

# **Erstellung einer offiziellen Android-App für die Hochschule Hof**

Lars Gaidzik und Lukas Mahr

## Contents

<b>1</b>	<b>IST-Zustand zu Semesterbeginn</b>	<b>3</b>
<b>2</b>	<b>Organisatorisches</b>	<b>3</b>
<b>3</b>	<b>Schnittstelle</b>	<b>4</b>
<b>4</b>	<b>Entwicklung der App</b>	<b>5</b>
4.1	Fragmente . . . . .	6
4.2	Adapter . . . . .	7
4.3	Datenhandler . . . . .	8
4.4	Datenstrukturen . . . . .	9
4.5	Experimental . . . . .	9
<b>5</b>	<b>Design</b>	<b>10</b>

# 1 IST-Zustand zu Semesterbeginn

Im letzten Semester wurde bereits eine App als Studienarbeit entwickelt. Diese hatte folgende Funktionen: Stundenplan, Stundenplanänderungen, Speiseplan, Notenbekanntgabe und Notenblatt. Da zu diesem Zeitpunkt keine Programmierschnittstelle existierte, wurden alle Daten über das Auslesen verschiedener Webseiten erreicht. Unser Ziel ist es, diese App in Zusammenarbeit mit der Hochschule Hof soweit anzupassen, dass diese als offizielle App der Hochschule Hof im Google Play Store eingestellt werden kann.

# 2 Organisatorisches

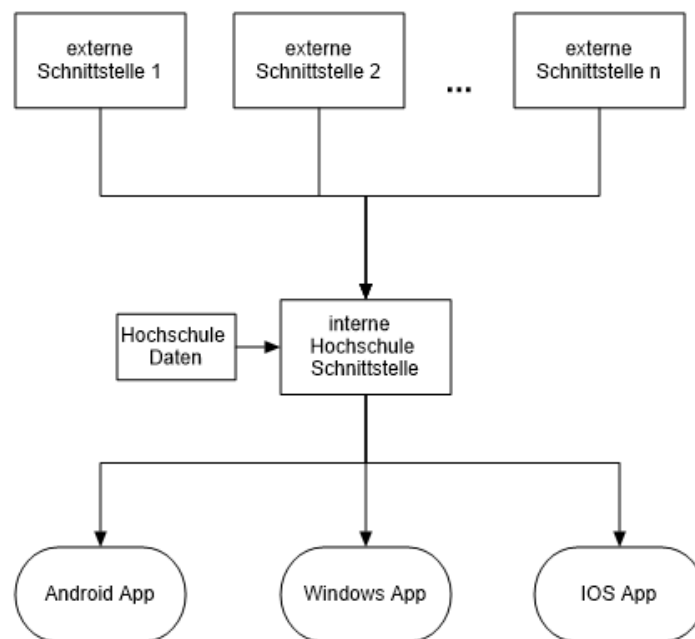
Zu Beginn wurde unsere Bestrebung an das Web-Team der Hochschule herangetragen. Hierbei stellten wir fest, dass es bereits mehrere Studentengruppen gibt, welche daran arbeiteten einen App auf den verschiedenen Plattformen zu entwickeln. Im nächsten Schritt wurden all diese Teams kontaktiert und ein gemeinsames Treffen arrangiert, um die einzelnen Gruppen zusammenzuführen und so die vorhandenen Ressourcen zielgerichtet einsetzen zu können. Bei einem weiteren Gespräch mit der Hochschulleitung wurde von offizieller Seite beschlossen, diese Vorhaben fortzuführen und eine App für die unterschiedlichen Plattformen, unter der Leitung von Herrn Prof. Dr. Stepping zu erstellen. Da es bereits eine Gruppe für Windows Phone gibt und von unserer Seite aus Android abgedeckt wird, werden zunächst für diese beiden Plattformen Apps entwickelt. Eine App für iOS soll später nachgezogen werden, sobald sich ein Studentengruppe dazu bereiterklärt. Während der Projektdurchführung wurde wöchentlich ein Meeting durchgeführt, um sich über Fortschritt des Projekts auszutauschen und offene Punkte zu klären. Bei allen Treffen waren jeweils ein Mitglied aus den Gruppen Android und Windows Phone, Prof. Dr. Stepping, Dipl.-Inf. Peter Rill und ein Vertreter von der internen Hochschule IT anwesend. Bei den Besprechungen einigte man sich darauf, in der ersten Version folgende Funktionen umzusetzen:

