

Stundenplan–App v4 für iOS

Dokumentation, Spezifikation, Konstruktion

xx.xx.2018

Inhaltsverzeichnis

1	Projekt Ablauf text vom include	1
1.1	Überschrift A	1
1.2	Überschrift B	2
1.3	Überschrift B	2
2	Design	3
3	Erweiterung durch eine Siri Integration	4
3.1	Einleitung	4
3.2	Istzustand	4
3.3	Beispielhafte Anfragen	4
3.4	Umsetzung	4
3.5	Voraussetzung für SiriKit	5
3.6	Siri Problem	5
4	Push Notifications	7
4.1	Einleitung	7
4.2	Projektfortschritt	7
4.3	Umsetzung	8
4.3.1	SQL Create Statement	8
4.3.2	Swift register	8
4.4	Aufgetretene Probleme	8
4.5	Anhang	9
5	Testen der Anwendung	10

1 Projekt Ablauf text vom include

1.1 Überschrift A

Hier könnte etwas über den Pr

1.2 Überschrift B

1.3 Überschrift B

2 Design

Text...

3 Erweiterung durch eine Siri Integration

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.

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.

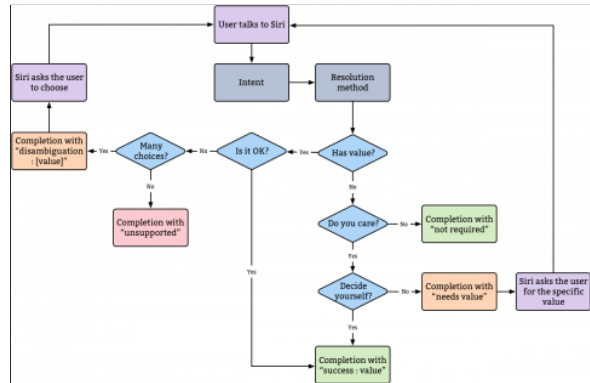


Abbildung 3.1: Siri Flowchart

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 Siri Problem

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

- 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-outsma>

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

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: <https://github.com/HochschuleHofStundenplanapp>

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.

4.3 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.

4.3.1 SQL Create Statement

Ein SQL Statement zum Erzeugen einer Tabelle:

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

4.3.2 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)")")
}
```

4.4 Aufgetretene Probleme

- 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.
- Abgelaufenes Zertifikat

4.5 Anhang

5 Testen der Anwendung

Text der Tests...