuerelemente

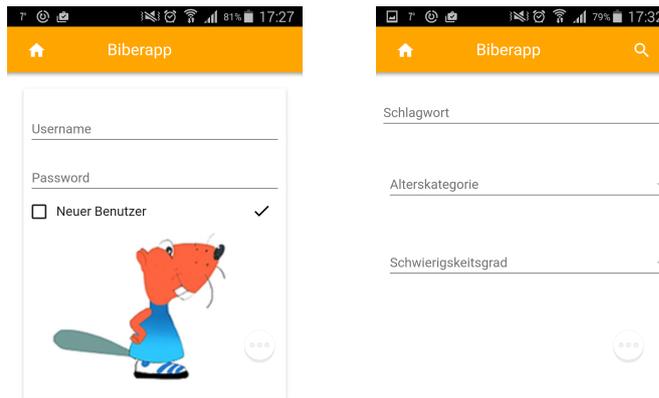
Die Steuer-Icons wurden alle konsistent im Header platziert, der als Hintergrundfarbe orange hat. Daher wurden die Steuericons in weiß gehalten.

R22: Gestalte das Interface so, dass die ungenaue Touchfunktion nicht ins Gewicht fällt. Verwende eine Schaltflächen-Größe von einer Größe zwischen 7-10 mm.

R23: Optimierte die Eingabe von Daten für mobile Geräte.

Bei allen klickbaren Elementen wurde darauf geachtet, dass sie den Konventionen entsprechen, die in der Richtlinie R22 angegeben wurden. Buttons, Icons sowie Listenelemente werden in einer für Finger angenehm klickbaren Größe dargestellt. Noch dazu wurden die einzelnen Elemente in einem genügend großen Abstand zueinander platziert, damit nicht versehentlich das falsche Element geklickt werden kann.

Da das Tippen auf Touchscreens sehr umständlich und fehleranfällig ist, wurden wenn immer möglich Eingabevarianten gewählt, die viel Tipparbeit verhindern. Die einzigen Texteingabefelder in der BiberApp sind die beiden auf der Loginseite und das Eingabefeld für das Schlagwort bei der Suche.



(a) Textfelder Login

(b) Textfeld Suche

Abbildung 9.11: Textfelder in der App

R24: Passe den Inhalt für das mögliche Einblenden der Tastatur an.

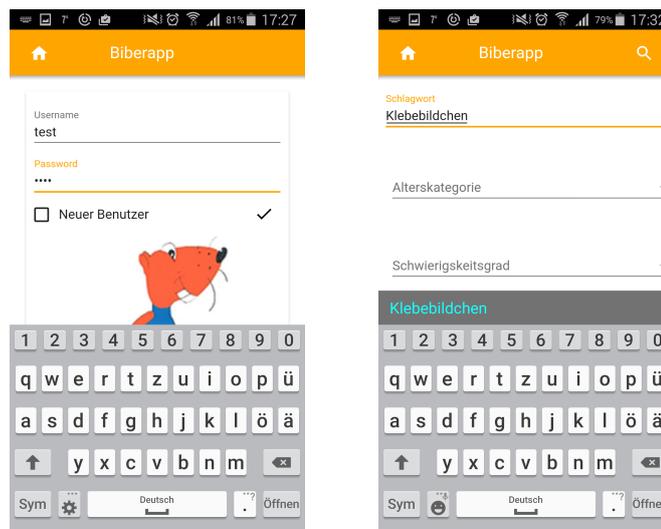
Wie schon im Kapitel der Guidelines erwähnt gibt es mehrere Möglichkeiten die Texteingabe bei Eingabefeldern zu gestalten.

Für den Prototypen wird beim Einblenden der Tastatur ein Teil der Seite absichtlich verdeckt. Das kann deshalb gemacht werden, weil die Eingabefelder so auf der Seite platziert sind, dass kein für die Eingabe wichtiger Teil der App dabei verdeckt wird.

Im Fall der Loginseite wird nur das Bild des Bibers verdeckt, bei der Suche werden zwar in dem Moment weitere Eingabeelemente verdeckt, es ist aber erst sinnvoll diese zu verwenden, wenn das Schlagwort eingegeben worden ist.

R25: Unterstütze bekannte Gesten.

Richtlinie R25 wurde in der App nicht beachtet. Es wurde gänzlich darauf verzichtet Gesten umzusetzen, um eine einheitliche Applikation für beide Betriebssysteme zu gestalten.



(a) Tastatur Login

(b) Tastatur Suche

Abbildung 9.12: Tastatur

R26: Stelle automatische Vervollständigung zur Verfügung.

R27: Stelle automatische Tippfehler-Korrektur zur Verfügung.

Für die einzigen drei Texteingabefelder in der BiberApp wurde die automatische Vervollständigung und Tippfehlerkorrektur nicht explizit umgesetzt, da das Betriebssystem hier schon eine Möglichkeit zur Verfügung stellt, wie man aus der vorher angezeigten Abbildung 9.12b entnehmen kann. Über der Tastatur wird ein dunkelgrau hinterlegter Bereich angefügt, in dem der Benutzer passende Wörter zu seiner Eingabe erhält. Somit werden indirekt auf mögliche Tippfehler von Userinnen und Usern korrigiert, aber auch einiges an Tipparbeit kann eingespart werden.

R28: Aktionen von Userinnen und Usern sollten rückgängig-machbar sein.

R29: Bei Fehlern sollen die Nutzerinnen und Nutzer gewarnt werden und die App in den letzten stabilen Zustand zurückkehren. Wenn der stabile Zustand nicht einfach zu erreichen ist, soll die Kontrolle an die Benutzerinnen und Benutzer übergeben werden.

Aufgrund der Übersichtlichkeit und Unkompliziertheit der App gibt es nur sehr wenig Gelegenheiten, wo Userinnen und User einen Fehler machen könnten, der rückgängig gemacht werden sollte. Ein Beispiel dafür wäre eine versehentlich falsche Auswahl einer Antwortmöglichkeit.

In der derzeitigen Umsetzung wird eine Antwort sofort gewertet, wenn sie geklickt wird. Man könnte hier den Einwand machen, dass man die Antwort zuerst wählt und dann erst mit einem weiteren Klick auf einen Button zur nächsten Frage kommt. Dies würde aber in der Summe eine große Menge an zusätzlichen Klicks produzieren, was zur Umsetzung der Variante mit der direkten Wertung geführt hat. Bei sonstigen Problemen, die auftreten können, haben immer die Userinnen und User die Kontrolle darüber was geschehen soll.

R30: Verlange eine Bestätigung für Änderungs- und Editieraufgaben.

R31: Erinnere die Userinnen und User an ungespeicherte Veränderungen.

Die App ist so konzipiert, dass es keine Möglichkeiten gibt, wo Userinnen und User Einstellungen verändern können, die einer Speicherung bedürfen. Es gibt aber den Fall des Contests, bei dem die Punkte der beantworteten Fragen summiert werden und in der Resultate-Tabelle abgelegt werden. Klicken die Benutzerinnen und Benutzer während des Contest auf den „Home“-Button, so wird der in Abbildung 9.13 angezeigte Dialog dargestellt. Userinnen und User werden in diesem Dialog darüber informiert, dass der bisher erreichte Spielstand verloren geht, wenn sie auf die Navigationsseite zurückwechseln und somit den laufenden Contest abbrechen. Wie schon vorhin bei Richtlinie R29 erwähnt, entscheiden nun die Benutzerinnen und Benutzer, wie weiter vorgegangen werden soll.



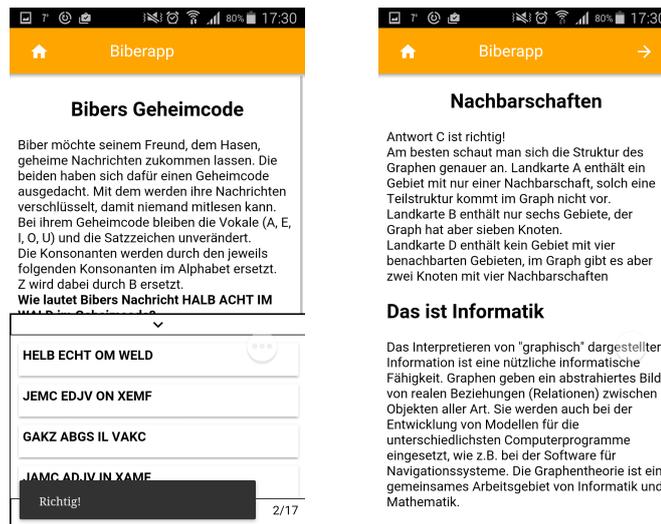
Abbildung 9.13: Dialog

R32: Implementiere Feedback für die Userinteraktion.

Einerseits wurde die Userinteraktion in der App auf Ebene der Bedienelemente umgesetzt, so wird ein Button, der geklickt wird, seine Farbe verändern, aber auch auf der Ebene des Inhalts.

Beim Beantworten der Frage, hier aber nur beim Fragen üben und Falsch beantworteten Fragen üben, bekommen Userinnen und User Information darüber, ob die

Frage richtig beantwortet wurde, oder ob die ausgewählte Antwort die falsche war. In der BiberApp sollte hier nicht mit Farben oder mit nervenden Geräuschen als Feedback gearbeitet werden. Daher wurde für eine richtig beantwortete Frage ein dezentes Alert mit dem Inhalt „Richtig!“ angezeigt (Abbildung 9.14a). War die Lösung falsch, wird eine weitere Seite angezeigt, die die Information über die richtige Lösung, eine Erklärung dazu und noch weitere Information zum Thema enthält (Abbildung 9.14b).



(a) Aufpoppendes Alert für richtige Antwort (b) Zusatzfrage für falsche Antwort

Abbildung 9.14: Feedback

R33: Erlaube es den Userinnen und Usern, die App an ihre Bedürfnisse anzupassen.

Da die App sehr kompakt und übersichtlich gehalten wurde, wurde auch keine Möglichkeit eingebaut, dass Userinnen und User die App an ihre Bedürfnisse anpassen können.

R34: Fasse wichtige Arbeitsschritte als Feature zusammen, um ungeübte Userinnen und User zu unterstützen.

Die einzelnen Funktionen der App sind so gehalten, dass sie nur mit wenigen Klicks gestartet werden können. Die App enthält weder komplizierte Prozesse, die durchlaufen

werden müssen, noch muss im Übermaß Text eingegeben werden. Es wird auch nicht verlangt, spezielle, umfangreiche Einstellungen zu treffen. Daher ist diese Richtlinie für die App nicht anwendbar.

R35: Gib den Userinnen und Usern die Möglichkeit, Elemente direkt zu manipulieren.

So wie schon die Richtlinie davor wurde diese Richtlinie für die App nicht beachtet. Die Fragen sind so umgesetzt worden, dass keine direkte Manipulation der Elemente nötig ist.

Inhalt

Dieser Abschnitt behandelt die Umsetzung der Richtlinien, die für die Inhaltsgestaltung von mobilen Applikationen ausschlaggebend sind.

R36: Verwende für die Desktop-Version und für die App ähnlichen Inhalt und ähnliche Funktionen.

Da der Biber der Informatik auf Multiple-Choice-Fragen basiert, wurde die App auch mit diesen Fragen umgesetzt. Es gibt in der Desktop-Version eine Möglichkeit für den Selbsttest, was in der BiberApp im Contest umgesetzt worden ist.