- Stundenplan
- Mein Stundenplan
- Stundenplanänderungen
- Speiseplan

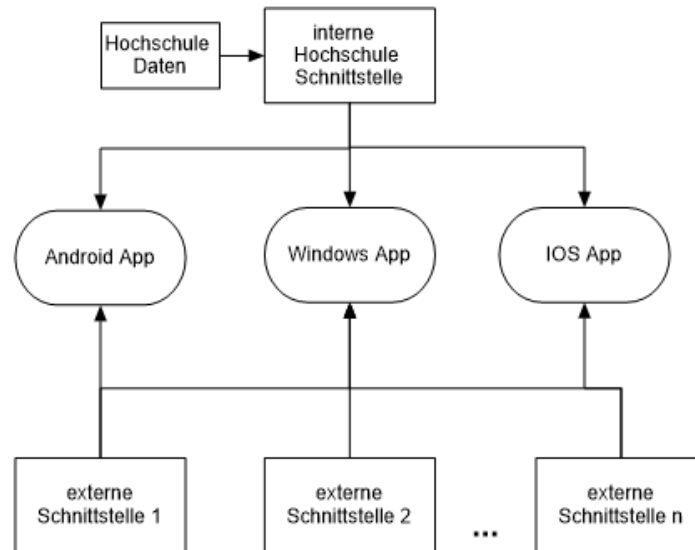
Bereits zu diesem Zeitpunkt stand jedoch fest, dass in folgenden Versionen die App um weitere Funktionen ergänzt werden soll. Dafür wurden die Entwickler des Primuss Portals kontaktiert um eine Programmierschnittstelle für die Noten zu erhalten, sodass in einer der zukünftigen Versionen direkt das Notenblatt als auch die Notenbekanntgabe einzusehen sind. Des Weiteren wurde eine Anfrage an die Deutsche Bahn gestellt um Zugriff auf die Fahrplandaten von Bus und Bahn zu erhalten. Denn es wurde angedacht, eine Funktion zu bieten, in der immer die nächstmögliche Verbindung vom aktuellen Standort aus zur Hochschule angezeigt wird.

### 3 Schnittstelle

Für eine saubere Lösung ist eine gute Programmierschnittstelle unerlässlich. Von einer anderen Gruppe, die an einer App arbeitet, wurde bereits in Zusammenarbeit mit der IT-Abteilung der Hochschule eine Schnittstelle für den Stundenplan und Stundenplanänderungen erstellt. Diese wurde von uns in Absprache mit Herrn Dienstbier, der diese mit der Programmiersprache PHP erstellt hat, weiterentwickelt und entsprechend der neuen Anforderungen angepasst. Die Schnittstelle sollte als zentrale Anlaufstelle für alle Apps zum Holen von Daten dienen. Wodurch man mehrere Vorteile vereint hätte, zum Einen eine vereinfachte Programmierung der App und zusätzlich müsste bei einer Änderungen von einer externen Schnittstellen ausschließlich die betroffenen Funktion auf dem Server angepasst werden und nicht alle Apps auf den verschiedenen Plattformen. Deshalb sollte der Speiseplan der Mensa, der von einer Schnittstelle des Studentenwerks bereitgestellt wird, durch unsere Schnittstelle verarbeitet und aufbereitet weitergegeben werden.



Nachdem dieses vollständig implementiert wurde, beschloss die IT-Abteilung jedoch, dass das in dieser Form nicht durchgeführt werden soll. Deshalb wird der Speiseplan trotzdem wieder direkt vom Studentenwerk über eine aufwändige XML-Struktur geholt, statt ihn von einer zentralen Schnittstelle bereitzustellen. Somit ist davon auszugehen, dass auch alle weiteren Schnittstellen kompliziert in jeder App einzeln umgesetzt werden müssen.



Die Schnittstelle des Speiseplans kann über folgende URL aufgerufen werden:  
[https://www.studentenwerk-oberfranken.de/?eID=bwrkSpeiseplanRss&tx\\_bwrkspeiseplan\\_pi2%5Bbar%5D=340&tx\\_bwrkspeiseplan\\_pi2%5Bdate%5D=2016-04-28](https://www.studentenwerk-oberfranken.de/?eID=bwrkSpeiseplanRss&tx_bwrkspeiseplan_pi2%5Bbar%5D=340&tx_bwrkspeiseplan_pi2%5Bdate%5D=2016-04-28)  
 Neben dem Speiseplan der Mensa in Hof, kann auch der Speiseplan von Münchberg durch Änderung des Parameters 'bar' von 340 auf 370 angezeigt werden. Durch den Parameter 'date' kann die Woche, für den der Speiseplan gelesen werden soll, ausgewählt werden. Dabei muss das Datum eines Tages aus der gewünschten Woche übergeben werden.

