Technische Hochschule Köln

Fakultät 10

-

Institut für Ingenieurwissenschaften und Informatik

Implementationsdokumentation

„InTime“

Gruppe 45

Stefan Geier, 11112826

Vadim Demizki, 11112832

Studiengang Medieninformatik

Entwicklungsprojekt interaktive Systeme

WS17/18

Dozenten

Prof. Dr. Gerhard Hartmann

Prof. Dr. Kristian Fischer

Betreuer

Robert Gabriel
Sheree Saßmannshausen

Inhaltsverzeichnis

1. Vorwort

Der Implementationsteil des Projektes „InTime“ besteht aus zwei Hauptkomponenten: Der Dienstgeber und der Dienstnutzer.
Wir entschieden uns, den Dienstnutzer für Android-Geräte und in der Programmiersprache Java zu entwickeln, wohingegen die Implementation des Dienstgebers in JavaScript erfolgte.

2. Dienstgeber

Um den Dienstgeber korrekt zu starten sind nur einige wenige Schritte von Nöten. Bevor diese Schritte jedoch durchgeführt werden sollten, ist die Installation von NodeJS[1] notwendig, da dies für die Ausführung des Dienstgebers unabdingbar ist.
Sollte dies erfolgt sein, ist es zu Beginn erforderlich die Daten des Dienstgebers aus dem Git-Repository herunterzuladen[2]. Im Anschluss darauf muss mittels Konsole (bspw. Git Bash) der Befehl *„node dienstgeber.js“* eingegeben werden, sofern sich zuvor mittels Kommandos in den heruntergeladenen Dienstgeber-Ordner begeben wurde.

Als Bestätigung, dass der Dienstgeber nun erfolgreich gestartet wurde, wird in selbiger Konsole genannt, dass dieser nun auf Port 6458 verfügbar ist. Sollte stattdessen eine Fehlermeldung angezeigt werden, ist es nötig, die in der *dienstgeber.js* vorausgesetzten packages zu installieren. Dies ist in der Konsole mittels *npm install <package_name>* zu erreichen.

3. Dienstnutzer

3.1 Vor der Inbetriebnahme

Da die Entwicklung des Dienstnutzers in Java und für Android-Geräte angesetzt war, wurde sich nach Rat und Vorschlag von Betreuern dazu entscheiden die Entwicklung mittels der Android Studio IDE[3] durchzuführen.
Um nun einen vollen Einblick in den Code des Dienstnutzers gewährleisten zu können, ist es möglich, das gesamte Projekt als komprimierte .zip-Datei aus unserem Git-Repository herunter zu laden[4] und an einem gewünschten Ort zu entpacken. Um den Code nun allerdings korrekt betrachten und im Anschluss diesen in einer Virtuellen Android-Maschine nutzen zu können, ist eine Installation der offiziellen Android Studio IDE notwendig.

[1] https://nodejs.org/en/
[2] https://github.com/sgeier96/EISWS1718DemizkiGeier/tree/master/MS3/Implementation/Dienstgeber
[3] https://developer.android.com/studio/index.html
[4] https://github.com/sgeier96/EISWS1718DemizkiGeier/blob/master/MS3/Implementation/Dienstnutzer/InTime.zip

3.2 Inbetriebnahme des Dienstnutzers

Nach erfolgreicher Installation der Android Studio IDE ist es nun erforderlich diesen zu starten und gegebenenfalls zum Willkommensfenster des Programmes zu navigieren. In diesem ist es möglich, ein neues Projekt anzulegen, bereits bekannte erneut zu betreten oder ein existierendes Projekt zu öffnen. