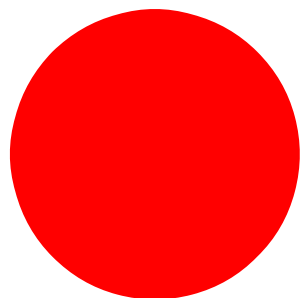
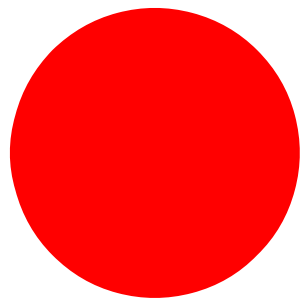


iSchadau

Die iPhone App für das Gymnasium Thun-Schadau

Eine Produktarbeit von Benjamin Messerli, 1g
Betreut durch Ralf Kretzschmar

18.8.2011
Gymnasium Thun-Schadau



1 Abstract

Mit der Einführung des iPhone im Jahr 2007 hat Apple die mobile Telefonie nachhaltig revolutioniert. Als Kombination von mobiler Telefonie und mobilem Computing können auf diesen Smartphones kleine Programme – sogenannte App's – installiert werden. Die vorliegende Maturaarbeit erläutert den gesamten Entwicklungsprozess einer solchen App von der Ideenfindung bis zum fertigen Produkt und zeigt auf, wie für das Gymnasium Thun-Schadau eine eigene App entsteht. Die fertige App mit den Hauptfunktionen ‚Stundenplan‘, ‚Bistro‘, ‚Termine‘, ‚Lehrer‘ und ‚Homepage‘ zeigt eindrücklich, wie mit Grundkenntnissen des objektorientierten Programmierens eine iPhone App zu einem marktreifen Produkt entwickelt werden kann. Die App ‚iSchadau‘ ist die Erfolgsgeschichte im Kleinen des Autors, der als iPhone Fan von seiner eigenen App geträumt hat.

2 Vorwort

Diese Maturaarbeit ist in ihren Grundzügen für alle interessierten Leser verständlich. Es gibt jedoch kleinere Abschnitte, welche ohne grundlegende Software- und Programmierkenntnisse nicht nachvollziehbar sind. In den entsprechenden Abschnitten ist dies noch vermerkt.

Der Rahmen einer gewöhnlichen Maturaarbeit beträgt ca. 100 Stunden. Der Aufwand dieser Arbeit hat diesen Rahmen überschritten und beträgt ca. 160 Stunden. Davon sind ungefähr 130 Stunden für die Softwareentwicklung verwendet worden.

Ich habe mich für die Entwicklung einer eigenen App entschieden, da ich an der Informatik interessiert und ein grosser Apple Fan bin. Mein Bruder und mein Vater sind beide im Bereich der Informatik tätig und ich besuche seit einem Jahr das Ergänzungsfach Informatik am Gymnasium. Zudem bin ich ein begeisterter Apple Kunde, somit bin ich auch im Besitz eines iPhone. Die Vermarktung des revolutionären iPhone repräsentiert für mich eine zukunftsorientierte Einstellung der Gesellschaft und kann als Musterbeispiel für die Informatisierung angesehen werden. Diese Faktoren führten letztendlich dazu, dass ich die ehrgeizige Aufgabe angepackt habe.

Während der Arbeit kam ich definitiv zum Schluss, dass ich die komplexe Entwicklung einer iPhone App unterschätzt habe. Trotz dieser Feststellung nahm die Motivation die App fertigzustellen kaum ab. Für gewisse Funktionen meiner App zog ich Hilfe von meinem Vater bei, da Programmierfragen auftraten, die mit meinen Kenntnissen schlicht als unlösbar schienen. Dass ich Hilfe von meinem Vater beziehe, wurde bereits vor Beginn der Entwicklung mit meinem Betreuer vereinbart. In meiner Arbeit werden die entsprechenden Teile angesprochen.

Mein Dank richtet sich an Alle, die mich unterstützt haben, ohne je eine Gegenleistung von mir verlangt zu haben. Herzlichen Dank an Jean-Pierre Messerli - mein Vater - Arzt und Informatiker, Tommaso Pace – ein guter Freund – Dipl. Arch Ba, HSLU, der mit mir ein Design entworfen hat, den Verantwortlichen von WebUntis von Untis und von SV-Group, welche mir freundlicherweise alle notwendigen Schnittstellen zur Verfügung stellten und schliesslich meinem Betreuer Dr. Ralf Kretzschmar, welcher mir fast komplette Eigenständigkeit

zutraute, mich fast nie unter Stress setzte und durch seine Flexibilität immer erreichbar war.

Ich widme diese Arbeit Stefanie Desirée Wullschleger.

3 Inhaltsverzeichnis

1	<i>Abstract.....</i>	<i>2</i>
2	<i>Vorwort.....</i>	<i>3</i>
3	<i>Inhaltsverzeichnis</i>	<i>5</i>
4	<i>Softwareapplikation für iPhone.....</i>	<i>7</i>
4.1	Die IT Firma Apple Incorporation.....	7
4.2	iPhone	8
4.3	App	9
4.4	Erfolgsgeschichte iPhone	10
5	<i>Entwicklungsprozess.....</i>	<i>11</i>
5.1	Hardware.....	11
5.2	Software	12
5.3	Programmierkenntnisse	12
5.4	Ideenfindung, Entwicklung, Testing.....	13
5.5	Deployment.....	13
5.6	Wartung, Updates.....	14
6	<i>Beschrieb Entwicklung</i>	<i>15</i>
6.1	Developer Lizenz.....	15
6.2	Xcode	15
6.3	Interface Builder	16
6.4	iOS Simulator.....	16
6.5	Beispiel	17
7	<i>Werkbericht: Eigene App herstellen</i>	<i>23</i>
7.1	Ideenfindung.....	23
7.2	Benutzerumfrage.....	24
7.3	Application Definition Statement (ADS)	27

7.4	Prototyping	28
7.5	Einrichtung Entwicklungsumgebung	28
7.6	Erlernen des Handwerks	29
7.7	GUI-Navigation	30
7.8	Beschreibung und Implementation der Funktionen.....	32
7.9	Testing	36
7.10	Deployment.....	36
7.11	Möglichkeiten zur Weiterentwicklung	37
8	<i>Beschreibung der App</i>	<i>38</i>
9	<i>Erfahrungswerte - Fazit.....</i>	<i>47</i>
10	<i>Quellenverzeichnis.....</i>	<i>49</i>
11	<i>Abbildungsverzeichnis</i>	<i>50</i>
12	<i>Anhang</i>	<i>52</i>
12.1	Bewertungskriterien.....	52
12.2	Codebeispiel Hello World	55
12.3	Webservice WebUntis	55
12.4	Webservice SV-Group.....	55
12.5	Beschreibung Anwendungsfälle.....	55
12.6	Zeitplan.....	55
13	<i>Eidesstattliche Erklärung.....</i>	<i>56</i>

4 Softwareapplikation für iPhone

Die neue Generation von Mobiltelefonen wird durch das Smartphone iPhone eingeläutet. Das iPhone ist ein Produkt von der IT Firma Apple. Apple hat zusätzlich zum iPhone den App-Store entwickelt, welches eine Plattform von sogenannten Apps fürs iPhone darstellt. Softwareapplikation für iPhone erklärt den ganzen Zusammenhang von Apple und iPhone, damit diese Maturaarbeit von Grund auf verstanden werden kann.

4.1 Die IT Firma Apple Incorporation



Apple Inc. ist ein amerikanisches Unternehmen mit Hauptsitz in Kalifornien, welches Computer, Unterhaltungselektronik, Betriebssysteme und Anwendungssoftware herstellt. Apple wird

Abbildung 1: momentan durch den CEO Steve Jobs geleitet und ist grösster Konkurrent von der IT Firma Microsoft. Apple wurde 1976 gegründet und entwickelte sich rasch. Heute gehört Apple zu einer der umsatzstärksten Firmen der Welt. 2010 erzielten sie einen Umsatz von 66.225 Milliarden US-Dollar. Apple entwickelte eine breite Palette an Personal Computern, den Mac's. Zu ihnen gehören der Mac Pro, der Mac Mini, der iMac, das MacBook Pro und das MacBook Air. Weitere Produkte sind der iPod, das iPad und schliesslich das iPhone. In dieser Arbeit spielt das iPhone eine zentrale Rolle [1,3].



Abbildung 2: Produktpalette Apple

4.2 iPhone

Das Apple iPhone ist das revolutionäre Smartphone von Apple. Ein Smartphone ist ein Mobiltelefon, welches nicht zum telefonieren optimiert ist. Es besitzt viele Funktionen wie zum Beispiel das Abspielen von Musik, von Videos, den Zugang zum Internet, ein GPS und vieles mehr. Mit dem iPhone hat Apple das Smartphone revolutioniert, da die Bedienung des iPhones über einen sogenannten Wärmeempfindlichen Touchscreen erfolgt. Das iPhone kann also fast komplett intuitiv durch Berührung des hochempfindlichen Bildschirms gesteuert werden und kommt mit einem einzigen Knopf zurecht. Eine weitere Eigenschaft des iPhones ist die individuelle Gestaltung des Gerätes. Es können über eine Plattform, die sich App-Store nennt, Programme heruntergeladen werden und somit kann jedes iPhone individuell zusammengestellt werden. Die Programme nennt man Apps. Zurzeit gibt es 2 verschiedene iPhone - Modelle auf dem Markt: Das etwas ältere iPhone 3G S und das neueste iPhone 4 [1].



Abbildung 3: iPhone 4, schwarz

4.3 App-Store

Der App-Store ist eine von Apple zur Verfügung gestellte Plattform, auf der iPhone-Benutzer Programme, sogenannte Apps, auf das iPhone herunterladen können. Anfangs wollte Apple nicht, das Dritte Applikationen für den App-Store

entwickeln, doch unter starkem Druck der Medien entschied sich Apple um, so dass es für Dritte nun möglich ist für den App-Store Apps zu entwickeln. Jedoch wird jede eingesendete App vorerst inspiziert und überprüft, damit qualitativ gute und keine anstössigen Applikationen auf dem App-Store aufzufinden sind. Auf den App-Store kann vom Computer oder vom iPhone aus zugegriffen werden. Dazu benötigt man eine sogenannte Apple-ID. Diese ID ist eine Registrierung bei Apple und ist kostenlos. Mit der Apple ID können Apps auf dem App-Store heruntergeladen werden. Es gibt kostenlose sowie kostenpflichtige Apps. Die meisten Apps sind jedoch nicht teurer als 1.10 Fr [1].



Abbildung 4: Printscreen aus dem App Store

4.4 App

Eine Applikation, kurz App, ist ein Teilprogramm des iPhones. Jedes iPhone besteht aus Apps und verleiht jedem Gerät Individualität. Apps können wie schon beschrieben vom App-Store heruntergeladen werden. Die Apps werden auf dem iPhone sowie auf dem Computer gespeichert und verwaltet. Momentan sind über 425'000 Applikationen auf dem App-Store erhältlich. Es gibt unzählige Apps, welche verschiedenste Funktionen erfüllen. Es wird in verschiedene Kategorien

von Applikationen unterscheidet. Diese umfassen Spiele, Unterhaltung, Dienstprogramme, Soziale Netze, Musik, Produktivität, Lifestyle, Referenz, Reisen, Sport, Navigation, Gesundheit & Fitness, Nachrichten, Fotografie, Finanzen, Wirtschaft, Bildung, Wetter, Bücher und Medizin. Welche App in welche Kategorie gehört, entscheidet der Entwickler [1].



Abbildung 5: Typischer iPhone Bildschirm mit Apps

4.5 Erfolgsgeschichte iPhone

Das ältere iPhone 3G, welches heutzutage nicht mehr vermarktet wird, war die erste Generation des iPhone mit hochsensiblen Touchscreen und dem Zugriff auf den App-Store. Das iPhone 3G wurde 2007 von Steve Jobs vorgestellt. In den ersten zwei Tagen wurden 270'000 Geräte in Amerika verkauft. Seit 2007 wurden weltweit ganze 128'962'000 iPhone verkauft [4,5,6]!

5 Entwicklungsprozess

Dieses Kapitel beschreibt den gesamten Entwicklungsprozesses einer iPhone Applikation. Um eine App zu entwickeln benötigt man bestimmte Hardware und Software. Die Hardware ist zum programmieren, testen und verwenden des Produkts notwendig. Die Software wird zum Programmieren und für Tests verwendet. Wenn die Software und Hardware stimmen, werden Programmierkenntnisse in Objective-C vorausgesetzt. Schliesslich folgt die Ideenfindung, die Entwicklung und die Publikation der eigentlichen App.

5.1 Hardware

Zum Entwickeln einer iPhone App braucht es die richtige Hardware. Einerseits wird ein Mac Computer benötigt, andererseits ein iPhone. Nur auf dem Mac kann die richtige Software für die Programmierung der App installiert werden. Dabei spielt es keine Rolle wie leistungsfähig der Mac ist. Gegebenenfalls kann die App auch auf einem PC mit Windows entwickelt werden, dabei ist aber ein Mac Emulator notwendig. Ein Emulator beschreibt ein System in der Computertechnik, welches Teilaspekte eines anderen Systems nachahmt. Emulatoren sind jedoch für die Entwicklung von Apps eher ungeeignet, da sie nicht optimal auf die Software abgestimmt und somit fehleranfällig sind. Auf dem iPhone kann die jeweils aktuelle Version der App laufend getestet werden. Das Endprodukt wird ausschliesslich auf dem iPhone benutzt.



Abbildung 6: Ein MacBook Air

5.2 Software



Abbildung 7:
Logo des SDK's

Auf dem Mac wird die Software zum entwickeln der App installiert. Dazu stellt Apple das iPhone SDK (Software Developer Kit) als Download zur Verfügung. Diese Entwicklungsumgebung enthält unter anderem die Programmierungsumgebung Xcode, den Interface Builder und den iOS Simulator. Kommerziell wird Xcode als

Programmierungsumgebung benutzt, jedoch gibt es noch andere Entwicklungsumgebungen, welche von Drittanbietern erstellt wurden (z.B. Monotouch). Mit Xcode wird die Applikation in Objective-C programmiert, Interface Builder ist ein Tool um das GUI zu erstellen, iOS Simulator ist eine Software, welche ein iPhone simuliert. Die drei Tools sind weiter unten ausführlicher beschrieben. Um die App auf dem iPhone selbst zu testen und um sie auf dem App-Store zu veröffentlichen, benötigt es eine zusätzliche Developer Lizenz, welche online für ca. 100 Franken beantragt werden kann. Auf dem Mac App Store können weitere Tools heruntergeladen werden, welche jedoch nicht zwingend für die Entwicklung einer App notwendig sind [2].

5.3 Programmierkenntnisse

Apps werden in der objektorientierten Programmiersprache Objective-C programmiert. Das ist eine Ableitung von C, bzw von C++. Die von Apple zur Verfügung gestellte Entwicklungsumgebung setzt Programmierkenntnisse in Objective-C voraus. Falls man diese Sprache nicht beherrscht, muss sie zuerst erlernt werden. Dazu werden Kenntnisse in den Programmiersprachen C, bzw. C++ und/oder grundlegende Kenntnisse einer objektorientierten Programmiersprache wie zum Beispiel Java vorausgesetzt. Das Erlernen der Sprache beansprucht viel Zeit und kann mittels Lernbücher, Lernvideos, Kursen und online Tutorials erreicht werden. Objective-C ist eine anspruchsvolle Programmiersprache und das Erlernen der Programmiersprache ist mit viel Aufwand verbunden, vor allem wenn noch kaum Programmierkenntnisse vorhanden sind. Während der Entwicklung treten immer wieder Problemstellungen und Stolpersteine auf, welche mit Hilfe von Büchern, die sich mit SDK und iPhone Programmierung befassen oder mit einschlägigen Internetforen gelöst und behoben werden können.

5.4 Ideenfindung, Entwicklung, Testing

Wenn die gesamte Hardware und Software vorhanden und die Programmierkenntnisse erworben sind, steht nichts mehr im Weg um die Applikation zu erstellen. Als Erstes wird eine Idee für eine Applikation benötigt. Dabei kann es sich um alles mögliche Handeln. Wenn eine Idee vorhanden ist, empfiehlt es sich ein sogenanntes ADS (Application Definition Statement) und einen GUI Prototyp zu erstellen. Ein Application Definition Statement drückt aus, worum es in einer Anwendung geht. Dabei hilft die Anwendungsdefinition eine fokussiertere und sauberere App zu erstellen, da Entscheidungen bezüglich der Features und UI-Elemente besser getroffen werden können. Inhalt einer solchen Definition umfasst die Festlegung der Zielgruppe, welches Problem mit der geplanten Applikation gelöst werden soll und schliesslich, was die App besonders macht. Ein GUI Prototyp wird vorzugsweise auf Papier erstellt. Dieser Prototyp gibt eine Idee, wie die Benutzeroberfläche und die Navigation schliesslich aussehen wird.

Sind das ADS und der GUI-Prototyp erstellt, kann mit der Entwicklung begonnen werden. Zuerst wird ein erstes, grobes GUI programmiert, wobei die Funktionen nicht ausprogrammiert werden. Zum Beispiel wenn man auf einen Knopf „Aktualisieren“ klickt, kommt eine Meldung mit „Hier aktualisieren“, wobei noch gar nichts aktualisiert wird. In weiteren Schritten werden die Funktionen programmiert und das GUI angepasst. Die einzelnen Funktionen werden iterativ erstellt und laufend getestet. Ist die Applikation fertig gestellt und das GUI fertig gestaltet, beginnt die Testphase. In der Testphase werden für alle Funktionen der Applikation Testfälle definiert. Ein Testfall umfasst die Voraussetzungen zum Testen der Funktion, die auszuführenden Schritte und die erwarteten Resultate. Jeder Testfall wird auf der Applikation durchgeführt und protokolliert. Anhand dieser Protokolle wird die Applikation stabilisiert. Das korrigieren der protokollierten Fehler nennt man Debugging.

5.5 Deployment

Nach der Testphase kommt das Veröffentlichen der App auf dem App Store. Dazu wird die Applikation an Apple gesendet. Apple entscheidet, ob sie die Applikation annimmt oder ob die Applikation noch überarbeitet werden muss. Die

Applikationen können nur über den App Store publiziert und verkauft werden. Apple behält sich vor 1/3 des Verkaufspreises für sich zu beanspruchen. Dies ist ein hoher Prozentsatz, jedoch fair, da Apple den ganzen App Store verwaltet.

Wenn man die Applikation anmeldet, muss man sich darauf vorbereiten. Man braucht eine Beschreibung, einen Titel, Keywords und Screenshots. Im App Store werden Suchanfragen nur nach den Keywords gefiltert. Dabei sind die Keywords auf 100 Zeichen beschränkt und müssen sorgfältig und überlegt festgelegt werden.

5.6 Wartung, Updates

Wenn die App auf dem App Store publiziert wurde, folgt die Pflege der Applikation. Aufgrund von Rückmeldungen, Bewertungen, Wünschen und Kritiken von den Usern wird eine Unterhaltung erwartet. Die Wartung einer Applikation umfasst Anpassungen an Wünschen Debugging und Implementationen neuer Funktionen. Im App Store kann jeweils nur eine Version der App veröffentlicht werden. Um eine neuere Version aufzuschalten, muss das Update neu angemeldet werden. Updates sind für User, welche bereits ältere Versionen der App erworben haben, kostenlos erhältlich. In den Updates können auch Keywords, Screenshots, Titel und Beschreibung verändert werden, jedoch muss Apple die Applikation von neuem annehmen.

6 Beschrieb Entwicklung

In diesem Kapitel werden alle vorher geschilderten Entwicklertools und die Developer Lizenz beschrieben. Diese Tools und die Lizenz sind das Minimum an Voraussetzungen, welche Entwickler benötigen um erfolgreich eine Applikation zu entwickeln. Für das komplette Verständnis dieses Abschnitts benötigt man, wie im Vorwort angekündigt, vertiefte Programmier- und Softwarekenntnisse.

6.1 Developer Lizenz

Für den Zugang zur notwendigen Entwicklungsumgebung, das sogenannte Software Developer Kit von Apple, muss eine Developer Lizenz erworben werden. Die Kosten betragen 99\$ pro Jahr. Mit der Developer Lizenz kann der Entwickler [2] :

- die aktuellste Version von Xcode herunterladen
- die aktuellste Version von Interface Builder herunterladen
- die aktuellste Version vom iOS Simulator herunterladen
- Die App während der Entwicklung auf dem iPhone testen
- Zwei Mal im Jahr Hilfe von Apple für Programmierprobleme anfordern (weitere Hilfe kann kostenpflichtig erworben werden)
- Zugang zur umfassenden iOS-Dokumentation und Lernvideos
- Zugang zu einem von Apple erstellten Forum

6.2 Xcode



Abbildung 8:

Xcode ist die Programmierungsumgebung zur Entwicklung von Programmen für jegliche Produkte von Apple. Für die iPhone-App Programmierung wird zusätzlich die Bibliothek Cocoa-Touch mit den spezifischen Interaktionsmöglichkeiten zur Verfügung gestellt. Wie bereits mehrmals erwähnt wird in Xcode mit der Programmiersprache Objective-C gearbeitet. In diesem Tool wird das eigentliche Programm codiert. Es ist das meist verwendete Tool für die Entwicklung und ist mit dem Interface Builder und dem iOS Simulator gekoppelt. Der erstellte Code wird durch Xcode kompiliert und kann auf das iPhone geladen werden [2].

6.3 Interface Builder



Abbildung 9:

Interface Builder ist das Tool zur vereinfachten Erstellung von Bildschirmen und deren Elemente. Die im Interface Builder erstellten Elemente werden mit dem Code verknüpft. Es ist das Tool um das GUI auf einer grafischen Oberfläche zu erstellen, ohne dass es programmatisch erstellt wird [2].

6.4 iOS Simulator



Abbildung 10:

iOS Simulator ist ein Tool, welches komplett ein iPhone auf dem PC simuliert. Damit kann die Applikation fortlaufend getestet und begutachtet werden. Zusätzlich zum testen auf dem Simulator sind auch Tests auf dem iPhone notwendig, da der Simulator ein Modell des iPhone darstellt und somit nicht das gesamte iPhone nachbildet. Zum Beispiel können

Schüttelbewegungen nur auf dem iPhone getestet werden [2].

6.5 Beispiel

Zur Veranschaulichung der beschriebenen Tools möchte ich anhand einer kleinen Applikation die Verwendung des SDK's aufzeigen. Die App trägt den Namen „Hello World“ und ist eine simple Applikation, welche ein Eingabefeld, einen Output Button und ein Ausgabefenster enthält. Der Benutzer kann in das Eingabefenster einen Namen eingeben, den Output Button drücken und es erscheint im Ausgabefeld eine kurze Textzeile mit „Hallo *Name*“. Der gesamte Code dieser kleinen Applikation ist im Anhang aufzufinden.

Im Xcode wird als Erstes ein neues Projekt angelegt. Dabei erstellt Xcode automatisch die erforderliche Programmierumgebung mit der Grundausstattung an Klassen und Frameworks.

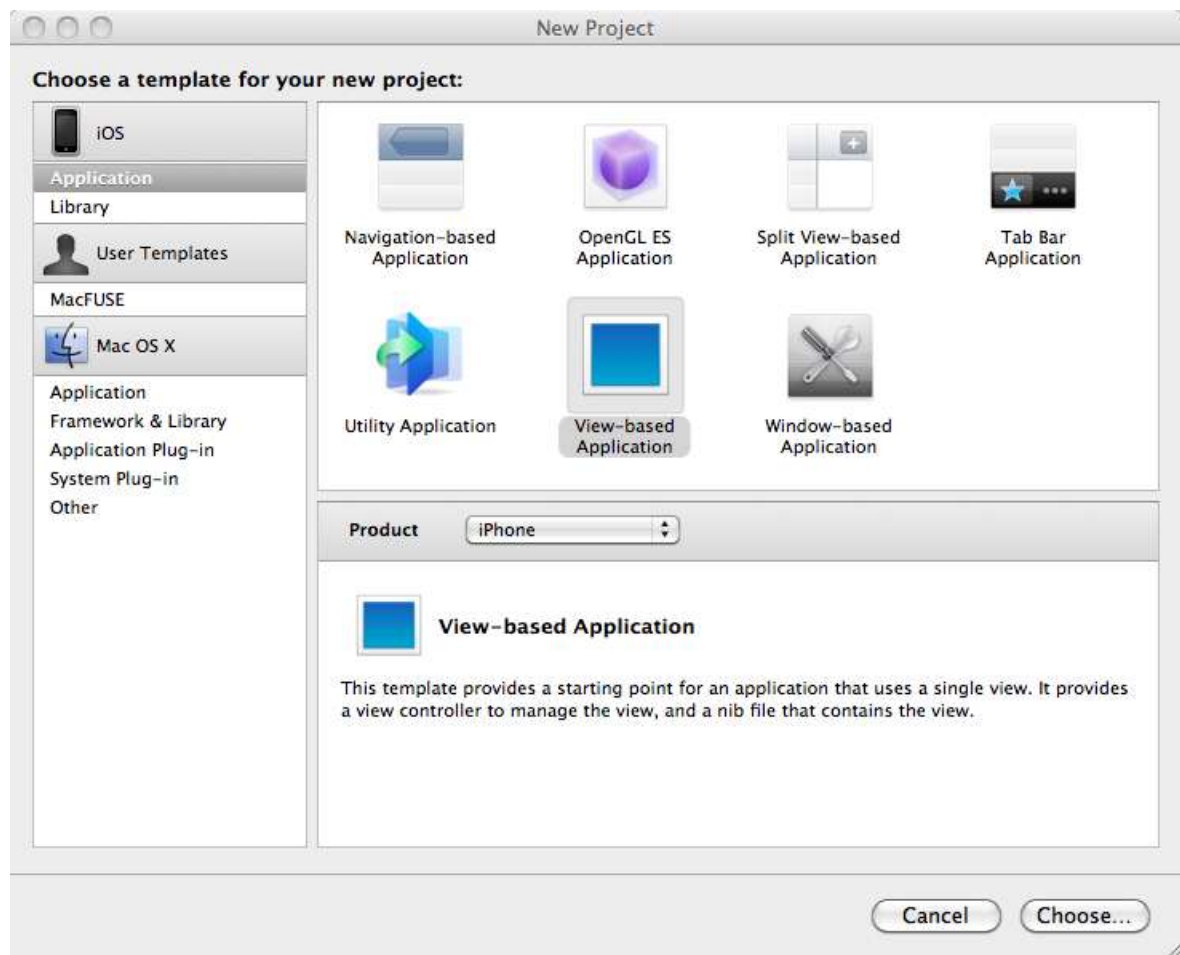


Abbildung 11: Printscreen Erstellen von einem neuen Projekt in Xcode

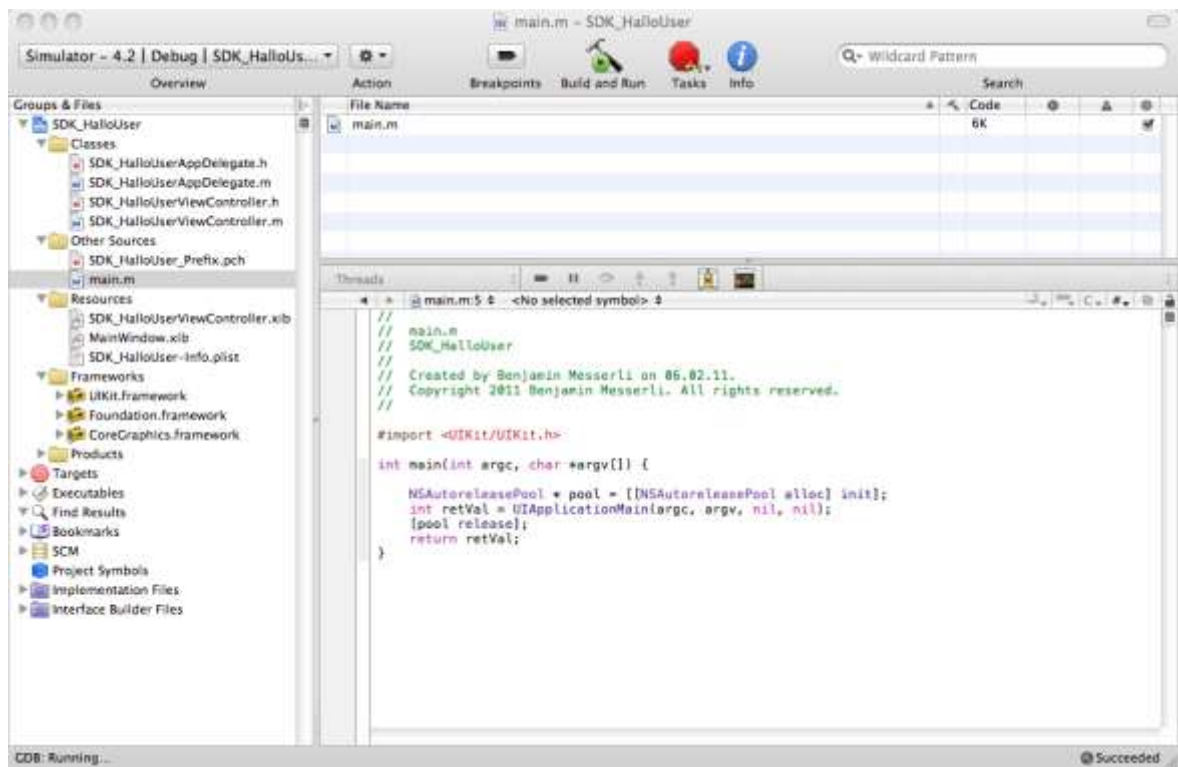


Abbildung 12: Printscreen automatische Erstellung Grundausrüstung an Klassen und Frameworks

Als nächstes wird der Interface Builder geöffnet und die gewünschten UI-Elemente werden aus der Bibliothek eingebaut und mit dem Inspector angepasst.



Abbildung 13: Printscreen Interface Builder

Danach werden im Xcode in der header Datei (SDK_HalloUserController.h) die grafischen Elemente im Code angelegt und die Methode sayHello deklariert.

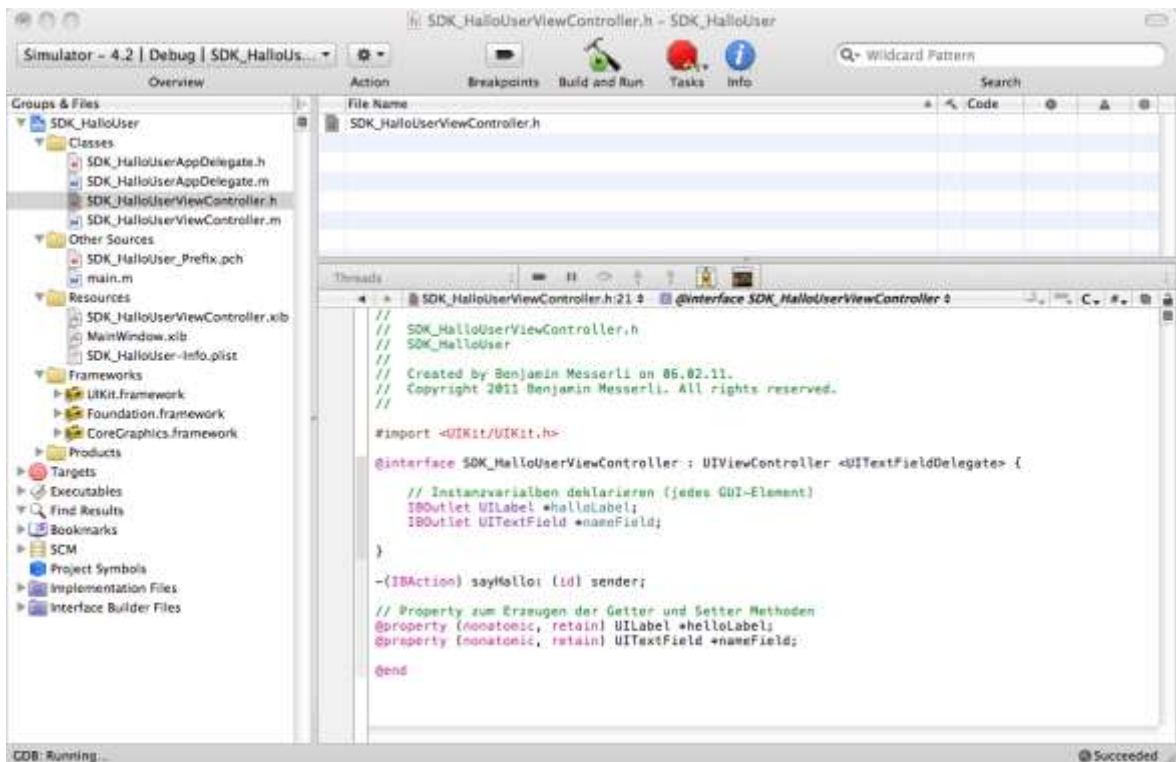


Abbildung 14: header Datei, Anlegung der grafischen Elemente

In der Implementationsdatei (SDK_HelloUserController.m) wird die Methode programmiert. Diese Methode wird beim Drücken des Buttons „sag Hallo“ aufgerufen.

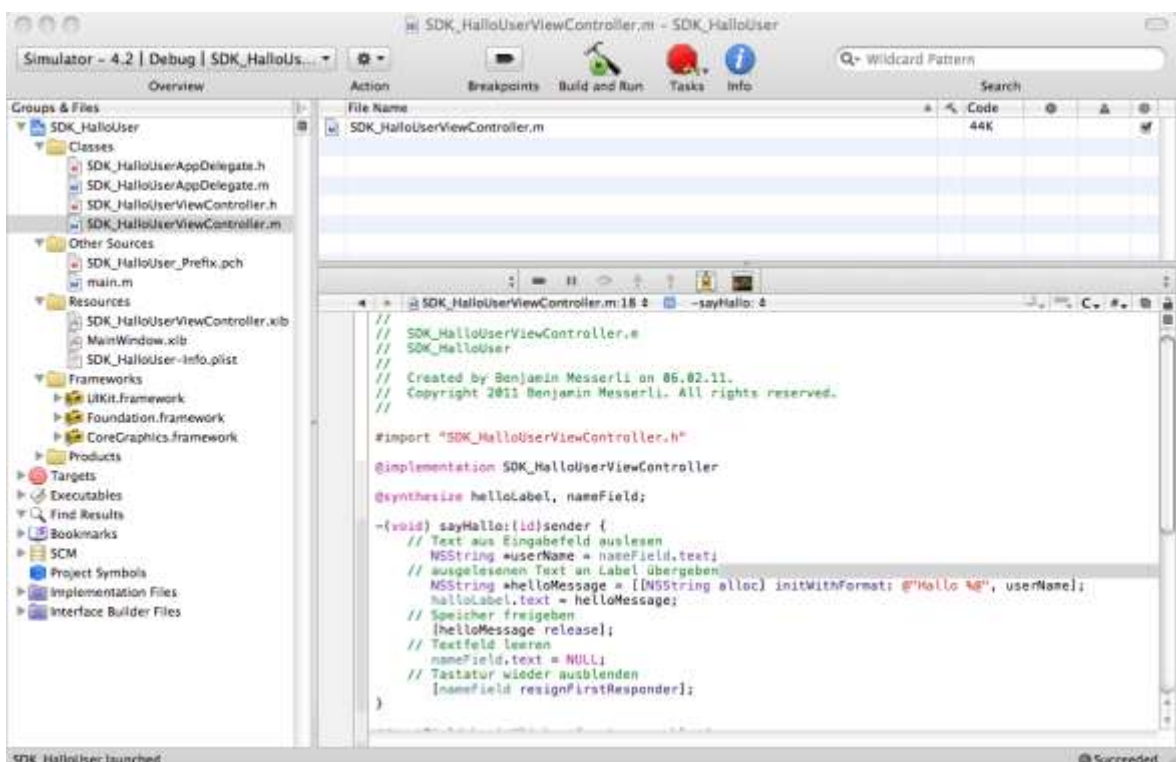


Abbildung 15: Printscreen Ausprogrammieren der Methode in der Implementationsdatei

Nun müssen im Interface Builder die im Programm angelegten Objekte mit den UI-Elementen verknüpft werden.



Abbildung 16: Printscreen Verknüpfung der angelegten Objekte mit den UI-Elementen

Nun kann das Programm im iOS Simulator getestet werden. Und es funktioniert!



Abbildung 17: Printscreen Test auf 1 dem Simulator



Abbildung 18: Printscreen Test auf dem Simulator 2

7 Werkbericht: Eigene App herstellen

Dieser Abschnitt enthält die gesamte Dokumentation der Entwicklung meiner eigenen iPhone App. Dabei habe ich versucht mich am kommerziellen Entwicklungsprozess von Applikationen zu orientieren. Eine erste Version der App iSchadau ist soweit fertig. Die App benötigt bestimmt weitere Wartung, also Updates. Ein Deployment auf dem App-Store erscheint derzeit noch nicht sinnvoll, da die Webservices zu instabil sind. In meiner Arbeit habe ich teilweise von erfahrenen Programmieren Hilfe bezogen, da der Programmieraufwand bereits den Rahmen einer Maturaarbeit weit sprengte und Teile der Programmierung mit meinen Erfahrungen schlicht als unlösbar schienen. Die Hilfe von meinem Vater als Arzt und Informatiker war mit dem Betreuer bereits vor Beginn der Arbeit vereinbart worden.

7.1 Ideenfindung

Als Schadaus Schüler ist man darauf angewiesen, dass man Zugang zu der Gymerhomepage hat. Fast alle Informationen können auf der Homepage aufgefunden werden. Da ich häufig auf die Homepage zugreife dachte ich mir, dass es praktisch wäre, wenn ich dies alles von meinem iPhone tun könnte. Also die notwendigen Informationen mobil abrufen kann. So kam mir die Idee, ich entwickle selbst eine App für das Gymnasium Thun-Schadau. Schnell stellten sich mir die Fragen, wie entwickelt man eine App? Was benötige ich dazu? Was muss ich können? Ahnungslos begann ich mit diesen Fragen und Gedanken meine Arbeit.

Als ich mir Klaren war, dass ich eine iPhone App für das Gymnasium entwickle, fragte ich mich, welche Funktionen meine App enthalten soll. Wenn ich auf die Homepage gehe, dann konsultiere ich meistens meinen Stundenplan. Weiter schlage ich die Termine nach, sehe mir die Ranglisten an und manchmal muss ich eine Telefonnummer eines Lehrers nachschauen. Den Webmail Exchange Server benutze ich praktisch nie, da alle meine Mails weitergeleitet werden. So sehen meine ersten persönlichen Überlegungen bezüglich des Funktionsumfangs aus:

- Bistro mit Menuplan
- Terminkalender
- Anreisemöglichkeiten und Fahrplan (STI,BLS)

- Sport(Ranglisten etc.)
- Events
- Evt. Abrufen Schadau Webmail
- Stundenplan
- Lehrerverzeichnis(E-mail, Telefon)
- Wetter in Thun

Da ich die Entscheidung heikel fand, entschied ich mich eine interne Umfrage zu erstellen.

7.2 Benutzerumfrage

Ich habe intern eine Benutzerumfrage am Gymnasium Thun-Schadau gestellt. Die Umfrage dient der Marktforschung, also ob eine App für das Gymnasium Thun Schadau überhaupt bei den potentiellen Benutzern Anklang findet und um den Funktionsumfang zu bestimmen. Zusätzlich wollte ich die Benutzer entscheiden lassen, welchen Namen meine Applikation tragen wird und somit mit Hilfe meiner Umfrage einen populären Namen für die App ermitteln. Zum Schluss soll ermittelt werden, wie viele Schüler Smartphones, beziehungsweise iPhones, besitzen. Meine Umfrage beabsichtigte auch eine Usergroup zu erstellen, von dem ich abgesehen habe.

Ich habe die Umfrage online mit Google Documents erstellt. Dies ist eine von Google zur Verfügung gestellte Plattform, wo ich als registrierter Benutzer einfach und praktisch Umfragen erstellen kann. Diese Umfrage wurde vom Sekretariat an alle Schüler gesendet und konnte freiwillig online ausgefüllt werden. Ich habe den Schülern Fragen zum Funktionsumfang gestellt

Von den rund 440 Schülern haben 221 Schüler sich an der Umfrage beteiligt. Die Umfrage hat folgende Resultate ergeben.

7.2.1 Fragen zum Funktionsumfang

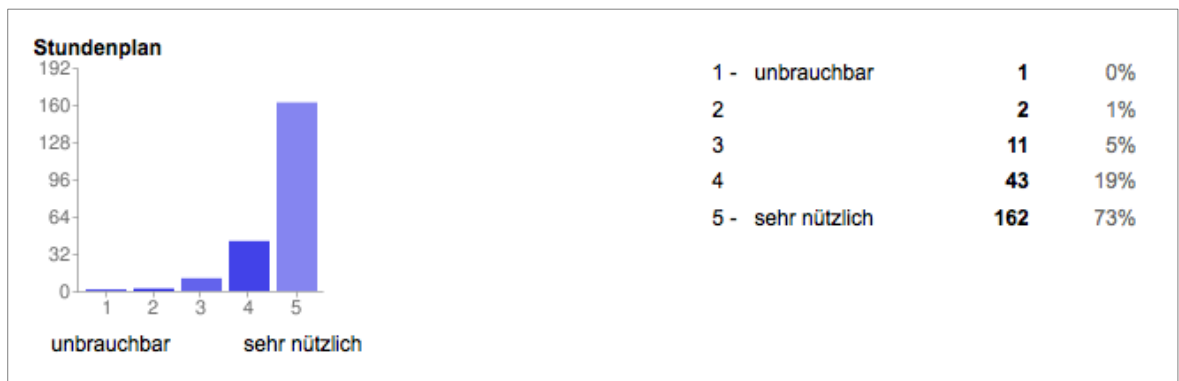


Abbildung 19: Resultate Benutzerumfrage Stundenplan Y-Achse: Anz. Schüler

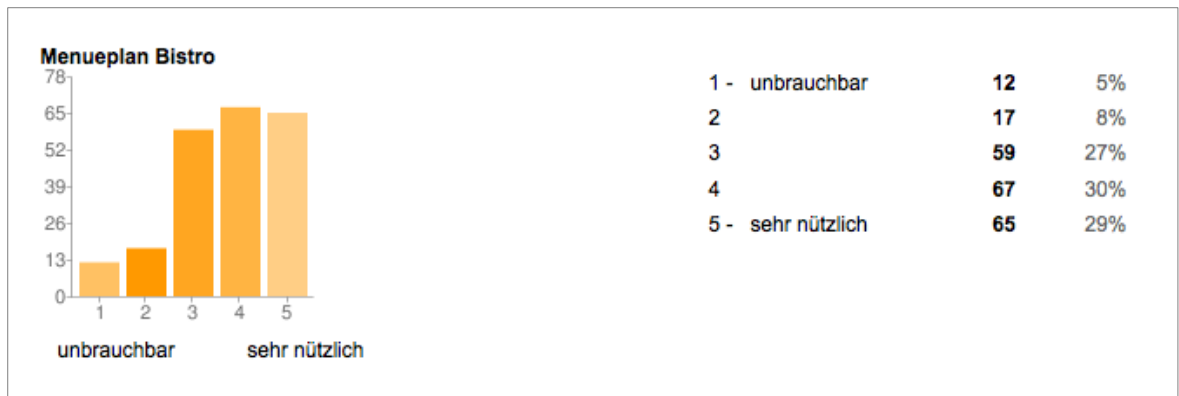


Abbildung 20: Resultate Benutzerumfrage Bistro, Y-Achse: Anz. Schüler

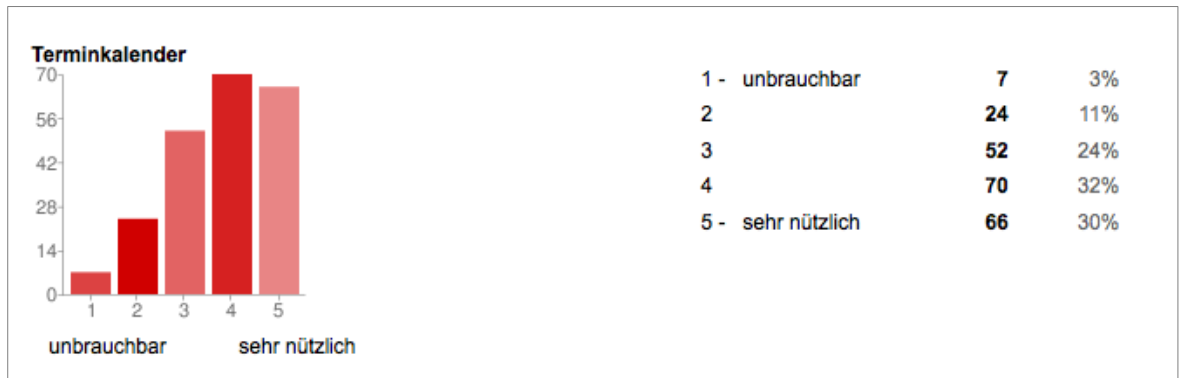


Abbildung 21: Resultate Benutzerumfrage Terminkalender Y-Achse: Anz. Schüler

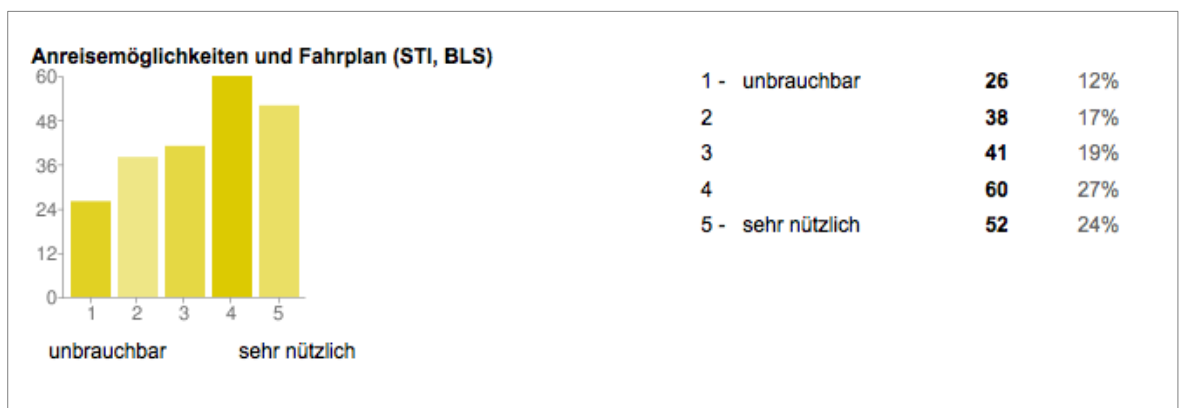


Abbildung 22: Resultate Benutzerumfrage Anreisemöglichkeiten Y-Achse: Anz. Schüler

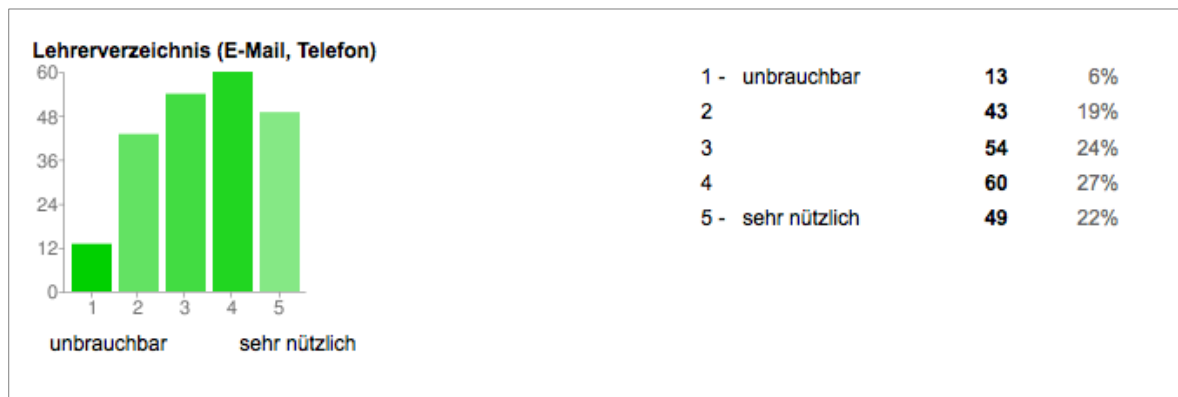


Abbildung 23: Resultate Benutzerumfrage Lehrerverzeichnis Y-Achse: Anz. Schüler

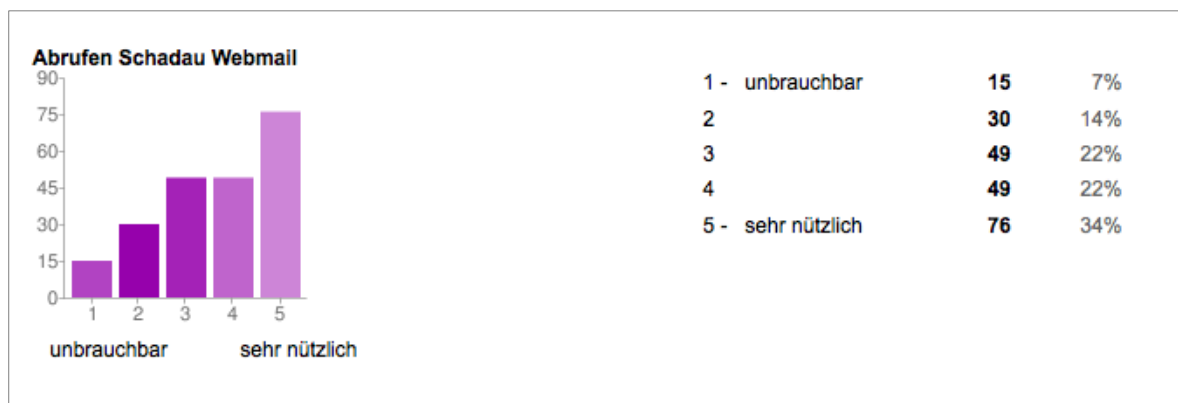


Abbildung 24: Resultate Benutzerumfrage Webmail Y-Achse: Anz. Schüler

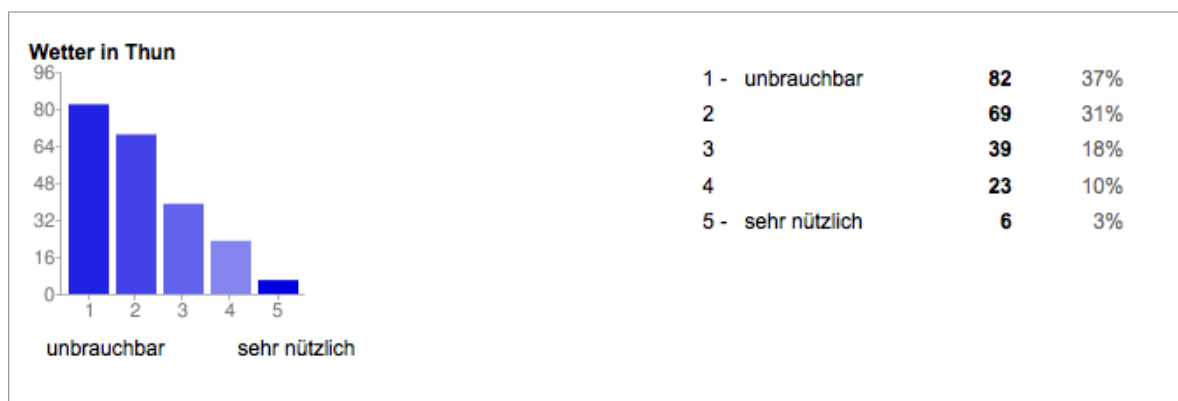


Abbildung 25: Resultate Benutzerumfrage Wetter Y-Achse: Anz. Schüler

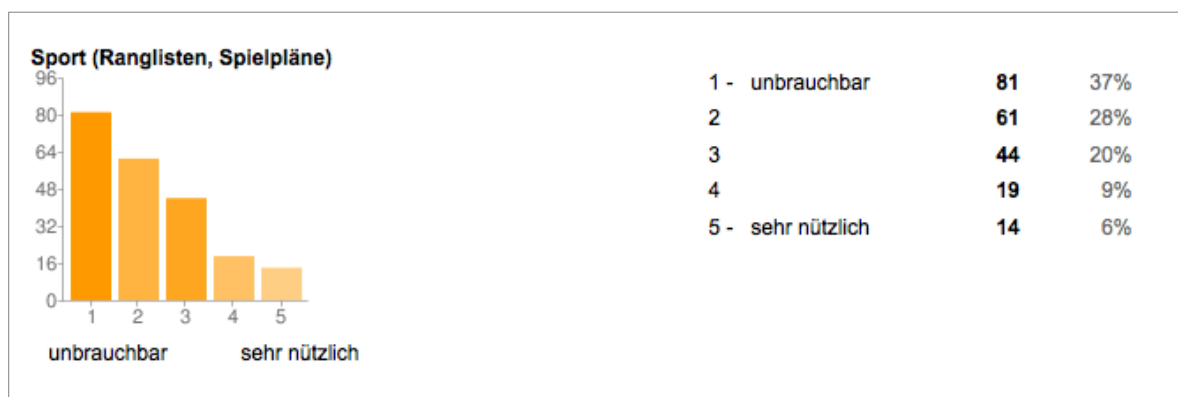


Abbildung 26: Resultate Benutzerumfrage Sport Y-Achse: Anz. Schüler

7.2.2 iPhone Besitzer

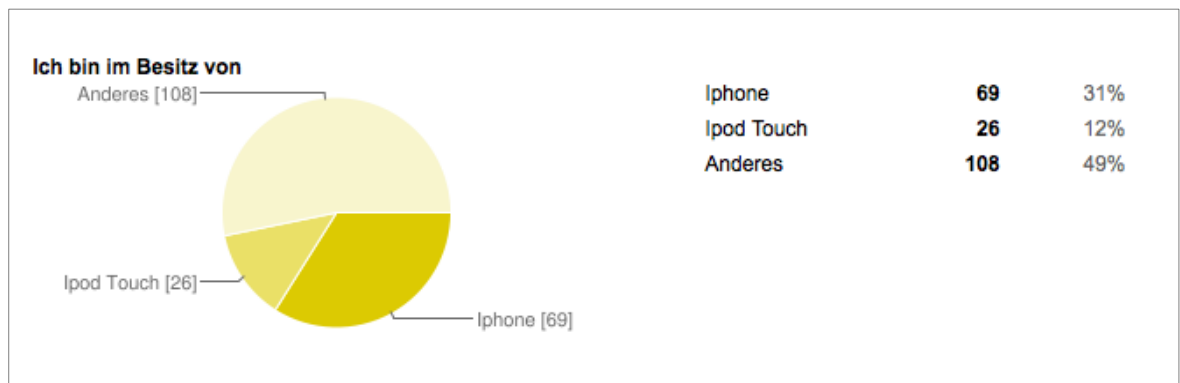


Abbildung 27: Resultate Benutzerumfrage iPhone Besitzer

7.2.3 Diskussion der Resultate

Rund 50 Prozent aller Schüler haben freiwillig die Umfrage ausgefüllt, was ein grosses Interesse an einem Smartphone Programm unterstreicht. Die Gewichtung der einzelnen Resultate decken sich ungefähr mit meinen Erwartungen. Absoluter Favorit ist der mobile Stundenplan. Am wenigsten gefragt ist das Wetter in Thun sowie die Sportresultate. Webmail, Terminkalender und Bistro sind über 50 Prozent der Stimmen bei vier oder mehr Punkten. Das Lehrerverzeichnis und die Anreisemöglichkeiten sind weniger gefragt.

Erstaunlich ist die Erkenntnis, dass rund ein Drittel der Schüler, welche sich an der Umfrage beteiligten, im Besitz eines iPhone sind.

Anhand dieser Umfrage habe ich den definitiven Funktionsumfang meiner Applikation festgelegt. Dieser habe ich anschliessend in das Application Definition Statement einfliessen lassen.

7.3 Application Definition Statement (ADS)

Aufgrund der Resultate der Benutzerumfrage, habe ich für meine App ein ADS zurechtgelegt.

Die Schadau-iPhone-Applikation soll optisch ansprechen und übersichtlich gestaltet werden. Wesentliche Funktionen sollen effizient und effektiv umgesetzt werden. Die Schadau-Applikation ist von einem Schadauschüler, für einen Schadauschüler.

Meine Applikation begleitet einen Schadaus Schüler durch den Alltag. Die WebUntis-Computer auf denen man den Stundenplan (mit den dazugehörigen Zimmernummern des jeweiligen Fachs) nachsieht, sind oftmals zu weit entfernt und viele Schüler wollen zeitgleich darauf zugreifen. Diese alltägliche Situation ist mühsam und zeitraubend. Die Schadau-Applikation löst dieses Problem, indem dieser Zugriff direkt auf dem iPhone durchgeführt werden kann. Dies spart wertvolle Zeit und Nerven.

Ein Schadaus Schüler führt sich tagtäglich Unmengen an Informationen zu Gemüte. Dabei gehen oft wichtige Termine vergessen und man kann sich als Schüler nicht optimal auf die jeweiligen Termine vorbereiten. Die Schadau-Applikation umfasst einen übersichtlichen Kalender, der Schadaus Schüler frühzeitig auf bevorstehende Termine aufmerksam macht und dies alles bequem vom iPhone aus.

Als weitere Gadgets sind praktische Funktionen vorgesehen. Ein Menüplan soll dem Schüler einen wöchentlichen Überblick geben. Ein Lehrerverzeichnis soll eine rasche Kommunikation zu spezifischen Lehrern ermöglichen.

Meine Applikation ist für Schüler und Schülerinnen des Gymnasiums und der Handelsmittelschule Thun-Schadau konzipiert. Die Applikation deckt optimal die wesentlichen Bedürfnisse der meisten Schüler ab, die 5 Tage die Woche zur Schule gehen, auf Verpflegung im Schulhaus angewiesen sind und natürlich ein iPhone besitzen.

7.4 Prototyping

Bevor ich meine Arbeit auf dem Mac begonnen habe, erstellte ich erstmals einen Prototyp meiner Applikation auf Papier. Dafür schneidete ich mir iPhone-grosse Karten aus und zeichnete darauf mein gesamtes GUI inklusive Navigation. Dies hat den Vorteil, dass man durch Programmierschwierigkeiten nicht in der Fantasie eingeschränkt ist und dass man nicht unnötig Schirme programmiert, welche man später verwerfen muss. Somit kann man später zielgerichtet ein optimales GUI erstellen.

7.5 Einrichtung Entwicklungsumgebung

Die Einrichtung der Entwicklungsumgebung war sehr praktisch und einfach. Ich musste mich dafür als iOS Developer auf der Developer Plattform von Apple

registrieren und die Developer Lizenz für 108 Franken lösen. Ich konnte mich mittels meiner Apple ID registrieren und hatte Zugang zum Download des iOS Software Developer Kit (iOS SDK) und weiteren praktischen Inhalten wie zum Beispiel Sample Codes und Tutorials von Apple. Als meine Lizenz bestätigt wurde konnte ich das iOS SDK herunterladen und auf meinem MacBook Air erfolgreich installieren. Es funktionierten gleich alle Tools einwandfrei.

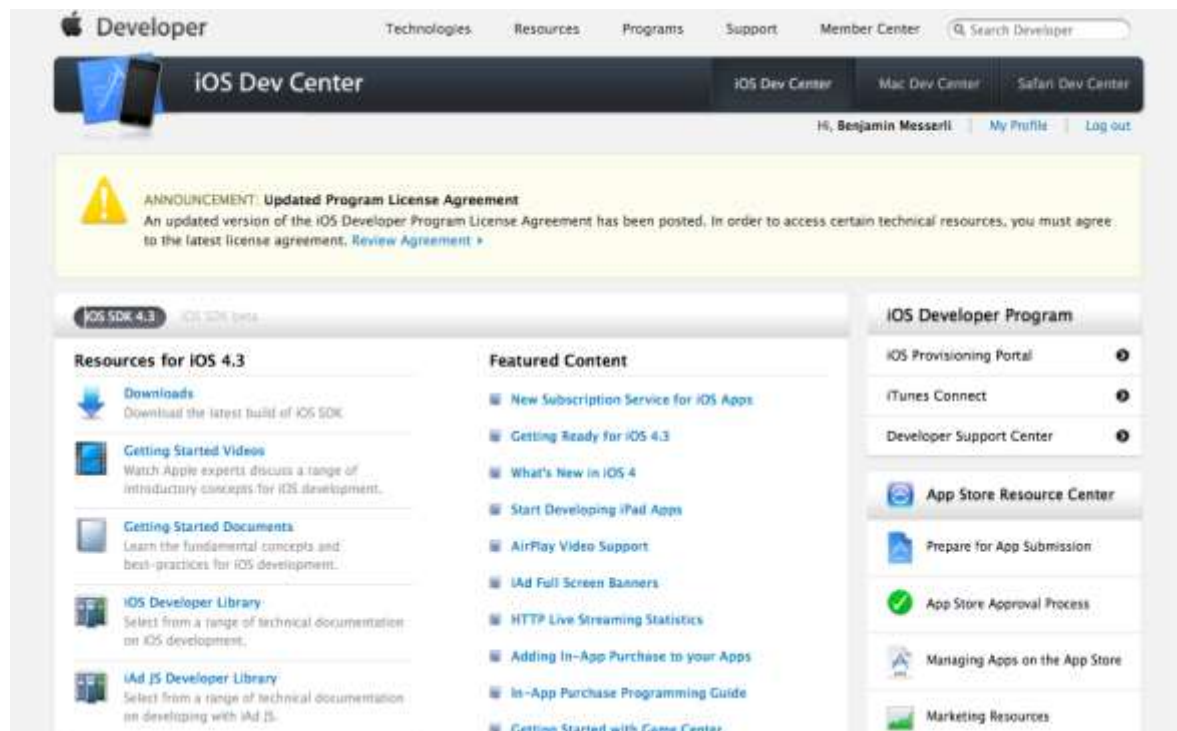


Abbildung 28: Printscreen iOS Developer Center Homepage

7.6 Erlernen des Handwerks

Ich besuche am Gymnasium seit einem Jahr das Ergänzungsfach Informatik, wo wir die Grundlagen der objektorientierten Sprache Java erlernten. Somit begann ich meine Arbeit mit dem Grundverständnis einer objektorientierten Programmiersprache. Im Rahmen meiner Arbeit musste ich zusätzlich Objective-C erlernen. Ich besorgte mir dafür vier Bücher und zwei Lern-DVD's. Ich schlug mich durch mehrere Kapitel der Bücher und sah mir die zwei mal siebenstündigen Lern-DVD's an. Die ganze Arbeit um Objective-C zu erlernen erschien mir nach getaner Arbeit ziemlich abstrakt und schwierig. So versuchte ich ein Testprojekt nach Anleitung der Lern-DVD nachzubauen.

Die Applikation trägt den Namen „Hello World“ und ist eine simple Applikation, die lediglich ein Textfeld zum eingeben eines Textes, ein Output Button und ein

Ausgabefeld enthält. Man kann in das Textfeld einen kurzen Text eingeben, den Button drücken und es der Text im Ausgabefeld angezeigt. Dies waren meine ersten Erfahrungen mit programmieren von Objective-C und Xcode. Schnell stellte ich fest, dass bei mir in der kurzen Zeit während dem Erstellen der App „Hello World“ mehr hängen geblieben ist, als beim mühseligen durcharbeiten von trockener, abstrakter Substanz wie die Lernbücher. Der grösste Lerneffekt bewährte sich auch bei weiteren Studien mit Learning-By-Doing am effektivsten. So entschied ich mich mit der Programmierung meiner Applikation zu beginnen.

