

Weiterentwicklung der iOS Stundenplan App der Hochschule Hof

Dokumentation, Spezifikation, Konstruktion

Christian G. Pfeiffer, Johannes Franz, Normen Krug, Marcel Hagmann,
Patrick Niepel, Carl Philipp Knoblauch

22.01.2018

Inhaltsverzeichnis

1	Projektablauf	1
1.1	Ausgangssituation	1
1.2	Überlegungen zu Projektbeginn	2
1.2.1	GitHub / Branches	2
1.2.1.1	Ausgangslage	2
1.2.1.2	Verbesserungen	2
1.3	Ziele für Version 4	3
1.4	Teams	3
1.5	Kommunikation	3
1.6	Fazit des Projektes	4
1.7	Projektfortschritt dokumentiert	4
1.8	Ausblick	4
2	Design	5
2.1	Einleitung	5
2.2	Mockups	5
2.2.1	Screen1	5
3	Erweiterung durch eine Siri Integration	7
3.1	Einleitung	7
3.2	Istzustand	8
3.3	Beispielhafte Anfragen	8
3.4	Umsetzung	8
3.5	Voraussetzung für SiriKit	8
3.6	Einschränkungen	8
4	Push Notifications	11
4.1	Einleitung	11
4.2	Projektablauf	11
4.3	Vorbereitung einer Testumgebung	13
4.3.1	MAMP als Virtuelle Umgebung	13
4.3.1.1	MAMP	13
4.3.1.2	MySQL Datenbank	13
4.4	Umsetzung	13
4.4.1	Server Installation	14
4.4.2	Zertifikate	14
4.4.3	Code zum Registrieren am Server	14
4.5	Aufgetretene Probleme	15
4.6	lessons learned	15
4.7	Weitere Arbeiten	15

5	Testen der Anwendung	17
6	Gruppe PMP	18
6.1	Einleitung	18
6.2	Überblick	18
6.3	Erster Bug	21
7	Aufgaben Erweiterung	22
7.1	Einleitung	22
7.2	Planung und Mockup	23
7.3	Funktionen	24
8	Kalenderschnittstelle	25
8.1	Einleitung	25
8.2	Überarbeitung	25
9	Onboarding	26
9.1	Einleitung	26
9.2	Umsetzung	26
10	Universal App Colors	27
10.1	Beschreibung	27

1 Projektablauf

Johannes Franz & Christian Pfeiffer

1.1 Ausgangssituation

Im Wintersemester 2016/2017 begann die Entwicklung der iOS Stundenplanapp mit Studierenden des Studiengangs Mobile Computing im 5. Semester im Rahmen des Moduls "Fortgeschrittene Themen der Swift 3 Programmierung". Am Ende des Semesters wurde die Version V1 in den AppStore veröffentlicht. Im Sommersemester 2017 wurde die App weiterentwickelt und schlussendlich wurde Version 2.0 veröffentlicht, welche eine Background Fetch Funktionalität einführte, lokale Notifications bei Vorlesungsverlegungen und Verbesserungen bei der Erkennung von einzelnen Vorlesungsterminen einführte.

Im Wintersemester 2017/2018 übernahmen die Studierenden des 5. Semesters im Rahmen des Moduls "Fortgeschrittene Themen der Swift 3 Programmierung" die Pflege und Weiterentwicklung der iOS Stundenplanapp.

1.2 Überlegungen zu Projektbeginn

1.2.1 GitHub / Branches

GitHub ist ein Plattform zur effektiven Versionsverwaltung von Softwareprojekten. Branches stellen gewisse Softwareversionen da. Das Verwenden mehrerer Branches ermöglicht es größeren Softwareteams gleichzeitig an verschiedenen Softwarefeatures zu arbeiten. Für die iOS Stundenplanapp wurde ein passendes Branch Konzept vom Projekt Team ausgearbeitet.

Der Großteil der Kommunikation lief über Issues und den Project Tab von Github ab.

1.2.1.1 Ausgangslage

Zu Projektbeginn wurden folgende Branches vorgefunden:

- master
- development (obsolete)
- v2 (depricated)
- v3 (aktuell auf Deployment Target 10.0 entspricht iOS Version 10)

Von den vorherigen Projektteams wurde in GitHub ein Wiki angelegt. Link zum Wiki:

<https://github.com/HochschuleHofStundenplanapp/iOS-App/wiki>

Darin wurden im Notizartig und in kurzer Form einige Auszüge aus den jeweiligen PDF Dokumenten zusammengetragen.

GitHub Branch Übersicht:

<https://github.com/HochschuleHofStundenplanapp/iOS-App/branches>

1.2.1.2 Verbesserungen

Es wurde beschlossen, die vorherige Struktur beizubehalten. Dabei wurden Branches jeweils mit der Versionsnummer beschriftet.

Der "master" Branch wird immer mit der aktuellsten Version gefüllt.

Neue Branches:

- v3.1 (Bugfixes, kleinere neue "Features". Der Branch bleibt auf Swift 3.1 / iOS 10.0)
- v3.2 Weitere Bugfixes. Der Branch bleibt auf Swift 3.1 / iOS 10.0
- v4 in Rahmen der Studienarbeit entwickelte Erweiterungen der App (Swift 4 / iOS 11)

Es wurde sich dafür entschieden beim iOS 10 deployment Target zu bleiben, um möglichst viele unter den Studierende verbeitete, alte Geräte zu unterstützen.

Version 4 wurde im laufe des Projekts auf Swift 4 geupgradet, bevor weitere Funktionen eingebaut wurden.

Zu Projektbeginn entstand die Idee, Branches nach Features zu benennen. Dabei war Design, Siri, Kalendersynchronisation, etc. angedacht. In der Projektphase hat sich zwischenzeitlich bewährt, die Branches weiterhin nach Versionsnummern zu benennen und zwischenzeitlich bei großen Änderungen ein Thema anzuhängen.

