

HOCHSCHULE HEILBRONN / UNIVERSITÄT HEIDELBERG
Studiengang Medizinische Informatik



RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG
EXZELLENZUNIVERSITÄT



Diplomarbeit

Eine Anwendung zur Darstellung von Stundenplänen mit dem
Android Betriebssystem

Eingereicht von: Hekmat Al-Shenewi

Matrikelnummer: 164466

E-Mail: halshane@hs-heilbronn.de

16.09.2011

Referent: Herr Prof. Dr. Pfeifer

Co-Referent: Herr Prof. Dr. Heuzeroth

Zusammenfassung

Mobile-Anwendungen nehmen eine zentrale Rolle in der IT-Branche ein und gewinnen zunehmend an Bedeutung. Unzählige Softwarelösungen, wie beispielsweise in den Bereichen Schulung, Banking, Navigation oder Logistik, stehen den privaten und kommerziellen Nutzern als Mobile-Anwendung zur Verfügung.

An der Hochschule Heilbronn steht derzeit das Stundenplan-System „SPlan“ zur Verfügung, welches den Studierenden, Mitarbeitern und Dozenten einer Suchanfrage nach Stundenpläne und dem Raumbelungspläne des jeweiligen Semesters ermöglicht. Für eine benutzerfreundlichere und effektivere Nutzung dieses Stundenplan-Systems entstand die Idee, dieses System im Rahmen meiner Diplomarbeit als Mobile-Anwendung bereit zu stellen. Das für diese Diplomarbeit konkretisierte Ziel ist nun, das Stundenplan-System „SPlan“ in eine Mobile-Anwendung, basierend auf dem Android-Betriebssystem, umzusetzen. Mithilfe der mobilen-Anwendung kann der Anwender die Stundenpläne des betreffenden Semesters mit den entsprechenden Studiengängen und die jeweils zugehörigen Studentengruppen anzeigen lassen. Genauso können für den Raumbelungspläne des betreffenden Semesters, die Standorte und die jeweils zugehörigen Räume definiert werden. Optional bietet sich die Möglichkeit, Einzelbuchungen und Blockveranstaltungen der beiden Suchanfragen anzeigen zu lassen. Außerdem werden die Suchanfragen automatisch anhand der gewählten benutzer-spezifischen Eingaben als Profil in der Anwendung gespeichert. Somit lässt sich die Anwendung personalisieren.

Danksagung

Ich möchte mich ganz besonders bei Herrn Prof. Dr. Pfeifer bedanken, der mir die Chance gegeben hat, dieses Thema zu bearbeiten, ebenso für seine Unterstützung und die Betreuung meiner Diplomarbeit.

Ebenfalls möchte ich mich bei Frau Katrin Göttke und Frau Barbara Sterzenbach ganz herzlich für die sprachlichen Korrektur- und Formulierungsvorschläge zu meiner Diplomarbeit bedanken.

Inhaltsverzeichnis

Abkürzungsverzeichnis	6
1. Einleitung.....	7
1.1. Motivation	7
1.2. Ziel dieser Arbeit	8
1.3. Aufbau der Arbeit.....	8
2. Grundlagen	9
2.1. Android.....	9
2.1.1. Android-System-Architektur	9
2.1.2. Android-Anwendungs-Komponenten	10
2.1.3. Aufbau der Android Anwendung.....	13
2.1.4. Lebenszyklus einer Activity.....	15
2.1.5. Weiterführende Quellen zu Android OS	17
2.2. Webservice	18
2.2.1. Hessian	18
2.3. Technisches Umfeld / Entwicklungswerkzeuge	20
2.3.1. Development Kits	20
2.3.2. Eclipse	20
2.3.3. Android-Plattform	20
2.3.4. Eclipse-Plug-Ins.....	21
2.3.5. SQLite Database - Browser	22
2.3.6. Visual Paradigm	22
2.3.7. Together	22
2.3.8. Paint.Net.....	22
2.3.9. Online Umfrage Dienst	22
2.3.10. Schnittstellen des Online Dienstes „SPlan“	23
3. Anforderungsanalyse.....	24
3.1. Anforderungen an die Benutzeroberfläche.....	24
3.2. Funktionale Anforderungen an die Applikation	24
3.3. Use-Case-Diagramm	26
3.3.1. Suche nach Stundenplan	27
3.3.2. Suche nach Raumbellegung	28
3.3.3. Suche nach freien Räumen.....	29
3.3.4. Profile verwalten	30
3.3.5. Profil auswählen und Suche durchführen	31

4.	Konzeption und Implementierung	32
4.1.	Programmablaufplan.....	32
4.2.	Konzept der Benutzeroberfläche	34
4.2.1.	Hauptmenü / Suche nach Stundenplan.....	34
4.2.2.	Hauptmenü / Suche nach Raumbelugung.....	36
4.2.3.	Anzeige der Stundenpläne	38
4.2.4.	Profilliste / Profilverwaltung	40
4.2.4.1.	Profil anzeigen / Stundenplan Anzeige.....	41
4.2.4.2.	Profil bearbeiten.....	41
4.2.4.3.	Profil löschen	41
4.2.5.	Suche nach freien Räumen.....	42
4.2.6.	Anzeige der freien Räume	43
4.3.	Implementierung.....	44
4.3.1.	Klassendiagramme	44
4.3.2.	Sequenzdiagramme.....	49
5.	Qualitätssicherung.....	51
6.	Fazit und Ausblick.....	52
7.	Literaturverzeichnis und Referenzen	54
8.	Anhang.....	56
8.1.	Abbildungsverzeichnis.....	56
8.2.	Tabellenverzeichnis	57
8.3.	Online-Umfrage	58
8.4.	Testprotokoll	62
9.	Inhalt der CD.....	65

Abkürzungsverzeichnis

App	Applikation
Apk	Android Programm Format - Android Package
ADT	Android Development Tools
bzw.	beziehungsweise
DB	Datenbank
DVM	Dalvik Virtual Machine
GUI	Graphical User Interface
JAR	Java Archive
JDK	Java Development Kit
MS	Microsoft
OS	Operating System
PAP	Programmablaufplan
PC	Personal Computer
RIM	Research In Motion
RPC	Remote Procedure Call
SDK	Software Development Kit
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
UDDI	Universal Description, Discovery and Integration
UML	Unified-Modeling-Language
URL	Uniform Resource Locator
WSDL	Web Services Description Language
XML	Extensible-Markup-Language
z.B.	Zum Beispiel

1. Einleitung

1.1. Motivation

In der heutigen Zeit werden Smartphones immer beliebter. Sie bieten den Anwendern mehr Flexibilität, Einfachheit, Personalisierung und viele Kommunikationsmöglichkeiten. Außerdem kann man damit Aufgaben effiktiver bearbeiten, Zeit gewinnen und Routinearbeit reduzieren [1]. Die Internetnutzungsmöglichkeit ist ebenfalls ein wichtiger Punkt der Beliebtheit. Eine Studie des Managementberatungsunternehmens Accenture zeigt, dass ca. 14 Millionen Menschen in Deutschland das mobile Internet nutzen [2]. Im Vergleich zu 2008 hat sich die Zahl verfünffacht. Smartphones sind vielseitig und decken fast alle Bereiche und Funktionen ab, die auch in herkömmlichen Desktop-PCs oder Notebooks zu finden sind [3]. Alles in allem kann man mit dem Smartphone den allgemeinen Gebrauch erleichtern.

Zu den bekanntesten in der Reihe der Smartphones gehört Apple mit seinem iPhone und dem eigenen Apple-Betriebssystem iOS [4], das Google-BS Android für mobile Geräte, MS Windows Mobile 7, Nokia Symbian OS, Blackberry OS und HP Web OS. Dennoch ist es Google gut gelungen, sich an die Spitze dieser Wettbewerber zu setzen. Laut einer Studie des Marktforschungsunternehmens Gartner wird berichtet und behauptet: „Android ist mit Abstand weltweit das beliebteste Betriebssystem bei Smartphones. Symbian kommt auf Platz 2, iOS von Apple nur auf Platz 3“ [5]. Android OS ist eines der am meisten wachsenden Betriebssysteme für Anwendungen auf mobilen Endgeräten und führt den Markt seit dem vierten Quartal 2010 an. Es hat einen Anteil von fast 50 % am weltweiten Smartphone-Markt. Laut Prognosen soll es diesen Anteil bis Ende 2015 beibehalten [6]. Die hohen Nutzerzahlen der Smartphones mit einem Android OS beruhen auf der Tatsache, dass dieses Betriebssystem allen Smartphone-Herstellern frei zur Verfügung steht [7], abgesehen davon, dass geringe Lizenzkosten verrechnet werden [27]. Mit Android OS besteht zugleich die Möglichkeit, viele Endanwender zu erreichen.

Für eine benutzerfreundlichere und effektivere Nutzung des „SPlan“-Projektes der Hochschule Heilbronn entstand die Idee, eine Mobile-Anwendung für das Stundenplan-System zu entwickeln. Sie soll eine ortsunabhängige Suchanfrage der Stundenpläne ermöglichen. Die vorliegende Arbeit soll hierfür ein Konzept erstellen und das Vorhaben entsprechend realisieren.

1.2. Ziel dieser Arbeit

Ziel dieser Arbeit ist die Entwicklung und Implementierung einer Applikation auf Basis der Android-Plattform. Der praktische Nutzen besteht darin, Studierenden, Mitarbeitern und Dozenten der Hochschule Heilbronn einen personalisierten Stundenplan auf einem mobilen Endgerät grafisch verfügbar zu machen. Der Zugriff auf die Daten des Stundenplans erfolgt über einen Web Service der Hochschule Heilbronn.

Um die Anwendung personalisieren zu können, wird das Anlegen von Profilen ermöglicht. Die erforderlichen benutzerspezifischen Daten werden in einer Datenbank der Anwendung zur Verfügung gestellt.

1.3. Aufbau der Arbeit

Im nachfolgenden Kapitel werden die Grundlagen des Betriebssystems Android vorgestellt und ein Überblick über die verwendeten Entwicklungswerkzeuge und das technische Umfeld gegeben. Anschließend werden im dritten Kapitel die funktionalen und grafischen Anforderungen an die Applikation analysiert und diskutiert.

Im vierten Kapitel werden der Programmablaufplan, das Konzept der Benutzeroberfläche, die Implementierung sowie die fertiggestellte Applikation vorgestellt. Eine kurze Übersicht über den Test der Applikation gibt das fünfte Kapitel. Im sechsten Kapitel werden abschließend ein Fazit gezogen und ein Ausblick auf Erweiterungsmöglichkeiten der Stundenplanapplikation gegeben.

2. Grundlagen

Dieses Kapitel beschäftigt sich mit dem Aufbau des Betriebssystems Android, seiner Systemarchitektur und seinen Komponenten. Des Weiteren werden die verwendeten Entwicklungswerkzeuge vorgestellt.

2.1. Android

Android ist ein freies Betriebssystem des Unternehmens Google. Es umfasst alle Bibliotheken und Tools, die zum Entwickeln von Mobile-Anwendungen benötigt werden[8]. Diese Anwendungen werden in der Programmiersprache JAVA entwickelt und können beispielsweise in Smartphones oder Tablets-PCs eingesetzt werden.

2.1.1. Android-System-Architektur

Die System-Architektur von Android (vgl. Abb. 1) besteht aus vier Schichten. Die unterste Schicht ist die Hardwareschnittstelle, sie stellt alle notwendigen Treiber bereit. Diese basiert auf einem Linux-Kernel 2.6. Das Betriebssystem ist unter anderem für die Prozesssteuerung, die Kommunikation und das Dateisystem zuständig [8].

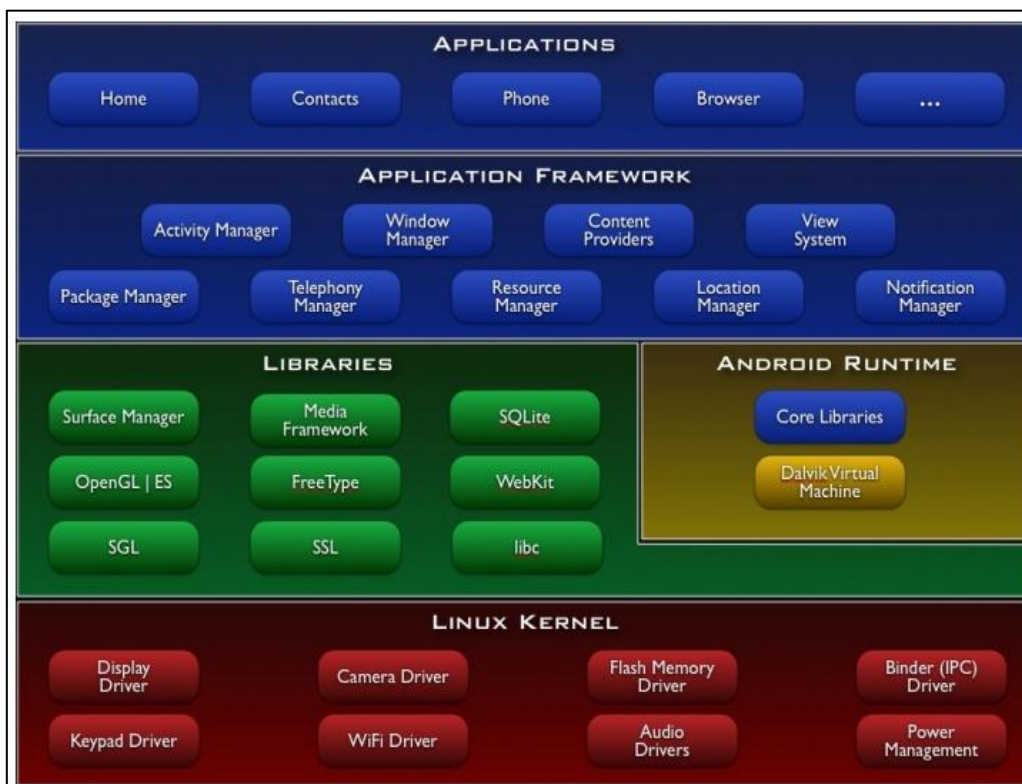


Abbildung 1: Android-System-Architektur [8]

In der nächsten Schicht liegen die Standardbibliotheken, sie stellen den Applikationen grundlegende Funktionen bereit. Beispielsweise umfasst die Standardbibliothek SQLite die Funktionen zur Speicherung und Abfrage von persistenten Daten. Die Programmierschnittstellen in der *Application Framework*-Schicht darüber ermöglichen den Entwicklern, eigene Anwendungen zu entwickeln [9, S.16]. Die Programmierungssprache ist JAVA, daher ist ein aktuelles Java-SDK für die Programmierung notwendig. „Kern der Laufzeitumgebung bildet die Dalvik Virtual Machine (DVM). Wird eine Android-Anwendung gestartet, so läuft sie in einem eigenen Betriebssystemprozess“ [9, S.15]. Die oberste Schicht umfasst alle Android-Anwendungen, die von Google oder von Dritt-Anbietern zur Verfügung gestellt werden.

2.1.2. Android-Anwendungs-Komponenten

Eine Android-Anwendung besteht aus mehreren Komponenten, die eng zusammenarbeiten. Im Folgenden sind die wichtigsten Komponenten aufgelistet [9, S.20, 21]:

- **Activity**

ist die meist verwendete Komponente bei einer Android-Anwendung. Sie ist für die Präsentation der Benutzeroberfläche zuständig. Eine Anwendung kann aus mehreren Activities bestehen, aber es muss mindestens eine Activity für eine Anwendung vorhanden sein. Jede Activity ist genau einer Bildschirmmaske zugeordnet. Der Inhalt einer Bildschirmmaske kann entweder in der Activity codiert oder in einem **View** als XML definiert sein. In einer Activity können die Aktionen des Anwenders entgegengenommen werden und weitergeleitet bzw. verarbeitet werden. Der Wechsel zwischen den Activities geschieht über **Intents**.