In der folgenden intensiven Programmierphase haben sich folgende Methoden am meisten bewährt. Die Lern-DVD hat sich als äusserst hilfreich erwiesen, da anhand von konkreten Beispielen der Umgang mit Objective C und dem iOS SDK verständlich aufgezeigt wird. Diese Beispiele können auch nachgebaut und somit noch besser verstanden werden. Diverse gezeigte Programmiertechniken konnte ich auch in meine eigene App übernehmen. Weiter konnte ich mit dem gekauften Buch, Das Grosse iPhone Entwicklerbuch, wo die Codes im Internet heruntergeladen werden können, weitere Probleme lösen. Bei Problemstellungen, die ich weder mit Bücher noch DVD lösen konnte, durchforschte ich das Internet. Es gibt zu fast jedem Problem zahlreiche Anleitungen und Internetforen, die das gesuchte Problem behandeln. Oft konnte ich ohne die Anleitungen im Detail zu verstehen Teile übernehmen und in mein Programm einbauen. Beim durchforsten des Webs haben sich die von Apple zur Verfügung gestellten „Sample Codes“ am meisten bewährt. Die „Sample Codes“ sind ganze Apps zu zahlreichen Programmierfragen, die heruntergeladen und angeschaut werden können.

7.7 GUI-Navigation

Anhand meines GUI Prototypen auf Papier habe ich als erster Entwicklungsschritt ein GUI programmatisch zusammengestellt. Ein GUI (Graphical User Interface) beschreibt die Benutzeroberfläche und umfasst alle Elemente (UI-Elemente) zur Interaktion zwischen Benutzer und Programm. Der Prototyp auf Papier erwies sich als sehr gut entwickelt und ich konnte ihn mehrheitlich übernehmen. Ich habe eine sogenannte Tab-Bar Applikation erstellt. Dies ist eine Applikation, die über eine Auswahlleiste verschiedene Programme zusammenfasst. Um das Programm innerhalb der App zu wechseln muss lediglich ein anderer Tab gedrückt werden. Das wesentliche Element einer Tab-Bar ist, dass jedes Programm autonom läuft

und das beim Wechsel der aktuelle Zustand des Programms nicht verändert wird. Beim erstellen des GUI's sind die verschiedenen Programme vorerst noch leer und müssen in weiteren Schritten ausprogrammiert werden.

Nach dem erstellen des GUI's konnte ich die Funktionen Implementieren und schrittweise Programm um Programm erstellen. Während dem Entwickeln der Funktionen wurde das GUI schliesslich endgültig fertiggestellt.

Im folgenden Printscreen sind einige UI-Elemente mit Beschriftung dargestellt, damit folgendes eingehend Verstanden werden kann.

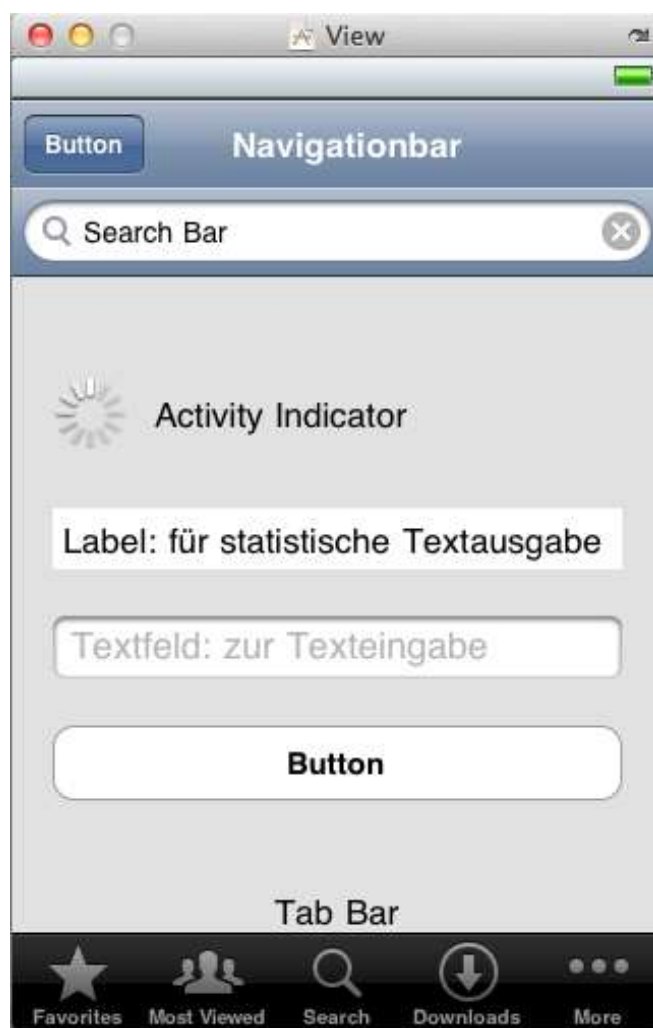


Abbildung 29: Printscreen grafische Elemente

Für das Design zog ich Hilfe von einem guten Freund bei. Ich entwickelte zusammen mit Tommaso Pace, Dipl. Arch Ba, HSLU, das GUI-Design meiner App. Ziel des Designs war ein eigen entwickeltes Design mit gewünschtem Wiedererkennungswert an den Schadau zu erstellen. Wir entschieden uns für ein durchgängiges Design mit ähnlichem Erscheinungsbild, welches durch die

Farbgebung an die Homepage anlehnt. Die Farben Rot, Weiss, Grau und Schwarz geben der App den erwünschten Wiedererkennungswert. Die Farbe Rot wird für den Navigationsbalken oben verwendet. Die Farbe Weiss wird als Beschriftung des Navigationsbalken und der Tabellenzeilen verwendet. Die Hintergründe der Infoschirme sind ebenfalls weiss, damit sie sich von der eigentlichen App abgrenzen. Die Farbe Grau dient als Hintergrund für die übrigen Bildschirme sowie als Hintergrund für die Tabellenzeilen. Die Schriftfarbe ist schwarz. Als Schrifttyp haben wir den zeitlosen Typ Helvetica gewählt. Als nächstes entwickelten wir ein Icon für meine Applikation. Das Icon ist mit einem roten Balken mit zwei Punkten aufweissem Hintergrund schlicht gehalten. Der Splashscreen ist ein vergrösserter Ausschnitt des Icons mit Beschriftung des Schadau's und einer Notiz, dass die App geladen wird.

Ich habe mich entschieden die Applikation nicht mehrsprachig zu entwickeln, da diese ausschliesslich für den deutschsprachigen Schadau konzipiert ist.

Ausserdem ist meine App nur vertikal angezeigt. Die App reagiert nicht auf ein drehen des iPhone. In einer Tab-Bar Applikation erschien mir dieses Feature als ungeeignet.

7.8 Beschreibung und Implementation der Funktionen

In diesem Abschnitt werden die verschiedenen Anwendungsfälle aufgelistet und kurz beschrieben. Zu jedem Programmteil sind ausserdem einige Sachverhalte näher erläutert, die für die Umsetzung eine entscheidende Rolle spielten. Eine detaillierte Beschreibung der einzelnen Anwendungsfälle befindet sich im Anhang. Die Anwendung an sich werden im nächsten Kapitel ausführlich beschrieben.

7.8.1 Stundenplan

Der Stundenplan wird über den Webservice WebUntis heruntergeladen und tabellarisch angezeigt. Es wird jeweils ein Tag des Schülers angezeigt. Der Stundenplan ist für die eingestellte Klasse und den eingestellten Schüler.

Der Stundenplan hat folgende Funktionen:

- Stundenplan des aktuellen Tages für eingestellte WebUntis Benutzer anzeigen
- Details einer Lektion anzeigen

- Einen Tag vorwärts blättern
- Einen Tag rückwärts blättern
- Datum aus Kalender auswählen
- Bildschirm aktualisieren (Refresh)
- Informationen über WebUntis anzeigen

Bemerkungen zur Implementation:

Die Hauptherausforderung beim Stundenplan war die Programmierung des Webservice von WebUntis. Ich habe als erstes die Schweizer Vertretung von WebUntis angeschrieben und eine Schnittstellenbeschreibung mit SOAP und XML erhalten. Zur Lösung dieses Problems habe ich eine der beiden Gratis-Anfragen bei Apple eingesetzt. Apple hat mir ein Beispielcode zugeschickt. In der Zwischenzeit habe ich auf Rückfrage vom Hersteller WebUntis in Oesterreich eine andere Schnittstellenbeschreibung mit JSON erhalten. Zusammen mit meinem Vater habe ich ein separates Testprojekt erstellt. Apple erlaubt nur asynchronen Download von Daten. Mit dem Beispiel-Code von Apple und aus dem Buch Rezepte konnten wir schrittweise eine Lösung finden.

Die zweite Schwierigkeit ist die Komplexität der WebUntis-Schnittstelle. Diese ist auf die Programmierung von Internetseiten ausgerichtet. Um den Stundenplan eines Tages abzurufen, müssen teilweise über hundert Abfragen erfolgen. Dadurch dauert die Aufbereitungszeit vergleichsweise lange. Hier wäre eine Smartphone optimierte Schnittstelle von Vorteil.

7.8.2 Bistro

Das Bistro lädt den Menüplan der aktuellen Woche vom Webservice von der SV Group herunter und wird in einer übersichtlichen Tabelle angezeigt.

Bistro enthält folgende Funktionen:

- Menüplan der aktuellen Woche der Mensa im Gymnasium Thun-Schadau anzeigen
- Bildschirm aktualisieren (Refresh)
- Informationen über die SV Group anzeigen

Bemerkungen zur Implementation:

Die grösste Herausforderung beim Bistro war die Programmierung des Webservice der SV-Group. Durch Kontaktaufnahme mit der SV-Group, welche die Mensa am Gymnasium Thun-Schadau führt, erhielt ich eine ausführliche Dokumentation eines sehr praktisch eingerichteten Webservices. Die Daten können vergleichsweise schnell heruntergeladen und geparkt werden. Mit der Lern-DVD und Büchern konnte ich die Daten synchron herunterladen, parsen und anzeigen. Die Programmierung des asynchronen Downloads und dem dazugehörigen Parsers, erreichte ich nur unter Aufsicht meines Vaters. Die Abfrage des Webservice erfolgt in zwei Schritten:

Zuerst wird ein URL abgefragt, aus dessen Antwort ich einen sogenannten Exporturl, also einen weiteren URL parsen musste. Danach musste ich diesen Exporturl abrufen und aus der Antwort die relevanten Daten parsen und anzeigen.

7.8.3 Termine

Termine ist eine Liste aller Termine des aktuellen Schuljahres des Gymnasiums und der Handelsmittelschule Thun-Schadau.

Der Tab Termine hat folgende Funktionen

- Termine des Gymnasiums Thun-Schadau in einer Tabelle anzeigen
- Alle Termine anzeigen
- Termine ab Heute anzeigen
- Details eines Termins anzeigen
- Vorwärts und rückwärts innerhalb eines Detail-Schirms blättern

Bemerkungen zur Implementation:

Die Terminliste ist keine Liste, welche laufend aktualisiert wird. Die Termine werden auch vom Gymnasium nicht laufend auf der Homepage aktualisiert, sondern es wird lediglich einmal im Jahr eine Excel Tabelle an die Internethosting Firma Internetgalerie gesendet, welche dann auf der Homepage angezeigt ist. Ich habe diese Excel Tabelle beim Sekretariat angefordert und mit dem Tool Windows Access ein XML File daraus erstellt. Danach habe ich das File eingelesen und in einer Tabelle anzeigen lassen. Der Vorteil der Funktion „Heute“ besteht darin, dass nur die aktuellen Termine angezeigt werden, und der Benutzer nicht immer durch die ganze Tabelle scrollen muss um den Aktuellsten Termin aufzufinden.

7.8.4 Lehrer

Der Tab Lehrer führt alle Lehrpersonen des Gymnasiums Thun-Schadau für das aktuelle Schuljahr auf. Für jede Lehrkraft sind E-Mail und Telefon hinterlegt. Mit diesen Angaben kann direkt Kontakt aufgenommen werden.

- Liste mit allen Lehrern anzeigen
- Einen Lehrer suchen
- Details für Lehrer anzeigen
- Lehrkraft anrufen
- E-Mail an Lehrperson senden

Bemerkungen zur Implementation:

Die Lehrer wurden wie zuvor die Termine über eine Excel Liste eingelesen. Diese Liste habe ich ebenfalls freundlicherweise vom Sekretariat des Gymnasiums erhalten. Auch die Lehrerliste wird nicht laufend aktualisiert und befindet sich auf der Homepage. Durch die Lehrerliste können die Lehrer schnell per Mail oder Telefon erreicht werden

7.8.5 Homepage

Die Homepage vom Gymnasium und der Handelsmittelschule Thun-Schadau wird automatisch geladen und angezeigt

Bemerkungen Implementation:

Die anfängliche Grösse des Homepage-Fensters passt nicht exakt auf den Bildschirm des iPhone. Die Ursache liegt in den unterschiedlichen Bildschirmformaten PC und iPhone.