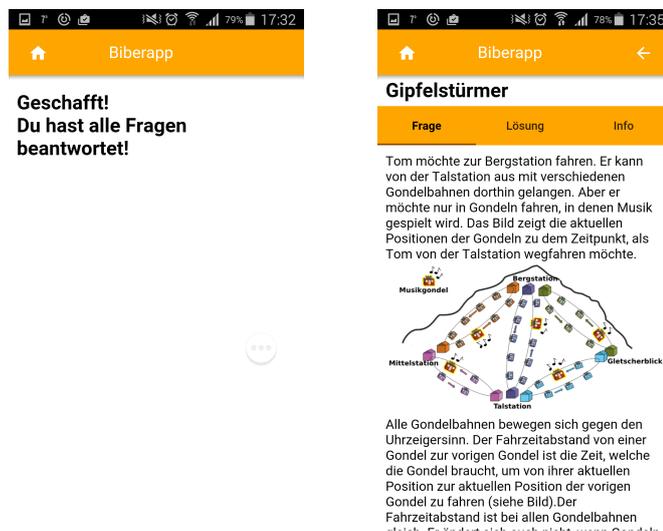
Weiters gibt es so wie in der Desktop-Version die Möglichkeit, die Fragen zu üben und ebenfalls eine Resultatsanzeige. Damit kann man sagen, dass diese Richtlinie R36 beachtet wurde.

R37: Inhalt sollte gut erkennbar und leicht zu überblicken sein.

Beim Umsetzen der Biber-Fragen in der mobilen Applikation, war es sehr wichtig, dass der Inhalt der Fragen bestmöglich zur Geltung kommt. Wie schon früher im Abschnitt Design beschrieben, gab es das Problem der sehr kleinen Anzeigemöglichkeit und den doch recht umfangreichen Fragen-Texten. Daher wurde darauf geachtet, dass sich in dem sehr kleinen Anzeigebereich der Inhalt sehr deutlich vom Hintergrund abhebt und die einzelnen Teile der Frage gut unterteilt und aufbereitet wurden.

R38: Vermeide lange Texte und verwende einfache Sätze.

In der Applikation wird Text überhaupt nur sehr spärlich verwendet. Text im allgemeinen Sinn kommt fast nicht vor - es gibt nur einzelne Wortgruppen. Die Ausnahmen sind der Dialog, der beim Beenden innerhalb des Contests auftaucht und die Endseite des Fragen üben und des Falsch beantwortete Fragen üben die Biber-Fragen selbst, deren Fragentext, Lösungserklärung und zusätzliche Information direkt aus den originalen Fragen übernommen wurden.



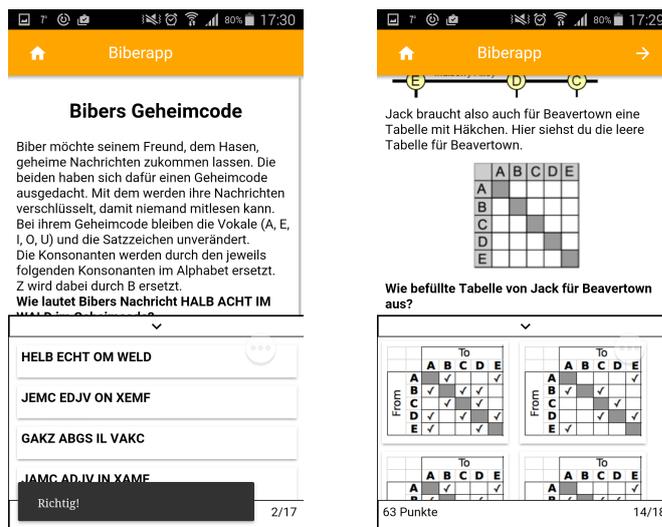
(a) Endseite Fragen üben

(b) Fragentext

Abbildung 9.15: Text in der BiberApp

R39: Hebe wichtige Information hervor.

Bei der Umsetzung der Biber-Fragen wird wichtige Information hervorgehoben, indem sie fett gedruckt wird. Daher wurden alle Antwortmöglichkeiten fett gedruckt (Abbildung 9.16a). Aber auch innerhalb der Biber-Fragen ist eine Hervorhebung notwendig. So wurde zum Beispiel die eigentliche Frage im Fragen-Text fett gedruckt, damit sie nicht überlesen werden kann (Abbildung 9.15b).



(a) Antwortmöglichkeiten fett (b) eigentliche Frage fett

Abbildung 9.16: Hervorhebung

R40: Auf der Startseite sollte nicht zu viel Information stehen.

Wie schon weiter oben erwähnt ist die erste Seite der BiberApp eine Loginseite und dann folgt die eigentliche Startseite der App - die Navigationsseite. Auf dieser Seite gibt es keine Sätze und keinen erzählenden Text. Sie beinhaltet nur die Links zu den in der App vorhandenen Funktionen. Daher kann man mit Sicherheit sagen, dass die Startseite nicht mit zu viel Information überladen ist.

R41: Überlege einen klaren, zielgerichteten Workflow.

Beim Gestalten der BiberApp wurde darauf geachtet, dass Userinnen und User mit wenigen Klicks alle Funktionen der App bedienen können und die App an den natürlichen Ablauf von Quizzes angepasst wird. Ebenfalls wurde sichergestellt, dass es keine Umwege oder für die Userinnen und User unverständlichen Abläufe gibt. Eine grafische Darstellung der Workflows/Funktionsweise der App wird in Kapitel Prototyp im Abschnitt Übersicht über die Funktionsweise des Prototypen dargestellt (siehe Abbildung 8.10).

R42: Strukturiere Inhalt nach seiner Wichtigkeit.

Wie schon weiter oben bei der Richtlinie R16 ausführlich beschrieben, wurden die Funktionen der App nach ihrer Wahrscheinlichkeit der Verwendung auf der Navigationsseite angeordnet. Das ist auch gleichermaßen eine Strukturierung des Inhaltes nach seiner Wichtigkeit. Bei den Userinnen und Usern wurde bei dieser Strukturierung von der Zielgruppe Schülerinnen und Schüler ausgegangen. Für Lehrerinnen und Lehrer könnte durchaus eine andere Strukturierung wie unten aufgelistet wichtig sein:

- | | |
|------------------------------------|--------------|
| 1. Suche | 4. Contest |
| 2. Fragen üben | 5. Resultate |
| 3. Falsch beantwortete Fragen üben | 6. Logout |

Text und Bild

Dieser Abschnitt behandelt das Thema Kombination von Text und Bild und wie im umgesetzten Prototypen damit umgegangen wurde.

R43: Zeige ein Bild und den dazugehörigen Text nahe beieinander an.

R44: Wenn eine gleichzeitige Anzeige von Text und Bild nicht möglich ist, zeige zuerst das Bild an.

Diese beiden genannten Richtlinien sind für die direkte Umsetzung der Biber-Fragen in der App sehr ausschlaggebend, denn fast jede der Fragen ist mit einem Bild kombiniert, das den erklärenden Text zum leichteren Verständnis noch veranschaulicht. Was auch, wie vorhin schon im Kapitel Aufbau der Aufgaben für den Biber Wettbewerb genauer beschrieben wurde, ein Kriterium für eine qualitativ hochwertige Frage ist.

Durch den sehr kleinen Anzeigebereich ist es schwierig, längere Texte und Bilder anzuzeigen. Beim ersten Versuch der Umsetzung ist man schnell an die Grenze gestoßen, wieviel gleichzeitig angezeigt werden kann. Vor allem waren die Bilder im Verhältnis zur Anzeigefläche extrem groß. Die Folge davon war: Zeigt man ein Bild an, kann fast kein Text mehr angezeigt werden. Die Bilder mussten als Folge daraus verkleinert werden, was aber wieder eine negative Auswirkung hat. Durch das Verkleinern sind viele der Bildinhalte nur noch sehr schwer erkennbar.

In den originalen Biber-Fragen wurden Bild und Text immer nebeneinander angezeigt. So konnten Userinnen und User das Bild beim Lesen des Textes immer im Auge behalten.

Für die App müsste aber aufgrund der ganz eigenen Bedingungen eine andere Lösung gefunden werden. Für die BiberApp konnte also die Richtlinie R43 nicht eingesetzt werden, aber Richtlinie R44 wurde für den Prototypen ausschlaggebend. Es wurde daher immer versucht, zuerst das zum Text gehörende Bild anzuzeigen und danach den erklärenden Text zu stellen. In einigen Fällen mussten Bilder und Grafiken extra neu überarbeitet werden und danach im Fragentext neu angeordnet werden, damit der Bezug zwischen Text und Bild immer bestehen bleiben konnte.

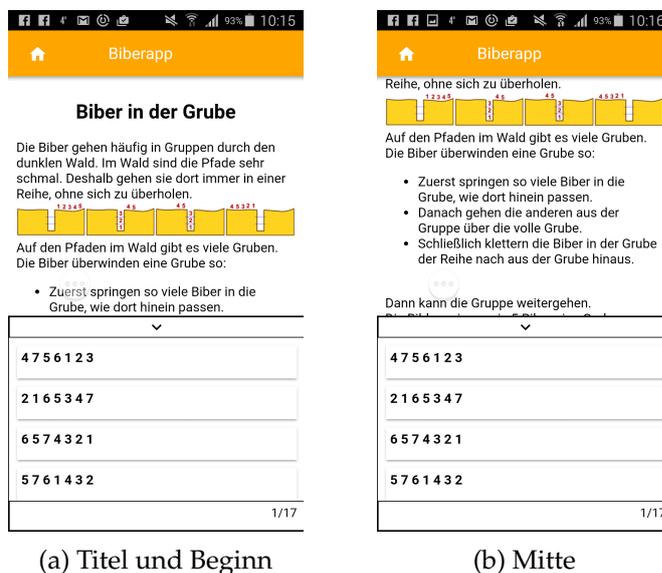
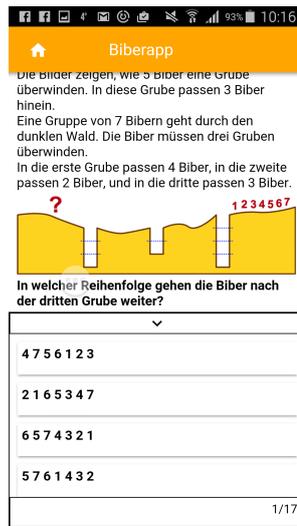


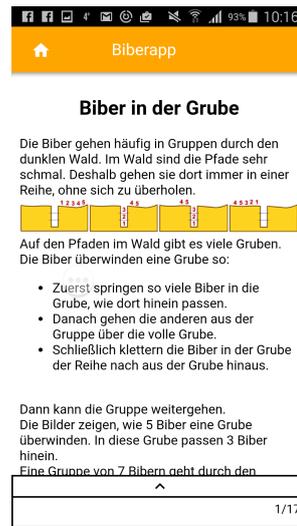
Abbildung 9.17: Bild und Text in BiberApp 1

Die Abbildungen 9.17 und 9.18 zeigen ein Beispiel dafür, wie die Anzeige von Text und Bild in der BiberApp umgesetzt wurde. Bei der Umsetzung dieser Frage steht das Bild nicht gleich zu Beginn, da der einleitende Text nicht direkt mit dem Bild in Bezug steht (Abbildung 9.17a).

Würde das Bild gleich ganz oben platziert werden, würde es beim Scrollen nach unten ziemlich schnell verschwinden. Bei der derzeit gewählten Position sieht man das Bild direkt vor dem Text, der sich darauf bezieht und kann als Leserin oder Leser beides gut in Verbindung bringen. Die weiteren Abbildungen zeigen den Rest der Frage beim Scrollverlauf. Das letzte Bild zum Text wird immer vor der eigentlichen Frage angezeigt (Abbildung 9.18a).



(a) Ende der Frage



(b) ganzer Anzeigebereich

Abbildung 9.18: Bild und Text in BiberApp 2

Body

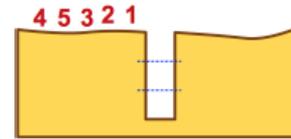
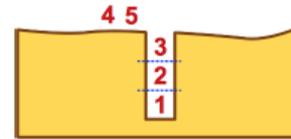
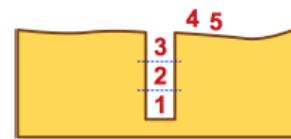
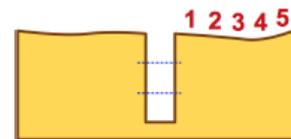
Die Biber gehen häufig in Gruppen durch den dunklen Wald. Im Wald sind die Pfade sehr schmal. Deshalb gehen sie dort immer in einer Reihe, ohne sich zu überholen.