- **Services**

sind Dienste und Prozesse, die im Hintergrund laufen. Sie können ohne Benutzeroberfläche ausgeführt werden, wie z.B. um SMS zu empfangen oder Musik zu hören. Viele Services können parallel laufen.

- **Content Provider**

ist eine Komponente, die den Android Betriebssystem ermöglicht, bestimmte oder beliebig viele gespeicherte Daten auf einem Gerät auf mehrere Anwendungen zu verteilen und zu verwenden.

- **Broadcast Receiver**

stellt einen Kommunikationsweg zwischen den Anwendungen und dem Android OS her. Beispielsweise erhält man eine Benachrichtigung über einen Anruf, während eine Anwendung im Vordergrund läuft, oder das Betriebssystem meldet einen niedrigen Status des Akkus.

Weitere wichtige Komponenten einer Android-Anwendung sind Intents, Views und AndroidManifest.xml

- **Intent**

ist eine Übertragungsmöglichkeit von Daten zwischen den einzelnen Activities sowie für den Wechsel zwischen den Bildschirmmasken. Services oder Broadcast Receivers können über Intents aufgerufen werden.

- **View**

stellt den Inhalt der Oberfläche einer Activity dar, wie z.B. Texte, Buttons oder Farbe. Eine View kann entweder in einer XML-Datei hart codiert (vgl. Abb. 2) sein oder über eine Activity dynamisch erstellt werden.

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:fillViewport="true">
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <TableLayout android:id="@+id/myTableLayout"
            android:layout_height="wrap_content"
            android:layout_width="match_parent">
        </TableLayout>
        <TextView android:text="Letzte Änderung: "
            android:id="@+id/status"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">
        </TextView>
    </LinearLayout>
</ScrollView>
```

Abbildung 2: Code Ausschnitt der XML-basierten Datei View

- **AndroidManifest.xml**

In dieser Datei (vgl. Abb. 3) werden alle notwendigen Informationen über die fertige Anwendung eingetragen, z.B. Name der Anwendung, das Logo, die verwendete Android-Plattform, alle verwendeten Activities und Services. Man kann das Layout einzelner Activities definieren, beispielsweise vertikales oder horizontales Layout. Außerdem werden hier die Berechtigungen der Anwendung definiert, z.B. das Verbinden mit dem Internet oder der Zugriff auf bestimmte Dateien auf dem Gerät. Zudem kann angegeben werden, ob die Anwendung verschiedene Bildschirmauflösungen unterstützen soll. Zusammengefasst beschreibt sie das Verhalten und die Reaktionen einer Anwendung auf die Außenwelt.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="de.splan" android:versionCode="1" android:versionName="1.0">
    <supports-screens android:largeScreens="false"
        android:normalScreens="true" android:smallScreens="false"
        android:anyDensity="true" />
    <uses-sdk android:minSdkVersion="8" />
    <uses-permission android:name="android.permission.INTERNET"></uses-permission>
    <application android:icon="@drawable/logo_32x32"
        android:label="@string/app_name">
        <activity android:name="de.views.SearchTimetable"
            android:screenOrientation="portrait" android:label="@string/app_name"
            android:theme="@android:style/Theme.Black.NoTitleBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name="de.views.ShowTimetableHHN" />
        <activity android:name="de.views.ShowProfile"
            android:screenOrientation="landscape" />
        <activity android:name="de.views.SearchFreeRooms"
            android:screenOrientation="portrait"
            android:theme="@android:style/Theme.Black.NoTitleBar" />
        <activity android:name="de.views.ShowFreeRooms"
            android:screenOrientation="landscape" />
        <activity android:name="de.database.SaveData" />
        <activity android:name="de.tabs.TabBar"
            android:screenOrientation="portrait" />
        <activity android:name="de.tabs.TabBarafterwards"
            android:screenOrientation="portrait" />
        <activity android:name="de.tabs.TabBarBefore"
            android:screenOrientation="portrait" />
    </application>
```

Abbildung 3: AndroidManifest.xml (2)

2.1.3. Aufbau der Android Anwendung

Die Tabelle 1 gibt groben Überblick über den Aufbau und die Struktur einer Android-Anwendung [10] [11]. Sie besteht aus einer Baumstruktur mit Ordnern und Unterordnern, die wiederum Dateien enthalten (vgl. Tabelle 1).

Src	Enthält alle Packages und deren Java Klassen.
Gen	Enthält eine Java-Klasse (R.java), in der alle Ressourcen-IDs festgelegt werden. Sie wird automatisch während der Implementierung generiert und darf nicht verändert werden.
Android Platform	Gibt an, welche Android Plattform in der Anwendung verwendet wird.
Referenced Libraries	Hier werden alle externen Bibliotheken für das Projekt aufgelistet.
Assets	
Res/drawable	Ordner zur Speicherung aller Bilder sowie Icons, die vom Projekt verwendet werden. Bilder können mit verschiedenen Auflösungen gespeichert werden. Diese werden in separaten Ordnern angelegt. Res/drawable-hdpi: Hochoauflösung Res/drawable-ldpi: Mittlere Auflösung Res/drawable-mdpi: Geringe Auflösung
Res/layout	Enthält alle XML-Dateien, in denen Views definiert werden.
Res/menü	Enthält alle XML-Dateien, in denen MenüMenüs definiert werden.
Res/values	Enthält alle XML-Dateien, in denen Strings und Arrays definiert werden.
AndroidManifest.xml	Enthält wichtige Informationen über die Android Anwendung.

Tabelle 1: Struktur eine Android Anwendung [10] [11]

In der Abbildung 4 sieht man die Struktur meines Projektes. Man kann erkennen, dass diese Abbildung geringfügig von der vorherigen Tabelle abweicht, da man das Projekt beliebig mit weiteren Ordnern sowie Dateien erweitern kann.

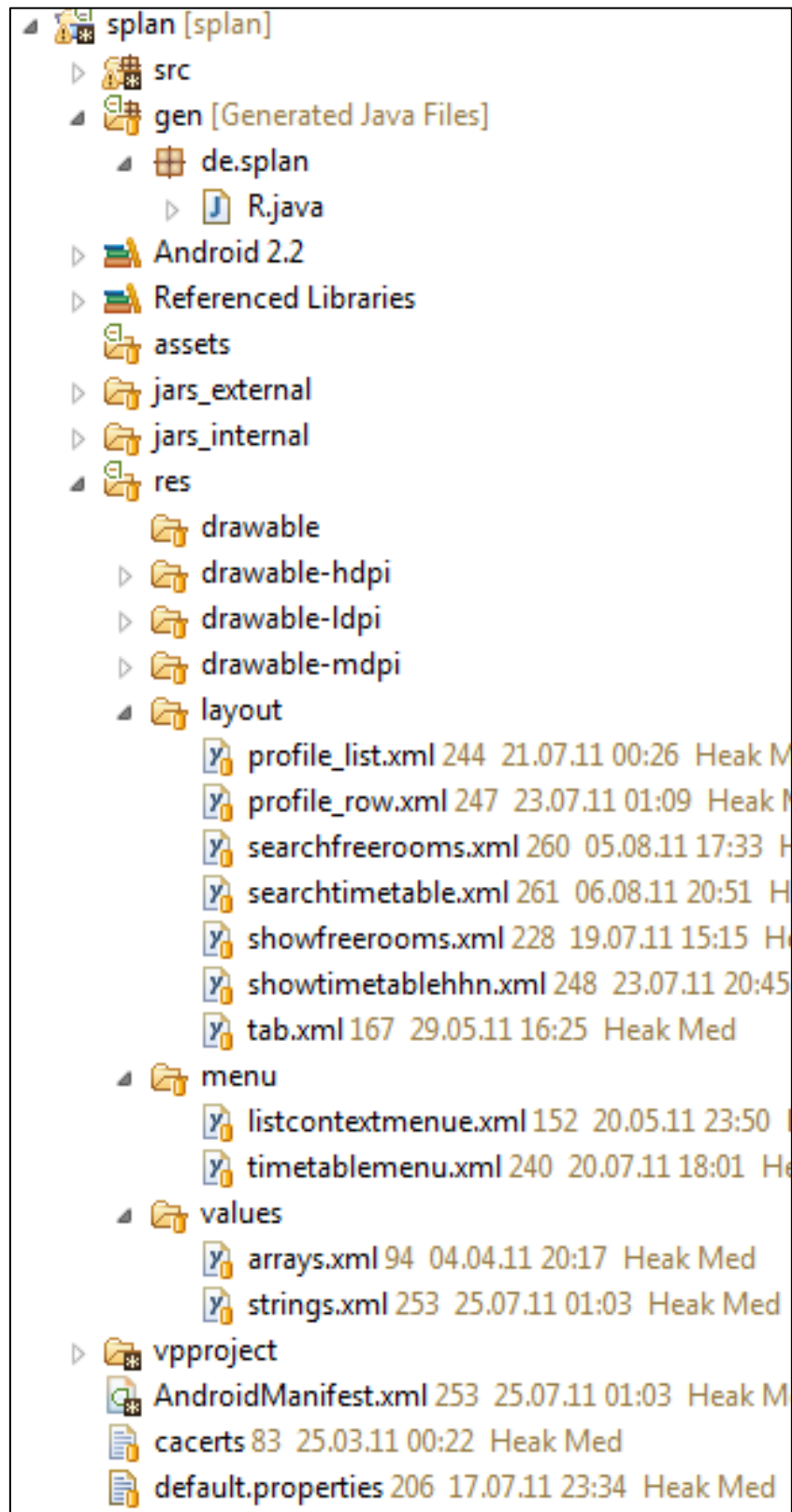


Abbildung 4: Aufbau einer Android Anwendung

2.1.4. Lebenszyklus einer Activity

Bereits im Kapitel 2.1.2 wurde die Activity als meist verwendete Komponente einer Android-Anwendung skizziert. Aus diesem Grund möchte ich sie hier genauer betrachten und den Lebenszyklus [12] [9, S.221-222] einer Activity (vgl. Abbildung 6) darstellen.

Sobald man eine Anwendung startet, wird die erste Activity aufgerufen. Sie ist in der AndroidManifest.xml definiert.

```
<application android:icon="@drawable/logo_32x32"
  android:label="@string/app_name">
  <activity android:name="de.views.SearchTimetable"
```

Abbildung 5: Codeausschnitt AndroidManifest.xml (2)

- Der Inhalt einer Activity wird durch die Methode **onCreate()** sichtbar gemacht. In der Methode werden alle Standardaufgaben deklariert, wie z.B. Views erstellen, Menüs initialisieren oder das Verbinden mit einer Datenbank.

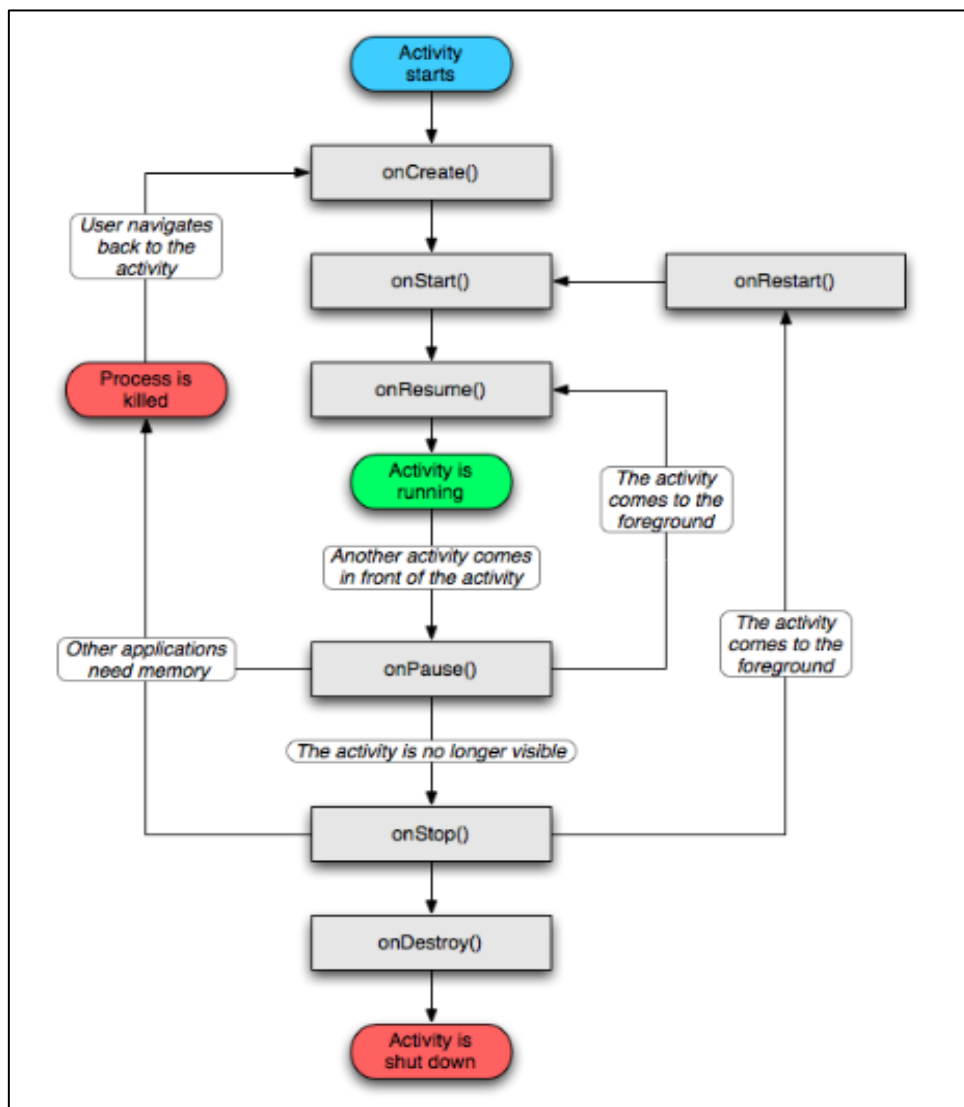


Abbildung 6: Lebenszyklus einer Activity [12]

- **onStart()** wird gleich nach der Methode **onCreate()** aufgerufen oder nach **onRestart()** wenn die Activity aus dem Hintergrund in den Vordergrund geholt wird.
- **onResume()** wird nach der Methode **onStart()** aufgerufen oder wenn davor die Methode **onPause()** aufgerufen wurde. Man könnte **onResume()** überschreiben, um Daten zwischendurch zu speichern.

```

@Override
protected void onPause () {
    super.onPause ();
    saveData ();
}

```

Abbildung 7: Codeausschnitt Methode onPause()

Nach dem Durchlauf dieser drei Methoden ist die Activity sichtbar und kann die Eingaben des Anwenders entgegennehmen.

- Sobald eine andere Activity gestartet wird, wird die Methode **onPause()** aufgerufen. Die erste Activity wird von der neuen Activity überlagert und ist nicht mehr sichtbar, jedoch im Hintergrund weiterhin aktiv.
- Erst wenn die Activity noch einmal aufgerufen wird, wird die Methode **onResume()** ausgeführt. Man könnte diese Methode überschreiben und beispielsweise beim Aktualisieren von Ergebnissen mit der Methode **onPause()** kombinieren.

```

@Override
protected void onResume () {
    super.onResume ();
    loadData ();
}

```

Abbildung 8: Codeausschnitt Methode onResume()

- **onStop()** wird aufgerufen, wenn eine andere Activity wieder fortgesetzt wird oder eine andere Activity gestartet wird.
- Will man eine gestoppte Activity wieder starten, werden **onRestart ()** und nachfolgend **onStart()** aufgerufen. Am Ende kann man mit der Methode **onDestroy()** die Activity beenden, um Speicherplatz freizugeben.

2.1.5. Weiterführende Quellen zu Android OS

Für weitere Informationen über das Android OS möchte ich auf die folgenden Bücher und Internet-Quellen hinweisen:

- <http://www.vogella.de/android.html>
- <http://developer.android.com/guide/topics/ui/menüs.html>
- Arno Becker, Marcus Pant. Android, Grundlagen und Programmierung, Dpunkt-Verlag, 1. Auflage 2009.
- Chris Haseman. Android Essentials. Apress, 1. Auflage 2008.
- Mark L. Murphy. Beginning Android 2. Apress, 1. Auflage 2010.
- <http://drdobbs.com/mobility/230600075>
- [http://de.wikipedia.org/wiki/Android_\(Betriebssystem\)](http://de.wikipedia.org/wiki/Android_(Betriebssystem))
- http://developer.android.com/guide/practices/screens_support.html

2.2. Webservice

Die Bezeichnung Webservice beschreibt eine Möglichkeit zur Zusammenarbeit zwischen verschiedenen web-basierten Anwendungen. Zur Kommunikation werden auf XML basierende offene Standards verwendet, wie z.B. SOAP, WSDL und UDDI. Über Internetprotokolle können die Daten ausgetauscht werden [13][14]. Webservices werden mit einem eindeutigen URI definiert. Somit ist es möglich, sie über entfernte Methoden unabhängig von der Plattform aufzurufen.

SOAP ist ein verbreiteter Standard, dennoch gibt es weitere Möglichkeiten für den Datenaustausch zwischen Webanwendungen, die zudem schneller und einfacher zu bedienen sind. Als Beispiel sei an dieser Stelle „Hessian“ genannt und nachfolgend beschrieben.

2.2.1. Hessian

"Hessian is a lightweight, self-describing binary RPC protocol. The Hessian binary web service protocol makes web services usable without requiring a large framework, and without learning yet another alphabet soup of protocols. Because it is a binary protocol, it is well-suited to sending binary data without any need to extend the protocol with attachments" [15] [16].

In den Abbildungen 9 und 10 werden verschiedene Protokolle anhand der Transportgeschwindigkeit von langen und kurzen Listen gemessen [17]. Hessian zeigt ein sehr gutes Ergebnis bzgl. der Performance bei großen und kleinen Datenmengen. Weiterhin bietet Hessian die Integration zwischen Java Klassen. Dies ermöglicht einen entfernten Methodenaufruf durch Implementierung von Schnittstellen. Schließlich hat man die Möglichkeit, eine modifizierte Drittanbieterversion (Hessian-Flamingo) für Android kostenlos zu verwenden, um eine optimale Kommunikation zur gewährleisten [18].

The following graph illustrates the response times of the various protocols for invocations returning smaller lists:

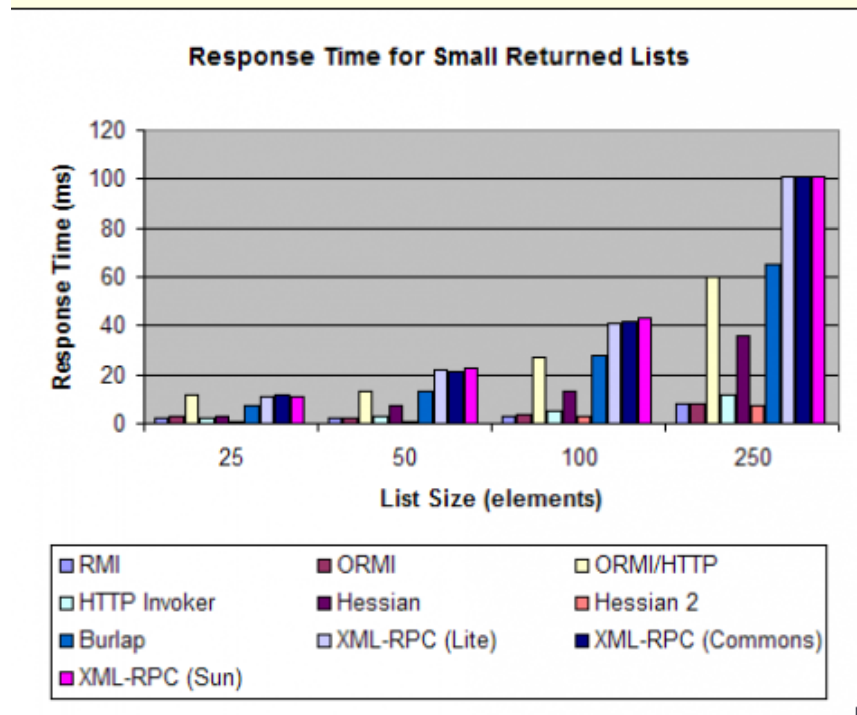


Abbildung 9: Vergleich der Transportgeschwindigkeit verschiedener Protokolle (1) [18]

The following graph illustrates the response times of the various protocols for invocations returning larger lists:

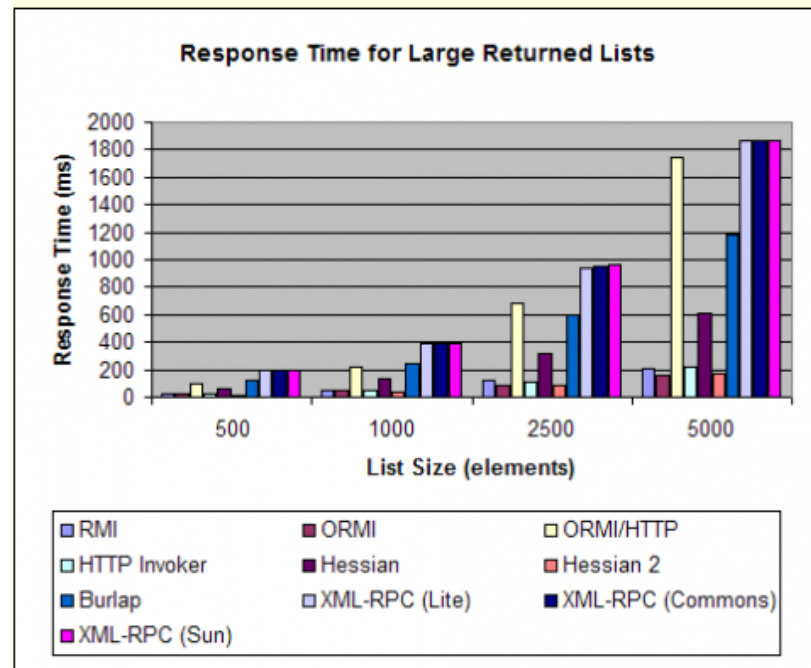


Abbildung 10: Vergleich der Transportgeschwindigkeit verschiedener Protokolle (2) [18]

2.3. Technisches Umfeld / Entwicklungswerkzeuge

In diesem Kapitel gebe ich einen kurzen Überblick über die Grundlagen meiner Diplomarbeit und erläutere die Entwicklungswerkzeuge, die ich zur Implementierung meiner Stundenplanapplikation verwendet habe.

2.3.1. Development Kits

- JAVA 6 JDK
enthält alle nötigen Bibliotheken für die Programmierung in der JAVA Sprache.
- Android SDK
ein wertvolles Entwicklungswerkzeug für die Programmierung beliebiger Smartphones, auf denen das Android OS installiert ist. Es beinhaltet die notwendigen Tools und Bibliotheken (vgl. Abb. 1)[9][19].

2.3.2. Eclipse

Im Rahmen meiner Arbeit wird „Eclipse for Java EE Developers 3.6“ zusammen mit der Android SDK und das Eclipse Plug-In ADT zur Entwicklung und zum Testen von Android-Applikationen eingesetzt [19]. Es lassen sich verschiedene Multi-Emulatoren mit unterschiedlichen Auflösungen und Bildschirmgrößen erstellen, um die entstandene Applikation zu testen, falls man kein Android-Gerät besitzt [9]. Auch die Finalisierung von Android-Applikationen, kurz auf „.apk“ endend, wird unterstützt. Zusätzlich können Zertifikate für die fertige Applikation signiert werden, um sie in den Android-Market hochzuladen [20]. Auf dieses Thema werde ich nicht weiter eingehen, da die fertige Applikation nur unter Android-Emulatoren getestet wird.

2.3.3. Android-Plattform

Die Android-Plattform sollte vor der Implementierung sorgfältig ausgewählt werden. Laut einer Statistik von Google sind derzeit Handys mit der Android Plattform 2.2 am weitesten verbreitet (bis zu 56 %) [21]. Daher empfiehlt sich die Programmierung mit dieser Version, denn alle Handys mit einer neueren Version als dieser werden ebenfalls unterstützt. So erreicht eine Android-Anwendung auf Basis von 2.2 über 81 % aller Android-Smartphones.

Platform	API Level	Distribution
Android 1.5	3	1.3%
Android 1.6	4	2.0%
Android 2.1	7	15.2%
Android 2.2	8	55.9%
Android 2.3 - Android 2.3.2	9	0.6%
Android 2.3.3 - Android 2.3.4	10	23.7%
Android 3.0	11	0.4%
Android 3.1	12	0.7%
Android 3.2	13	0.2%

Abbildung 11: Verteilung der verschiedenen Android Plattformen [21]

2.3.4. Eclipse-Plug-Ins

Nach der Installation von Eclipse wurden zusätzlich einige Plug-Ins installiert, um die Arbeit zu vervollständigen und zu vereinfachen.

- ADT

Android Development Tools (ADT) ist ein Plug-In für die Entwicklungsumgebung Eclipse. Es stellt eine Benutzeroberfläche mit einem Editor und einem Emulator zum Erstellen, Debuggen und Testen von Applikationen zur Verfügung.

- Subversion

Ist ein Versionsmanagementsystem, mit denen verschiedene Versionen von Dateien und Verzeichnissen verwaltet werden können.

2.3.5. SQLite Database - Browser

SQLite Database-Browser ist eine Software, mit der man SQLite-DBs anschauen und bearbeiten kann.

2.3.6. Visual Paradigm

Visual Paradigm ist ein sehr nützliches Tool für die Erstellung von UML-Diagrammen, wie z.B. Klassendiagramme und Aktivitätsdiagramme.

2.3.7. Together

Ist ebenfalls ein sehr praktisches Tool für die Erstellung von UML-Diagrammen.

2.3.8. Paint.Net

Aufgrund der verschiedenen Auflösungen der Handybildschirme wurden alle Bilder und Icons der Applikation mit Paint.Net bearbeitet und angepasst.

2.3.9. Online Umfrage Dienst

Für das Sammeln der Anforderungen wurde zuerst an eine herkömmliche papierbasierte Umfrage gedacht. Die Fragebögen werden gedruckt und an die Studenten verteilt, jedoch ist dies eine mühsame, zeitintensive und fehleranfällige Arbeit. Bei einem Online-Umfragedienst, wie z.B. <https://www.soscisurvey.de> [22], wird eine intuitive Web-Oberfläche angeboten, die beim Erstellen einer Umfrage behilflich ist. Es besteht die Möglichkeit, solche Umfragen unabhängig von Zeit und Ort durchzuführen. Entweder verschickt man einen Umfragelink per E-Mail, bindet ihn in die eigene Homepage ein, oder man veröffentlicht diese Umfrage online zum Ausfüllen im Internet-Browser. Die Umfrage wird anonym durchgeführt. Um den Datenschutz zu gewährleisten, wird diese Umfrage ohne Erfassung von persönlichen Informationen wie z.B. dem Namen des Teilnehmers durchgeführt. Denkt man an die Zukunft, kann man solche dieselbe Umfragen immer wieder verwenden. Ein wichtiger Aspekt ist auch die damit verbundene hohe Datenqualität auf Grund der vorgegebenen Auswahlmöglichkeiten.

2.3.10. Schnittstellen des Online Dienstes „SPlan“

Das Stundenplan-System „SPlan“ der Hochschule Heilbronn [23] ermöglicht allen Studierenden, Mitarbeitern und Dozenten eine Darstellung der Stundenpläne des jeweiligen Studiengangs und des Raumbelagungsplans. Auf diese Daten des Systems kann über Schnittstellen zugegriffen werden. Für den Zugriff auf die Schnittstellen des Onlinedienstes „SPlan“ benötigt man das Webservice-Protokoll Hessian in der Version 3.1.5 sowie weitere drei JARS des Projektes:

- SPlan-1.1.3
- Common-service-client-1.0.9
- Common-service-common-1.0.9

Diese Webservices beinhalten alle erforderlichen Klassen und Schnittstellen, welche für die Kommunikation und die Datenübertragung zwischen der geplanten Anwendung und dem Online-Dienst „SPlan“ notwendig sind. Jeder Webservice hat eine definierte URL, anhand derer die Verbindung aufgebaut wird. Beispielsweise ermöglichen die URL <https://splan.hs-heilbronn.de/splan/PlanningUnitService> den Zugriff auf die Semester-Daten, und die URL

<https://splan.hs-heilbronn.de/splan/OrgGroupService> den Zugriff auf die Studiengänge. In dem `package org.hhn.mi.splan.service` sind alle Webservices aufgelistet, die der Online-Dienst „SPlan“ zur Verfügung stellt. Die folgende Abbildung 12 zeigt einen Codeausschnitt für den Zugriff auf den Webservice PlanningUnitService (Semester).

```
public String[] getAllSemesterVisible() {
    String url3 = "https://splan.hs-heilbronn.de/splan/PlanningUnitService";
    PlanningUnitService servicePlanningUnit = null;
    try {
        servicePlanningUnit = (PlanningUnitService) factory.create(
            PlanningUnitService.class, url3);
    } catch (MalformedURLException e) {
        e.printStackTrace();
    }
    List<PlanningUnit> alleSemester = servicePlanningUnit.getAll();
}
```

Abbildung 12: Codeausschnitt Webservice Aufbau

3. Anforderungsanalyse

In diesem Kapitel werden die Anforderungen an die Benutzeroberfläche sowie funktionale Anforderungen an die Applikation aus Anwendersicht analysiert und ermittelt.

3.1. Anforderungen an die Benutzeroberfläche

Die Applikation soll den Studierenden eine Lösung bieten, den Stundenplan des Online-Dienstes „SPlan“ für Stunden- und Veranstaltungspläne der Hochschule Heilbronn auf einem mobilen Endgerät grafisch darzustellen. Der Bildbereich bzw. die Bildschirmmaske für die Suche wurde so konzipiert, dass er mit der Benutzeroberfläche des Online-Dienstes „SPlan“ im Wesentlichen übereinstimmt [23]. Der Bildbereich für die Anzeige der Stundenpläne wurde dagegen aufgrund der kleinen Bildschirmfläche des Smartphones völlig überarbeitet. Außerdem wurden von Zeit zu Zeit, nach Rücksprache mit Herrn Prof. Dr. Pfeifer, fortlaufend Verbesserungen an der Benutzeroberfläche durchgeführt. Detaillierte Informationen bzgl. der Benutzeroberfläche werden im Kapitel 4.1 beschrieben.

3.2. Funktionale Anforderungen an die Applikation

Um eine bestmögliche Auswahl für die wichtigsten Funktionalitäten des Anwenders zu treffen, wurde eine Onlineumfrage erstellt und ausgewertet [22]. Die Umfrage habe ich persönlich mit Studenten der Hochschule Heilbronn durchgeführt. An der Umfrage nahmen 30 Studenten teil. Als Ergebnis der Analyse und Auswertung dieser Umfrage wurde eine grafische Statistik (vgl. Abb. 13) erstellt, die eine Übersicht über die Wichtigkeit der ausgewählten Funktionalitäten gibt.