7.8.6 Einstellungen

Unter dem Tab Einstellungen können diverse globale Einstellungen vorgenommen werden. Diese umfassen:

- Stundenplan: WebUntis Benutzer festlegen und zurücksetzen
- Bistro: Zurücksetzen

Bemerkungen zur Implementation:

Der Vorteil des Tabs Einstellungen liegt darin, dass alle Einstellungen global, einmalig und alles an gleicher Stelle definiert werden kann. Für Benutzereinstellungen bietet das iPhone einen speziellen Speicherort, einfachheitshalber sind die Einstellungen jedoch in einer eigenen Datei gespeichert. Somit ist ein guter Überblick gewährleistet und die Benutzerangaben müssen nur einmalig festgelegt werden

7.8.7 Impressum

In der App ist eine Funktion Impressum vorhanden, welche die aktuelle Version der App, den Entwickler und weitere Informationen auflistet.

7.9 Testing

Jede Funktion wurde während der Entwicklung ausführlich ausgetestet und schliesslich entschied ich mich aufwendige Schlusstests mit Testfällen aus Zeitgründen nicht mehr durchzuführen.

Ich durchging nur Ausnahmesituationen. Zum Beispiel wenn der Internetzugang fehlt oder der falsche Benutzer eingestellt ist. Während den Tests wurde mein Account auf WebUntis gesperrt und ich musste ihn wieder entsperren lassen. Ausserdem fiel zeitweise der Webservice der SV-Group aus. Durch Kontaktaufnahme konnte dieses Problem rasch beseitigt werden. Anhand der Tests der Ausnahmesituationen konnte ich noch die letzten Feinheiten anbringen und meine App stabilisieren.

7.10 Deployment

Ich habe die erste Version von iSchadau fertiggestellt. Als nächster Schritt käme die Publikation der App auf dem App-Store. Ich habe die App noch nicht publiziert, da der Webservice von WebUntis zu unzuverlässig ist, da dieser nicht für die Anwendung auf einem Smartphone optimiert ist. Zudem müssten die Termine und die Lehrer als Webservice zur Verfügung stehen und laufend abrufbar sein. Die App funktioniert gut, doch müssten gewisse Voraussetzungen erfüllt sein, damit die App als stabile Einheit publiziert werden kann. Diese Voraussetzungen beziehen sich auf die vorher genannten instabilen oder fehlenden Webservices.

7.11 Möglichkeiten zur Weiterentwicklung

Ich habe für die gesamte Entwicklung der App sehr viel Zeit und Nerven investiert. Trotz langer und harter Arbeit gibt es noch Verbesserungs- und Erweiterungspotential. Ich könnte mir dank des riesigen Rücklaufs meiner Umfrage durchaus vorstellen, dass bei den Schülern ein grosses Bedürfnis herrscht, meine App zu verwenden. Somit ist ein Interesse der Publikation im App-Store oder über das Gymnasium selbst gegeben. Ich wäre dazu bereit während meiner verbleibenden Schulzeit am Gymnasium die App zu pflegen, doch nach meinem Abgang müsste die Wartung von jemand anderem übernommen werden. Durch die Verwendung meiner App am Schulhaus würden bestimmt weitere Wünsche und Rückmeldungen eingehen, was den Charakter der App durch Updates verändere. Durch Kontaktaufnahme mit der SV Group wurde ich in dessen Hauptsitz in Dübendorf für eine Präsentation meiner App eingeladen. Allenfalls ergibt sich hier die Möglichkeit zusammen mit der SV-Group ein Anschlussprojekt umzusetzen.

8 Beschreibung der App

Wenn die App auf dem iPhone installiert ist, kann sie über das Icon gestartet werden.

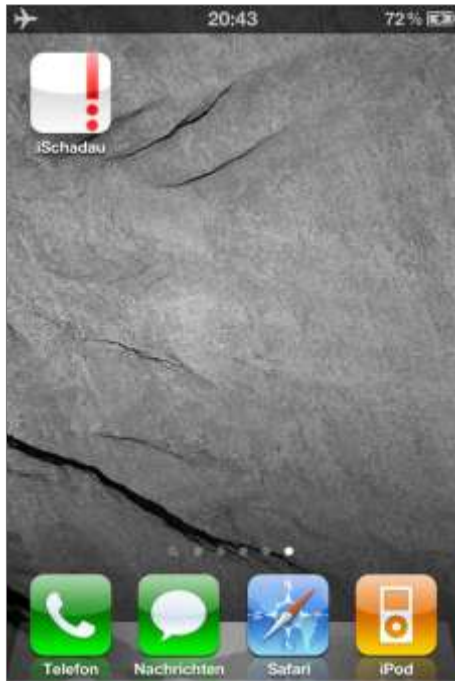


Abbildung 30: Icon iSchadau

Solange die App geladen wird, ist der Splashscreen zu erkennen.



Abbildung 31: Splashscreen iSchadau

Als erster Tab öffnet sich automatisch der Stundenplan. Der zuletzt geladene Stundenplan wird angezeigt und gleichzeitig im Hintergrund der aktuelle geladen.



Dienstag 16.08.2011		
07:45	Französisch	F
08:30	Mäder Regula	H209
08:40	Chemie Schwerpunkt	ChS
09:25	Jordi Daniel	H014
09:35	Chemie Schwerpunkt	ChS
10:20	Jordi Daniel	H014
10:40	Englisch	E
11:25	Sauter Kathrin	S103
11:35	Turnen und Sport Neigun...	TSN
12:20	Keller Martin	Pr

Abbildung 32: Tab Stundenplan iSchadau

Beim erstmaligen Starten der App muss zuerst der Benutzer unter Einstellungen erfasst werden. Die Buttons auf der Toolbar sind bis zur Erfassung eines Benutzers nicht bedienbar.

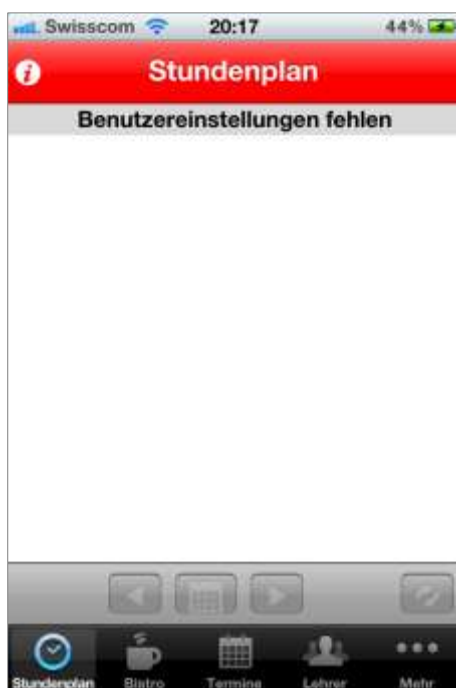


Abbildung 33: Benutzereinstellungen fehlen, iSchadau



Abbildung 34: WebUntis Benutzereinstellungen iSchadau

Ist der Benutzer eingestellt und die Buttons aktiviert, kann der Benutzer mit den Pfeil-Buttons den vorangehenden oder nächsten Stundenplan laden. Über den Kalender-Button kann der Stundenplan zu gewünschtem Datum geladen werden. Mit dem Refresh Button wird die aktuelle Anzeige aktualisiert.



Abbildung 35: Stundenplan, Kalender iSchadau

Klickt der Benutzer auf eine Tabellenzeile, werden die Details dieser Lektion eingeblendet.



Abbildung 36: Stundenplan, Details, iSchadau

Der Infobutton in der Navigationbar lädt einen WebUntis Infoschirm.



Abbildung 37: WebUntis Infoschirm, iSchadau

Wenn man auf den Tab Bistro klickt, wird automatisch der Menüplan der aktuellen Woche geladen, währendem der zuletzt geladene Plan angezeigt wird (falls vorhanden). Mit dem Refresh Button wird der Menüplan aktualisiert.



Abbildung 38: Tab Bistro, iSchadau

Mit dem Infobutton wird ein Infofenster mit Informationen zur SV Group angezeigt.



Abbildung 39: Infoschirm SV-Group 1 iSchadau



Abbildung 40: Infoschirm SV-Group 2 iSchadau

Beim klicken auf den Tab Termine werden alle Termine des aktuellen Jahres angezeigt. Klickt der Benutzer auf den Button Alle, werden alle Termine angezeigt. Klickt der Benutzer auf Heute, werden alle Termine ab dem aktuellen Datum angezeigt.



Abbildung 41: Tab Terminliste Alle iSchadau



Abbildung 42: Tab Terminliste Heute iSchadau

Klickt der Benutzer auf einen Termin, wird dessen Detailfenster geöffnet. Innerhalb des Detailschirms kann der Benutzer direkt zum vorangehenden oder nächsten Detailfenster schalten.



Abbildung 43: Termine, Details, iSchadau

Lehrer beinhaltet eine Liste mit allen Lehrern des aktuellen Jahres. Klickt der Benutzer auf das obige Suchfenster, kann ein Lehrer gesucht werden.



Abbildung 44: Lehrer Suchen, iSchadau

Wenn der Benutzer einen Lehrer auswählt, wird das Detailfenster des Lehrers eingeblendet. Auf diesem kann der Benutzer den Lehrer durch Drücken des Telefonbuttons anrufen oder durch Drücken des E-Mail Buttons eine Mail senden.

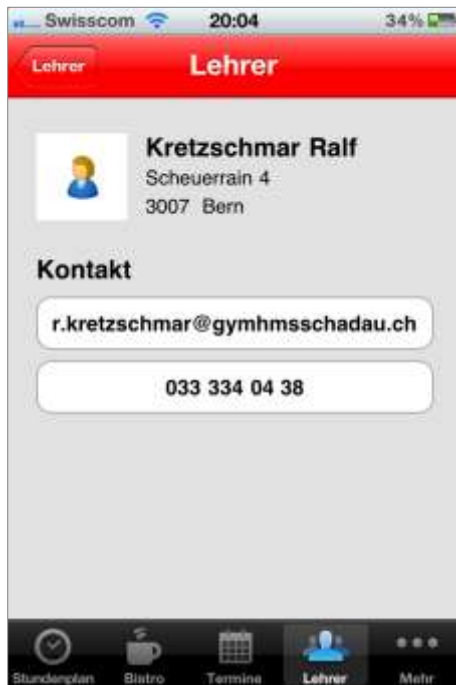


Abbildung 45: Lehrerdetail iSchadau

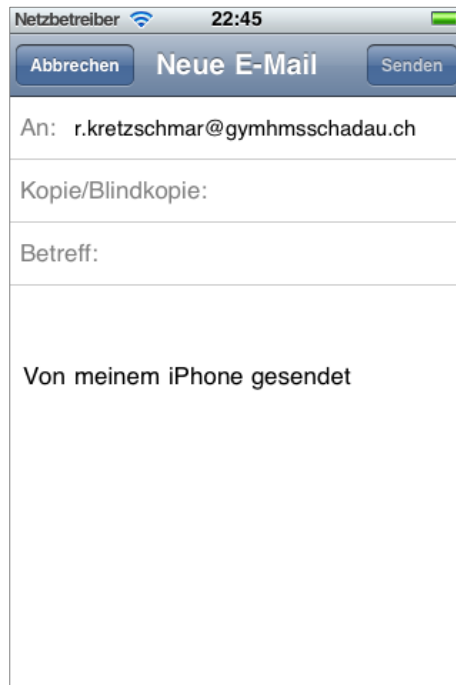


Abbildung 46: E-Mail an Lehrer senden iSchadau

Der Tab Mehr öffnet die weiteren zur Verfügung stehenden Programme meiner App.

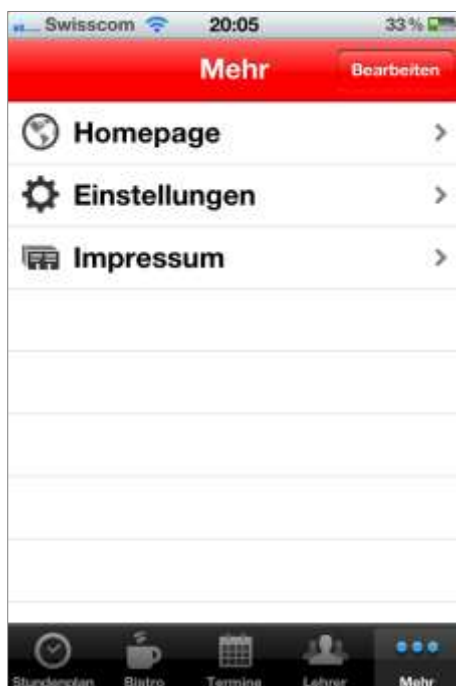


Abbildung 47: Tab Mehr, iSchadau

Drückt der Benutzer den Bearbeiten Button, kann er die Reihenfolge der verschiedenen Programme auf der Tab-Bar individuell konfigurieren.