Die Beschreibung der Schnittstelle für Stundenplan und Stundenplanänderungen wird beim Aufruf der "docs.php" angezeigt. Auf Wunsch des IT Services ist diese nur im Testsystem verfügbar.

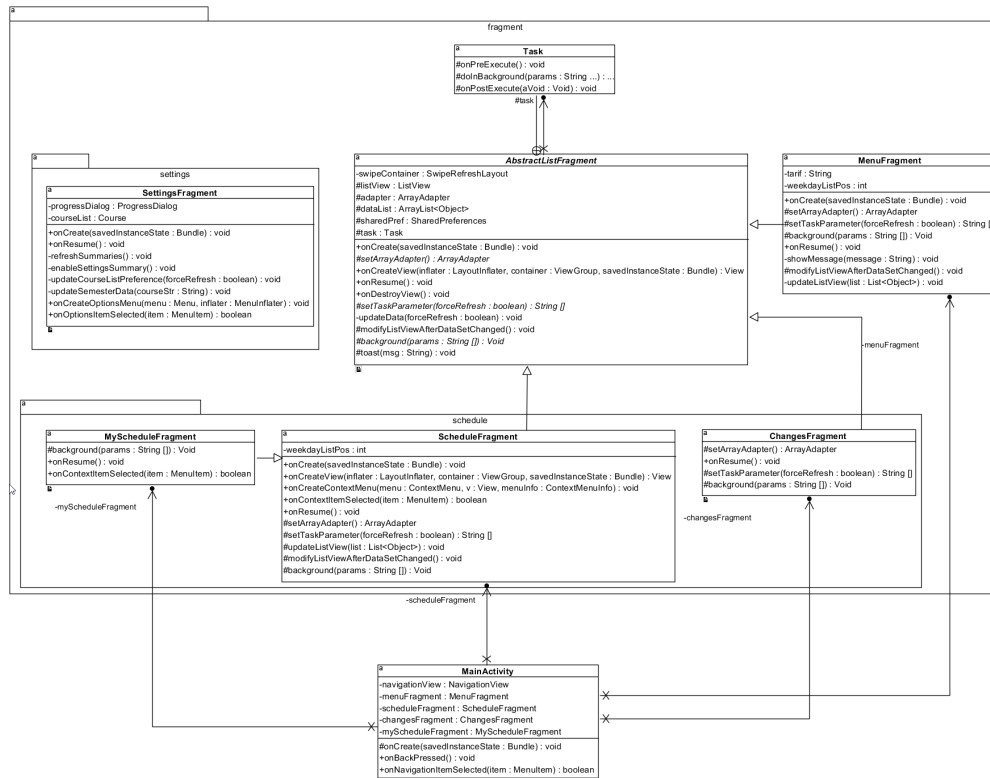
## 4 Entwicklung der App

Die bisherige App musste um einige Funktionen gekürzt werden. Dies lag daran, dass für diese Funktionen aktuell keine Programmierschnittstellen verfügbar sind. Das Parsen des Quelltexts ist nicht für den produktiven Einsatz mit mehreren tausend Studenten geeignet, da dies eine hohe Fehleranfälligkeit mit sich bringt. Des Weiteren wurden Konsolidierungen und Refactorings am bestehenden Code vorgenommen, um die Les- und Wartbarkeit zu erhöhen. Außerdem wurde die Funktion "Mein Stundenplan" implementiert.