1.3 Ziele für Version 4

Folgende Aufgaben wurden als Ziel für die Version 4 festgelegt:

- Onboarding
- Hausaufgaben Manager
- Push Notifications
- Widget
- iOS 11 Design
- Testkonzept für die App

1.4 Teams

Team 1: (Pfeiffer, Scheler)

- Design
- Buxfixes
- Swift 4 Konvertierung

Team 2: (Franz, Krug):

- Siri
- Push Notifications

Team 3: (Hagmann, Knoblauch, Niepel):

- Verschiedene Fehlerbehebungen
- Hausaufgaben Manager
- Onboarding
- Universal App Color
- Kalenderschnittstelle überarbeitet

Team 4: (Kusserow, Sonntag, Dümmlein):

- Stundenplan Verbesserungen
- Erstellen eines Widgets

Team 5: (Pöhlmann):

- Testkonzept ausarbeiten und Testen nach Protokoll

1.5 Kommunikation

- Chat Gruppe mit allen Beteiligten
- Kommunikation während der ZEit in der Hochschule
- GitHub (Issues, Project Sektion)
- Scrum ähnliche Vorstellung neu eingebauter Funktionen zu Beginn jeder Vorlesung

1.6 Fazit des Projektes

Herausforderungen:

- Überblick bewahren
- Git Projekt und Issues im Blick zu haben

1.7 Projektfortschritt dokumentiert

Projekt Tab in Github

<https://github.com/HochschuleHofStundenplanapp/iOS-App/projects> Scrum ähnliches arbeiten

1.8 Ausblick

2 Design

Angelina Scheler & Christian Pfeiffer

2.1 Einleitung

Eine weitere Aufgabe, die unsere Gruppe übernommen hat, war das Onboarding für die Stundenplan App. Im Onboarding kann der Nutzer gleich zum Start der App seine Einstellungen zu Fakultät, Studiengang, Semester, Vorlesung und Kalendersynchronisation vornehmen. Das Onboarding wird gestartet, wenn noch keine Angaben zu den genannten Einstellungen gemacht wurden.

- Neues Design
- besseres Design
- iOS Design

2.2 Mockups

Unsere Design Sachen sind sehr schön!

2.2.1 Screen1

1. Fakultät (die die Farbe der App bestimmt)

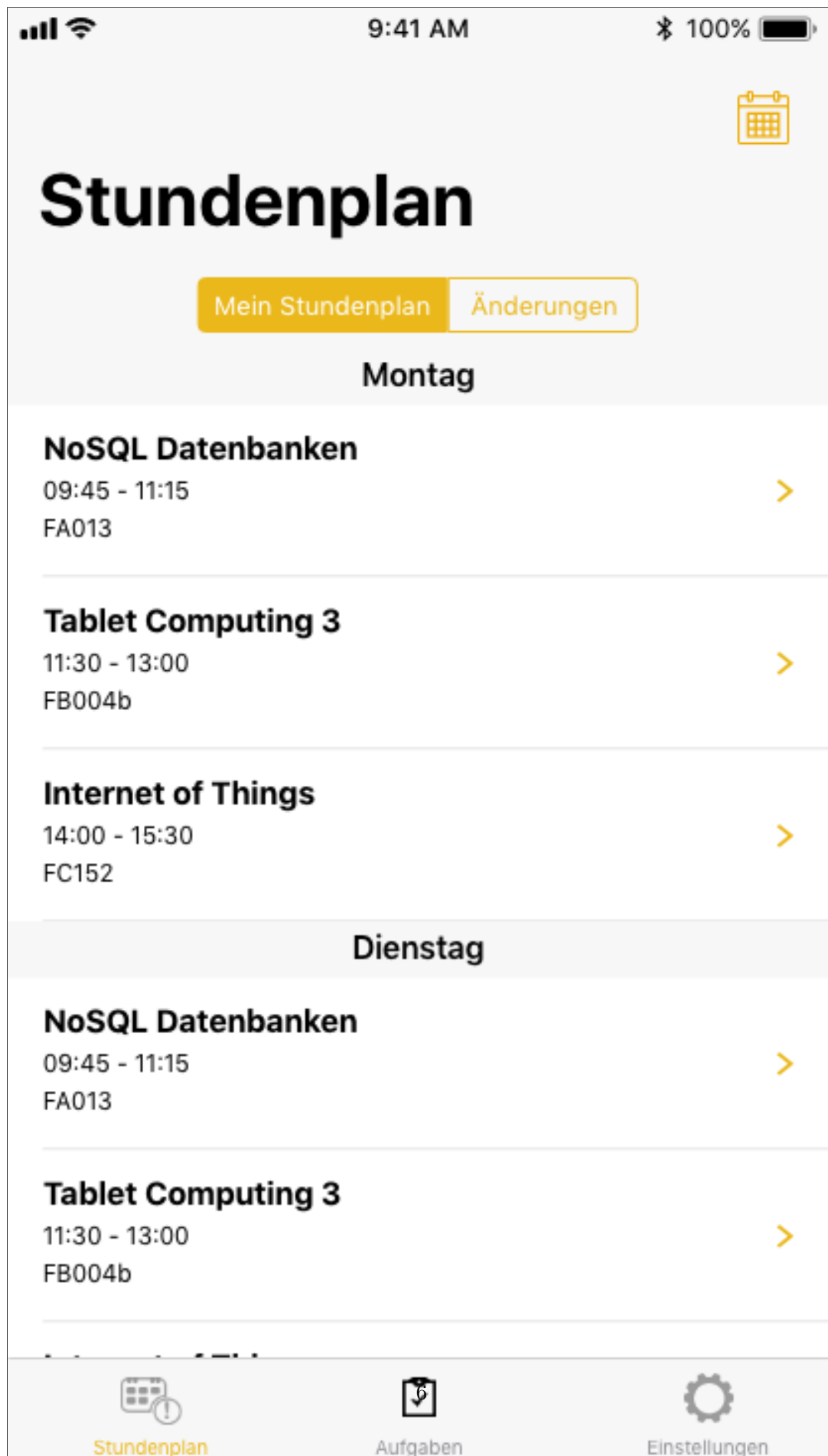


Abbildung 2.1: Mein schönstes Mockup

3 Erweiterung durch eine Siri Integration

Johannes Franz & Normen Krug

3.1 Einleitung

Da sprachbasierte Mensch-Maschinen-Interface immer beliebter und praktikabler werden, ist es naheliegend, dass die Stundenplan App um diese erweitert wird. Apple bietet mit Siri solch einen Sprachassistenten ein. Dieser ist tief im System eingebaut und wird daher von Apple gut unterstützt. Da es weiterhin das Ziel sein soll, die App zur Nutzung nicht öffnen zu müssen, ist eine Siri Integration der nächst logische Schritt zur Weiterentwicklung der Anwendung.

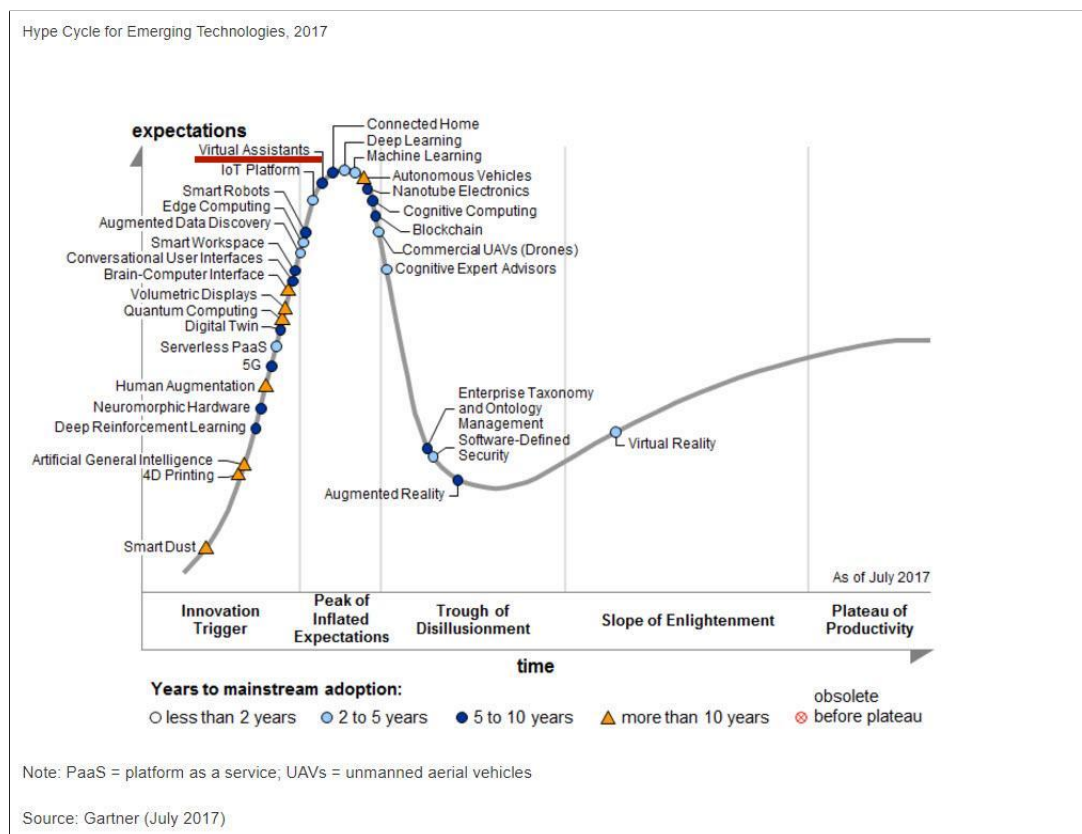


Abbildung 3.1: hype cycle for technologies 2017

3.2 Istzustand

Die App bietet aktuell keine Unterstützung für Sprachbefehle an. Eine Eingabe durch Sprachbefehle durch die Apple Watch ist so ebenfalls nicht möglich. Im Hinblick auf barrierefreie Bedienung hat die App daher noch Verbesserungspotential.

3.3 Beispielhafte Anfragen

Um die gängigen Anfragen abzudecken, müssen diese in der App vorher festgelegt werden. Um das Anliegen des Benutzers verstehen zu können müssen unterschiedliche Fragestellungen die zum selben Ergebnis führen abgedeckt werden.

Frage	Reaktion
Wann/Wo ist meine nächste Vorlesung?	Name der Vorlesung, Zeitspanne und Raumnummer vorlesen. UI zeigt diese Information noch einmal an.
Habe ich heute noch eine Vorlesung?	Ja/Nein Antwort, Name der Vorlesung mit Uhrzeit. Wenn ja wird im UI etwas angezeigt.
Fällt heute etwas aus?	Ja/Nein Antwort, Name der Vorlesung mit Uhrzeit mit Grund. Wenn ja wird im UI etwas angezeigt.
Welche Änderungen gibt es heute?	Aufzählung der Änderungen für heute. Auflisten dieser Änderungen im UI
In welchem Raum ist Vorlesung X? (optional/tricky)	Name der Vorlesung und Raum wird genannt. Informationen im UI werden angezeigt.

3.4 Umsetzung

Apple bietet für die Umsetzung des Sprachassistenten das SiriKit an. Mittels einer “Applications Extension” namens “Intents Extension” ist es möglich Hooks für eine Reaktion der Stundenplan App zu definieren. Für die Ausgabe im Lockscreen kann dabei ein angepasstes UI zur Verfügung gestellt werden, welches die Sprachausgabe ergänzt.

3.5 Voraussetzung für SiriKit

Stand: 10.10.2017

Um SiriKit verwenden zu können, müssen die Kernbereiche der App in ein Framework ausgelagert werden. Apple empfiehlt das bei allen Erweiterungen. Dieser Schritt ist notwendig, weil der Benutzer eine Interaktion mit Siri starten kann auch wenn die App zurzeit nicht läuft.

3.6 Einschränkungen

Apple gewährt keinen vollständigen Zugriff auf Siri. Die zu entwickelte App muss in eine der folgenden Kategorien/Domains fallen:

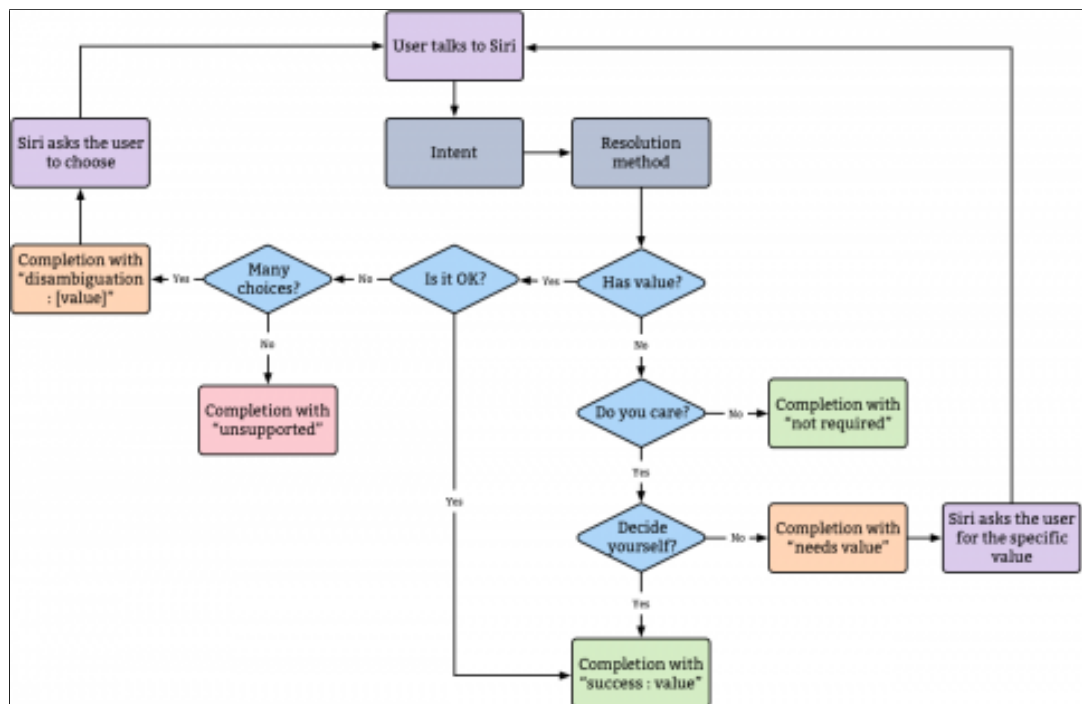


Abbildung 3.2: Siri Flowchart

- VoIP Calling
- Messaging
- Payments
- Lists and Notes
- Visual Codes
- Photos
- Workouts
- Ride Booking
- Car Commands
- CarPlay
- Restaurant Reservations

Die Schlüsselwörter welche Siri voraussetzt, um zu erkennen dass der Benutzer mit der App interagieren will, hängen von der Domain ab. Diese Schlüsselwörter müssen zwingend in der Anfrage des Benutzer enthalten sein.

Beispiel:

In der „Search Message“ Domain müssen die Wörter „Suche“ und „Nachrichten“ enthalten sein. Falls eines der beiden Wörter nicht in der Anfrage enthalten ist, erkennt Siri die Anfrage nicht.

Passender Blog-Post zu den Thema:

<https://swifting.io/blog/2016/07/18/20-sirikit-can-you-outsmart-provided-intents/>

Mögliche Workarounds:

Es ist theoretisch möglich mit der „Lists and Notes“ Domain, die Funktion für die Stundenplan hinzubiegen. Da aber die bestimmten Schlüsselwörter enthalten sein müssen, wird aber kein natürlich sprachliche Interaktion möglich sein. <https://developer.apple.com/documentation/sirikit>

4 Push Notifications

Johannes Franz & Normen Krug

4.1 Einleitung

Push Notifications sind ein fester Bestandteil moderner Apps. Deswegen soll die Stundenplan App um solche erweitert werden. Ziel soll es sein den Benutzer über Stundenplanänderungen aktiv zu informieren, um speziell auf kurzfristige Änderungen reagieren zu können. Ein Server überprüft dabei eine Stundenplan Datenbank auf Änderungen und sendet eine Push Notification an alle iOS Geräte, die sich für die jeweilige Vorlesung registriert haben. Ein Apache Webserver stellt dabei mit einer MySQL Datenbank und entsprechenden Chronjobs das Backend bereit.

GitHub repository: <https://github.com/HochschuleHofStundenplanapp/iOS-App/>

4.2 Projektablauf

Um den Projektfortschritt nachvollziehen zu können, wurde dieser tabellarisch in Verbindung mit dem aktuellen Datum aufgelistet.

Datum	Erreichter Meilenstein
Vorlesungsbeginn	Themenfindung
Siri	Einarbeitung in Siri. Erkennen erster Hürden.
Neue Themenfindung	Verwerfen von der Siri Projektidee und neue Themenfindung.
Festlegung	Thema Push Notifications festgelegt und Beginn der Einarbeitung.
28.10.2017	Push Notifications lassen sich per PHP Script an einen fest installierten Token schicken
30.10.2017	Eine Test iOS App kann per php Script mittels MAMP lokal eine Push Notification senden. Datenbank lokal in PhpMyAdmin angelegt. PHP Script schreibt bei Aufruf in die angelegte SQL Datenbank. Einarbeitung und Konvertierung der Dokumentation in Latex.
31.10.2017	Die Test iOS App wurde so erweitert, dass ein JSON File per POST Nachricht übermittelt werden kann. Das PHP Script parst nun das ankommende JSON File und fügt per insert die geparsten Daten in die Datenbank ein. Einarbeitung in bestehende Schnittstelle und Überlegungen wie das bestehende Backend erweitert werden muss.
Zwischenzeit	PHP Scripte der bestehenden (Android) Schnittstelle angepasst und getestet.
24.11.2017	Testserver wurde bereitgestellt.
2018	Push Notifications mit HTTP2 implementiert.

4.3 Vorbereitung einer Testumgebung

4.3.1 MAMP als Virtuelle Umgebung

4.3.1.1 MAMP

Die Testumgebung MAMP (Akronym steht für: Mac, Apache, MySQL, PHP) virtualisiert die genannten Komponenten, um lokale Tests zu ermöglichen.

4.3.1.2 MySQL Datenbank

Die Tabelle *fcm_nutzer* enthält eine Zuordnung von abonnierten Vorlesungsverlegungen mit dem jeweiligen Token des Gerätes. Dabei ist zusätzlich vermerkt welches Betriebssystem verwendet wird. Die *0* in der Spalte *os* steht dabei für Android während *1* iOS repräsentiert. Da die ausgewählte Sprache des Nutzers nicht bekannt ist, wird für die *language* Spalte an dieser Stelle auch null akzeptiert. Um bei einer späteren Version der App zwischen Nutzern die noch keine Sprache ausgewählt haben und allen anderen differenzieren zu können, wird hier trotz der aktuell nur in deutsch vorhandenen App der Wert nicht standardmäßig auf deutsch gesetzt.

```
CREATE TABLE 'fcm_nutzer' (  
  'id' int(11) NOT NULL AUTO_INCREMENT,  
  'token' varchar(255) CHARACTER  
    SET utf8 COLLATE utf8_unicode_ci NOT NULL,  
  'vorlesung_id' varchar(255) CHARACTER  
    SET utf8 COLLATE utf8_unicode_ci NOT NULL,  
  'os' int(11) DEFAULT '0',  
  'language' varchar(2) DEFAULT NULL,  
  PRIMARY KEY ('id'),  
  KEY 'token' ('token'),  
  KEY 'vorlesung_id' ('vorlesung_id')  
);
```

Da eine bestehende Infrastruktur die Grundlage dieses Projektes darstellt, musste diese Tabelle lediglich um *os* und *language* erweitert werden.

Auf Änderungen an anderen Tabellen konnte komplett verzichtet werden.

4.4 Umsetzung

Apple bietet für die Umsetzung von Push Nachrichten den Apple Push Notification service (APNs) an. Dabei handelt es sich um einen bei Apple gehosteten Dienst, der Push Notifications per API ermöglicht.

4.4.1 Server Installation

Zur Installation der Software sind Admin Rechte notwendig. Der *wget* Befehl lädt dabei eine zum aktuellen Zeitpunkt sehr neue *curl* Version herunter. Diese hat die Besonderheit HTTP2 zu unterstützen, welches für die Push Notification Schnittstelle von Apple vorausgesetzt ist.

```
wget https://curl.haxx.se/download/curl-7.54.0.tar.bz2
tar -xvjf curl-7.54.0.tar.bz2
cd curl-7.54.0/
./configure --with-nghttp2 --prefix=/usr/local --with-ssl=/usr/local/ssl
make
make install
ldconfig

ggf. php7.0-curl erneut installieren
// Apache neu starten
```

4.4.2 Zertifikate

Um mit einer gesicherten Verbindung auf die Apple Push Notification Schnittstelle zuzugreifen, werden zwei Zertifikate benötigt. Dabei handelt es sich um ein öffentliches und privates Zertifikat. Diese Zertifikate müssen vor der Verwendung in das Zielformat *.pem* umgewandelt werden, bevor sie benutzbar sind.

Um die Zertifikate zu anzulegen und zu verwalten, stellt Apple eine Übersicht dem bei Apple angemeldeten Entwickler bereit. Bei dem verwendeten Zertifikat wird zwischen *Development* und *Production* unterschieden.

<https://developer.apple.com/account/ios/certificate/>

Die MacOS App “Easy APNs Provider“ ermöglichte ersten Gehversuche, um Push Nachrichten an ausgewählte Token zu senden.

Quelle: <https://github.com/immobiliare/ApnsPHP>

4.4.3 Code zum Registrieren am Server

Um sich am Server für eine Vorlesung zu registrieren waren Veränderungen im Code notwendig.

```
func application(_ application: UIApplication,
    ↪ didRegisterForRemoteNotificationsWithDeviceToken deviceToken: Data) {
    print("DeviceToken1:␣\(deviceToken)")
    let token = deviceToken.map { String(format: "%02.2hhx", $0) }.joined()
    print("DeviceToken:␣\(token)")
}
```

4.5 Aufgetretene Probleme

Als Herausforderung zu sehende Punkte haben häufig sehr viel Zeit in Anspruch genommen oder den Projektfortschritt entschleunigt.

- Verzögerte Bereitstellung des Servers führte zu komplexen Reimplementierungen in schwer zu synchronisierenden getrennten VMs
- Beschränkte Rechte auf dem Testserver die nach und nach erweitert werden mussten
- Verschiedene Zertifikatarten sind sehr verwirrend
- Bundle Identifier und App Capabilities mussten nach jedem Git Pull wieder angepasst werden
- Komplexes Testen (Lokale Server, Erreichbarkeit im Labornetzwerk / Wi-Fi, unterschiedliche Datei- und Serverstände)
- Zu Beginn kein Zugriff auf die Git Schnittstellen Projekt
- Ablaufende Zertifikate
- Serverseitig unterschiedliche Übertragungsverfahren zwischen Android und iOS
- Erschwertes Debugging der PHP Skripte
- HTTP2 wurde von Apache2 und Curl nicht offiziell unterstützt
- Struktur der Datenbank und der PHP Skripte war durch die Android App vorgegeben
- Veraltete APNs Variante war stark verbreitet, brachte allerdings viele Probleme mit sich und musste als Ansatz letztendlich verworfen werden

4.6 lessons learned

Unser Team hat feststellen müssen, dass zu einem umfassenden iOS Projekt mehr als nur der Interface Builder und die Sprache Swift gehört. So sind Punkte zu nennen die positiv aus dem Projekt mitgenommen wurden.

- Linuxkenntnisse
- PHP und Debugging auf dem Server
- Beachten von Abhängigkeiten wie der Abwärtskompatibilität der Software zu früheren und bestehenden Android Versionen

4.7 Weitere Arbeiten

In diesem Projektabschnitt konnten alle gesteckten Ziele erreicht werden. Da solch ein relativ junges Projekt noch viele Möglichkeiten beinhaltet Funktionen zu verbessern und neue Funktionen einzuführen werden hier mögliche Punkte zur Anregung aufgelistet.

- Auswertungen die aktuell bei der Android und iOS App mehrfach auf dem Client Gerät stattfinden, können auf dem Server zentral bearbeitet werden. Dies verringert die Komplexität bei beiden Anwendungen und erleichtert die Wartung des Codes an zentraler Stelle.
- Erweiterung der Push Notification Information um eine Priorität, Ablaufdatum,...
- Aufkommende Issues und Ideen aus dem öffentlichen Git Projekt der iOS App und Schnittstelle
- Erweiterung der App und Schnittstelle um andere Sprachen wie z.B. Englisch was in der Android App bereits angeboten wird

5 Testen der Anwendung

Text der Tests...

6 Gruppe PMP

Patrick Niepel & Marcel Hagmann & Carl Philipp Knoblauch

6.1 Einleitung

Unsere Gruppe besteht aus Patrick Niepel, Marcel Hagmann und Carl Philipp Knoblauch. In diesem Team haben wir neben viel Debugging ...

- Aufgaben Erweiterung
- Überarbeitung der Kalenderschnittstelle
- Universal App Colors Funktion
- Onboarding

erarbeitet.

6.2 Überblick

Eine Übersicht über unsere Arbeit über das Semester für Fortgeschrittene Programmierung unter Swift 3.

Datum	Aufgaben/Vorlesung	Was wir gemacht haben
KW 40 02.10.17 - 08.10.17	<ul style="list-style-type: none"> • Einleitung der Vorlesung • GitHub Projekt vorgestellt • In Gruppen aufgeteilt • Erster Bug vorgestellt • Neue Aufgabe: Ersten Bug finden • Neue Aufgabe: Themensuche 	<ul style="list-style-type: none"> • Unsere Gruppe: Marcel Hagmann, Patrick Niepel, Carl Philipp Knoblauch • Marcel Hagmann findet ersten Bug • Themensuche Ideen: Aufgaben Erweiterung, Widget, Machine Learning, ... • Ausarbeitung der Idee Aufgaben Erweiterung (Vorstellungen, Aufbau, Mockup)
KW 41 09.10.17 - 15.10.17	<ul style="list-style-type: none"> • Neue Aufgabe: Aufgaben Erweiterung 	<ul style="list-style-type: none"> • Marcel Hagmann stellt ersten Bug vor und behebt ihn • Vorstellung der Aufgaben Erweiterung • Programmierung der Aufgaben Erweiterung
KW 42 16.10.17 - 22.10.17	<ul style="list-style-type: none"> • Weiter an der Aufgaben Erweiterung arbeiten 	<ul style="list-style-type: none"> • Push: BugFix von Marcel (16.10.17) • Push: Aufgaben Erweiterung (20.10.17)
KW 43 23.10.17 - 29.10.17	<ul style="list-style-type: none"> • Neue Aufgabe: Kalenderschnittstelle 	<ul style="list-style-type: none"> • Weiterer Bug entfernt, daySize (23.10.17) • Komplette Umstrukturierung der Kalenderschnittstelle
KW 44 30.10.17 - 05.11.17	<ul style="list-style-type: none"> • Weiter an der Kalenderschnittstelle arbeiten 	<ul style="list-style-type: none"> • Push: Kalenderschnittstelle (4.11.17)
KW 45 06.11.17 - 12.11.17	<ul style="list-style-type: none"> • Neue Aufgabe: Überarbeitung des Design und der Icons für die Aufgaben Erweiterung 	<ul style="list-style-type: none"> • Bug gefunden: Datumsberechnungsfehler (nahezu Endlosschleife) • Push (Bug): Datumsberechnungsfehler behoben (10.11.17)
KW 46 13.11.17 - 19.11.17	<ul style="list-style-type: none"> • Neue Aufgabe: App Color Design 	<ul style="list-style-type: none"> • Einarbeitung in Latex • Überarbeitung des Designs der Aufgaben Erweiterung • Push: des neuen Aufgaben Designs (13.11.17) • App Color Design (Farbe für App in Einstellungen auswählbar) • Push: App Color Design (17.11.17)

Datum	Aufgaben/Vorlesung	Was wir gemacht haben
KW 47 20.11.17 - 26.11.17	<ul style="list-style-type: none"> • Neue Aufgabe: Onboarding 	<ul style="list-style-type: none"> • Einarbeitung und Planung des Onboardings • Programmierung des Onboardings • Vorstellung des Onboardings (24.11.17) • Kalenderschnittstellen Debugging (KalenderID wurde nicht persistent gespeichert)
KW 48 27.11.17 - 03.12.17	<ul style="list-style-type: none"> • Weiter am Onboarding arbeiten 	<ul style="list-style-type: none"> • Programmierung des Onboardings • Debugging des Onboardings
KW 49 04.12.17 - 10.12.17	<ul style="list-style-type: none"> • Neue Aufgabe: Aufgaben auch im Kalender anzeigen (Termin) • Bug: Einstellungen Synchronisation: Man kann nichts drücken • Bug: Beim App schließen bricht Kalendersynchronisation ab (wird nur zum teil ausgeführt) 	<ul style="list-style-type: none"> • Programmierung: Aufgaben auch im Kalender anzeigen (In den Notizen) • Debugging • Einarbeitung in Latex
KW 50 11.12.17 - 17.12.17	<ul style="list-style-type: none"> • Bug: Beim Löschen einer Aufgabe wird diese nicht aus den Notizen im Kalender entfernt 	<ul style="list-style-type: none"> • Einarbeitung in Latex • Dokumentation schreiben
KW 51 18.12.17 - 24.12.17	<ul style="list-style-type: none"> • Bug: Farben im Onboarding nicht richtig übernommen 	<ul style="list-style-type: none"> • Push: Bei Änderungen an Tasks, werden diese nun auch in den Kalender übernommen (18.12.17) • Onboarding Farben gefixt
KW 2 08.01.18 - 14.01.18	<ul style="list-style-type: none"> • Neue Aufgabe: Push-Notifications in Onboarding integrieren • Neue Aufgabe: Roter Punkt bei jeder Vorlesung, für die eine Aufgabe offen ist. 	<ul style="list-style-type: none"> • Aufgaben Erweiterung überarbeitet (Rote Punkte) • Onboarding erweitert (Push-Notifications)
KW 3 15.01.18 - 21.01.18	...	test
KW 4 22.01.18 - 28.01.18	Abgabe	test

6.3 Erster Bug

- Wintersemester Sommersemester segmented Control ist verbuggt.
- Änderungen werden gelöscht.
- Alle Vorlesungen mit Kommentar werden in den Kalender geschrieben, aber alle anderen sind im Kalender nicht vorhanden.

7 Aufgaben Erweiterung

Patrick Niepel & Marcel Hagmann & Carl Philipp Knoblauch

7.1 Einleitung

Unsere erste Aufgabe war die Erweiterung der Stundenplan App um ein Aufgaben Feature. Mit dem Aufgaben Feature kann der Nutzer seine Aufgaben aus Vorlesungen in die App eintragen, die dann mit dem Kalender synchronisiert werden.

7.2 Planung und Mockup

Zuallererst machten wir uns Gedanken darüber, welche Informationen der Nutzer beim Hinzufügen seiner Aufgaben in der App angeben muss und möchte. Anhand dieser Erkenntnisse, fertigten wir das Mockup an.

