Stundenplan–App v4 für iOS

Dokumentation, Spezifikation, Konstruktion

xx.xx.2018

Inhaltsverzeichnis

1 Projekt Ablauf **1**
- 1.1 Ausgangssituation . . . 1
- 1.2 Überlegungen zu Projektbeginn . . . 2
  - 1.2.1 GitHub / Branches . . . 2
    - 1.2.1.1 Vorher . . . 2
    - 1.2.1.2 Nachher . . . 2
- 1.3 Ziele für Version 4 . . . 3
- 1.4 Teams . . . 3
- 1.5 Kommunikation . . . 3
- 1.6 Fazit des Pojektes . . . 4
- 1.7 Projektfortschritt dokumentiert . . . 4
- 1.8 Ausblick . . . 4

2 Design **5**

3 Erweiterung durch eine Siri Integration **6**
- 3.1 Einleitung . . . 6
- 3.2 Istzustand . . . 7
- 3.3 Beispielhafte Anfragen . . . 7
- 3.4 Umsetzung . . . 7
- 3.5 Voraussetzung für SiriKit . . . 7
- 3.6 Einschränkungen . . . 7

4 Push Notifications **10**
- 4.1 Einleitung . . . 10
- 4.2 Projektfortschritt . . . 10
- 4.3 Vorbereitung einer Testumgebung . . . 12
  - 4.3.1 MAMP als Virtuelle Umgebung . . . 12
    - 4.3.1.1 MAMP . . . 12
    - 4.3.1.2 MySQL Datenbank . . . 12
- 4.4 Umsetzung . . . 12
  - 4.4.1 Zertifikate . . . 12
  - 4.4.2 Swift register . . . 12
- 4.5 Aufgetretene Probleme . . . 13

5 Testen der Anwendung **14**

1 Projekt Ablauf

Johannes Franz & Christian Pfeiffer

1.1 Ausgangssituation

Hier könnte etwas über den Pr

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 mehren Branches ermöglicht es größeren Softwareteams gleichzeitig an verschiedenen Features der App zu arbeiten. Für die iOS Studnenplanapp wurde ein passendes Branch Konzept vom Projekt Team ausgearbeitet.

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

1.2.1.1 Vorher

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 Notiz Style und kurzer Form eine Auszüge aus den jeweiligen PDF Dokumenten zusammengetragen.

GitHub Branch Übersicht:
`https://github.com/HochschuleHofStundenplanapp/iOS-App/branches`

1.2.1.2 Nachher

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

Team1(Pfeiffer, Scheler):

- Design

Team2(Franz, Krug):

- Siri
- Push Notifications

Team3(Hagmann, Knoblauch, Niepel):

- Verschiedene Fehlerbehebungen
- Hausaufgaben Manager

Team4(Kusserow, Sonntag, Dümmlein):

- Stundenplan Verbesserungen
- Widget

Team5(Pöhlmann):

- Testkonzept ausarbeiten und Testen

1.5 Kommunikation

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

1.6 Fazit des Pojektes

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

Text...

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.