Nach Absprache mit Herrn Stepping wurde jedoch beschlossen die Funktionen des Primuss Portals und die Raumsuche wieder als experimentelle Funktionen zu imple-

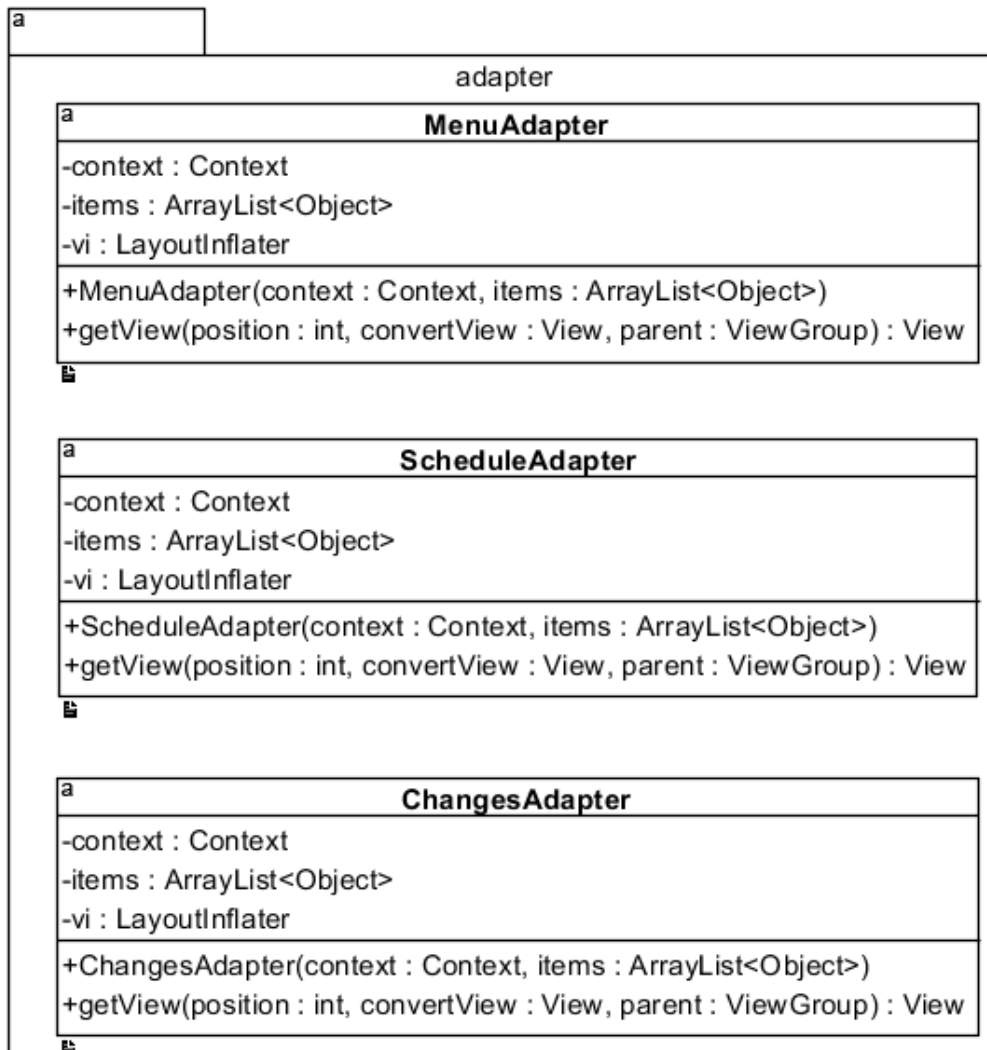
mentieren. Deshalb sind sie wieder in der App enthalten, müssen jedoch in den Einstellungen explizit aktiviert werden.