Diese Funktionalitäten werden im nächsten Kapitel anhand von Anwendungsfällen detailliert beschrieben. Es werden auch Funktionalitäten erläutert, die während meiner Arbeit zusätzlich hinzugekommen sind.

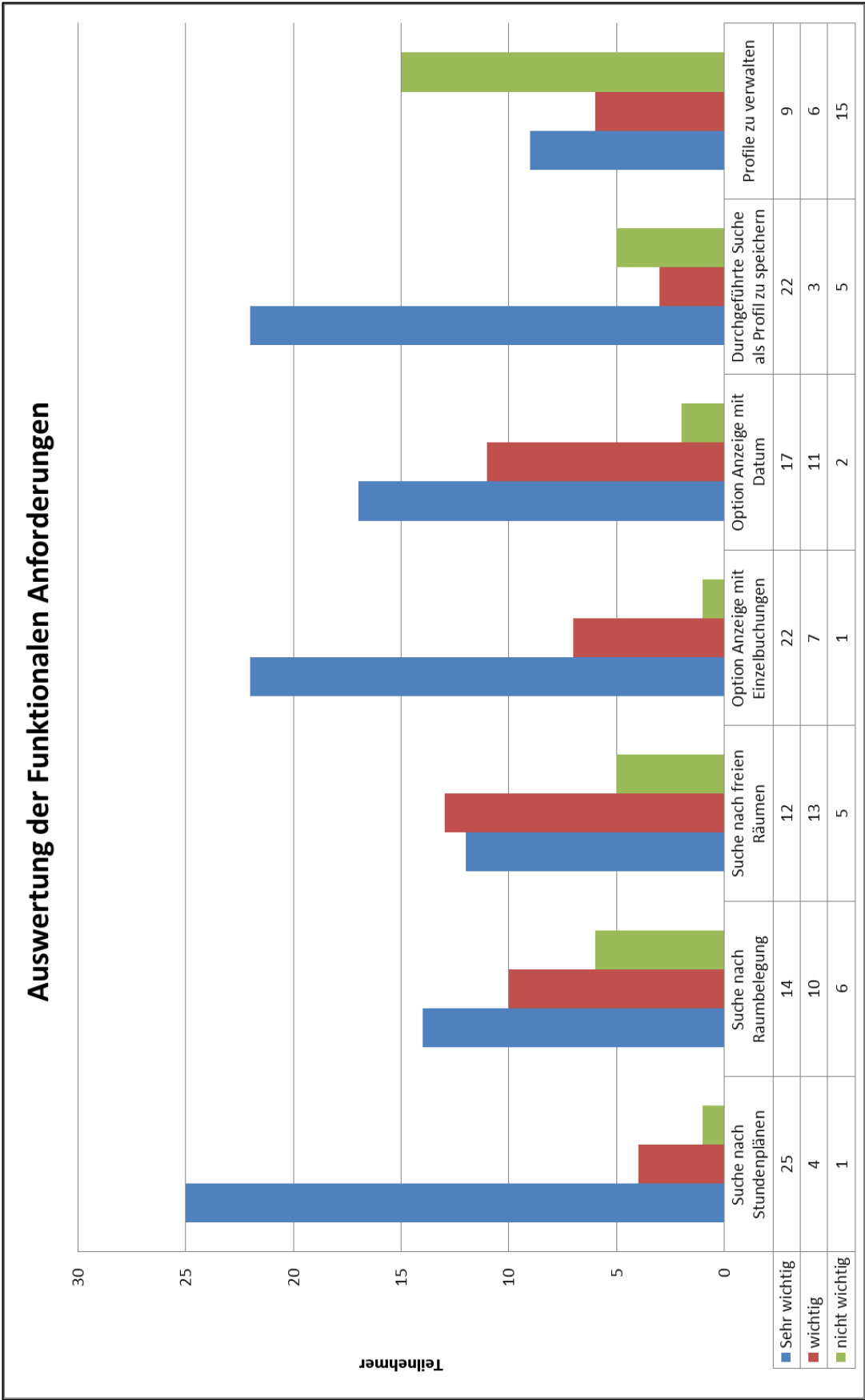


Abbildung 13: Auswertung der funktionalen Anforderungen

3.3. Use-Case-Diagramm

Das vorliegende Use-Case-Diagramm in Abbildung 14 zeigt einen groben Überblick über die Funktionalitäten, die dem Anwender zur Verfügung gestellt werden.

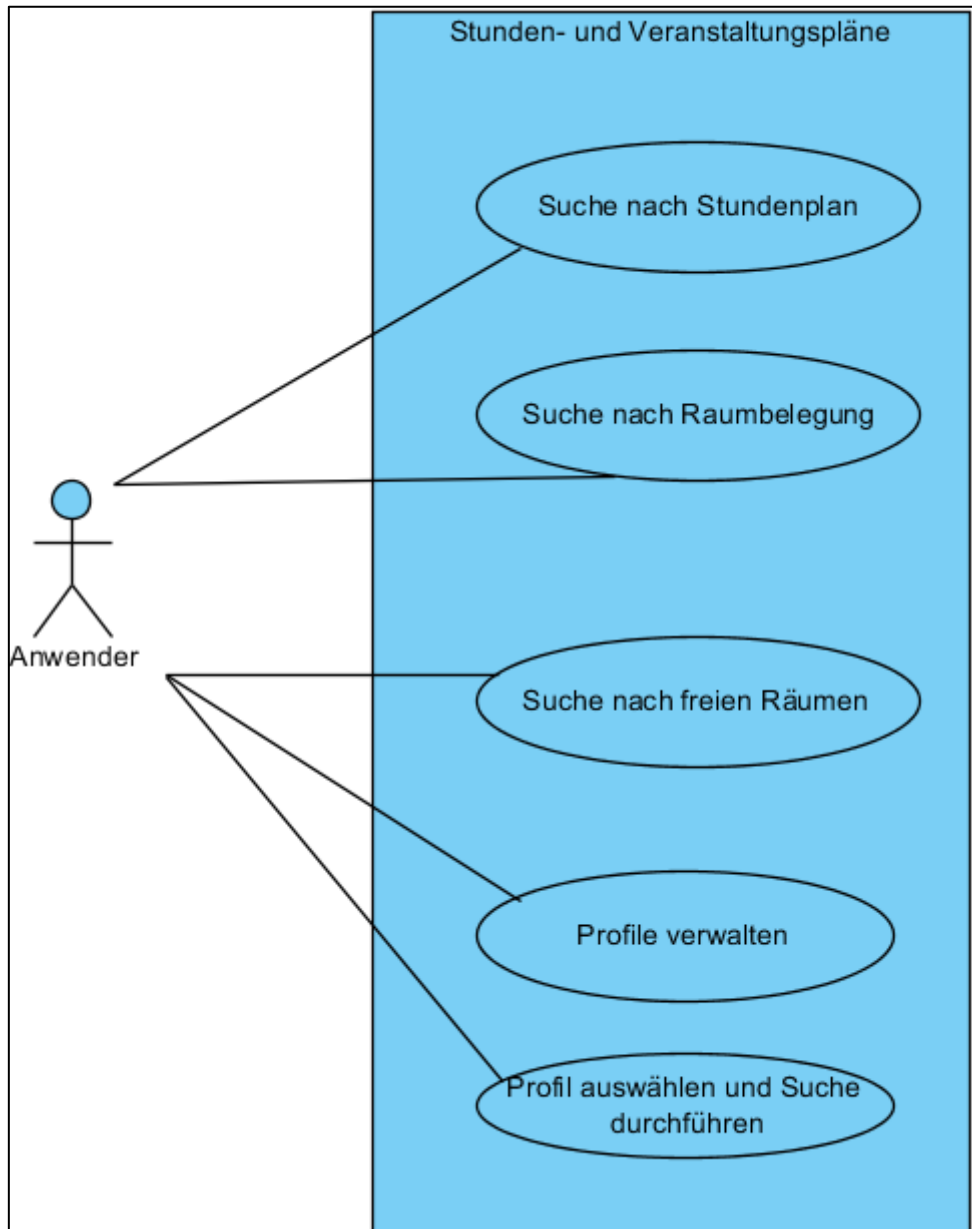


Abbildung 14: Use-Cases-Diagramm der Applikation

Anhand jedes Use-Case werden die Wünsche des Anwenders und das erwartete Ergebnis(se) genauer erläutert.

3.3.1. Suche nach Stundenplan

Beschreibung Use-Case	
Name	Suche nach Stundenplan
Kurzbeschreibung	Der Anwender kann seinen Stundenplan für das jeweilige Semester suchen und sich anzeigen lassen.
Akteure	Anwender
Vorbedingungen	Für das betreffende Semester müssen die Auswahlliste der Studiengänge und die jeweils zugehörigen Studentengruppen definiert sein.
Eingehende Informationen	Stundenplananforderung
Ergebnis(se)	Stundenplan gemäß Benutzereingaben anzeigen. Die Sucheigenschaften werden automatisch in der Datenbank als Profil gespeichert.
Auslöser	Anwender
Nachbedingungen	s. Ergebnis(se)
Ablauf	<ol style="list-style-type: none"> 1. Einen Studiengang auswählen. Alle Studentengruppen des ausgewählten Studiengangs werden in eine neue Auswahlliste geladen. 2. Unter „Studentengruppe“ ein Semester auswählen. 3. Es werden dem Anwender drei Optionen zur Verfügung gestellt, die mit „JA“ oder „NEIN“ ausgewählt werden: <ul style="list-style-type: none"> • Die Suche als Standard festlegen, d.h., sobald die Applikation neu gestartet wird, wird automatisch die Suche anhand der gespeicherten benutzer-spezifischen Eingaben ausgeführt und zeigt den Stundenplan an. • Anzeige „mit Datum“, um Blockveranstaltungen anzuzeigen. • Anzeige von Einzelbuchungen. 4. Auf „Speichern“ drücken. Der Stundenplan wird angezeigt. Gleichzeitig werden die Sucheigenschaften als Profil in der DB gespeichert und der Profilliste hinzugefügt.
Priorität	Hoch
Aufwand	Hoch
Zeitpunkt, Dringlichkeit	Sobald man mit der Implementierung begonnen hat.
Anmerkungen, Sonstiges	SPlan gilt nur während der Vorlesungszeit!

Tabelle 2: Suche nach Stundenplan

3.3.2. Suche nach Raumbellegung

Beschreibung Use-Case	
Name	Suche nach Raumbellegung
Kurzbeschreibung	Der Anwender kann seinen Raumbellegungsplan des jeweiligen Semesters suchen und sich anzeigen lassen.
Akteure	Anwender
Vorbedingungen	Für das betreffende Semester müssen die Auswahlliste der Standorte und die jeweils zugehörigen Räume definiert sein.
Eingehende Informationen	Informationen zur Suche der belegten Räume
Ergebnis(se)	Der Plan für die Belegung der Räume wird gemäß Wunsch angezeigt.
Auslöser	Anwender
Nachbedingungen	s. Ergebnis(se)
Ablauf	<ol style="list-style-type: none"> 1. Einen Standort auswählen. Alle Räume des ausgewählten Standorts werden in eine neue Auswahlliste geladen. 2. Unter „Räume“ einen bestimmten Raum auswählen. 3. Es werden dem Anwender drei Optionen zur Verfügung gestellt, die mit „JA“ oder „NEIN“ ausgewählt werden: <ul style="list-style-type: none"> • Die Suche als Standard festlegen, d.h., sobald die Applikation neu gestartet wird, wird automatisch die Suche anhand der gespeicherten benutzerspezifischen Eingaben ausgeführt und zeigt den Plan der belegten Räume an. • Anzeige „mit Datum“, um Blockveranstaltungen anzuzeigen. • Anzeige von Einzelbuchungen. 4. Auf „Speichern“ drücken. Der Plan der belegten Räume wird angezeigt und die Sucheigenschaften in der DB als Profil gespeichert sowie der Profilliste hinzugefügt.
Priorität	Mittel
Aufwand	Hoch
Zeitpunkt, Dringlichkeit	Sobald man mit der Implementierung begonnen hat.
Anmerkungen, Sonstiges	SPlan gilt nur während der Vorlesungszeit!

Tabelle 3: Suche nach Raumbellegung

3.3.3. Suche nach freien Räumen

Beschreibung Use-Case	
Name	Suche nach freien Räumen
Kurzbeschreibung	Der Anwender kann nach freien Räumen zu einem bestimmten Tag und Uhrzeit suchen und sich anzeigen lassen.
Akteure	Anwender
Vorbedingungen	Für das betreffende Semester muss die Auswahlliste der Standorte definiert sein.
Eingehende Informationen	Informationen zur Suche von freien Räumen
Ergebnis(se)	Freie Räume gemäß Wunsch angezeigt
Auslöser	Anwender
Nachbedingungen	s. Ergebnis(se)
Ablauf	<ol style="list-style-type: none"> 1. Der Anwender wählt einen Standort und eine Raum-Art (Vorlesungsraum, Labor, Übungsraum) aus. 2. Optional kann er Datum und Uhrzeit ändern, ansonsten werden automatisch das aktuelle Datum und die aktuelle Uhrzeit ausgewählt. 3. Auf „Suchen“ drücken. Die Liste aller freien Räume wird angezeigt.
Priorität	Mittel
Aufwand	Mittel
Zeitpunkt, Dringlichkeit	Sobald man mit der Implementierung begonnen hat.
Anmerkungen, Sonstiges	SPlan gilt nur während der Vorlesungszeit!

Tabelle 4: Suche nach freien Räumen

3.3.4. Profile verwalten

Beschreibung Use-Case	
Name	Profile verwalten
Kurzbeschreibung	Der Anwender kann die gespeicherten Profile ändern, löschen, nur das Datum des Profils aktualisieren oder das Profil auswählen und sich den Stundenplan anzeigen lassen.
Akteure	Anwender
Vorbedingungen	Zumindest muss eine „Suche nach Stundenplan“ oder „Suche nach Raumbelugung“ ausgeführt sein.
Eingehende Informationen	Informationen zur Bearbeitung der Profile.
Ergebnis(se)	Zu änderndes Profil gemäß Wunsch anzeigen bzw. zu löschendes Profil aus der Liste entfernen oder die Suche starten und den Stundenplan anzeigen
Auslöser	Anwender
Nachbedingungen	s. Ergebnis(se)
Ablauf	<p>Änderungen</p> <ol style="list-style-type: none"> 1. Zu änderndes Profil auswählen. 2. Änderungen vornehmen. 3. Auf „Speichern“ drücken. <p>Löschen</p> <ol style="list-style-type: none"> 1. Zu löschendes Profil auswählen. 2. Profil aus der DB löschen sowie aus der Profilliste entfernen. <p>Aktualisieren</p> <ol style="list-style-type: none"> 1. Zu änderndes Profil auswählen. 2. Aktualisierung wird automatisch durchgeführt.
Priorität	Mittel
Aufwand	Hoch
Zeitpunkt, Dringlichkeit	Nach der Implementierung von „Suche nach Stundenplan“ und „Suche nach Belegung der Räume“.
Anmerkungen, Sonstiges	SPlan gilt nur während der Vorlesungszeit!

Tabelle 5: Profile verwalten

3.3.5. Profil auswählen und Suche durchführen

Beschreibung Use-Case	
Name	Profil auswählen und Suche durchführen
Kurzbeschreibung	Der Anwender kann ein Profil auswählen und die Suche anhand der gespeicherten Sucheigenschaften ausführen.
Akteure	Anwender
Vorbedingungen	Zumindest muss eine „Suche nach Stundenplan“ bzw. „Suche nach Raumbelegung“ ausgeführt sein.
Eingehende Informationen	Informationen zur Bearbeitung der Profile.
Ergebnis(se)	Stundenplan bzw. Plan der Raumbelegung anzeigen.
Auslöser	Anwender
Nachbedingungen	s. Ergebnis(se)
Ablauf	<ol style="list-style-type: none"> 1. Profil auswählen 2. Suche wird automatisch durchgeführt, Stundenplan bzw. Plan der Raumbelegung wird angezeigt.
Alternative Abläufe	-
Priorität	Gering
Aufwand	Gering
Zeitpunkt, Dringlichkeit	Nach der Implementierung von „Suche nach Stundenplan“ und „Suche nach Belegung der Räume“.
Anmerkungen, Sonstiges	SPlan gilt nur während der Vorlesungszeit!