Auf den Pfaden im Wald gibt es viele Gruben. Die Biber überwinden eine Grube so:

- Zuerst springen so viele Biber in die Grube, wie dort hinein passen.
- Danach gehen die anderen aus der Gruppe über die volle Grube.
- Schließlich klettern die Biber in der Grube der Reihe nach aus der Grube hinaus.

Dann kann die Gruppe weitergehen.

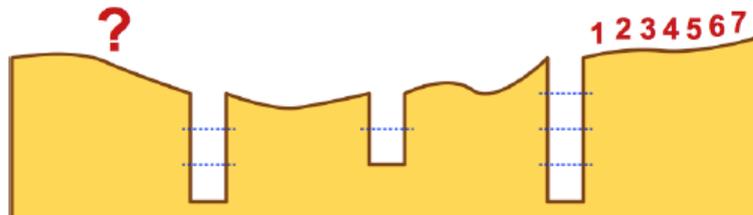
Die Bilder zeigen, wie 5 Biber eine Grube überwinden. In diese Grube passen 3 Biber hinein.



Question

Eine Gruppe von 7 Bibern geht durch den dunklen Wald. Die Biber müssen drei Gruben überwinden.

In die erste Grube passen 4 Biber, in die zweite passen 2 Biber, und in die dritte passen 3 Biber.



In welcher Reihenfolge gehen die Biber nach der dritten Grube weiter?

Abbildung 9.19: Originale Frage

Die Abbildung 9.19 zeigt den Aufbau der gleichen Frage im Original. Der sofort sichtbare Unterschied ist, dass im ersten Teil der Frage die Bilder rechts am Rand direkt neben dem beschreibenden Text platziert sind. Wie man in Abbildung 9.17 erkennen kann, wurden diese Bilder für die mobile Applikation in einer Grafik zusammengefasst, damit ein besserer Überblick ermöglicht werden konnte. Die eigentliche Frage steht im Original wie auch in der mobilen Version direkt nach der zur Frage gehörenden Abbildung.

Performance

Die Einhaltung der Richtlinien zum Bereich Performance sind Thema des nachfolgenden Abschnitts.

R45: Verringere Ladezeiten für Benutzerinnen und Benutzer. Die, die Applikation öfter verwenden, sind bei Ladezeiten toleranter.

Die BiberApp wurde, wie schon im Kapitel Verwendete Technologien beschrieben, mit einer clientseitigen Datenbank ausgestattet, die direkt auf dem Gerät der Userinnen und User aufgebaut wird. Daher kann die App ganz ohne Internetverbindung verwendet werden, nachdem sie einmal auf dem Gerät installiert wurde.

Die Funktionsfähigkeit ohne Internet bedeutet auch, dass keine Ladezeiten für einzelne Seiten und Bilder entstehen, weil sie direkt aus dem Speicher des Gerätes abgerufen werden können. Aus diesem Grund läuft die BiberApp sehr flüssig und ohne lange Ladezeiten.

9.2 Evaluation Suchfunktion der BiberApp

Um die zweite Forschungsfrage der Arbeit -

Wie gut einsetzbar ist eine Volltextsuche in den Aufgaben des Biber der Informatik um bestimmte Themen für den Unterricht zu suchen?

- zu beantworten bietet sich eine Evaluation der in der App implementierten Suchfunktion an.

Sie wurde, wie im Kapitel Prototyp schon angesprochen, als Volltextsuche umgesetzt. Die Suche in den Aufgaben des Biber der Informatik funktioniert aufgrund der besonderen Beschaffenheit der Aufgaben. Wie schon im Abschnitt Aufbau der Aufgaben für den Biber der Informatik detailliert ausgeführt, ist die Basis für die Fragen das

Computational Thinking. Dagiene und Futschek [10] erläutern in ihrem Artikel, dass in den Aufgaben keine Fachbegriffe der Informatik vorkommen. Daher ist eine Volltextsuche eine angemessene Lösung, da es keine Einschränkung dafür gibt, wonach gesucht werden kann.

Laut Pohl und Hein [38] sind die Aufgaben, wie schon erwähnt, in den in-contest-Bereich und den after-contest Bereich geteilt. Für die Suchfunktion und deren Nützlichkeit für den Unterricht ist im Besonderen der after-contest-Bereich ausschlaggebend. Die größte Bedeutung hat in diesem Bereich der Abschnitt Das ist Informatik, oder im Englischen it's informatics.

Im Template für die Gestaltung von Aufgaben der Bebras Challenge wird für die Befüllung dieses Bereiches folgendes vorgeschlagen:

„Explain the relevance in informatics of this task to the target age group. Use about 3 to 8 sentences. Do not explain the correct answers of a task, but give a larger picture.“ [37]

Es wird also empfohlen in einem kurzen Text auf die Relevanz der Frage für die Informatik, aber nicht auf die richtige Antwortmöglichkeit einzugehen. Über diese Beschreibung kann eine Verbindung mit den Konzepten und Fachbegriffen der Informatik hergestellt werden.

Ein weiterer Abschnitt im Aufbau eines Bebras tasks hilft bei der Verknüpfung der Aufgabe mit den Konzepten der Informatik, der Bereich Keywords. Im bereitgestellten Template heißt es dazu:

„Give some keywords, useful to find further information on the web. The target are the teachers and interested students.“ [37]

Hier kann noch einmal in Form von Schlagworten ein Bezug zu Fachbegriffen und Konzepten der Informatik hergestellt werden.

Laut der Definition von guten Aufgaben für die Bebras Challenge nach Pohl und Hein [38] ist jeder Bebras task so aufgebaut. Daraus kann man folgern, dass jede Aufgabe auch einen Bezug zu einem Konzept der Informatik hat.

Die Einsetzbarkeit der Suchfunktion für den Unterricht wird unterstützt durch folgende Fakten: Laut Futschek [15] sind für den Informatikunterricht die fundamentalen Ideen der Informatik eine sehr wichtige Basis. Die Abbildung 9.20 zeigt die fundamentalen Ideen/ Konzepte der Informatik, wie sie in der Didaktik der Informatik von Schubert und Schwill [42] enthalten sind.

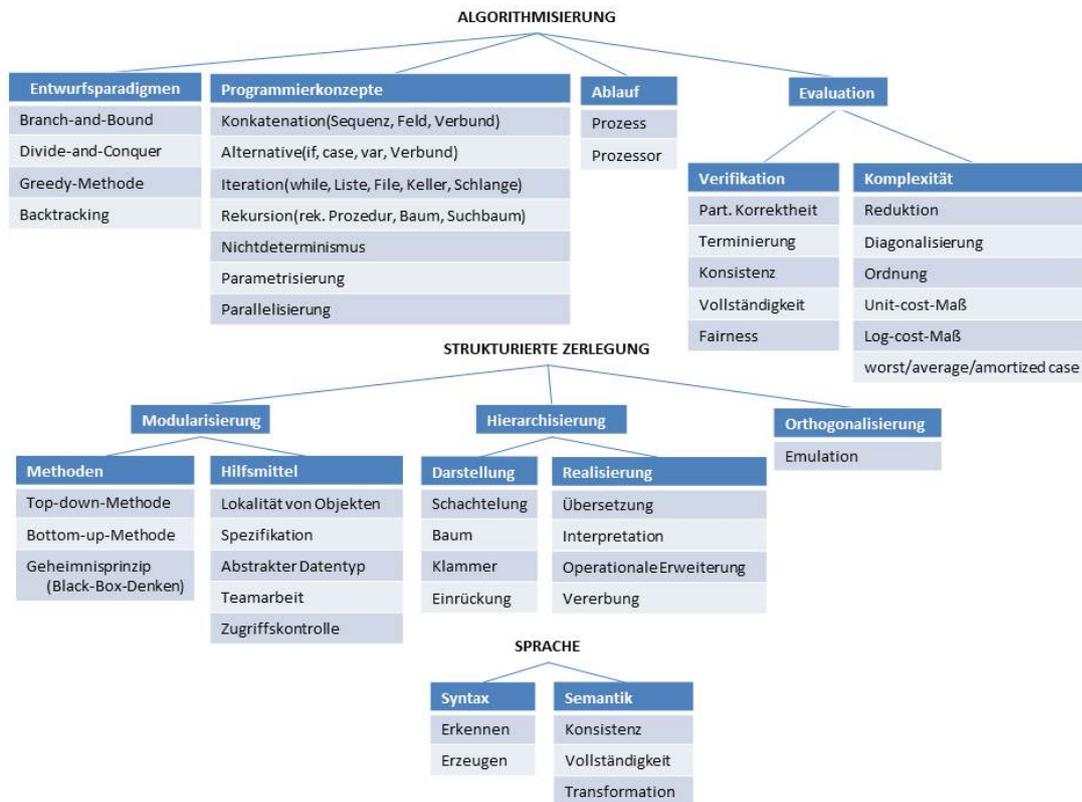


Abbildung 9.20: Fundamentale Ideen der Informatik nach Schwill [42]

Diese fundamentalen Ideen fließen nun auch in neue Lehrpläne mit ein. In den noch unveröffentlichten Lehrplänen für Informatik für die Oberstufe, die im Appendix zu finden sind, wird im Abschnitt praktische Informatik das Lehren von Konzepten der Informatik angeführt.

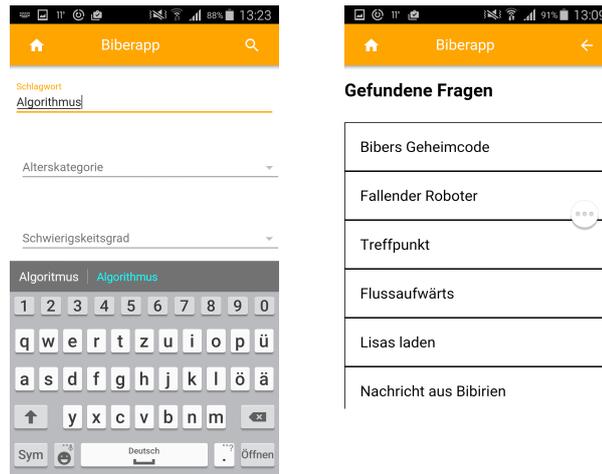
Daraus folgend kann man schließen, dass die Aufgaben des Biber der Informatik in Österreich im Unterricht einsetzbar sind. Pohl und Hein [38] schlagen vor, dass die Aufgaben im Unterricht als Einführung in ein spezielles Themengebiet verwendet werden können.

Beispiele der Suchfunktion

Basierend auf Katalogen fundamentaler Konzepte der Informatik, wie der oben genannte von Schwill [42] haben Zendler und Spannagel [54] einen empirisch belegten Katalog von 49 Konzepten der Informatik erstellt. Aus diesem Katalog wurden sechs Konzepte gewählt, die für nachstehenden Beispiele als Schlagwort in die Suchfunktion eingegeben wurden, um zu sehen, ob dazu Aufgaben des Biber der Informatik gefunden werden.

Schlagwort: Algorithmus

Zum Schlagwort Algorithmus werden mehrere Fragen gefunden. Wie man nach Kenntnis des Aufbaus einer Aufgabe des Biber der Informatik vermuten kann, findet sich dieser Fachbegriff im Abschnitt Info des Prototypen, der durch den Text aus dem Abschnitt der Biber Aufgabe Das ist Informatik umgesetzt wurde.



(a) Schlagwort: Algorithmus (b) Ergebnismenge

Abbildung 9.21: Suchbeispiel 1: Schlagwort und Ergebnismenge

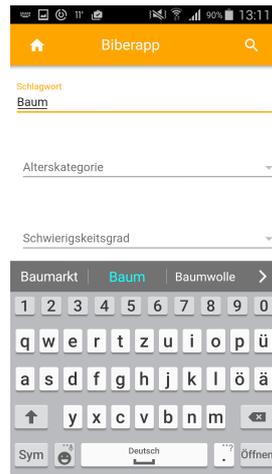


(a) Fallender Roboter: Das ist Informatik

Abbildung 9.22: Suchbeispiel 1: Vorkommnisse des Schlagwortes

Schlagwort: Baum

Zu diesem Begriff werden nur wenige Fragen gefunden. Was hier auffällt, ist dass auch Fragen angezeigt werden, die mit dem informatischen Fachbegriff Baum nicht verbunden sind. Der Begriff kommt nicht im Bereich Das ist Informatik der Frage vor. Im speziellen Fall der Aufgabe Wasserversorgung kommt das Wort Baum aber im Fragentext vor.



(a) Schlagwort: Baum

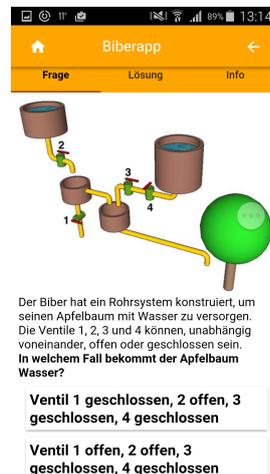


(b) Ergebnismenge

Abbildung 9.23: Suchbeispiel 2: Schlagwort und Ergebnismenge



(a) Wasserversorgung: Das ist Informatik

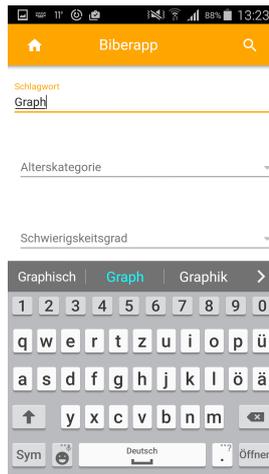


(b) Wasserversorgung: Fragentext

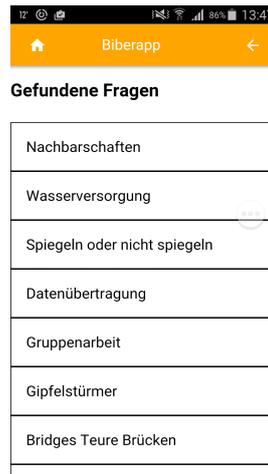
Abbildung 9.24: Suchbeispiel 2: Vorkommnisse des Schlagwortes

Schlagwort: Graph

Für den Begriff Graph werden unten angezeigte Fragen gefunden. Im Fall der Gruppenarbeit kommt der Fachbegriff zwar nicht im Text Das ist Informatik vor, aber in der Erklärung zur Lösung wird er verwendet.



(a) Schlagwort: Graph

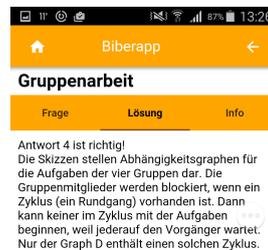


(b) Ergebnismenge

Abbildung 9.25: Suchbeispiel 3: Schlagwort und Ergebnismenge



(a) Nachbarschaften: Das ist Informatik



(b) Gruppenarbeit: Erklärung zur Lösung

Abbildung 9.26: Suchbeispiel 3: Vorkommnisse des Schlagwortes

Schlagwort: Information

Auch beim Begriff der Information werden viele Fragen angezeigt. Hier ist zu beachten, dass nicht nur Fragen gefunden werden, die das Wort Information enthalten, sondern auch Wörter wie Informationsdarstellung.

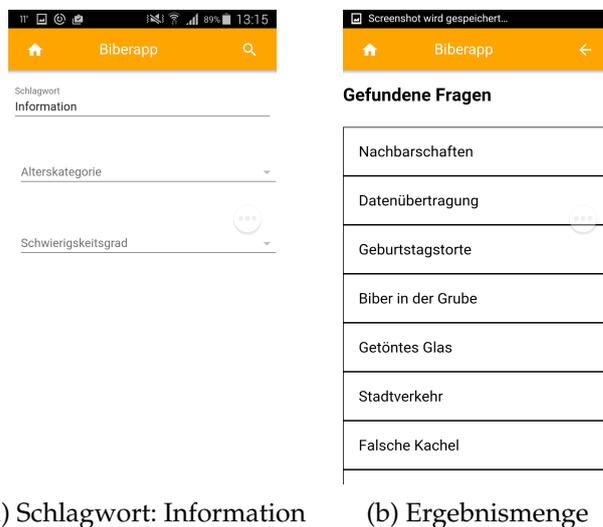


Abbildung 9.27: Suchbeispiel 4: Schlagwort und Ergebnismenge

Schlagwort: Modell

Das Schlagwort Modell liefert konkret sieben Aufgaben aus denen gewählt werden kann.

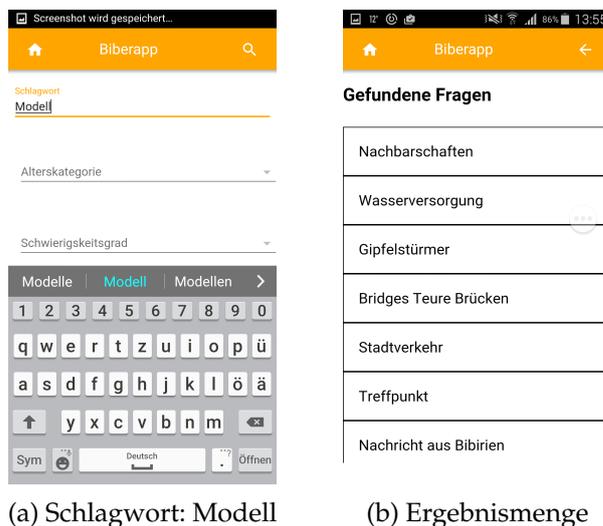
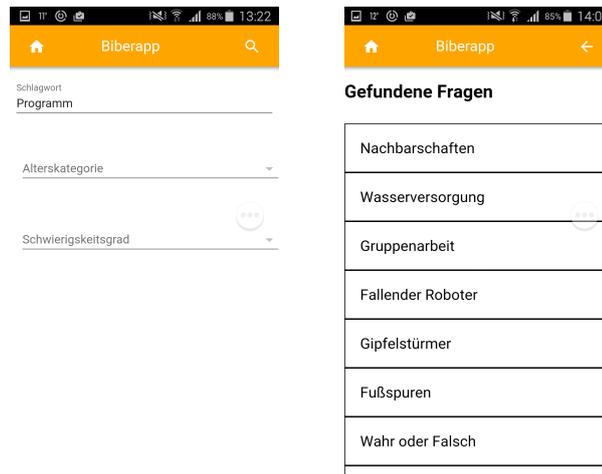


Abbildung 9.28: Suchbeispiel 5: Schlagwort und Ergebnismenge

Schlagwort: Programm

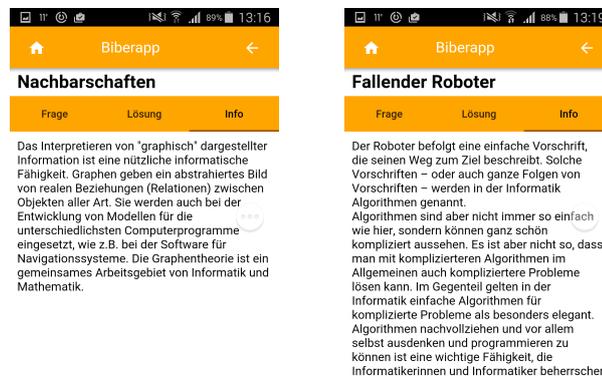
Für dieses Schlagwort werden mit Abstand die meisten Aufgaben gefunden. Das liegt auch daran, dass nicht nur Bebras tasks gefunden werden, die das Wort Programm enthalten und mit dem Konzept der Informatik Programm verbunden sind, sondern auch solche, die Wörter wie programmieren und Programmiersprachen enthalten.



(a) Schlagwort: Programm

(b) Ergebnismenge

Abbildung 9.29: Suchbeispiel 6: Schlagwort und Ergebnismenge



(a) Nachbarschaften: Das ist Informatik (b) Fallender Roboter: Das ist Informatik

Abbildung 9.30: Suchbeispiel 6: Vorkommnisse des Schlagwortes

Kapitel 10

Ergebnisse und Zusammenfassung

10.1 Ergebnisse Umsetzung der Biberfragen

Bei der Evaluation der Umsetzung der Biber-Fragen in der BiberApp hat sich herausgestellt, dass die meisten der in der Literaturrecherche gefundenen Richtlinien zum Gestalten einer mobilen Applikation auf den Prototypen anwendbar sind.

Eine große Erleichterung im Bereich der Navigationsgestaltung und der Aufteilung des Inhaltes, aber auch eine Einschränkung, war der von vorne herein schon sehr genau begrenzte Inhalt der App. Durch die von Beginn an schon sehr klare Struktur konnte eine klare, einfache Navigation aufgebaut werden. Würde die mobile Applikation noch weitaus mehr Funktionen beinhalten, wäre eine Strukturierung für die Navigation und Inhaltsaufteilung wahrscheinlich wesentlich schwieriger gewesen.

Eine beträchtliche Einschränkung bei der Umsetzung der Fragen in der App war auf jeden Fall der sehr kleine Anzeigebereich der mobilen Geräte. Die Fragen konnten nicht wie in der Desktop-Applikation strukturiert angezeigt werden. Bei manchen der Fragen nahm schon ein Bild den gesamten Anzeigebereich ein, oder durch einen langen Text musste sehr viel gescrollt werden, um das Gelesene mit dem Bild zur Frage zu vergleichen, weil das/die Bild/er nicht wie im Original einfach daneben platziert werden konnte/n. Die Fragen wurden laut den Richtlinien für Text und Bild an den eingeschränkten Anzeigebereich angepasst, was in manchen Fällen auch eine Umarbeitung von Grafiken und Bildern bedeutet hat. Nach der Bearbeitung der Grafiken und Bilder, was in den meisten Fällen eine Verkleinerung mit sich brachte, entstand aber das Problem, dass der Bildinhalt nur noch schwer zu erkennen war.

Was beim Verwenden der BiberApp auffällt ist, dass man durch zusätzliches Scrollen, störende Umgebung und der gleichen nicht so konzentriert lesen kann und man für das Erfassen des Inhalts, sowie zur Beantwortung einer einzelnen Frage wahrscheinlich mehr Zeit braucht, als beim Arbeiten mit der Desktop-Applikation.

In einigen wenigen Fällen wurde bewusst von vorgegebenen Richtlinien abgewichen. Ein Beispiel dafür war die Startseite mit dem Login. Es wurde bewusst entschie-

den das Login direkt an den Beginn der App zu setzen, weil die Login-Information später für die Resultate des Contests wichtig wurden. Für eine Abweichung musste es aber immer einen triftigen Grund geben, der in der Evaluation selbst ausführlich beschrieben wurde. Die letzte Entscheidung einer Abweichung wurde aber immer mit Bedacht auf die Usability für Userinnen und User getroffen.

Andererseits wurden in der App Richtlinien außer Acht gelassen, die den Userinnen und Usern die Möglichkeit geben, die App an eigene Vorlieben und Bedürfnisse anzupassen, da die App keine reine Spiele-Applikation ist, bei der diese Features sicher sehr wichtig sind. Die BiberApp ist hauptsächlich ein Tool, um sich auf den Biber der Informatik vorzubereiten. Der zweite Aspekt gegen dieses Customizing ist, dass auch Lehrerinnen und Lehrer die App für den Unterricht einsetzen können und diese Features eher störend, eventuell Zeit fressend sein könnten. Was man mit Sicherheit sagen kann ist, dass dieser entstandene Prototyp auf Geräten, die ein kleineres Display als Touchphones haben nicht sinnvoll einsetzbar ist.

10.2 Ergebnisse Suchfunktion

Unterstützt durch die Literatur zur Gestaltung von Aufgaben für die Bebras Challenge und die Artikel zur Qualität einer guten Aufgabe von Dagiene und Futschek [10] sowie Pohl und Hein [38] kann man folgern, dass die Volltextsuche gut funktioniert um nach Aufgaben zu suchen.

Um die Einsetzbarkeit der Suchfunktion für den Unterricht zu evaluieren, konnten Aspekte aus der Literatur speziell von den Autoren Futschek [15] und Pohl und Hein [38] als auch der Autorin Wing [52] gefunden werden, die die Nützlichkeit der Suchfunktion im Unterricht untermauern.

Nachdem gezeigt wurde, dass die Suchfunktion gut funktioniert und Aufgaben zu Konzepten der Informatik gefunden werden können, muss aber auch noch ein Nachteil genannt werden. Bei der Suche nach Konzepten, wie in der Evaluation am Beispiel Schlagwort Baum gezeigt, werden auch Aufgaben gefunden, die nicht direkt mit dem informatischen Fachbegriff Baum zu tun haben. Die Ursache darin liegt in der Implementation der Suchfunktion als Volltextsuche. Die Entscheidung welche Frage nun sinnvoll im Unterricht verwendet werden kann und wirklich passend ist, muss letztendlich also von den Benutzerinnen und Benutzern getroffen werden.

Anhang A

Unveröffentlichter Lehrplan Informatik AHS 5. Schulstufe

Informatik 5. Klasse

Präambel

Bildungsziele und Bildungsinhalte sind immer ein Spiegelbild des gesellschaftlichen, politischen und ökonomischen Umfeldes. Gegenwärtig bildet die Informatik den Wesenskern des digitalen Zeitalters und damit auch das Fundament moderner Informations- und Kommunikationstechnologien.

Ihre Inhalte sind daher allgemeinbildend und dienen sowohl einem fundierten Weltverständnis als auch der fachlichen Basis für zukünftige Berufsbilder. Der Informatik kommt als Wissenschaft und als schulisches Fachgebiet eine Schlüsselrolle zu, da sie die automatische Datenverarbeitung und digitale Informationsrepräsentation zum Gegenstand hat und diese mit Hilfe von Informatiksystemen nutzbar macht.

Das Fach Informatik eröffnet allen Schülerinnen und Schülern einen gleichberechtigten Zugang zu informatischen Denk- und Arbeitsweisen als Voraussetzung für den produktiven Umgang mit digitalen Informations- und Kommunikationstechnologien.

Beiträge zu den Bildungsbereichen:

Sprache und Kommunikation:

Konstruktiver Informatikunterricht ist auch Sprachunterricht. Der Mensch-Maschine-Kommunikation liegt im Gegensatz zu natürlichen Sprachen eine abstrakte formale Sprache zugrunde.

Informatiksysteme tragen wesentlich zu Veränderungen der Kommunikationskultur bei. Unterschiedliche digitale Repräsentationsformen von Information ergänzen die traditionelle Verständigung und erfordern neue technologische und methodische Kompetenzen.

Die vielfältigen Möglichkeiten der elektronischen Kommunikation ermöglichen einen Austausch über Grenzen hinweg und erleichtern die virtuelle Begegnung mit anderen Kulturen. Die davon ausgehende Motivation, Fremdsprachenkenntnisse zu erwerben, wird durch die Verfügbarkeit aktueller und authentischer fremdsprachlicher Informationen und das Fachvokabular verstärkt

Mensch und Gesellschaft:

Arbeitswelt und privates Umfeld der Menschen verändern sich durch den Einfluss der Informationstechnologien permanent. Durch die Beschäftigung mit diesen Technologien lernen Schülerinnen und Schüler deren Auswirkungen, Möglichkeiten, Grenzen und Gefahren kennen.

Die Schülerinnen und Schüler erkennen das Potenzial ihrer eigenen Fähigkeiten als denkende, handelnde, fühlende und sich entwickelnde Menschen im Unterschied zu einer lernenden Maschine. Dies erfordert einen verantwortungsvollen Umgang mit Informationstechnologien.

Natur und Technik:

Durch Modellbildung, Formalisierung und Abstraktion leistet die Informatik einen wesentlichen Beitrag zur Auseinandersetzung mit Natur und Technik und führt zu einer besseren Entscheidungs- und Handlungskompetenz.

Kreativität und Gestaltung:

Der Umgang mit Informationstechnologie gibt den Schülerinnen und Schülern Gelegenheit, selbst kreativ tätig zu sein und Gestaltungserfahrungen zu machen.

Gesundheit und Bewegung:

Die Verantwortung für den eigenen Körper erfordert als Ausgleich zur Arbeit am Computer gezielte Bewegung. Den Schülerinnen und Schülern soll die Bedeutung eines ergonomisch gestalteten Arbeitsplatzes bewusst werden.

Der Einsatz von Informationstechnologien zur Erfassung und Analyse von Daten im Sport- und Gesundheitsbereich bietet die Möglichkeit zur kritischen Reflexion.

Didaktische Grundsätze:

Der Lehrplan ermöglicht die eigenständige und verantwortliche Unterrichts- und Erziehungsarbeit der Lehrerinnen und Lehrer des vorgegebenen Umfangs. Der Unterricht ist daher kontinuierlich zu planen, durchzuführen sowie laufend zu reflektieren und anzupassen. Es ist notwendig, Inhalte so auszuwählen und zu organisieren, dass sie die Vorkenntnisse und Vorerfahrungen der Schülerinnen und Schüler berücksichtigen und daran anknüpfen. Die Themen sind dabei so auszuwählen, dass sie vielfältige Bezüge zur Lebens- und Begriffswelt der Jugendlichen herstellen. In der Übergangsphase von der 8. zur 9. Schulstufe sind im Informatikunterricht besondere fachdidaktische Überlegungen anzustellen um Defizite auszugleichen und individuelle Stärken einzubinden und zu fördern.

Die Unterrichtsplanung hat sich an für Schülerinnen und Schüler transparenten Lehrzielen zu orientieren. Die Unterrichtsdurchführung soll beispielgebend für die eigene Lern- und Arbeitsorganisation auch außerhalb des Informatikunterrichts sein. Variierende Arbeitsformen wie Einzelarbeit, Gruppenarbeit und Teamarbeit geben Schülerinnen

und Schülern Gelegenheit, Neues zu erforschen und bereits Gelerntes in verschiedenen kommunikativen und inhaltlichen Kontexten anzuwenden. Selbsttätigkeit und Eigenverantwortung sind zu fördern und Möglichkeiten zur persönlichen Lernzielkontrolle anzubieten.

Gemeinschaftliches Problemlösen in einem projektorientierten Unterricht soll gefördert werden. Dabei ist kooperativen Entscheidungsstrukturen entsprechender Platz einzuräumen. Methodische Überlegungen sollen sich an den spezifischen Anforderungen von Einstieg, Entwicklung und Abschluss von Unterrichtsphasen orientieren. Explorative, systematische und exemplarische Vorgehensweisen sollen zur Vertiefung von Wissen und Erweiterung von Kompetenzen in der Informatik führen.

Schülerinnen und Schülern ist Gelegenheit zu geben, durch Transfer und Analogiebildung den Lernertrag zu sichern. Der Informatikunterricht soll beispielhaft für den sinnvollen Einsatz verfügbarer Technologien sein. Dem Erwerb einer wissenschaftlichen Arbeits- und Dokumentationsweise ist die Erstellung eines Portfolios dienlich.

Der Erwerb informatischer Kompetenzen erfordert passende Formen der Wissensdarstellung und -verarbeitung. Grundlegende Strukturen und Prozesse in Gesellschaft, Natur und Technik werden aus Sicht der Informatik veranschaulicht. Dazu sind Methoden der Visualisierung und der Abstraktion zu verwenden. Die zyklische Vorgangsweise des Sammelns, Auswählens, Strukturierens, Abstrahierens, Auswertens und Interpretierens von Daten ist beim Problemlösen zu berücksichtigen.

Zur Motivation und zur Sicherung des Unterrichtsertrags sind den Schülerinnen und Schülern im Rahmen des Informatikunterrichts vielfältige Möglichkeiten anzubieten, ihr Wissen zu präsentieren, sich der Kritik anderer zu stellen und ihre Arbeit zu argumentieren.

Die Gestaltung eines angenehmen und erfolgreichen Lernklimas beruht auf Vertrauen, auf der Förderung der individuellen Stärken und des kreativen Potenzials. Auf die unterschiedlichen Interessen sowohl der Schülerinnen als auch der Schüler ist durch Auswahl entsprechender Inhalte und Aufgabenstellungen einzugehen.

Exkursionen und Einladungen von Expertinnen und Experten zu Vorträgen und zur Diskussion sollen den Erfahrungshorizont erweitern.

Bildungs- und Lehraufgabe

Informatische Bildung ist das Ergebnis von Lernprozessen, in denen fachliche Grundlagen verdeutlicht und Anwendungskompetenzen durch planvolle Arbeitsweisen systematisch erworben werden. Sie befähigt Schülerinnen und Schüler, die gesellschaftliche und wirtschaftliche Dimension digitaler Informations- und Kommunikationstechnologien zu erfassen. Aufgabe des Informatikunterrichts ist es, die Schülerinnen und Schüler zum Erwerb informatischer und informationstechnischer Kompetenzen hinzuführen, um sie zu befähigen, diese zur Lösung verschiedener Problemstellungen einzusetzen.

Durch die Analyse realer Probleme vor allem aus ihrer Erfahrungswelt sollen sie Strukturen und Zusammenhänge erkennen und die Notwendigkeit von Abstraktion und Reduktion bei der Modellbildung von einfachen realen Systemen erfahren und diese Modelle auf empirische Daten anwenden lernen. Sie sollen kooperative und kommunikative Arbeitsweisen unter Einsatz von Kommunikationstechnologien anwenden lernen. In allen Bildungsbereichen stehen dabei Erweiterung und Festigung von Sach-, Selbst- und Sozialkompetenz im Mittelpunkt.

Die Schülerinnen und Schüler sollen erkennen, dass die Informatik einer wissenschaftlichen Systematik unterliegt und Interesse und Wertschätzung verdient. Der Informatikunterricht fasst die vorhandenen Fähigkeiten von Schülerinnen und Schülern in der Informatik durch Beschäftigung mit Entwurf, Gestaltung und Anwendung von Informationssystemen zusammen und baut sie aus. Bei der kritischen Auseinandersetzung mit den dabei ablaufenden Prozessen und deren Ergebnissen sollen die Schülerinnen und Schüler ihr kognitives, emotionales und kreatives Potenzial nützen. Dies soll die Jugendlichen bei der Entwicklung und Festigung einer persönlichen Werthaltung und Weltsicht unterstützen und einen tieferen Einblick in die gesellschaftlichen Zusammenhänge und Auswirkungen moderner Informationstechnologie ermöglichen.

Kompetenzen

5. Klasse

Informatik, Mensch und Gesellschaft

- Die Bedeutung von Informatik in der Gesellschaft beschreiben, die Auswirkungen auf die Einzelnen und die Gesellschaft einschätzen und Vor- und Nachteile an konkreten Beispielen abwägen können.
- Maßnahmen und rechtliche Grundlagen im Zusammenhang mit Datensicherheit, Datenschutz und Urheberrecht kennen und anwenden können.
- Die Entwicklung der Informatik beschreiben und bewerten können.
- Informatikberufe und Einsatzmöglichkeiten der Informatik in verschiedenen Berufsfeldern benennen und einschätzen können.

Informatiksysteme

- Den Aufbau von digitalen Endgeräten beschreiben und erklären können.

- Die Funktionsweise von Informatiksystemen erklären können.
- Grundlagen von Betriebssystemen erklären, eine graphische Oberfläche und Dienstprogramme bedienen können.
- Grundlagen der Vernetzung von Computern beschreiben und lokale und globale Computernetzwerke nutzen können.

Angewandte Informatik

- Standardsoftware zur Kommunikation und Dokumentation sowie zur Erstellung, Publikation und multimedialen Präsentation eigener Arbeiten einsetzen können.
- Standardsoftware für Kalkulationen und zum Visualisieren anwenden können.
- Informationsquellen erschließen, Inhalte systematisieren, strukturieren, bewerten, verarbeiten und unterschiedliche Informationsdarstellungen verwenden können.
- Digitale Systeme zum Informationsaustausch, zur Unterstützung der Unterrichtsorganisation und zum Lernen auch in kommunikativen und kooperativen Formen verwenden können

Praktische Informatik

- Begriffe und Konzepte der Informatik verstehen und Methoden und Arbeitsweisen anwenden können
- Algorithmen erklären, entwerfen, darstellen und in einer Programmiersprache implementieren können.
- Grundprinzipien von Automaten, Algorithmen, Datenstrukturen und Programmen erklären können.
- Datenbanken benutzen und einfache Datenmodelle entwerfen können.

Anhang B

Unveröffentlichter Lehrplan Informatik Wahlpflichtfach AHS 6.-8. Schulstufe

INFORMATIK

Bildungs- und Lehraufgabe (6. bis 8. Klasse):

Das Wahlpflichtfach Informatik schließt an das Pflichtfach in der 5. Klasse an und soll die Schülerinnen und Schüler zu einer erweiterten und vertieften informatischen Bildung führen. Diese umfasst sowohl sachliche und fachsystematische als auch methodische und soziale Komponenten.

Unter Berücksichtigung der fachwissenschaftlichen Systematik soll der Unterricht zu einem zielorientierten, konstruktiven Problemlösen mit informatischen Werkzeugen und Methoden führen. Durch eine präzise, strukturierte und vollständige Beschreibung sowohl von Problemstellungen als auch von Abläufen, sowie durch die Modularisierung und Dekomposition komplexer Aufgaben soll die Informatik zur Schulung abstrakten Denkens beitragen. Schließlich soll das Wahlpflichtfach wesentlich zu einem vertieften technischen und soziokulturellen Verständnis unserer digital geprägten Welt führen und interessierte Schülerinnen und Schüler auf ein einschlägiges Studium vorbereiten.

Didaktische Grundsätze:

Die didaktischen Grundsätze für das Pflichtfach Informatik dienen auch im Wahlpflichtfach als Leitlinien. Die Hinführung zu modul- und projektartigen, vernetzten Arbeitsformen ist Grundlage für das Fach selbst und auch für fächerübergreifende Kooperationen. Ein Schwerpunkt des Informatikunterrichts soll in der formalen Modellierung von Sachverhalten und in der Problemlösung liegen, welche aus Analyse, Beschreibung in verschiedenen Darstellungsformen, algorithmischer Lösung, Implementation, Überprüfung und Interpretation sowohl vom informatischen als auch sachlichen Standpunkt bestehen. Die unterschiedliche Kenntnislage der Schülerinnen und Schüler erfordert in einzelnen Fachbereichen eine differenzierte und individualisierte Unterrichtsgestaltung. Die Auswahl der Themen soll zeitgemäß und gendersensibel sein und möglichst die Interessen der Schülerinnen und Schüler berücksichtigen.

Bildungs- und Lehraufgabe, Lehrstoff:

Kompetenzen

Das Kompetenzmodell für Informatik ist in der Inhaltsdimension in vier Bereiche, in der Handlungsdimension in drei Bereiche gegliedert. Die Kompetenzen wurden in den Schnittpunkten von Inhalts- und Handlungsbereich formuliert und auf die einzelnen Semester aufgeteilt. Der Unterricht im Wahlpflichtfach Informatik soll im Sinne eines konsistenten Kompetenzaufbaus in allen Bereichen zu einer Vertiefung und Erweiterung der Kompetenzen führen. Der Abstraktionsgrad und der Erwerb konzeptuellen Verständnisses sollen im Laufe der aufsteigenden Semester gesteigert werden.

Gliederung der Inhaltsdimension

- Informatik, Mensch und Gesellschaft
- Informatiksysteme
- Angewandte Informatik
- Praktische Informatik

Gliederung der Handlungsdimension

- Wissen und Verstehen
- Anwenden und Gestalten
- Reflektieren und Bewerten

Die Anforderungen aus der Handlungsdimension können in Hinblick auf die mündliche Reifeprüfung den Aspekten Reproduktion, Transfer, Reflexion und Problemlösung zugeordnet werden.

Es obliegt den Lehrerinnen und Lehrern in Abstimmung mit den Schülerinnen und Schülern in bestimmten Bereichen Schwerpunkte zu setzen oder Bereiche exemplarisch zu behandeln.

6- Klasse

3. Semester – Kompetenzmodul 3

Informatiksysteme

Technische Grundlagen und Funktionsweisen

- Komponenten von Informatiksystemen beschreiben und ihre Funktionsweise und ihr Zusammenwirken erklären können

- Ein Computersystem samt Peripheriegeräten sachgerecht aufbauen und nutzen können

Betriebssysteme und Software

- Die Kernaufgaben und Arbeitsweisen von Betriebssystemen beschreiben und erklären können
- Kategorien von Software nennen und deren Anwendung beschreiben können
- Ein Betriebssystem installieren, Systemkonfigurationen vornehmen und seine wichtigsten Funktionen nutzen können

Angewandte Informatik

Produktion digitaler Medien

- Gängige Medienformate und ihre Eigenschaften beschreiben können.
- Grundlegende Richtlinien, die bei der Produktion digitaler Medien von Bedeutung sind, erläutern können
- Digitale Medien in Form von Text, Ton, Bildern und Filmen sachgerecht bearbeiten, produzieren und auch im Web publizieren können. Digitale Produkte (Artefakte) in Bezug auf inhaltliche Relevanz, Wirkung und Design einschätzen und bewerten können

Kommunikation und Kooperation

- Wichtige Webanwendungen für den Informationsaustausch und die Zusammenarbeit benennen und ihre Grundlagen erklären können
- Netzwerke mit geeigneten Webanwendungen zum Informationsaustausch, zur Diskussion und zur Zusammenarbeit sinnvoll und verantwortungsbewusst nutzen können

Praktische Informatik

Algorithmen, Datenstrukturen und Programmierung

- Den Algorithmusbegriff erklären können
- Grundlegende Aufgaben und Problemstellungen algorithmisch und formalsprachlich in geeigneten Datenstrukturen beschreiben können
- Grundlegende Algorithmen entwerfen, diese formal darstellen, implementieren und testen können

4. Semester – Kompetenzmodul 4

Informatiksysteme

Netzwerke

- Netzwerke und ihre Protokolle beschreiben und ihre Funktions- und Wirkungsweise erklären können
- Ein einfaches Computernetzwerk konzipieren, aufbauen, verwalten und nutzen können
- Verschiedene Internetdienste nennen und ihre Einsatzmöglichkeiten und Funktionsweisen beschreiben und erklären können. Verschiedene Internetdienste nutzen können

Angewandte Informatik

Kalkulationsmodelle und Visualisierung

- Den (informatischen) Funktionsbegriff erklären können
- Kalkulationsmodelle zur Lösung von Problemen gestalten und implementieren können
- Die Korrektheit von Kalkulationsmodellen und Berechnungsmethoden reflektieren und Alternativen prüfen können

Praktische Informatik

Algorithmen, Datenstrukturen und Programmierung

- Aufgaben mit Mitteln der Informatik modellieren können
- Komplexere Algorithmen entwerfen, diese formal darstellen, implementieren und testen können (Erweiterung, Vertiefung)

7. Klasse

5. Semester – Kompetenzmodul 5

Informatiksysteme

Technische Grundlagen und Funktionsweisen

- Technische Grundlagen und Funktionsweisen von Informatiksystemen verstehen und erklären können (Erweiterung und Vertiefung)

Angewandte Informatik

Suche, Auswahl und Organisation von Information

- Unter Verwendung passender Dienste und Angebote und Wahl geeigneter Suchmethoden Informationen und Medien gezielt suchen und auswählen können.
- Informationen hinsichtlich ihrer Relevanz und Qualität einschätzen und bewerten können.
- Adäquate Werkzeuge und Methoden der Daten- und Informationsorganisation beurteilen können.

Kalkulationsmodelle und Visualisierung

- Grundbegriffe strukturierter und tabellarisch erfasster Daten und Operationen benennen können sowie Datenbestände mit entsprechender Software auswerten und den Anforderungen entsprechend visualisieren können
- Varianten von Visualisierungen bewerten können

Praktische Informatik

Algorithmen, Datenstrukturen und Programmierung

- Aspekte der Prozeduralen, Funktionalen und Objektorientierten Programmierung nennen und an Beispielen erläutern können
- Aufgaben mit Mitteln der Informatik modellieren können. Vielfältige Algorithmen entwerfen, diese formal darstellen, implementieren und testen können (Erweiterung, Vertiefung)

Datenmodelle und Datenbanksysteme

- Den Begriff Datenbanken und andere in diesem Kontext wichtige Fachbegriffe beschreiben und an Beispielen erklären können
- Datenbankmodelle, Tabellen und ihre Beziehungsmuster sowie weitere Datenbankobjekte erklären können
- Daten strukturiert (in Tabellen) erfassen, abfragen, auswerten sowie Datenbanken modellieren und einfache automatisierte Datenbanklösungen entwickeln können
- Datenmodelle hinsichtlich der Datentypen, Redundanz, Integrität und Relevanz bewerten können

6. Semester – Kompetenzmodul 6

Informationstechnologie, Mensch und Gesellschaft

Verantwortung, Datenschutz und Datensicherheit

- Für den Schutz und die Sicherheit von Informatiksystemen, mit denen man arbeitet, sorgen können

Geschichte der Informatik

- Meilensteine in der Entwicklung der Computertechnik beschreiben und maßgebliche dahinterstehende Persönlichkeiten nennen können
- Das geschichtliche Wissen in Beziehung zur aktuellen Situation setzen und daraus gegebenenfalls mögliche Zukunftsszenarien ableiten können

Berufliche Perspektiven

- Berufsfelder benennen können, in denen die Anwendung der IT eine bedeutende Rolle spielt, und die Vielfalt an IT-Berufen kategorisieren können
- Das eigene Wissen und die eigenen schulischen Erfahrungen im Zusammenhang mit IT für die eigene künftige Erwerbsbiographie nutzen können
- Die wirtschaftliche Bedeutung der IT in den diversen Berufsfeldern einordnen und die Chancen von IT-Berufen abschätzen können

Informatiksysteme

Technische Grundlagen und Funktionsweisen

- Unterschiedliche digitale Endgeräte bzw. Informatiksysteme in Bezug auf ihre technischen Eigenschaften und ihre Leistungsfähigkeit bewerten können
- Einfache Fehler diagnostizieren und beheben können

Netzwerke

- Maßnahmen zur Netzwerksicherheit umsetzen können
- Technische Aspekte von Netzwerken hinsichtlich der Verfügbarkeit und Qualität einschätzen können
- Einsatzmöglichkeiten verschiedener Internetdienste bewerten können

Praktische Informatik

Konzepte der Informationsverarbeitung

- Wesentliche informatische Konzepte und fundamentale Ideen der Informatik benennen und an Hand von Beispielen erklären können
- Bei der Lösung konkreter Aufgaben Heuristiken, Grundprinzipien und Konzepte der Informatik anwenden und informatische Modelle gestalten können

Algorithmen, Datenstrukturen und Programmierung

- Wesentliche Aspekte der Prozeduralen, Funktionalen und Objektorientierten Programmierung nennen und an Beispielen erläutern können
- Vielfältige Aufgaben mit Mitteln der Informatik modellieren können
- Vielfältige Algorithmen entwerfen, diese formal darstellen, implementieren und testen können

8. Klasse – Kompetenzmodul 7

7. Semester

Informationstechnologie, Mensch und Gesellschaft

Bedeutung von Informatik in der Gesellschaft

- Wissen über Informatiksysteme im digitalen privaten und schulischen Umfeld zielgerichtet anwenden und nutzen können
- Den Einfluss von Informatiksystemen auf den Alltag, auf die Gesellschaft und Wirtschaft einschätzen und an konkreten Beispielen Vor- und Nachteile abwägen können

Verantwortung, Datenschutz und Datensicherheit

- Persönliche Rechte und Pflichten in der Nutzung von Informatiksystemen beschreiben und wesentliche Aspekte des Datenschutzes und der Datensicherheit erklären können
- Beim Einsatz von Informatiksystemen das Wissen um Pflichten und Rechte in Bezug auf die eigene Person und ihre Arbeitsumgebung, auf persönliche und fremde Daten verantwortungsbewusst anwenden können

Geschichte der Informatik

- Anhand der Entwicklung der IT zwischen kurzlebigen Hard- und Softwareprodukten und langlebigen Prinzipien unterscheiden können

Informatiksysteme

Technische Grundlagen und Funktionsweisen

- Grundlegende technische Konzepte von Informatiksystemen verstehen

Betriebssysteme und Software

- Sich in die Bedienung neuer Software selbstständig einarbeiten können
- Software (inklusive Betriebssysteme) zur Bewältigung von Aufgaben bewerten und die Wahl für einen Lösungsweg begründen können

Mensch-Maschine-Schnittstelle

- Die Benutzerfreundlichkeit von MMS einschätzen und die Bedeutung für die Anwender bewerten können

Angewandte Informatik

Kommunikation und Kooperation

- Den situationsgerechten Einsatz von Kommunikations- und Kooperationssystemen bewerten können und ihre Bedeutung für mich und die Gesellschaft reflektieren können

Praktische Informatik

Konzepte der Informationsverarbeitung (Erweiterung, Vertiefung)

- Wesentliche informatische Konzepte und fundamentale Ideen der Informatik benennen und an Hand von Beispielen erklären können
- Bei der Lösung konkreter Aufgaben Heuristiken, Grundprinzipien und Konzepte der Informatik anwenden und informatische Modelle gestalten können
- Unterschiedliche Lösungsansätze in Bezug auf zugrunde liegende Konzepte reflektieren und in konkreten Handlungssituationen bewerten können

Algorithmen, Datenstrukturen und Programmierung

- Wesentliche Aspekte und Methoden der Softwareentwicklung und des Softwareprojektmanagements erklären können
- Ein Softwareprojekt planen und durchführen können
- Die Schritte der Softwareentwicklung reflektieren können
- Die Angemessenheit der Entwicklungswerkzeuge grob einschätzen können
- Die Effizienz von Algorithmen bewerten können
- Gezielt nach Programmfehlern suchen und diese korrigieren können

Intelligente Systeme

- Bereiche beschreiben können, in denen sich Informatiksysteme bzw. Computer intelligent verhalten
- Den Unterschied zwischen menschlicher und maschineller Intelligenz erklären können, Merkmale menschlicher Intelligenz und künstlicher Intelligenz vergleichen und einschätzen können.
- Intelligente Informatiksysteme anwenden können

8. Semester

Sicherung der Nachhaltigkeit

- Wiederholen, Vertiefen von Fähigkeiten und Vernetzen von Inhalten, um einen umfassenden Überblick über die Zusammenhänge unterschiedlicher informatischer Gebiete zu gewinnen

Glossar

Adobe Phonegap ist eine Distribution von Apache Cordova, erweitert durch Tools wie Command-Line-Tools, Phonegap-Build und die PhoneGap Developer App. 22, 57

Apache Cordova ist eine Sammlung von Geräte APIs, die das Programmieren von hybriden Apps und die Verwendung der nativen Gerätefunktionen erlaubt. 57

App ist ein Programm oder eine Software mit einem speziellen Zweck, die auf ein mobiles Gerät heruntergeladen werden kann. 19

Barren-Handys Handys mit sehr kleinem Bildschirm und einer Tastatur, bei der die Zahlentasten mit mehreren Buchstaben belegt sind. 17

Bebras Challenge internationale Bezeichnung für den Biber der Informatik. 1

Biber der Informatik ist ein Wettbewerb zum Thema Informatik. Dabei liegt der Schwerpunkt der Fragen auf Konzepten der Informatik. 1

Cascading Style Sheets ist eine Technologie zur Steuerung der Anzeige von Elementen auf Bildschirmen. 55

Computational Thinking ist die mentale Aktivität, die passiert, wenn ein Problem formuliert wird um eine berechenbare Lösung zu finden. 8

Crossplatformtool ist ein Tool, das es den Entwicklerinnen und Entwicklern ermöglicht, eine App nur einmal in einer bestimmten Programmiersprache zu schreiben. Mit einem Crossplatformtool kann die App dann für verschiedene Betriebssysteme bereitgestellt werden. 56

CSS Cascading Style Sheet. 22, 56

Distant Learning Lernen über weite Strecken hinweg. 30

E-Learning Lernen mit Hilfe von elektronischen Medien. 29, 31

Feature Phone mobile Telefone mit etwas größerem Display und größerem Funktionsumfang haben; wird hauptsächlich für die Kommunikation verwendet, hat aber eine Verbindung für den Datenaustausch über GPRS oder UMTS. Die Tastatur besteht aus einem Nummernblock und wenigen zusätzliche Buttons. 16

GMS Geostationary Meteorological Satellite. 15

GPRS General Packet Radio Service. 16

Handhelds Handhelds sind wie der Name schon sagt, mobile Geräte, die in eine Hand passen. Sie haben sich aus den Kalender und Agendas entwickelt und haben diesen auch als Hauptzweck behalten. Sie stellen im Unterschied zu Handys eine Internetverbindung nur über das WLAN her. 15

HTML Hypertext Markup Language. 21, 22, 55, 56

hybride App Apps, die in bekannten Webtechnologien als WebApp umgesetzt werden und von einem Crossplattformtool für die verschiedenen Betriebssysteme übersetzt wird. 22

Hypertext Markup Language ist eine Beschreibungssprache, die von Browsern interpretiert wird. 55

Information Foraging ist eine Theorie, die sich damit beschäftigt, wie der Mensch bei der Informationssuche im Internet vorgeht. 41

Informationsfährte bezieht sich auf die Benennung und Betitelung von Elementen, die der UserIn schon einen Hinweis darauf gibt, worum es sich handelt. 42

informelles Lernen Der Lernende lernt durch eigenen Informationsbeschaffung. 32

Microlearning kleine, kurze, kompakte Lerneinheiten, die mit direktem Feedback verbunden sind. 33

Mini-IA Mini-Informationsarchitektur. 51

Mini-Informationsarchitektur ist ein Begriff für die Strukturierung von Inhalt zu einem einzelnen Thema. 51

M-Learning Mobiles Lernen. 13, 30

MMS Multimedia Messaging Service. 16

mobile Geräte kleines elektronisches mit Rechenleistung versehenes Gerät, dass eine unterbrochene oder permanente Verbindung zum Internet hat. 13

Mobile Learning Lernen anhand von mobilen Geräten. 13, 28, 30, 31

Mobile User Experience ist ein Begriff, der die Wahrnehmungen und Reaktionen einer Person auf die Verwendung oder erwartete Verwendung von einem Produkt beschreibt. 16

Multiple-Choice-Fragen Fragen, bei denen mehrere Antwortmöglichkeiten zur Auswahl vorgegeben werden. 87

native App Apps die speziell für ein Betriebssystem in der dem System eigenen Programmiersprache verfasst ist. 20

Netbook ist eine kleinere, leichtere Version des Notebooks. Es fehlen aber meistens CD und DVD Laufwerk und sie haben auch eine kleinere Tastatur. 13, 14

Notebook Notebooks sind mobile Rechner mit normaler Tastatur und CD/DVD Laufwerk, die aber nicht im Stehen und Gehen benutzbar sind. 15

PDA Person Digital Assistent. 15

personalized learning ist eine Form des Lernens, die speziell auf die Bedürfnisse eines Lernenden zu geschnitten ist. 29

Prinzip der Kontiguität Beschreibt den Zusammenhang von Dingen. Es gibt die räumliche Kontiguität, die sich auf das nah beinander liegende Anzeigen von Bild und Text bezieht, aber auch die zeitliche Kontiguität, die den Zusammenhang zwischen Audio und Bildern beleuchtet. 52

Prinzip der Reihenfolge das Prinzip der Reihenfolge legt fest, dass Text und Bild, die zusammengehören, aber nicht gleichzeitig angezeigt werden können, immer in der Reihenfolge Bild dann Text dargestellt werden sollen. 52

situiertes Lernen der Lernende lernt in einer vorgegebenen Umgebung, unter vorbereiteten Bedingungen, das Lernen ist eingebettet in eine bestimmte Situation. 32

Smartphone mobiles Telefon mit mittelgroßem Bildschirm, einer Tastatur mit allen Buchstaben von A-z und einer Anbindung an das Internet (z.B.: Blackberry-Modelle). 15, 17

SMS Short Message Service. 16

Split-Attention-Prinzip ist das Prinzip der geteilten Aufmerksamkeit. 52

SQLite-Datenbank ist eine eingebettete Datenbank, die als eigenständiger Prozess innerhalb einer Applikation läuft. 57, 58, 61

- Tablet PC** Tablet PCs sind mobile Rechner mit Touchscreen. Die Tastatur wird auf dem Screen eingeblendet. Diese Geräte können mit nur einer Hand bedient werden. 15
- Thin Client** Thin Clients sind mobile Internet Geräte, die meistens über eine Tastatur bedient werden und eine ständige Internetverbindung brauchen. 15
- Touchphone** mobiles Telefon mit sehr großem Bildschirm, Touchscreen und einer sehr guten Verbindung zum Internet. iii, 18, 38, 73, 102
- Touchscreen** ein auf Berührung reagierender Bildschirm; Mittels Tippen und Gesten können Applikationen gesteuert werden. 18
- Ultra Mobile PC** sehr kleines, leichtes Tablet, an welches auch eine Tastatur angeschlossen werden kann. 14
- UMPC** Ultra Mobile PC. 15
- UMTS** Universal Mobile Telecommunications System. 15, 16
- Usability** ist der Begriff der beschreibt, in welchem Ausmaß ein Produkt von bestimmten Userinnen und User verwendet wird, um vordefinierte Ziele in einer bestimmten Benutzungsumgebung effektiv, effizient und befriedigend zu erreichen. 17, 18, 26, 38, 53
- WebApp** Apps, die als Webseite umgesetzt sind. 21
- Web-Tablet** Web-Tablets sind den Tablet PCs sehr ähnlich, brauchen aber eine ständige Internetverbindung und werden hauptsächlich wie PDAs genutzt. 15
- WLAN** Wireless Local Area Network. 18, 20

Literaturverzeichnis

- [1] Oxford dictionary: App. <http://www.oxforddictionaries.com/definition/english/app>. Online; Stand 4. März 2016.
- [2] Boonlit Adipat, Dongsong Zhang, and Lina Zhou. The Effects of Tree-View Based Presentation Adaption on mobile Web Browsing. *MIS Quarterly*, 35(1), 2011.
- [3] Grant Allen and Mike Owens. *The Definitive Guide to SQLite* -. Apress, New York, 2. aufl. edition, 2011.
- [4] David Alonso-Ríos, David Raneburger, Roman Popp, Hermann Kaindl, and Jürgen Falb. A User Study on Tailoring GUIs for Smartphones. In *Proceedings of Society for Information Technology & Teacher Education International Conference 2014*, pages 186–192. ACM, 2014.
- [5] Ben Bachmair, Maren Risch, Katja Friedrich, and Katja Mayer. Eckpunkte einer Didaktik des mobilen Lernens. *MedienPädagogik - Zeitschrift für Theorie und Praxis der Medienbildung Theorie und Praxis der Medienbildung*, 19:1–38, 2011.
- [6] Peter Baumgartner. Microlearning - Vier didkatische Herausforderungen. <http://www.procontext.com/aktuelles/2010/03/usability-und-user-experience-unterscheiden.html>, 2013. Online; Stand 29. Juli 2015.
- [7] Marco Bergmann. Hybrid Apps–Definition, Eigenschaften, Einsatzorte, Vorteile und Beispiele. <http://www.flyacts.com/blog/hybrid-apps-definition-eigenschaften-einsatzorte-vorteile-und-beispiele/>, 2013. Online; Stand 29. Juli 2015.
- [8] Kathleen Buettner and Anna M. Simmons. Mobile Web and Native Apps : How One Team Found a Happy Medium. In Aaron Marcus, editor, *Design, User Experience, and Usability. Theory, Methods, Tools and Practice*, pages 549–554. Springer Science & Business Media, 2011.
- [9] Österreichische Computergesellschaft. Biber der informatik. <http://www.ocg.at/de/biber>. Online; Stand 20. Dezember 2015.