## 4.1 Fragmente



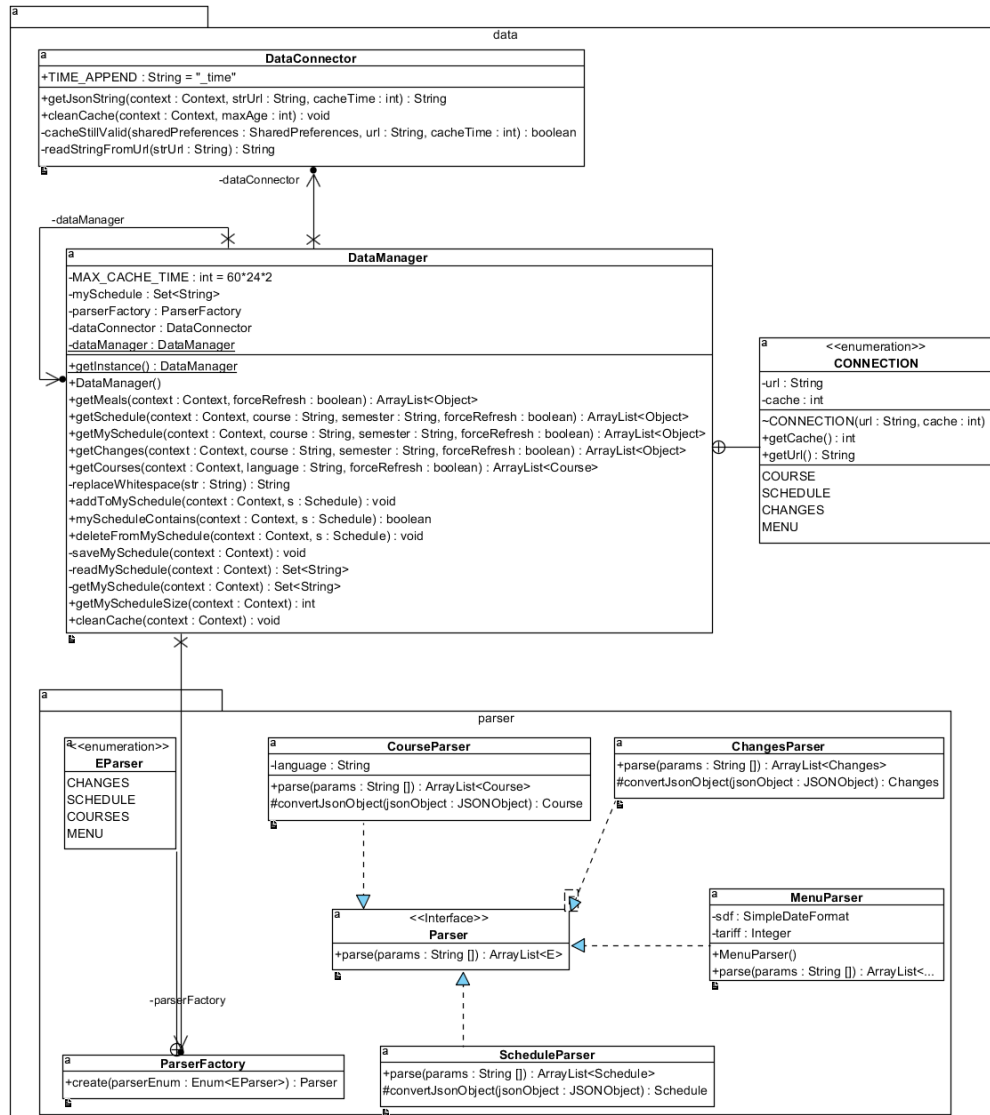
In diesem Package befinden sich alle Fragmente. Da in den meisten Fällen, in den unterschiedlichen Ansichten Daten in Form von einer Liste dargestellt werden, wurde bei der Erstellung der App-Architektur eine weitere Abstraktionsebene in Form einer abstrakten Oberklasse hinzugefügt. Dies ermöglicht es den Fragmenten "Schedule-Fragment", "MyScheduleFragment", "ChangesFragment" und "MenuFragment" die Daten in der beschriebenen Form anzuzeigen. Alle diese Klassen müssen ausschließlich die gewünschten Adapterklassen und den Task mit der benötigten Routine zum Holen der Daten definieren.

## 4.2 Adapter



An dieser Stelle werden alle Adapterklassen zusammengefasst. Sie dienen dazu in den verschiedenen Ansichten die ListViews und deren Listeneinträge zu erzeugen und mit den zur Verfügung stehenden Daten zu befüllen.