Tabelle 6: Profil auswählen und Suche durchführen

4. Konzeption und Implementierung

Nachdem die Anforderungen an die Applikation gesammelt wurden und diese den Leser kurz dargestellt wurden, behandelt dieses Kapitel den Programmablaufplan der Applikation sowie das Konzept der Benutzeroberfläche. Diese wird anhand einiger Screenshots näher erklärt. Anschließend wird die Implementierung mithilfe von UML-Diagrammen erläutert.

4.1. Programmablaufplan

Der Programmablaufplan (vgl. Abbildung 15) dieser Applikation stimmt fast mit dem PAP des Online Dienstes „SPlan“ überein. Für eine bessere Bedienbarkeit wurde der PAP um einige Optionen erweitert.

Beim Start der Applikation wird die DB auf bereits gespeicherte Profile durchsucht. Wird ein Profil gefunden, werden dessen Eigenschaften überprüft.

Der Anwender kann das Profil mit dem Attribut „Standard“ kennzeichnen, um der Applikation mitzuteilen, dass das Profil automatisch beim Start auszuführen ist und die entsprechenden Ergebnisse anzeigt.

Ist der Begriff „Standard“ nicht gewählt, gelangt man zum Hauptmenü. Von hier aus kann der Anwender eine neue Suche durchführen oder bereits gespeicherte Profile verwalten. Dem Anwender werden drei Sucharten zur Auswahl geboten:

1. Suche nach Stundenplan des jeweiligen Studiengangs
2. Suche nach Raumbellegung
3. Suche nach freien Räumen

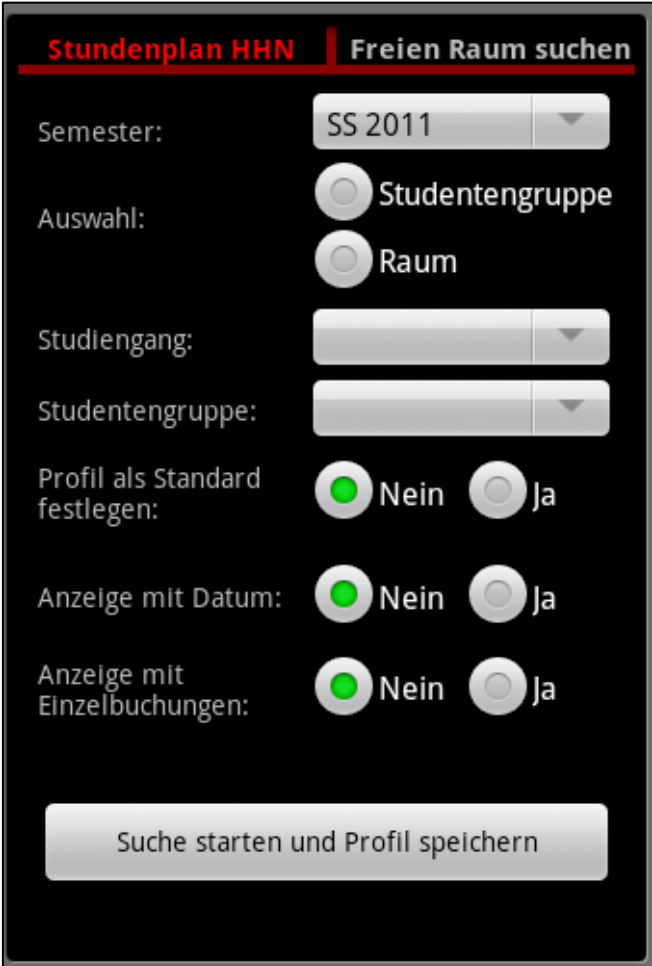
Bei den ersten beiden Sucharten kann der Anwender die durchgeführte Suche als Profil in der DB speichern. Die Suche wird anhand ihrer ausgewählten Eigenschaften der Profilliste hinzugefügt. Von dort aus kann der Anwender zu einem späteren Zeitpunkt eine gleichartige Suche durch Aufrufen des Profils ausführen, sie bearbeiten, oder sie aus der DB entfernen.

4.2. Konzept der Benutzeroberfläche

Im Kapitel 3.2 wurde bereits erwähnt, dass die Benutzeroberfläche meiner Applikation so konzipiert wurde, dass sie der Benutzeroberfläche des Online-Dienstes „SPlan“ für Stunden- und Veranstaltungspläne der Hochschule Heilbronn im Wesentlichen entspricht [23]. Die Applikation umfasst sechs Bildschirmmasken. Diese werden nachfolgend näher erläutert.

4.2.1. Hauptmenü / Suche nach Stundenplan

Beim Start der Applikation gelangt der Anwender in das Hauptmenü (vgl. Abb. 16). Im Hauptmenü kann zwischen der Suche nach Stundenplan oder nach Raumbellegung gewählt werden. Um freie Räume zu suchen, wählt man „Freien Raum suchen“. In der aktuell ausgewählten Sicht ist der Titel rot dargestellt.



The screenshot shows a search interface for finding free rooms. At the top, there are two tabs: 'Stundenplan HHN' (highlighted in red) and 'Freien Raum suchen'. Below the tabs, there are several input fields and radio buttons:

- Semester:** A dropdown menu showing 'SS 2011'.
- Auswahl:** Two radio buttons: 'Studentengruppe' (selected) and 'Raum'.
- Studiengang:** A dropdown menu.
- Studentengruppe:** A dropdown menu.
- Profil als Standard festlegen:** Two radio buttons: 'Nein' (selected) and 'Ja'.
- Anzeige mit Datum:** Two radio buttons: 'Nein' (selected) and 'Ja'.
- Anzeige mit Einzelbuchungen:** Two radio buttons: 'Nein' (selected) and 'Ja'.

At the bottom, there is a large button labeled 'Suche starten und Profil speichern'.

Abbildung 16: Hauptmenü - Suche nach Stundenplan

Nachdem die Studentengruppe ausgewählt wurde, startet man die Suche nach Stundenplänen.

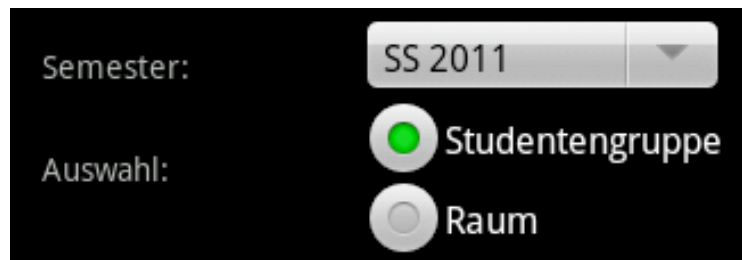


Abbildung 17: Hauptmenü - Auswahlmöglichkeiten

Darauf basierend wird automatisch die entsprechende Liste der Studiengänge für das jeweilige Semester geladen.

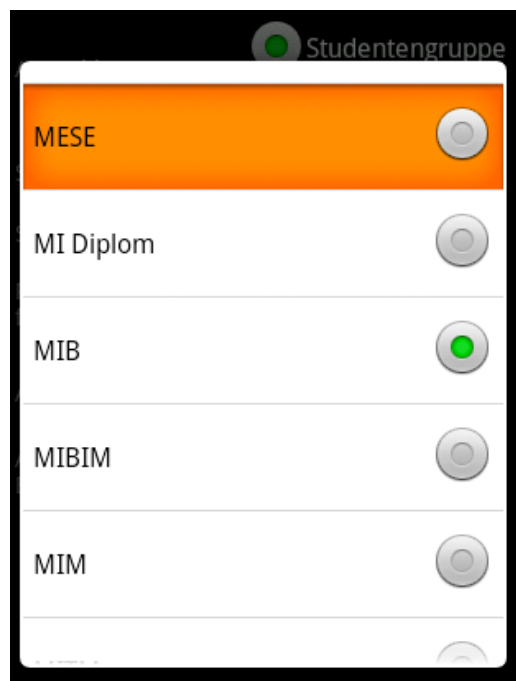


Abbildung 18: Hauptmenü/ Anzeige der Liste der Studiengänge

Wählt man einen Studiengang aus, wird wiederum automatisch die entsprechende Liste der Studentengruppen für den jeweiligen Studiengang geladen.

Zunächst hat der Anwender drei Optionen zur Verfügung, die mit JA oder NEIN ausgewählt werden:

1. Die Suche als Standard festlegen.
2. Das Datum für die Suche ändern.
3. Einzelbuchungen anzeigen.

Um die Suche zu starten, wird anschließend „Suche starten und Profil speichern“ gewählt. Gleichzeitig werden die Sucheigenschaften als Profil in der Profilliste gespeichert und der

Stundenplan wird wie in Abbildung 21 angezeigt. Die Suche und das Ergebnis können zu einem späteren Zeitpunkt erneut aufgerufen werden, ohne dass weitere Eingaben erforderlich sind.

4.2.2. Hauptmenü / Suche nach Raumbelegung

Wie im vorherigen Kapitel bereits erwähnt wurde, gelangt der Anwender beim Start der Applikation in das Hauptmenü. Um einen freien Raum zu finden, wählt der Anwender „Freien Raum suchen“.

Stundenplan HHN | **Freien Raum suchen**

Semester: SS 2011

Auswahl: Studentengruppe Raum

Standort: Bitte wählen S

Raum: Bitte wählen S

Profil als Standard festlegen: Nein Ja

Anzeige mit Datum: Nein Ja

Anzeige mit Einzelbuchungen: Nein Ja

Suche starten und Profil speichern

Abbildung 19: Hauptmenü - Suche nach Raumbelegung

Anschließend wird automatisch die Liste der Standorte für das jeweilige Semester geladen. Wählt man einen Standort aus, wird die Liste der Räume für den ausgewählten Standort angezeigt.

Analog zur Suche nach Stundeplänen hat der Anwender drei Optionen zur Verfügung, die er mit JA oder NEIN auswählen kann:

1. Die Suche als Standard festlegen.
2. Das Datum für die Suche ändern.
3. Einzelbuchungen anzeigen.

Um die Suche zu starten, wird anschließend „Suche starten und Profil speichern“ gewählt. Gleichzeitig werden die Sucheigenschaften als Profil in der Profilliste gespeichert, und es wird der Stundenplan mit Raumbelegung (vgl. Abb. 21) angezeigt. Die Suche und das Ergebnis können zu einem späteren Zeitpunkt erneut aufgerufen werden, ohne dass weitere Eingaben erforderlich sind.

Optionsmenü für das Hauptmenü

In den beiden vorhergehenden Kapiteln wurden die grundlegenden Funktionen des Hauptmenüs dargestellt. Über das Optionsmenü können zusätzliche Aktionen ausgeführt werden. Drückt man auf den Menü-Button, wird ein Menü wie in Abbildung 21 angezeigt. Mit einem weiteren Klick auf „Alle Profile anzeigen“ gelangt man vom Hauptmenü zur Profilliste, in der man die Profile verwalten kann.



Abbildung 20: Hauptmenü – Anzeige des Optionsmenüs

4.2.3. Anzeige der Stundenpläne

Nachdem die Suche gestartet wurde, wie bereits in den Kapiteln 4.2.1 und 4.2.2 beschrieben wurde, gelangt man auf die Bildschirmmaske, auf der die Stundenpläne der Studenten und die Raumbellegung dargestellt sind (vgl. Abb. 21).

SS 2011	MIB	MIB1	6-8-2011	
Montag	Dienstag	Mittwoch	Donnerstag	Freitag
12:15	AN1			F140
13:45	MIB1			Werner
14:00	LOG			F027, F140
17:15	MIB1			Schmidt
13:15	ErstSemBeg			F140
15:15	MIB1			

Abbildung 21: Anzeige der Stundenpläne

In der oberen Zeile werden folgende Informationen zum Profil angezeigt:

1. Das Semester, zu dem die Suche gestartet wurde
2. Der Studiengang oder Standort
3. Die Studentengruppe oder Raum
4. Das Datum

SS 2011	MIB	MIB1	6-8-2011
---------	-----	------	----------

Abbildung 22: Statusleiste

In der Mitte befindet sich eine Tableiste, wobei jeder Tab einen Tag repräsentiert. Unter der Tableiste wird der Stundenplan angezeigt. Im linken Bereich befinden sich die Uhrzeiten, sowie Beginn und Ende der Vorlesungen. Im Mittelbereich werden die Abkürzungen für die Vorlesungen in Großschrift und grün hervorgehoben angezeigt. Einzelbuchungen und Blockveranstaltungen sind gelb gekennzeichnet. Im rechten Bildbereich werden die Raumbezeichnungen und die Namen der Professoren und Dozenten angezeigt.

12:15	AN1	F140
13:45	MIB1	Werner
14:00	LOG	F027, F140
17:15	MIB1	Schmidt
13:15	ErstSemBeg	F140
15:15	MIB1	

Abbildung 23: Anzeige der Stundenpläne- Hauptansicht

Optionsmenü

Über das Optionsmenü können zusätzliche Aktionen ausgeführt werden. Wird der Menü-Button gewählt, wird eine entsprechende Menü-Liste angezeigt (vgl. Abb. 24). Der Anwender hat nun die Möglichkeit, zum Hauptmenü zurück zu kehren oder zur Profilliste zu navigieren.

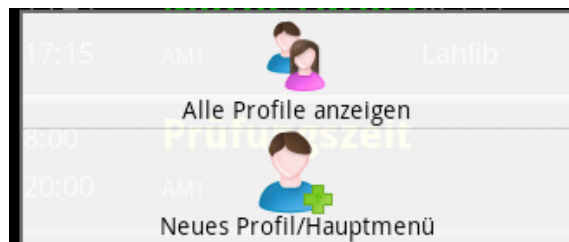


Abbildung 24: Anzeige der Stundenpläne - Optionsmenü

4.2.4. Profilliste / Profilverwaltung

Über die Profilliste (vgl. Abb. 25) kann der Anwender seine Profile verwalten. Jedes Profil hat einen Namen, der folgende sechs Bezeichnungen in dieser Reihenfolge beinhaltet:

1. Das jeweilige Semester
2. Den Studiengang oder den Standort
3. Die Studentengruppe oder den Raum
4. Die Auswahl des Profils, ob als Standard festgelegt
5. Die Auswahl bei der Anzeige von Einzelbuchungen
6. Das Datum

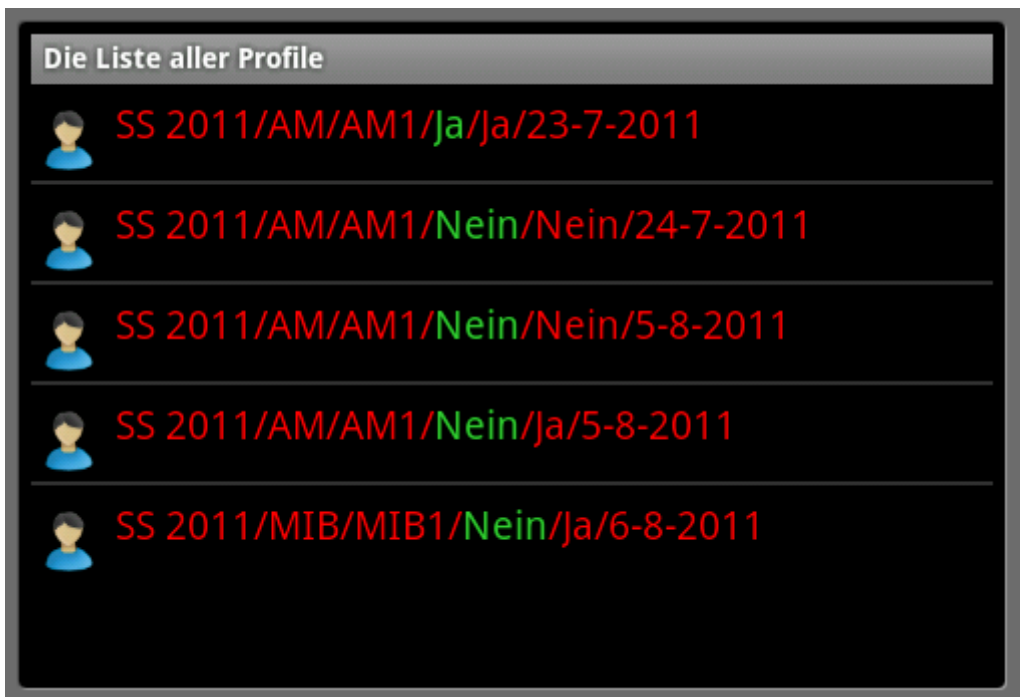


Abbildung 25: Anzeige der Profilliste

Die Funktionen zur Verwaltung der Profile wurden aufgeteilt und anschließend dem Optionsmenü sowie dem Kontextmenü zugeordnet.