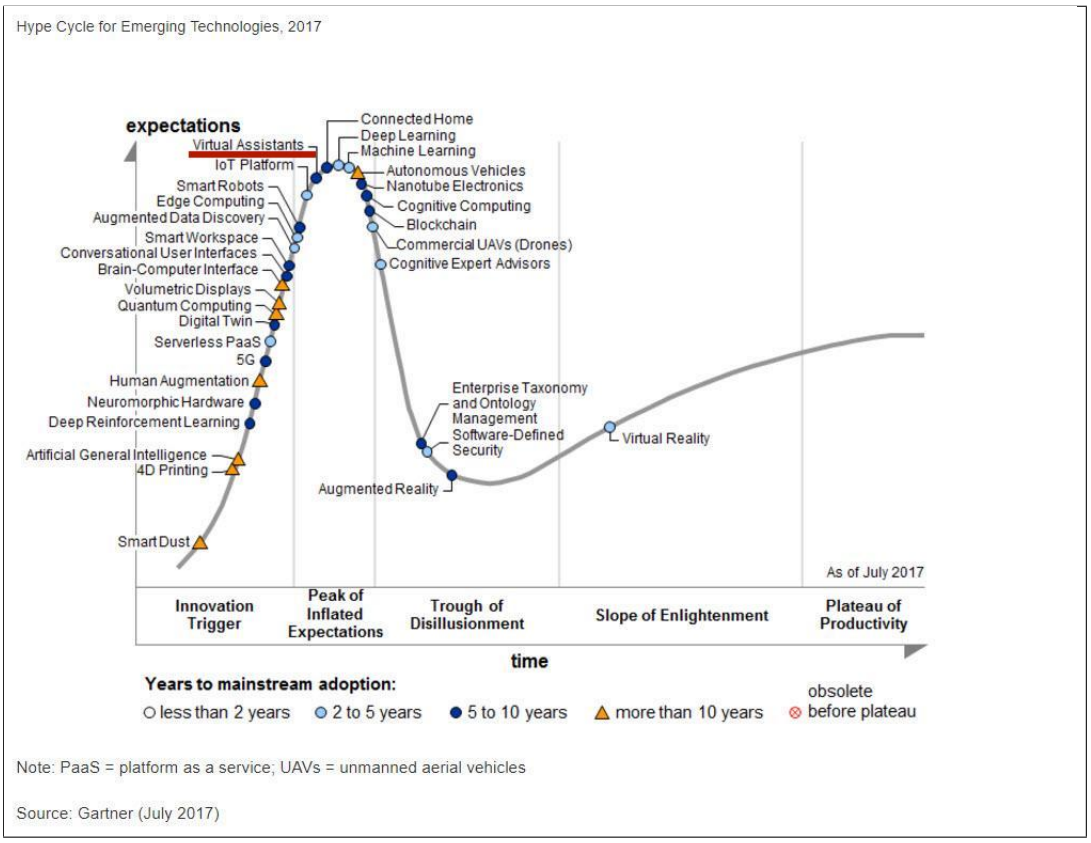

Abbildung 3.1: hype cycle for technologies 2017

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:

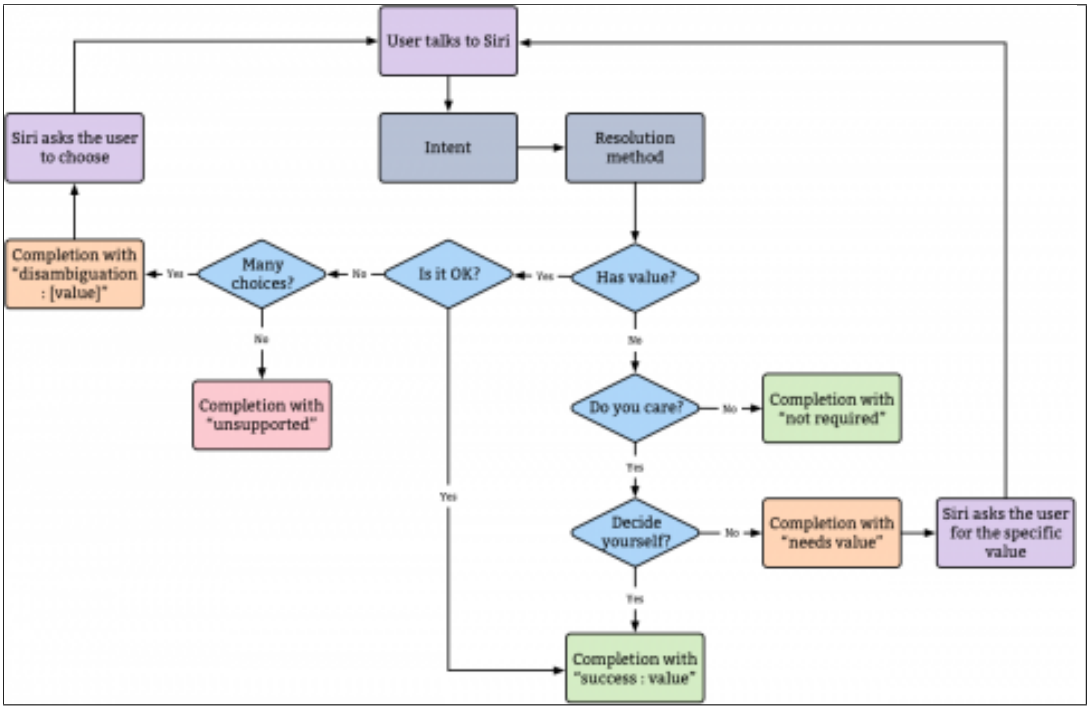

Abbildung 3.2: Siri Flowchart

- 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 möglichen mit der „Lists and Notes“ Domain, die Funktion für die Stundenplan hinzubiegen. Da aber die bestimmten Schlüsselwörter enthalten sei 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 vieler 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 Projektfortschritt

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 eingestellten 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	Server erhalten.

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

Ein SQL Statement zum erzeugen der Tabelle:

```
create database apns_user
       use apes_user
       create table apns_user(
              id int NOT NULL AUTO_INCREMENT,
              token varchar(255),
              lecture_id varchar(255) NOT NULL,
              PRIMARY KEY (id)
       );}
```

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 Zertifikate

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

Die MacOS App Ëasy APNs Providerërmöglicht die ersten Gehversuche, um Push Nachrichten an ausgewählte Token zu senden. Quelle: `https://github.com/immobiliare/ApnsPHP`

4.4.2 Swift register

Ein Swift Beispiel:

```
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 Herausforderungen zu sehende Punkte haben häufig sehr viel Zeit in Anspruch genommen oder den Projektfortschritt gebremst.

- Zertifikate sehr verwirrend
- Bundle identifier und App Capabilities nach einem Git Pull
- Komplexes Testen (Lokale Server, unterschiedliche Datei- und Serverstände)
- Kein Zugriff auf die Git Schnittstelle.
- Abgelaufnes Zertifikat

5 Testen der Anwendung

Text der Tests...

Abbildungsverzeichnis

3.1 hype cycle for technologies 2017 6
3.2 Siri Flowchart 8