### 4.3 Datenhandler

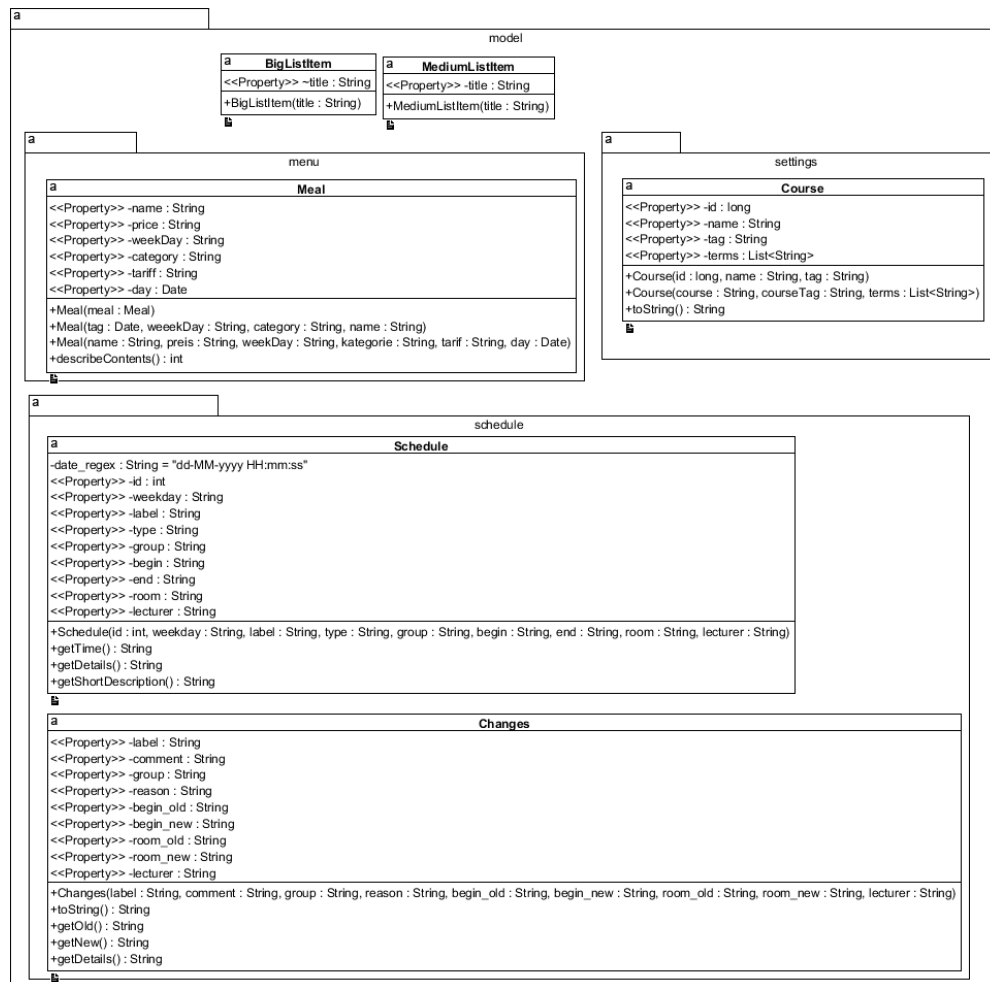


Die Klasse "DataManager" ist der zentrale Punkt, über den auf alle externen Inhalte zugegriffen wird. Deshalb wird diese Klasse über das Singleton-Pattern bereitgestellt. Darüber hinaus verfügt die Klasse über verschiedene Parser, auf welche einheitlich über das Interface "Parser" zugegriffen werden kann. Die konkreten Parser sind jeweils auf eine spezielle Datenstruktur optimiert und geben die gewonnenen Daten zurück. Der "DataConnector" übernimmt die Aufgabe die externen Quellen abzurufen und die dort vorliegenden Daten als String zurückzuliefern. Des Weiteren wird die zuletzt geladene



Version für einen angegebenen Zeitraum gecached und anstelle der zeitaufwendigen Anfrage am Fremdsystem bevorzugt.

## 4.4 Datenstrukturen



Alle Klassen dienen der Datenhaltung und werden in den oben beschriebenen Fragmenten, Adaptern und Datenhandler verarbeitet.

## 4.5 Experimental

Im Package "experimental" wurden alle Funktionen der "alten" App eingefügt, die noch nicht auf eine Programmierschnittstelle und das neue Programmkonzept umgesetzt wurden. Der Aufbau dieser Klassen unterscheidet sich nicht von dem bereits in

der Studienarbeit zu Aspekte der Androidprogrammierung aus dem letzten Semester erläuterten. Deshalb wird dieser Teil hier nicht weiter betrachtet. Es ist generell zu empfehlen, diese Programmteile bei Verfügbarkeit einer Programmierschnittstelle nicht an diese anzupassen, sondern besser entsprechend dem obig beschriebenen Schema komplett neu umzusetzen. Damit entsteht sukzessiv eine konsistente, wartbare Applikation.

## **5 Design**

In Absprache mit der Marketingabteilung der Hochschule sollte das Design der Apps an die Corporate Identity der Hochschule angepasst werden. Im bisherigen CI-Handbuch wurden Apps jedoch noch nicht behandelt, weshalb eine genaue Umsetzung der Vorgaben relativ schwierig war. Gemeinsam mit der Hochschule wurde deshalb ein neues Logo entworfen, was nun als Logo für die Apps dient. Die Hochschulfarben wurden entsprechend des Handbuchs in die App integriert.