Um zum Optionsmenü zu gelangen, wählt man den Menü-Button des Android-Gerätes. Für die Anzeige des Kontextmenüs hält der Anwender die entsprechende Zeile einige Sekunden gedrückt, bis das Menü erscheint.

4.2.4.1. Profil anzeigen / Stundenplan Anzeige

Damit der Stundenplan angezeigt wird, tippt man nur kurz auf die Beschreibung des Profils.

4.2.4.2. Profil bearbeiten

Im Gegensatz zur Stundenplan-Anzeige muss man einige Sekunden auf die Beschreibung des Profils drücken, damit das Menü für das gewählte Profil angezeigt wird.

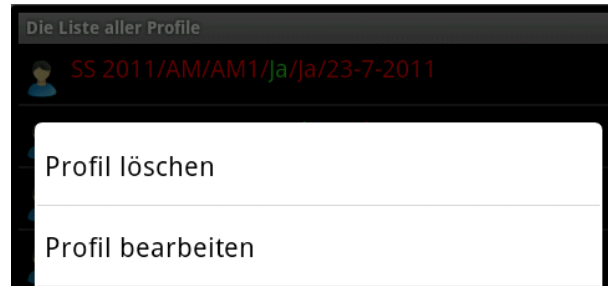


Abbildung 26: Anzeige der Profilliste - Contextmenü

Wird „Profil bearbeiten“ ausgewählt, gelangt man zum Hauptmenü. Es werden alle Informationen für dieses Profil angezeigt. Nach dem Bearbeiten des Profils wählt man „Speichern“. Die Änderungen werden in der Datenbank gespeichert, und man gelangt automatisch wieder zur Profilliste.

4.2.4.3. Profil löschen

Mit der Auswahl „Profil löschen“ werden das Profil in der DB gelöscht und die Liste aktualisiert.

Optionsmenü

Über das Optionsmenü können zusätzliche Aktionen ausgeführt werden. Wird der Menü-Button gewählt, wird eine entsprechende Menü-Liste angezeigt. Der Anwender hat nun die Möglichkeit, zum Hauptmenü zurück zu kehren oder zur Profilliste zu navigieren.

4.2.5. Suche nach freien Räumen

Wie bereits im Kapitel 4.2.1 erwähnt wurde, wählt man „Freien Raum suchen“, um freie Räume zu finden. In der aktuell ausgewählten Sicht ist der Titel rot dargestellt.



The screenshot shows a mobile application interface for searching for free rooms. At the top, there are two tabs: 'Stundenplan HHN' and 'Freien Raum suchen', with the latter being highlighted in red. Below the tabs, there are several input fields and controls:

- Standort:** A dropdown menu with the text 'Bitte wählen Sie' and a downward arrow.
- Raumart:** A dropdown menu with the text 'Bitte wählen Sie' and a downward arrow.
- Anfang:** A text input field containing '15:31' with a clock icon to its right.
- Ende:** A text input field containing '16:31' with a clock icon to its right.
- Datum:** A date picker showing 'Aug 09 2011' with '+' and '-' buttons for each part of the date.
- Suchen:** A large button at the bottom labeled 'Suchen'.

Abbildung 27: Bildschirmmaske der Suche nach freien Räumen

Zunächst wählt man einen Standort und eine Raumart aus. Die Applikation zeigt stets das aktuelle Datum und unter „Anfang“ und „Ende“ die aktuellen Uhrzeiten an. Das Intervall zwischen Anfang und Ende ist auf eine Stunde festgelegt. Das Datum und die Uhrzeiten können geändert werden. Durch Wahl des „Suchen“-Buttons werden alle freien Räume angezeigt.

4.2.6. Anzeige der freien Räume

Im Ergebnis der Suche nach freien Räumen werden alle freien Räume mit zusätzlichen Informationen tabellarisch angezeigt.

Freie Räume an Standort Campus HN - Uhrzeit: 15:31-16:31					
Kürzel	Gebäude	Standort	Raumart	Privater Raum	Sitzplätze
A107	A	Campus HN	Vorlesungsraum	Nein	72
A108	A	Campus HN	Vorlesungsraum	Nein	60
A109	A	Campus HN	Vorlesungsraum	Nein	60
A207	A	Campus HN	Vorlesungsraum	Nein	72
A208	A	Campus HN	Vorlesungsraum	Nein	60
A209	A	Campus HN	Vorlesungsraum	Nein	60
A210	A	Campus HN	Vorlesungsraum	Nein	60
A211	A	Campus HN	Vorlesungsraum	Nein	60
A212	A	Campus HN	Vorlesungsraum	Nein	60
A307	A	Campus HN	Vorlesungsraum	Nein	60
A308	A	Campus HN	Vorlesungsraum	Nein	60
A310B	A	Campus HN	Vorlesungsraum	Nein	12

Abbildung 28: Bildschirmmaske für Anzeige der freien Räume

Optionsmenü

Über das Optionsmenü kann man zum Hauptmenü zurück navigieren.

A209	A	Campus HN	Vorlesungsraum	Nein	60
A210	A	Campus HN	Vorlesungsraum	Nein	60
A211	A	Campus HN	Vorlesungsraum	Nein	60
A212	A	Campus HN	Vorlesungsraum	Nein	60
 Neues Profil/Hauptmenü					

Abbildung 29: Bildschirmmaske für Anzeige der freien Räume - Optionsmenü

4.3. Implementierung

Dieser Kapitel gibt anhand einiger Klassendiagramme einen groben Überblick über die wichtigsten Klassen, die die Struktur und den Programmablaufplan der Applikation bilden, und ihre Benutzeroberflächen. Anschließend wird der Anwendungsfall „Suche nach Stundenplänen“ mit einem Sequenzdiagramm beschrieben. Es zeigt die Kommunikation zwischen Objekten in einem ausgewählten Szenario. Es wird erläutert, welche Objekte in diesem Szenario verwendet werden, welche Informationen (Nachrichten) zwischen den Objekten ausgetauscht werden, und in welcher chronologischen Reihenfolge der Informationsaustausch stattfindet [24].

4.3.1. Klassendiagramme

Das Projekt besteht aus vier Paketen (vgl. Abbildung 30), die miteinander kommunizieren.

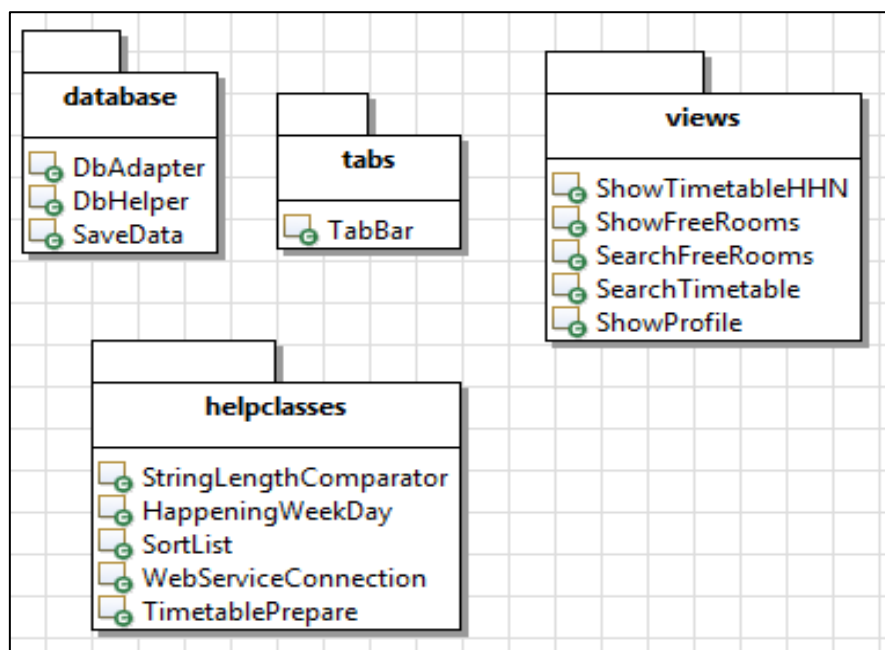


Abbildung 30: Übersicht über alle Klassen

Zusätzlich sind Jars in das Projekt integriert. Zum einen beinhalten sie alle erforderlichen Klassen sowie Schnittstellen für die Kommunikation mit dem SPlan Dienst, zum anderen verfügen sie über modifizierte Klassen des RPC Protokolls Hessain (Hessian-Flamingo), um eine optimale Kommunikation mit Android zu gewährleisten [18].

Das Paket „views“ ist für die Darstellung der GUI sowie für das Entgegennehmen der Anwenderaktionen verantwortlich. Jeder Klasse ist einer der fünf Bildschirmmasken, wie in Kapitel 4.2 erklärt wurde, zugeordnet. Das Paket „tabs“ dient der 5-Tage-Sicht beim Anzeigen der Stundenpläne und der Raumbellegung.

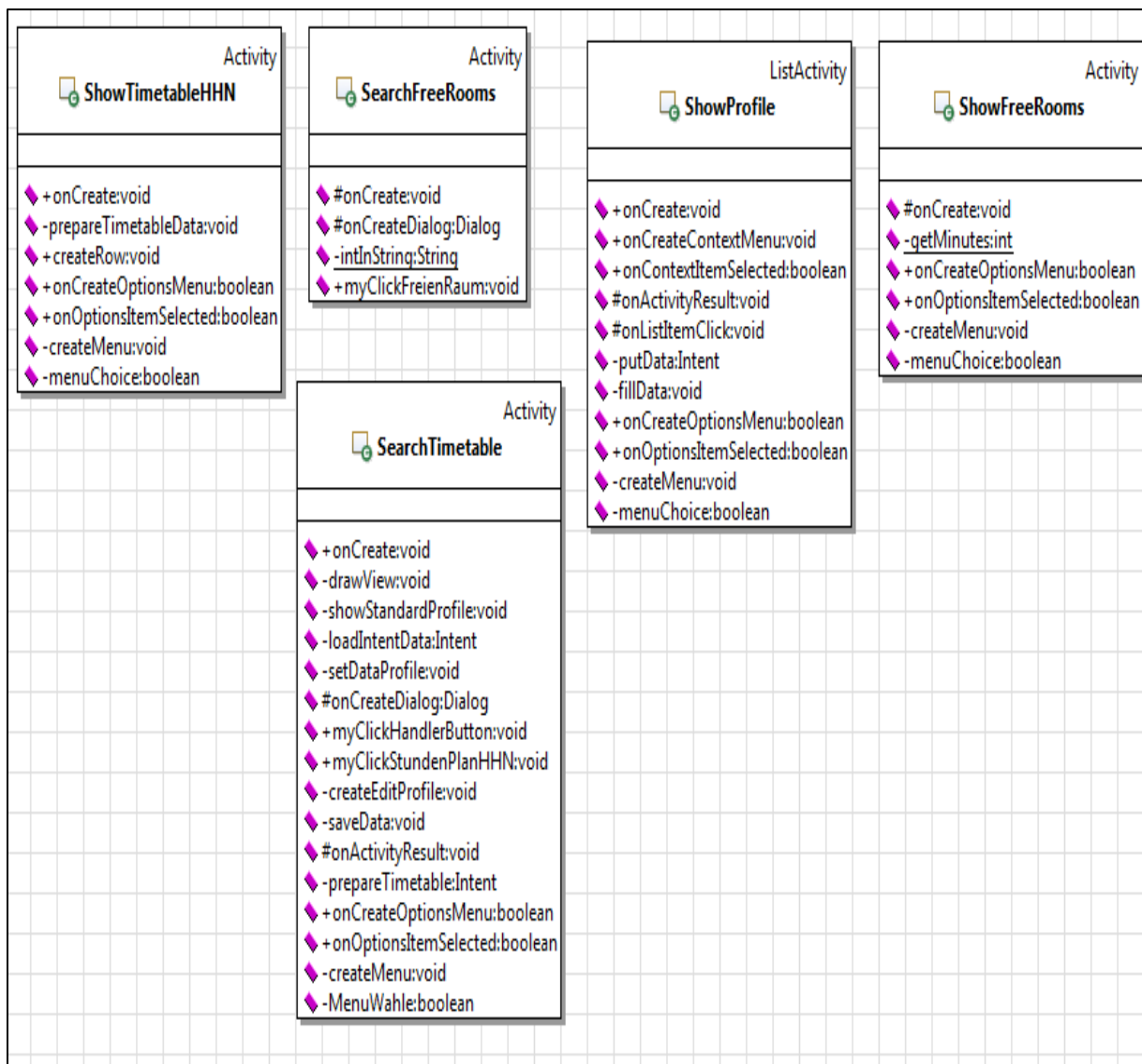


Abbildung 33: Klassendiagramm zur Benutzeroberfläche

Für die Herstellung einer Verbindung zwischen dem Client und dem Webserver wird die Methode `connectWebserver()` der Klasse „WebServiceConnection“ vom Paket „helpclasses“ eingesetzt. Außerdem beinhaltet sie weitere Methoden, die mit dem Webserver der Hochschule HN kommunizieren, um die nötigsten Daten für die GUI vorzubereiten.

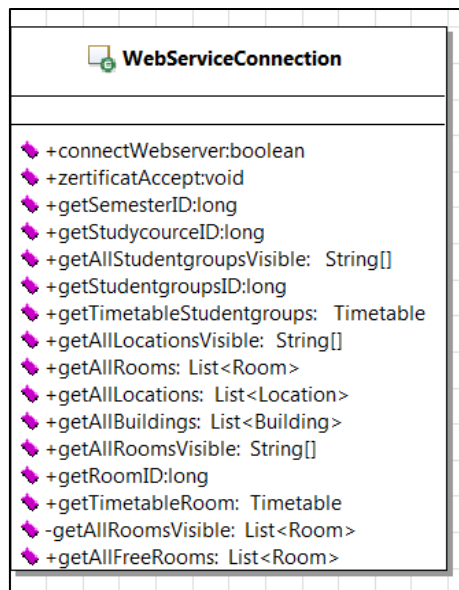


Abbildung 34: Darstellung der Klasse WebServiceConnection

Im Paket „helpclasses“ gibt es weitere Klassen, die eine detailliertere Darstellung der Stundenpläne ermöglichen.