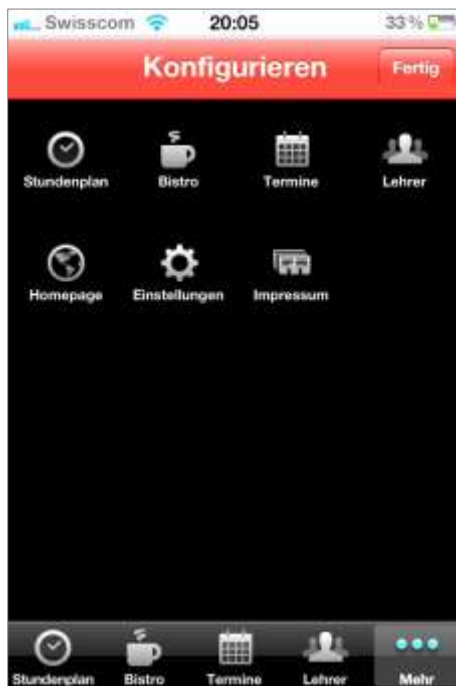


Abbildung 48: Konfigurieren der Tabs, iSchadau

Der Tab Homepage ladet automatisch die Homepage des Schadaus und zeigt sie an.



Abbildung 49: Tab Homepage, iSchadau

Der Tab Einstellungen bietet dem Benutzer verschiedene Einrichtungsmöglichkeiten. Der WebUntis Benutzer wurde bereits weiter oben eingeblendet. Mit dem Zurücksetzen Button beim Stundenplan und Bistro können alle geladenen Daten und Benutzerangaben gelöscht werden.



Abbildung 50: Tab Einstellungen iSchadau



Abbildung 51: Einstellungen Zurücksetzen iSchadau

Zuletzt ist noch ein Impressum aufgeführt.



Abbildung 52: Impressum 1 iSchadau



Abbildung 53: Impressum 2 iSchadau

9 Erfahrungswerte - Fazit

Von Grund auf eine eigene App entwickeln ist ein ehrgeiziges Vorhaben. Der Aufwand dafür ist entsprechend gross und sprengt den Rahmen einer gewöhnlichen Maturaarbeit deutlich. Meine Arbeitsstunden habe ich vorwiegend in die Entwicklung der App investiert.

Den Aufwand, eine Software zu entwickeln, habe ich grob unterschätzt. So musste ich viel Arbeit, entgegen meinen Zeitplan, nach hinten verschieben. Dies führte teilweise zu Stresssituationen. Trotz diesen schwierigen Umständen habe ich die Motivation für meine Arbeit nie verloren, was mich schliesslich auch zum Ziel führte. Erst bei Fertigstellung der App wurde mir bewusst, wie ich eigentlich genau hätte vorgehen sollen und wie ich unangenehme Stresssituationen hätte vermeiden können. Es zeichnete sich ab, dass ich mir schon vor der Entwicklung der App, sogar bereits vor der Implementation aller Funktionen den Zugang zu den gewünschten Daten hätte sicherstellen müssen. Ich habe dies zu spät getan, jedoch hatte ich riesiges Glück, da die Verantwortlichen von WebUntis und vor Allem die von der SV Group mir sehr rasch und zuvorkommend antworteten. Nur Dank dem freundlichen Kontakt konnte ich meine Arbeit wie gewünscht zu Ende führen.

Trotz meinen vertieften Programmier- und Softwarekenntnissen stiess ich deutlich an meine Grenzen. Die Softwareentwicklung ist harte Knochenarbeit, welche ich ebenfalls unterschätzte. Im Anbetracht dieser Faktoren will ich jedem Einsteiger der App-Entwicklung ans Herz legen, dass diese Arbeit mit sehr grossem Aufwand verbunden ist. Programmieren auf diesem Niveau kann nervenzerreissend sein. Ich geriet oft in Situationen, da ich mehr als zwei Tage benötigte, um eine Lösung auf ein Problem ausfindig zu machen. Die Lösung war entsprechend nur wenige Zeilen Code. Ausserdem sollte man sich genügend Reserven einrechnen, da in der Informatik immer wieder unerwartete Probleme und Stolpersteine auftreten können.

Ich habe durch diese Arbeit viele Erfahrungen gesammelt. Das Verfolgen eines Softwareprojekts über ein halbes Jahr ist anspruchsvoll. Diese Arbeit lehrt mich, dass Apps programmieren extrem komplex ist und ich achte nun dementsprechend die Apps mehr als zuvor. Im Bezug auf das Programmieren

habe ich gelernt, dass das Protokollieren innerhalb des Codes extrem wichtig ist, da man mit dem ein und selben Code doch ein halbes Jahr arbeitet. So kann man sich nicht Alles merken. Auch das lesbare Programmieren und eine logische Namensgebung habe ich zu schätzen gelernt. Das Verfassen einer Arbeit in diesem Umfang ist ebenfalls neu für mich. Das Erstellen dieser Arbeit benötigt im Anbetracht des grossen Umfangs zu Beginn grosse Überwindung, doch in Zukunft werde ich Arbeiten in diesem Ausmass mit weniger Hemmung angehen können.

Die Belohnung für all meine Bemühungen und Entsagungen ist eine überwältigende Freude über die funktionierende iPhone App iSchadau.

10 Quellenverzeichnis

1. <http://www.apple.com/de/>
2. <http://developer.apple.com/>
3. <http://de.wikipedia.org/wiki/Apple>
4. http://www.business-on.de/nordbaden/geraet-apple-smartphone-verkauf-iphone-smartphones-_id1408.html
5. <http://www.muluku.de/das-iphone-apples-erfolgsgeschichte/>
6. http://de.wikipedia.org/wiki/Apple_iPhone

Bücher und DVD's zu Objective-C und iPhone Programmierung:

- **Einstieg in Objective-C 2.0 und Cocoa inkl. iPhone-Programmierung**
Kai Surendorf, Markus Hardt
ISBN 978-3-8362-1310-3
- **Objective-C 2.0**
Anwendungen entwickeln für Mac und iPhone
ISBN 978-3-8273-6276-6
- **iPhone Entwicklung**
Anwendungen entwickeln mit dem iPhone SDK 3.0
ISBN 978-3-8273-2746-8
- **Objective-C 2.0: Anwendungen entwickeln für Mac und iPhone**
Stephen G. Kochan
ISBN 978-3-8273-2746-8
- **Das iPhone-Entwicklerbuch: Rezepte für
Anwendungsprogrammierung mit dem iPhone SDK**
Erica Sadun
ISBN 978-3-8273-2816-8
- **Entwickeln mit dem iPhone SDK**
Bill Dudney (Autor), Chris Adamson (Autor), Thomas Demmig
ISBN 978-3-89721-951-9

11 Abbildungsverzeichnis

Abbildung 1: Apple Logo.....	6
Abbildung 2: Produktpalette Apple	7
Abbildung 3: iPhone 4, schwarz	8
Abbildung 4: Printscreen aus dem App Store.....	9
Abbildung 5: Typischer iPhone Bildschirm mit Apps	10
Abbildung 6: Ein MacBook Air.....	11
Abbildung 11: Printscreen Erstellen von einem neuen Projekt in Xcode.....	17
Abbildung 12: Printscreen automatische Erstellung Grundausrüstung an Klassen und Frameworks	18
Abbildung 13: Printscreen Interface Builder	19
Abbildung 14: header Datei, Anlegung der grafischen Elemente.....	20
Abbildung 15: Printscreen Ausprogrammieren der Methode in der Implementationsdatei	20
Abbildung 16: Printscreen Verknüpfung der angelegten Objekte mit den UI-Elementen.....	21
Abbildung 17: Printscreen Test auf dem Simulator 1.....	22
Abbildung 18: Printscreen Test auf dem Simulator 2	22
Abbildung 19: Resultate Benutzerumfrage Stundenplan Y-Achse: Anz. Schüler ..	25
Abbildung 20: Resultate Benutzerumfrage Bistro, Y-Achse: Anz. Schüler.....	25
Abbildung 21: Resultate Benutzerumfrage Terminkalender Y-Achse: Anz. Schüler	25
Abbildung 22: Resultate Benutzerumfrage Anreisemöglichkeiten Y-Achse: Anz. Schüler.....	25
Abbildung 23: Resultate Benutzerumfrage Lehrerverzeichnis Y-Achse: Anz. Schüler.....	26
Abbildung 24: Resultate Benutzerumfrage Webmail Y-Achse: Anz. Schüler	26
Abbildung 25: Resultate Benutzerumfrage Wetter Y-Achse: Anz. Schüler	26
Abbildung 26: Resultate Benutzerumfrage Sport Y-Achse: Anz. Schüler	26
Abbildung 27: Resultate Benutzerumfrage iPhone Besitzer.....	27
Abbildung 28: Printscreen iOS Developer Center Homepage.....	29
Abbildung 29: Printscreen grafische Elemente.....	31
Abbildung 30: Icon iSchadau.....	38
Abbildung 31: Splashscreen iSchadau.....	38
Abbildung 32: Tab Stundenplan iSchadau	39
Abbildung 33: Benutzereinstellungen fehlen, iSchadau.....	39
Abbildung 34: WebUntis Benutzereinstellungen iSchadau	39
Abbildung 35: Stundenplan, Kalender iSchadau	40
Abbildung 36: Stundenplan, Details, iSchadau	40
Abbildung 37: WebUntis Infoschirm, iSchadau	41
Abbildung 38: Tab Bistro, iSchadau	41
Abbildung 39: Infoschirm SV-Group 1 iSchadau.....	42
Abbildung 40: Infoschirm SV-Group 2 iSchadau	42
Abbildung 41: Tab Terminliste Alle iSchadau.....	42
Abbildung 42: Tab Terminliste Heute iSchadau	42
Abbildung 43: Termine, Details, iSchadau	43
Abbildung 44: Lehrer Suchen, iSchadau.....	43
Abbildung 45: Lehrerdetail iSchadau.....	44
Abbildung 46: E-Mail an Lehrer senden iSchadau	44
Abbildung 47: Tab Mehr, iSchadau	44

Abbildung 48: Konfigurieren der Tabs, iSchadau	45
Abbildung 49: Tab Homepage, iSchadau.....	45
Abbildung 50: Tab Einstellungen iSchadau.....	46
Abbildung 51: Einstellungen Zurücksetzen iSchadau	46
Abbildung 52: Impressum 1 iSchadau.....	46
Abbildung 53: Impressum 2 iSchadau.....	46

12Anhang

Der Anhang besteht aus zwei Teilen. Das Kapitel 12.1 ist in der schriftlichen Arbeit aufgeführt. Die Kapitel 12.2 – 12.6 sind auf einer CD, welche dieser Arbeit beiliegt, einzusehen. Auf der CD befindet sich ausserdem das gesamte Xcode Projekt mit allen Quellcodes der App iSchadau.

12.1 Bewertungskriterien

12.1.1 Form der Arbeit

Produkt mit Werkbericht

Aspekte	Maximalpunktzahl
Inhalt <ul style="list-style-type: none">• Problemstellung• Aufbau, logische Gliederung• Vollständigkeit, Überblick• Themenbeschränkung, - vertiefung, Differenziertheit• Relevanz der Aussagen• Richtigkeit der Schlussfolgerungen• Kreativität, Eigenständigkeit• Illustration (Abbildungen, Tabellen u.a.)• Zusammenfassung / Abstract / Fazit• Literatur, Dokumentationsbasis• Fachspezifische Kriterien	55
Sprache <ul style="list-style-type: none">• Klarheit, Verständlichkeit• Textkohärenz und Logik der Formulierungen• Textsorten-Angemessenheit• Fachspezifische Wortwahl• Syntaktische Richtigkeit• Interpunktion, Rechtschreibung	10
Form - Werkbericht <ul style="list-style-type: none">• Einhaltung der formalen Anforderungen• Gliederung und Übersichtlichkeit• Unterscheidung Eigenes und Übernommenes• Zweckmässigkeit der Darstellung• Optischer Gesamteindruck	5

Erarbeitungsprozess <ul style="list-style-type: none"> • Einhalten der Rahmenbedingungen • Eigeninitiative und Selbstständigkeit des Arbeitens • Arbeitstechnik • Selbstkritische Haltung, Motivation • Teamarbeit (bei Gruppenarbeiten) • Arbeitsprozessbericht 	10
Werk <ul style="list-style-type: none"> • Design • Affordance – intuitive Bedienbarkeit • Funktionalität 	20
	Total — 100

Notenskala

Punkte	Note
100-95	6.0
94-85	5.5
84-75	5.0
74-65	4.5
64-55	4.0
54-45	3.5
44-35	3.0
34-25	2.5
24-15	2.0
14-5	1.5
4-0	1.0

Gewichtung der mündlichen Präsentation

Die mündliche Präsentation trägt zu **40%** zur Gesamtnote bei.

Unterschriften

Die folgenden Personen erklären sich mit den Bewertungskriterien, der Notenskala und der eidesstattlichen Erklärung einverstanden:

Der Betreuer: Ralf Kretzschmar

Benjamin Messerli

12.2 Codebeispiel Hello World

12.3 Webservice WebUntis

12.4 Webservice SV-Group

12.5 Beschreibung Anwendungsfälle

12.6 Zeitplan

13 Eidesstattliche Erklärung

Hiermit erkläre ich die vorliegende Arbeit selbständig und unter Angabe aller nötigen Quellen verfasst zu haben.

Spiez den 17. August 2011

Unterschrift

Benjamin Messerli