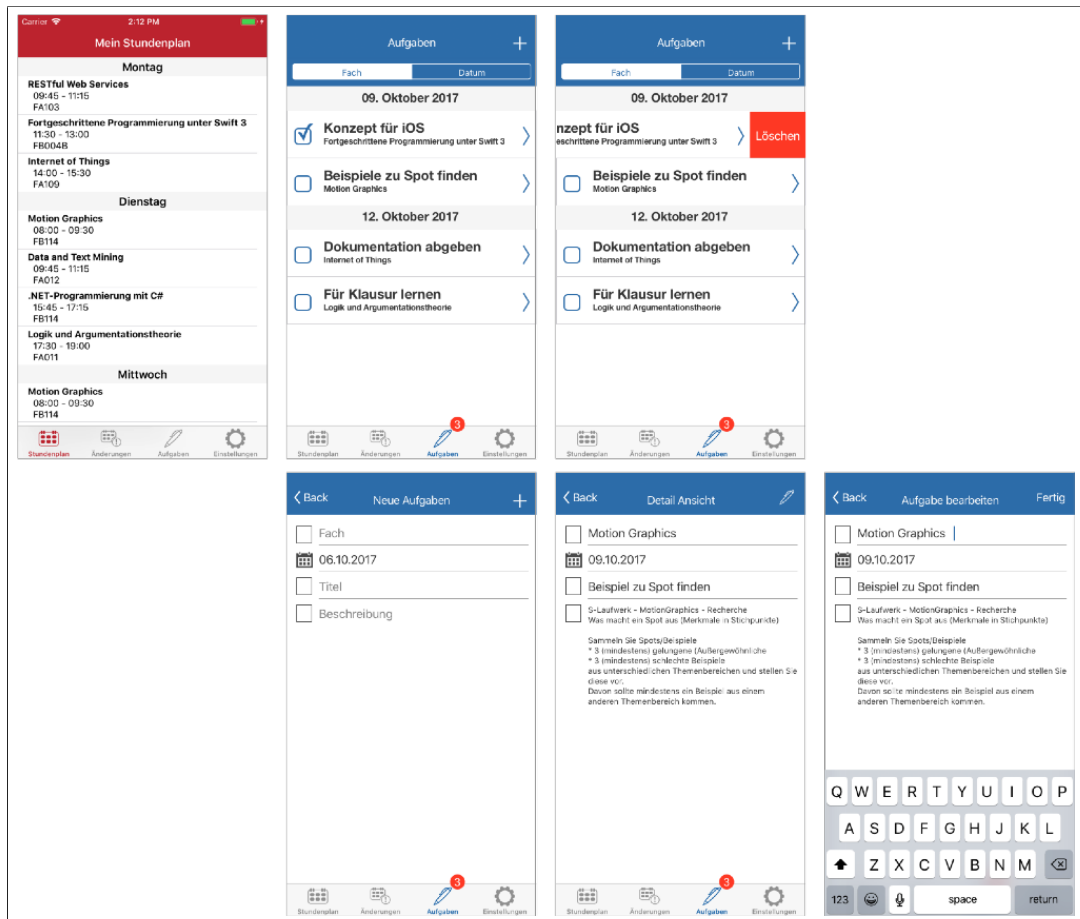


Abbildung 7.1: Mockup unserer Aufgaben Erweiterung

7.3 Funktionen

- Sortierung nach Datum und Fach
- Hinweis der noch zu erledigenden Aufgaben mit einem Badge in der Tab-Bar
- Hinzufügen, Löschen und Bearbeiten einer Aufgabe
- Vorlesungen in der Wochenübersicht, denen eine Aufgabe zugeteilt wurde hervorheben
- Aufgabe mit dem Kalender synchronisieren
- Speichern der Aufgaben in UserData

8 Kalenderschnittstelle

Patrick Niepel & Marcel Hagmann & Carl Philipp Knoblauch

8.1 Einleitung

Da Code öfters über die Zeit an Übersichtlichkeit und Korrektheit verliert, wie es in der Kalenderschnittstelle der Fall war, muss dieser in regelmäßigen Abständen kontrolliert und überarbeitet werden.

8.2 Überarbeitung

Eigentlich wollten wir nur die folgenden Klassen überarbeiten:

- CalendarController
- CalendarInterface

Allerdings waren unsere Änderungen so groß, dass folgende Klassen auch davon betroffen waren:

- DateExtension
- NotificationNameExtension
- SettingsController
- SettingsTableViewController

Als erstes überarbeiteten wir den einfachen Teil, die Übersichtlichkeit des Codes. Da Code von oben nach unten gelesen wird und beim Lesen des alten Codes viel hin und her gesprungen werden musste, ordneten wir zu allererst die Reihenfolge der einzelnen Methoden an.

Danach überprüften wir jede Methode auf Logikfehler und bemerkten dabei, dass unsere Vorgänger an der ein oder anderen Stelle gefuscht haben, wodurch sich dann Folgefehler durch das gesamte Programm zogen. Einer dieser Fehler war beispielsweise der, dass in gewissen Vorlesungsstunden der Vorlesungsbeginn in der Klasse **JsonLectures**, falsch berechnet wird und so anstelle von 2017 das Jahr 0017 ausgegeben wurde, was dazu führte, dass die Semestervorlesungsstunden-Berechnung nicht von 9 Millionen (dass ein paar Sekunden dauert) von 6 Milliarden berechnet wurde und somit dazu führte, dass die App für den Nutzer “einfriert”. Wir fanden diesen Fehler und entfernten diesen und damit auch die dadurch unnötig gewordenen Notifications, die die Vorgänger dazu verwendeten um vorzeitig aus dieser “fast Endlosschleife” auszubrechen.

Des Weiteren sorgten wir für eine bessere Behandlung der Fehler die während der Ausführung der Operationen auftreten konnten, dass die Kalendersynchronisation im Hintergrund auch nach dem Schließen der App weiter läuft, der Kalender anhand seiner ID gesichert wird und der alte Kalender dementsprechend durch den neuen ausgetauscht wird.

9 Onboarding

Patrick Niepel & Marcel Hagmann & Carl Philipp Knoblauch

9.1 Einleitung

Eine weitere Aufgabe, die unsere Gruppe übernommen hat, war das Onboarding für die Stundenplan App. Im Onboarding kann der Nutzer gleich zum Start der App seine Einstellungen zu Fakultät, Studiengang, Semester, Vorlesung und Kalendersynchronisation vornehmen. Das Onboarding wird gestartet, wenn noch keine Angaben zu den genannten Einstellungen gemacht wurden.

9.2 Umsetzung

Zuerst überlegten wir uns, welche Informationen die App vom Nutzer benötigt, damit der Dienst im vollen Umfang verwendet werden kann. Dabei kamen wir auf folgendes Ergebnis:

1. Fakultät (die die Farbe der App bestimmt)
2. Abfrage der Fakultät
3. Abfrage des Studiengangs
4. Abfrage des Semesters
5. Abfrage der besuchten Vorlesungen
6. Ob eine Synchronisation mit dem Kalender gewünscht ist
7. Ob Push-Benachrichtigungen gewünscht sind

Das gesamte Onboarding befindet sich in einem separaten Storyboard (Onboarding.storyboard), damit die Übersichtlichkeit des aktuellen Projektes weiterhin gewährleistet wird. Jeder Schritt im Onboarding (Ausnahme: Kalendersynchronisation) kann nur fortgesetzt werden, wenn eine Auswahl getroffen worden ist, die für die weiteren Schritte zwingend notwendig ist.

10 Universal App Colors

Patrick Niepel & Marcel Hagmann & Carl Philipp Knoblauch

10.1 Beschreibung

Bisher hatte die Stundenplan-App für jeden Tab eine andere Farbe. Wir haben uns darum gekümmert, dass der Nutzer in den Einstellungen zwischen den Hochschulfarben auswählen kann und diese dann für die ganze App übernommen werden. Die Hochschulfarben repräsentieren die Fakultät an der Hochschule. Vorher wurden die verschiedenen Farben im Storyboard gesetzt. Das hat zur Folge, dass bereits bei einer kleinen Farbänderung jedes Element einzeln im Storyboard geändert werden muss. Deshalb setzten wir alle Design Element im Code, die über die von uns neu angelegte Klasse AppColor fungiert.

Abbildungsverzeichnis

2.1	Mein schönsten Mockup	6
3.1	hype cycle for technologies 2017	7
3.2	Siri Flowchart	9
7.1	Mockup unserer Aufgaben Erweiterung	23