Das Paket „database“ ist für die persistente Speicherung der Daten (erstellte Profile) in einer SQL-Datenbank verantwortlich. In Android wird eine SQLite-Datenbank verwendet. Die in meiner Arbeit verwendete Datenbankstruktur wurde analog zum Tutorial von Herrn Lars Vogel aufgebaut, da diese optimal strukturiert ist [25].

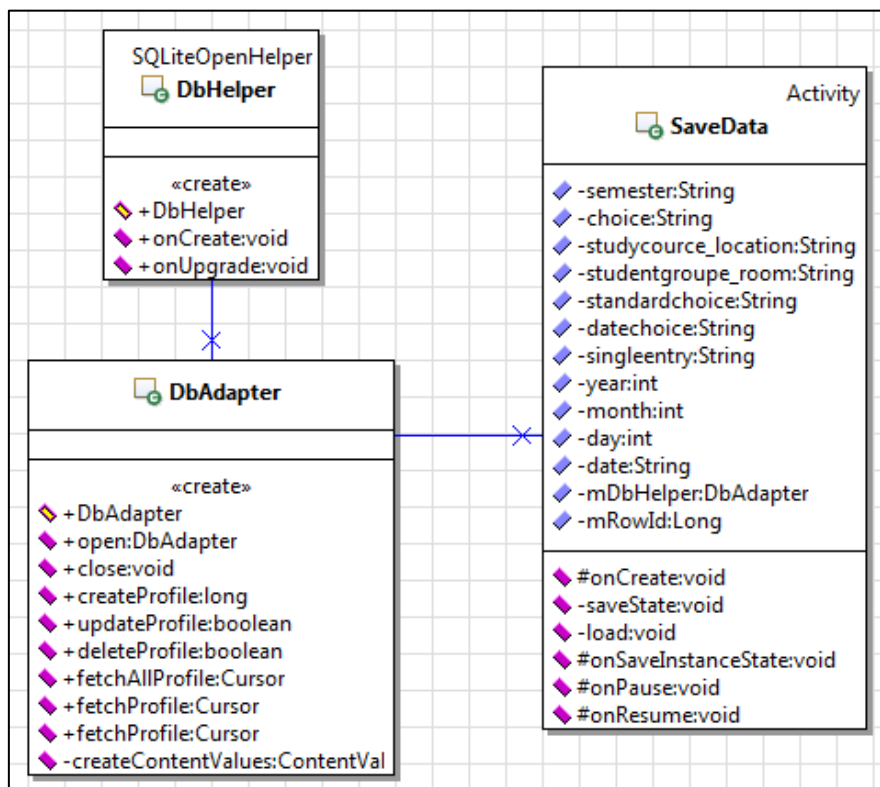


Abbildung 35. Klassendiagramm Datenbank

Datenbankschema

Die Datenbankstruktur (vgl. Abb. 36) ist sehr einfach und umfasst nur eine Tabelle. Anhand des SQL Statements zur Erzeugung der Tabelle *profile* lässt sich der folgende Tabellenaufbau erkennen:

```
private static final String DB_CREATE = "create table profile
(_id integer primary key autoincrement,
" + "semester text not null,
choice text not null,
studycource text not null,
studentgroupe text not null,
standardchoice text not null,
datechoice text not null,
singleentry text not null,
year integer,
month integer,
day integer,
date text not null);";
```

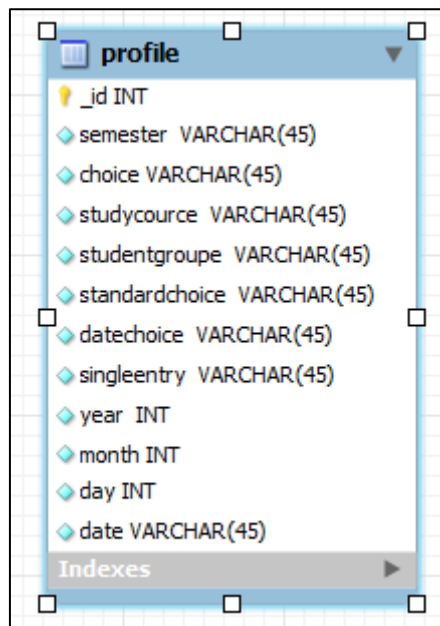


Abbildung 36: Datenbankstruktur

4.3.2. Sequenzdiagramme

In diesem Kapitel werden die Abläufe des Use Cases „Suche nach Stundenpläne“ anhand eines Sequenzdiagramms (vgl. Abb. 37) unter der Annahme, dass die Datenbank leer ist, Schritt für Schritt, erklärt.

Zunächst wählt der Anwender das Hauptmenü „Stundenplan HHN“ aus. Daraufhin werden die Activity für die Suche erzeugt und die Verbindung mit dem Web-Server hergestellt. Die Daten für die Suche werden vom Web-Server heruntergeladen und der Anzeigebildschirm mit den Inhalten angezeigt. Sobald der Benutzer den Button „Suche starten und Profil speichern“ wählt, wird die Suche gestartet. Es wird ein neues Profil erstellt und mit seinen Eigenschaften in der Datenbank gespeichert. Daraufhin werden die Daten vom Web-Server abgerufen, bearbeitet und für die Darstellung formatiert und strukturiert. Schließlich wird der Stundenplan angezeigt.

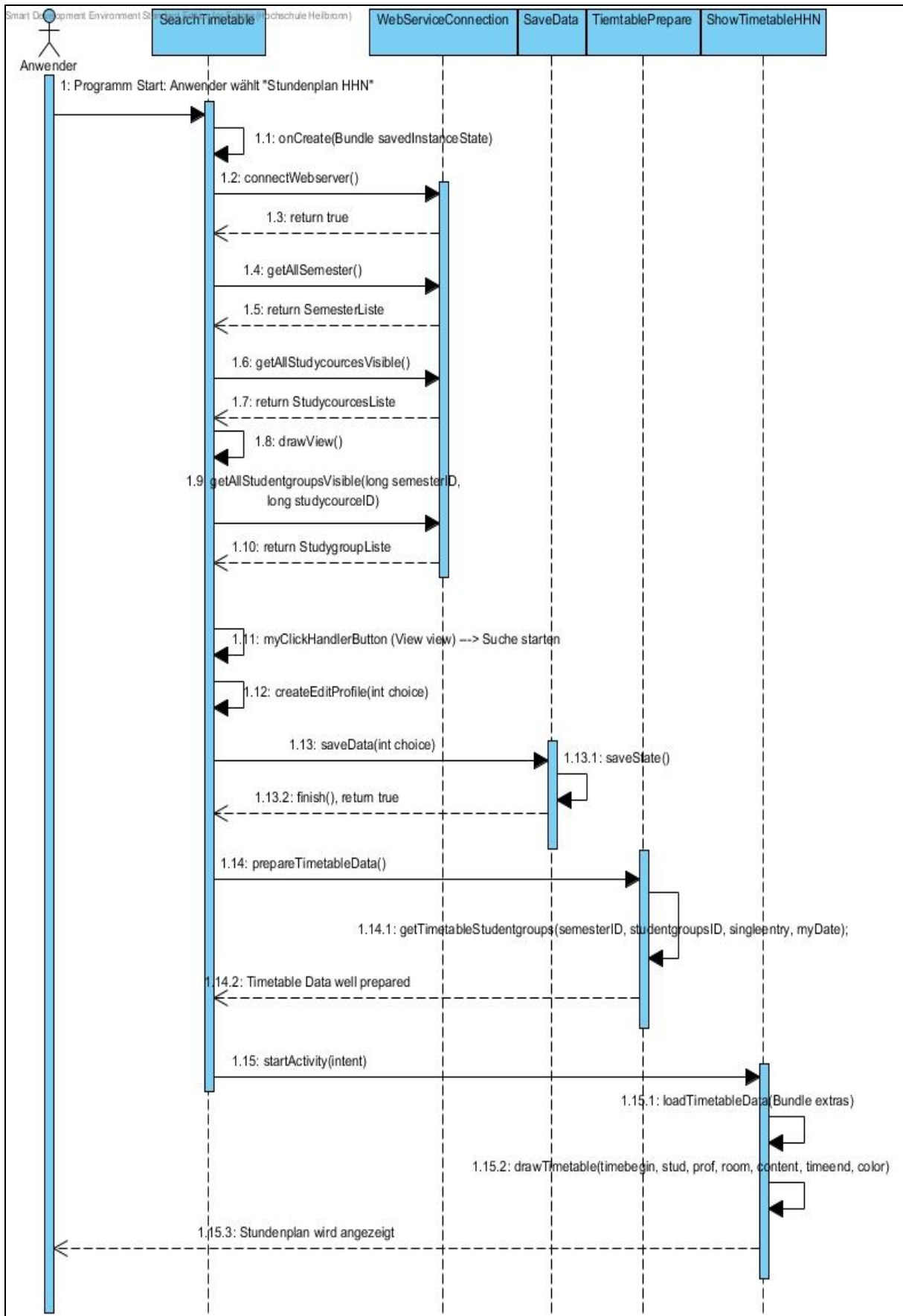


Abbildung 37: Sequenzdiagramm „Suche nach Stundenpläne“

5. Qualitätssicherung

Um die Qualität der fertiggestellten Applikation zu überprüfen, wurden alle beschriebenen Use-Cases aus Kapitel 3.3 mit verschiedenen Emulatoren in unterschiedlichen Bildschirmauflösungen getestet. Es wurde ein sogenannter „Black-Box-Test“ durchgeführt. Das Ergebnis wurde manuell mit dem Online-Dienst „SPlan“ für Stunden- und Veranstaltungspläne der Hochschule Heilbronn verglichen. Im Nachhinein wurden einige dieser Use-Cases auf einem Android-Smartphone getestet, um sicherzustellen, dass die Anwendung praxistauglich ist.

Folgende Testeigenschaften wurden für alle Testfälle, die auf „Suche nach Stundenplan“ beruhen, festgelegt:

Semester	2011						
Studiengang	MIB			AM		ASE	
Studentengruppe	MIB1	MIB2	MIB4	AM1	AM2	ASE1	ASE2

Tabelle 7: Testmerkmale Teil Eins

Folgende Testeigenschaften wurden für alle Testfälle, die auf „Suche nach Raumbellegung“ beruhen, festgelegt:

Semester	2011						
Standort	Campus HN			Campus KÜZ			
Studentengruppe	A206	A207	A208	KA204			

Tabelle 8: Testmerkmale Teil Zwei

6. Fazit und Ausblick

Das Ziel dieser Diplomarbeit war es, mich in das Android OS einzuarbeiten, hierüber einen Überblick zu geben, des weiteren eine Anwendung für die Plattform Android zu entwickeln, mit der Studenten und Mitarbeiter der Hochschule Heilbronn personalisierte Stundenplan und Raumbelungspläne auf einem mobilen Endgerät online abrufen können. Auf Basis der verwendeten Entwicklungsumgebung Eclipse mit Android SDK und dem Eclipse Plug-In „ADT“ ist mir diese Aufgabe gut gelungen. Es gibt zahlreiche Dokumentationen und Tutorials, sowohl von Google als auch Drittanbietern, die den Anfängern das Erlernen erleichtern [26].

Während der Entwicklungsphase ich die Schwächen und die Stärken des Android Betriebssystems erkennen. Mit den Schwächen möchte ich beginnen, da sie für meine Diplomarbeit keine großen Hindernisse darstellten. Einerseits kann man aufgrund der unterschiedlichen Android-Plattformen die fertiggestellte Anwendung nicht auf allen Smartphones, die auf dem Android OS basieren, installieren. Dennoch werden über 80% der Smartphones unterstützt. Zum anderen habe ich festgestellt, dass die Oberfläche eines Emulators nicht 100%ig der Oberfläche eines Android Smartphones entspricht. Daher können minimale Unterschiede bei der Implementierung einer Android-Anwendung auftreten. In Bezug auf die Entwicklungsumgebung Eclipse mit Android SDK erfordern die Implementierung und das Testen mit den Emulatoren, sehr leistungsstarke Prozessoren und viel Arbeitsspeicher.

Die große Stärke des Android OS kann wie folgt zusammengefasst werden: „Android ist freie Software und quelloffen“ [27]. Jeder kann auf den Quellcode zugreifen und damit Anwendungen entwickeln [19]. Damit war es mir leicht möglich, meine Anwendung für meine Diplomarbeit zu programmieren. Darüber hinaus konnte ich die fertiggestellte Anwendung ohne zusätzliche Kosten testen und auf einem Android Smartphone installieren. Ebenso konnte ich mit dem Eclipse Plug-In „ADT“ meine Anwendung individuell gestalten und problemlos implementieren. Es bietet sich sogar das Erstellen von Komponenten an, die man wiederholt verwenden kann. Das Eclipse Plug-In „ADT“ nimmt Entwicklern viel Routinearbeit ab und reduziert den Implementierungsaufwand. Auf den ersten Blick erscheint die Implementierung einer Android-Anwendung etwas komplizierte, sie ist jedoch leicht nachvollziehbar. Alles in allem kann man sowohl einfache als auch sehr

professionelle Android-Anwendungen mit wenig Aufwand und wenig Lizenzkosten programmieren.

Kritisch betrachtet ist meine Anwendung hinsichtlich der Performance verbesserungswürdig. Bei der Darstellung „Alle freien Räume“ kann der auf dem Bildschirm angezeigte Text vergrößert oder verkleinert werden. Leider benötigen einige Smartphones-Gerätetypen mit leistungsschwachen Prozessoren einige Sekunden Rechenzeit, bis der Inhalt auf dem gesamten Bildschirm zu sehen ist. Da in naher Zukunft immer leistungsstärkere Prozessoren auf den Markt kommen, besteht dieses Problem nur vorübergehend.

Alle geforderten Funktionen wurden fertiggestellt. Dennoch decken diese Funktionen nicht alle Bedürfnisse der Anwender ab. Daher wäre es wünschenswert, weitere Funktionen in die Anwendung einzubinden. Als Beispiel sei das Speichern der Stundenpläne in einer Datenbank genannt, um sie offline aufrufen zu können. Ein weiteres Feature wäre das Aufbereiten von Stundenplänen für mehrere Wochen, sodass man eine Gesamtübersicht über den Monat oder das Semester erhält.

Zum Schluss möchte ich darauf hinweisen, dass sich die fertiggestellte Anwendung noch in der Testphase befindet, in der ich bereits Testläufe auf verschiedenen Emulatoren durchgeführt habe. Daher sollten vor Veröffentlichung der Anwendung eine Release-Version und weitere Testläufe auf verschiedenen Smartphones mit unterschiedlichen Auflösungen und Bildschirmgrößen durchgeführt werden.