- [10] Valentina Dagiene and Gerald Futschek. Bebras International Contest on Informatics and Computer Literacy: Criteria for Good Tasks. In Matevž Jekovec, editor, *Proceedings of the 3rd International Conference on Informatics in Secondary Schools - Evolution and Perspectives: Informatics Education - Supporting Computational Thinking*, ISSEP '08, pages 19–30. Springer, 2008.
- [11] Ana Luiza Dias, Renata De Pontin, Mattos Fortes, Paulo Cesar Masiero, Willian Massami Watanabe, and Matheus Edson Ramos. An Approach to Improve the Accessibility and Usability of Existing Web System. In *Proceedings of the 31st ACM international conference on Design of communication, SIGDOC '13*, pages 39–48. ACM, 2013.
- [12] Xiang Fang and Clyde W. Holsapple. Eckpunkte einer Didaktik des mobilen Lernens. *Decision Support Systems*, 43(2):476–491, 2006.
- [13] Florian Franke and Johannes Ippen. *Apps mit HTML5 und CSS3 : für iPad, iPhone und Android ; [Geodaten, Videos, Sound, Grafiken, Bewegungssensoren u.v.m. ; Arbeit mit jQuery Mobile, Sencha Touch, PhoneGap ; inkl. Entwicklung von Tablet-Magazinen]* -. Galileo Press, Bonn, 2. aufl. edition, 2013.
- [14] Kristin Skeide Fuglerud and Till Halbach Røssvoll. An evaluation of web-based voting usability and accessibility. *Universal Access in the Information Society*, 11(4):359–373, 2012.
- [15] Gerald Futschek. Österreich braucht gut ausgebildete informatiklehrerinnen. In *Best Spirit : Best Practice, Lehramt an Österreichischen Universitäten*, pages 100–104. Braumüller, 2011.
- [16] Thomas Geis. Usability und User Experience unterscheiden. <http://peter.baumgartner.name/2013/06/23/microlearning-vier-didaktische-herausforderungen/>, 2010. Online; Stand 10. März 2015.
- [17] Arjan Geven, Reinhard Sefelin, and Manfred Tscheligi. Depth and breadth away from the desktop: the optimal information hierarchy for mobile use. In *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services, MobileHCI '06*, pages 157–164. ACM, 2006.
- [18] Redaktion Gründerszene. App: Was ist eine app? <http://www.gruenderszene.de/lexikon/begriffe/app>, 2014. Online; Stand 3. März 2016.
- [19] Miran Han and Peom Park. A Study of Interface Design for Widgets in Web Services Through Usability Evaluation. In *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, pages 1013–1018. ACM, 2009.

- [20] Andreas Holzinger and Maximilian Errath. Mobile computer Web-application design in medicine: some research based guidelines. *Universal Access in the Information Society*, 6(1):31–41, 2007.
- [21] Hwaci. SQLite. <http://www.sqlite.org/about.html>. Online; Stand 29. April 2015.
- [22] Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems. Standard, International Organization for Standardization, March 2010.
- [23] Rodolfo Inostroza, Silvana Roncagliolo, Cristian Rusu, and Virginica Rusu. Usability Heuristics for Touchscreen-based Mobile Devices: Update. In *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, ChileCHI '13, pages 24–29. ACM, 2013.
- [24] Rodolfo Inostroza and Cristian Rusu. Mapping Usability Heuristics and Design Principles for Touchscreen-based Mobile Devices. In Philip Tsang, SimonK.S. Cheung, VictorS.K. Lee, and Ronghuai Huang, editors, *Proceedings of the 7th Euro American Conference on Telematics and Information Systems*, EATIS '14. ACM, 2014.
- [25] Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability. Standard, International Organization for Standardization, Geneva, CH, March 1998.
- [26] Desmond Keegan. The future of learning: From elearning to mlearning. http://learning.ericsson.net/mlearning2/project{_}one/book.html, 2002. Online; Stand 29. Juli 2015.
- [27] Michael O. Leavitt and Ben Shneiderman. *Research-based Web Design & Usability Guidelines* -. U.S. Department of Health and Human Services, u.s. government printing office edition, 2007.
- [28] Desmond Lobo, Kerem Kaskaloglu, ChaYoung Kim, and Sandra Herbert. Web Usability Guidelines For Smartphones: A Synergic Approach. *International Journal of Information and Electronics Engineering*, 1(1):33–37, Juli 2011.
- [29] Olibário Machado Neto and Maria Da Graça Pimentel. Heuristics for the Assessment of Interfaces of Mobile Devices. In *Proceedings of the 19th Brazilian symposium on Multimedia and the web*, WebMedia '13, pages 93–96. ACM, 2013.
- [30] Helmut M. Niegemann, Steffi Domagk, Silvia Hessel, Alexandra Hein, Matthias Hupfer, and Annett Zobel. *Kompendium multimediales Lernen* -. Springer Science & Business Media, Berlin Heidelberg, 2008. aufl. edition, 2008.

- [31] Jakob Nielsen. {CHAPTER} 1 - What Is Usability? In Chauncey Wilson, editor, *User Experience Re-Mastered - Your Guide to Getting the Right Design*, pages 3 – 22. Morgan Kaufmann, 2010.
- [32] Jakob Nielsen and Raluca Budiu. *Mobile Usability*. MITP-Verlags GmbH & Co. KG, Heidelberg, 1. Aufl. edition, 2013.
- [33] Erik G. Nilsson. Design patterns for user interface for mobile applications. *Advances in Engineering Software*, 40:1318–1328, 2011.
- [34] Norbert Pachler, Ben Bachmair, John Cook, and Gunther Kress. *Mobile Learning - Structures, Agency, Practices*. Springer Science & Business Media, Berlin Heidelberg, 2010.
- [35] Yong S. Park, Sung H. Han, Jaehyun Park, and Youngseok Cho. Touch Key Design for Target Selection on a Mobile Phone. *International journal of industrial ergonomics*, 40(1):68–76, 2010.
- [36] Serena Pastore. Touch Key Design for Target Selection on a Mobile Phone. *Advances in Computer Science: an International Journal*, 3(1):27–36, 2014.
- [37] Wolfgang Pohl. Bebras contest. <http://bebras.org>. Online; Stand 1. Feber 2016.
- [38] Wolfgang Pohl and Hans-Werner Hein. Aspects of Quality in the Presentation of Informatics Challenge Tasks. In Matevž Jekovec, editor, *Proceedings of International Conference on Informatics in Schools: Situation, Evolution and Perspectives*, ISSEP '15, pages 21–32. University of Ljubljana, Faculty of Computer and Information Science, 2015.
- [39] Polymer Authors. Welcome - Polymer 1.0. <https://www.polymer-project.org>, 2015. Online; Stand 17. November 2016.
- [40] Gunther Schiefer and Michael Decker. Taxonomy for mobile Terminals. In *Proceedings of the International Conference on e-Business*, ICB-E 2008, pages 255–258, 2008.
- [41] Tim Schneidermeier, Florian Maier, and Johannes Schricker. *Design, User Experience, and Usability. Web, Mobile, and Product Design: Second International Conference, DUXU 2013, Held as Part of HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013, Proceedings, Part IV*, chapter Human-Centered Communication Planning: A Conceptual Approach, pages 94–102. Springer Berlin Heidelberg, 2013.
- [42] Sigrid Schubert and Andreas Schwill. *Didaktik der Informatik*. Springer-Verlag, Berlin Heidelberg New York, 2. Aufl. edition, 2011.

- [43] Karsten Seipp and Kate Devlin. Landscape Vs Portrait Mode: Which Is Faster To Use On Your Smart Phone? In *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services*, MobileHCI 13, pages 255–258. ACM, 2013.
- [44] Tom Seymour, Jasmine Zakir Hussain, and Sharon Reynolds. How To Create An App. *International Conference on ICT for Smart Society*, 18(2):27–36, 2014.
- [45] Maria Shitkova, Justus Holler, Tobias Heide, Nico Clever, and Jörg Becker. Towards Usability Guidelines for Mobile Websites and Applications. In *Wirtschaftsinformatik Proceedings 2015*, 2015.
- [46] Ben Shneiderman, Catherine Plaisant, Maxine Cohen, and Steven Jacobs. *Designing the User Interface - Strategies for Effective Human-computer Interaction*. Addison-Wesley, Amsterdam, international ed of 5th revised ed edition, 2010.
- [47] Tapanee Tirapat and Tiranee Achalakul. Usability Assessment for Hyperlink Methods. In *Proceedings of the 2006 International Conference on Hybrid Information Technology - Volume 01, ICHIT '06*. IEEE Computer Society, 2015.
- [48] John Traxler. Defining, Discussing and Evaluating Mobile Learning: The moving finger writes and having writ *The International Review of Research in Open and Distance Learning*, 8(2), 2007.
- [49] Verein zur Förderung Mathematischer Interessen und Begabungen. Känguru der Mathematik. <https://www.kaenguru.at>. Online; Stand 3. März 2016.
- [50] John M. Wargo. *Apache Cordova 4 Programming* -. Addison Wesley, Amsterdam, 2015.
- [51] Jeannette M. Wing. Computational Thinking. *Communications of the ACM*, 49(3):33–35, März 2006.
- [52] Jeannette M. Wing. Research Notebook: Computational Thinking–What and Why? <http://http://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>, 2011. Online; Stand 31. März 2016.
- [53] Claudia Witt, Almut (Eds.) Reiners, Claudia de Witt, and Almut Reiners. *Mobile Learning - Potenziale, Einsatzszenarien und Perspektiven des Lernens mit mobilen Endgeräten*. Springer-Verlag, Berlin Heidelberg New York, 1. Aufl. edition, 2013.
- [54] Andreas Zendler and Christian Spannagel. Zentrale Konzepte im Informatikunterricht: eine empirische Grundlegung. *Notes on Educational Informatics - Section A: Concepts and Techniques*, 2(1):1–21, 2006.

- [55] Dongsong Zhang and Boonlit Adipat. Challenges, Methodologies, and Issues in the Usability Testing of Mobile Applications. *International Journal of Human-Computer Interaction*, 18(3), 2005.
- [56] Yazhen Zhang and Jian Li. Application of 3G-based Mobile Learning in Teacher Training. In Philip Tsang, SimonK.S. Cheung, VictorS.K. Lee, and Ronghuai Huang, editors, *Proceedings of the 3rd International Conference of Human Learning, ICHL 2010*, pages 27–29. Springer, 2010.