7. Literaturverzeichnis und Referenzen

- [1] Jan Borns. Studie zur Smartphone-Nutzung 2012: Die „Smart-Natives“ kommen! [Online] 16.11.2010 [Zuletzt zugegriffen am: 28. August 2011]
<http://t3n.de/news/studie-smartphone-nutzung-2012-smart-natives-kommen-285160/>
- [2] Newsiversum. Mobile Internetnutzung mit Handy immer beliebter. [Online] 01.08.2011 [Zuletzt zugegriffen am: 28. August 2011]
<http://www.newsiversum.de/lifestyle/internet/mobile-internetnutzung-mit-handy-immer-beliebter.html>
- [3] Smartphone Welt. Was ist ein Smartphone. [Online] 06.02.2009 [Zuletzt zugegriffen am: 28. August 2011] <http://www.smartphone-welt.de/was-ist-ein-smartphone>
- [4] Jean-Claude Frick. Apple gibt Quartalszahlen bekannt – Erfolg an allen Fronten. [Online] 20.04.2011 [Zuletzt zugegriffen am: 28. August 2011]
<http://www.ifrick.ch/2011/04/apple-gibt-quartalszahlen-bekannt-erfolg-an-allen-fronten/>
- [5] Arnulf Schäfer. Gartner: Android ist das beliebteste Betriebssystem. [Online] 11.08.2011 [Zuletzt zugegriffen am: 28. August 2011]
<http://www.connect.de/news/gartner-android-ist-das-beliebteste-betriebssystem-1168962.html>
- [6] Christoph H. Hochstätter. Gartner: Knapp 50 Prozent Marktanteil für Android bis 2012. [Online] 07.04.2011 [Zuletzt zugegriffen am: 28. August 2011]
<http://www.zdnet.de/news/41551444/gartner-knapp-50-prozent-marktanteil-fuer-android-bis-2012.htm>
- [7] Android Developers. Welcome to Android. [Online] [Zuletzt zugegriffen am: 28. August 2011]
<http://source.android.com/>
- [8] Android Developer. What is Android?. [Online] [Zuletzt zugegriffen am: 28. August 2011]
<http://developer.android.com/guide/basics/what-is-android.html>
- [9] Arno Becker, Marcus Pant. Android, Grundlagen und Programmierung. Dpunkt-Verlag, 1. Auflage 2009.
- [10] Chris Haseman. Android Essentials. Apress, 1. Auflage 2008.
- [11] Mark L. Murphy. Beginning Android 2. Apress, 1. Auflage 2010.
- [12] Android Developers. Activity Lifecycle. [Online] [Zuletzt zugegriffen am: 28. August 2011]
<http://developer.android.com/reference/android/app/Activity.html>
- [13] Webopedia, Inc. Webservices. [Online] [Zuletzt zugegriffen am: 28. August 2011]
http://www.webopedia.com/TERM/W/Web_services.html
- [14] Wikipedia, Inc. Webservice. [Online] [Zuletzt zugegriffen am: 28. August 2011]
<http://de.wikipedia.org/wiki/Webservice>
- [15] Caucho Technology, Inc. Hessian binary web service protocol. [Online] [Zuletzt zugegriffen am: 28. August 2011] <http://hessian.caucho.com/>

- [16] Caucho Hessian. RPC Hessian. [Online] [Zuletzt zugegriffen am: 28. August 2011]
<http://java-source.net/open-source/web-services-tools/caucho-hessian>
- [17] Daniel Gredler. Java Remoting: Protocol Benchmarks. [Online] 07.08.2008 [Zuletzt zugegriffen am: 28. August 2011]
<http://daniel.gredler.net/2008/01/07/java-remoting-protocol-benchmarks/>
- [18] Exadel Flamingo, Inc. Supported client technologies. [Online] [Zuletzt zugegriffen am: 28. August 2011] <http://exadel.org/flamingo>
- [19] Android Developer. SDK & Platform Tools. [Online] [Zuletzt zugegriffen am: 28. August 2011]
<http://developer.android.com/guide/developing/tools/index.html>
- [20] Android Developers. Installing the SDK. [Online] [Zuletzt zugegriffen am: 28. August 2011]
<http://developer.android.com/sdk/installing.html>
- [21] Android Developers. Plattform Versions. [Online] [Zuletzt zugegriffen am: 28. August 2011]
<http://developer.android.com/resources/dashboard/platform-versions.html>
- [22] oFb. der online Fragebogen. [Online] [Zuletzt zugegriffen am: 28. August 2011]
<https://www.socisurvey.de/splan>
- [23] SPlan. Das Stundenplan-System „SPlan“ der Hochschule Heilbronn. [Online] [Zuletzt zugegriffen am: 28. August 2011]
<https://splan.hs-heilbronn.de/splan/Dispatcher?act=tt>
- [24] Fachbereich Informatik - Hochschule Darmstadt. Sequenzdiagramm. [Online] [Zuletzt zugegriffen am: 28. August 2011]
<https://www.fbi.h-da.de/labore/case/uml/sequenzdiagramm.html>
- [25] Lars Vogel. Android SQLite Database – Tutorial. [Online] [Zuletzt zugegriffen am: 28. August 2011] <http://www.vogella.de/articles/AndroidSQLite/article.html>
- [26] Android Developer. Developer Resources. [Online] [Zuletzt zugegriffen am: 28. August 2011]
<http://developer.android.com/resources/index.html>
- [27] Android Developer. Android Open Source Project license. [Online] [Zuletzt zugegriffen am: 28. August 2011] <http://source.android.com/source/licenses.html>

8. Anhang

8.1. Abbildungsverzeichnis

Abbildung 1: Android-System-Architektur [8].....	9
Abbildung 2: Code Ausschnitt der XML-basierten Datei View	11
Abbildung 3: AndroidManifest.xml (2).....	12
Abbildung 4: Aufbau einer Android Anwendung	14
Abbildung 5: Codeausschnitt AndroidManifest.xml (2)	15
Abbildung 6: Lebenszyklus einer Activity [12].....	15
Abbildung 7: Codeausschnitt Methode onPause().....	16
Abbildung 8: Codeausschnitt Methode onResume()	16
Abbildung 9: Vergleich der Transportgeschwindigkeit verschiedener Protokolle (1) [18]	19
Abbildung 10: Vergleich der Transportgeschwindigkeit verschiedener Protokolle (2) [18]	19
Abbildung 11: Verteilung der verschiedenen Android Plattformen [21]	21
Abbildung 12: Codeausschnitt Webservice Aufbau	23
Abbildung 13: Auswertung der funktionalen Anforderungen	25
Abbildung 14: Use-Cases-Diagramm der Applikation	26
Abbildung 15: Programmablaufplan der Applikation	33
Abbildung 16: Hauptmenü - Suche nach Stundenplan	34
Abbildung 17: Hauptmenü - Auswahlmöglichkeiten.....	35
Abbildung 18: Hauptmenü/ Anzeige der Liste der Studiengänge	35
Abbildung 19: Hauptmenü - Suche nach Raumbellegung.....	36
Abbildung 20: Hauptmenü – Anzeige des Optionsmenüs.....	37
Abbildung 21: Anzeige der Stundenpläne	38
Abbildung 22: Statusleiste.....	38
Abbildung 23: Anzeige der Stundenpläne- Hauptansicht	39
Abbildung 24: Anzeige der Stundenpläne - Optionsmenü.....	39
Abbildung 25: Anzeige der Profilliste	40
Abbildung 26: Anzeige der Profilliste - Contextmenü	41
Abbildung 27: Bildschirmmaske der Suche nach freien Räumen.....	42
Abbildung 28: Bildschirmmaske für Anzeige der freien Räume.....	43
Abbildung 29: Bildschirmmaske für Anzeige der freien Räume - Optionsmenü.....	43
Abbildung 30: Übersicht über alle Klassen.....	44
Abbildung 31: Übersicht über alle referenzierten Bibliotheken	45
Abbildung 32: Klassendiagramm der gesamten Applikation	45
Abbildung 33: Klassendiagramm zur Benutzeroberfläche	46
Abbildung 34: Darstellung der Klasse WebServiceConnection	47
Abbildung 35: Klassendiagramm Datenbank	47
Abbildung 36: Datenbankstruktur.....	48
Abbildung 37: Sequenzdiagramm „Suche nach Stundenpläne“	50
Abbildung 38: Online Umfrage-1.....	58
Abbildung 39: Online Umfrage-2.....	59
Abbildung 40: Online Umfrage-3.....	59
Abbildung 41: Online Umfrage-4.....	60
Abbildung 42: Online Umfrage-5.....	61

8.2. Tabellenverzeichnis

Tabelle 1: Struktur eine Android Anwendung [10] [11]	13
Tabelle 2: Suche nach Stundenplan	27
Tabelle 3: Suche nach Raumbelugung.....	28
Tabelle 4: Suche nach freien Räumen	29
Tabelle 5: Profile verwalten.....	30
Tabelle 6: Profil auswählen und Suche durchführen	31
Tabelle 7: Testmerkmale Teil Eins	51
Tabelle 8: Testmerkmale Teil Zwei	51

8.3. Online-Umfrage

Im Rahmen meiner Diplomarbeit wurde eine Online Umfrage [4] gestellt und durchgeführt. Es ist auf den folgenden Seiten durch Screenshots zu sehen.

FHN
HOCHSCHULE HEILBRONN
SCHULE | LEHRE | BERUF | DURCHGANG

Allgemeines

1. Geschlecht [A001]
Bitte machen Sie eine Angabe zu Ihrem Geschlecht.

weiblich
 männlich

2. Welcher Studienrichtung gehören Sie an? [A002]

Technik
 Wirtschaft
 Informatik

3. In welchem Studienabschnitt befinden Sie sich gerade? [A003]

Grundstudium
 Hauptstudium
 Master

Abbildung 38: Online Umfrage-1

Verhalten der Nutzer in Bezug auf den Online-Dienst „Stunden- und Raumplanung“

Folgende Fragen beziehen sich auf das ganze Semester

4. Nutzen Sie diesen Online-Dienst? [A101]

- Ja
- Nein

5. Falls Ja, wie oft? [A102]

- 1 – 19
- 20 – 40
- mehr als 40

Abbildung 39: Online Umfrage-2

6. Wie häufig benutzen Sie die Option „Suche nach Stundenpläne“? [A103]

- 0
- 1 – 9
- 10 – 20
- mehr als 20

7. Wie häufig benutzen Sie die Funktionsbereich „Suche nach Raumbellegung“? [A104]

- 0
- 1 – 5
- 6 – 10
- mehr als 10

Abbildung 40: Online Umfrage-3

8. Wie häufig benutzen Sie die „Suche nach freien Räumen“? [A105]

- 0
- 1 – 5
- 6 – 10
- mehr als 10

9. Wie wichtig sind Ihnen folgende Funktionsbereiche bei „Suche nach Stundenpläne“ bzw. bei „Suche nach Raumebelegung“? [A106]

	Unwichtig						sehr wichtig
Option „Anzeige mit Einzelbuchungen“	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Option „Anzeige mit Datum“	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	

10. Wie wichtig wären Ihnen folgende Funktionsbereiche der HHN-App? [A107]


	gar nicht						sehr stark
	1	2	3	4	5		
Durchgeführte Suche als Profil zu speichern	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>		
Profile zu verwalten	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>		

Abbildung 41: Online Umfrage-4

67% ausgefüllt



Frage [A108]

Frage [A109]

Für weitere Anregungen und Ideen zur Gestaltung einer Hochschul-App bin ich Ihnen sehr dankbar.

(Bitte nur konstruktive
Vorschläge) Anmerkungen

Abbildung 42: Online Umfrage-5

8.4. Testprotokoll

Projekt: SPlan App
Autor: Hekmat Al-Shenewi
Letzte Änderung: 20.07.2011

Übersicht

Testobjekt(e)	Es werden alle Use-Cases getestet.
Tester	Hekmat Al-Shenewi
Testdatum	20.07.2011

Testfälle

1. Suche nach Stundenplan ohne Datum Angabe / ohne Anzeige von Einzelbuchungen

Erwartetes Ergebnis	Der reguläre Stundeplan wird angezeigt. Einzelbuchungen werden nicht angezeigt.
Tatsächliches Ergebnis	Der reguläre Stundeplan wird angezeigt. Einzelbuchungen werden nicht angezeigt.
Testergebnis	Das tatsächliche Ergebnis stimmt mit dem erwarteten Ergebnis überein.

2. Suche nach Stundenplan ohne Datum Angabe / mit Anzeige von Einzelbuchungen

Erwartetes Ergebnis	Der reguläre Stundeplan wird angezeigt. Einzelbuchungen werden angezeigt.
Tatsächliches Ergebnis	Der reguläre Stundeplan wird angezeigt. Einzelbuchungen werden angezeigt.
Testergebnis	Das tatsächliche Ergebnis stimmt mit dem erwarteten Ergebnis überein.

3. Suche nach Stundenplan mit Datum

Erwartetes Ergebnis	Der reguläre Stundeplan wird angezeigt. Datum kann geändert werden.
Tatsächliches Ergebnis	Der reguläre Stundeplan wird angezeigt. Datum kann geändert werden.
Testergebnis	Das tatsächliche Ergebnis stimmt mit dem erwarteten Ergebnis überein.

4. Suche nach Raumbellegung ohne Datum Angabe / ohne Anzeige von Einzelbuchungen

Erwartetes Ergebnis	Der Plan für die Raumbellegung wird angezeigt. Einzelbuchungen werden nicht angezeigt.
Tatsächliches Ergebnis	Der Plan für die Raumbellegung wird angezeigt. Einzelbuchungen werden nicht angezeigt.
Testergebnis	Das tatsächliche Ergebnis stimmt mit dem erwarteten Ergebnis überein.

5. Suche nach Raumbellegung ohne Datum Angabe / mit Anzeige von Einzelbuchungen

Erwartetes Ergebnis	Der Plan für die Raumbellegung wird angezeigt. Einzelbuchungen werden angezeigt.
Tatsächliches Ergebnis	Der Plan für die Raumbellegung wird angezeigt. Einzelbuchungen werden angezeigt.
Testergebnis	Das tatsächliche Ergebnis stimmt mit dem erwarteten Ergebnis überein.

6. Suche nach Raumbellegung mit Datum

Erwartetes Ergebnis	Der Plan für die Raumbellegung wird angezeigt. Datum kann geändert werden.
Tatsächliches Ergebnis	Der Plan für die Raumbellegung wird angezeigt. Datum kann geändert werden.
Testergebnis	Das tatsächliche Ergebnis stimmt mit dem erwarteten Ergebnis überein.

7. Suche nach freien Räumen

Erwartetes Ergebnis	Alle freien Räume werden tabellarisch angezeigt.
Tatsächliches Ergebnis	Alle freien Räume werden tabellarisch angezeigt.
Testergebnis	Das tatsächliche Ergebnis stimmt mit dem erwarteten Ergebnis überein.

8. Profile verwalten / Profil Ändern

Erwartetes Ergebnis	Das Profil soll änderbar sein. Zudem soll auch in DB parallel geändert. Die Profilliste wird auch geändert mit der neuen zugehörigen Informationen.
Tatsächliches Ergebnis	Profil konnte geändert werden. DB und Profilliste wurden auch geändert.
Testergebnis	Das tatsächliche Ergebnis stimmt mit dem erwarteten Ergebnis überein.

9. Profile verwalten / Profil Löschen

Erwartetes Ergebnis	Das Profil soll von der Profilliste, sowie von der DB gelöscht werden.
Tatsächliches Ergebnis	Das Profil konnte von der Profilliste, sowie von der DB gelöscht werden.
Testergebnis	Das tatsächliche Ergebnis stimmt mit dem erwarteten Ergebnis überein.

10. Profil auswählen und Suche durchführen

Erwartetes Ergebnis	Nach Auswahl von einen bestimmten Profil, wird die Suche anhand ihrer Eigenschaften durchgeführt und der Plan ist dann zu sehen.
Tatsächliches Ergebnis	Profile können gewählt werden. Die Stundepläne werden anhand der Sucheigenschaften angezeigt.
Testergebnis	Das tatsächliche Ergebnis stimmt mit dem erwarteten Ergebnis überein.

Bewertung

Der Test war erfolgreich, es wurden keine Abweichungen gefunden.

9. Inhalt der CD

Auf der beigefügten CD-ROM ist der Quellcode des SPlan-Projekts enthalten. Zusätzlich ist die Diplomarbeit als PDF in digitaler Form verfügbar.

Erklärung

Ich erkläre hiermit an Eides Statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen (einschließlich elektronischer Quellen) direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Für die sprachlichen Korrektur- und Formulierungsvorschläge zu meiner Diplomarbeit habe ich Unterstützungsleistung von folgenden Personen erhalten:

- Frau Katrin Göttke
- Frau Barbara Sterzenbach

Leimen, den 16.09.2011

Unterschrift: Hekmat Al-Shenewi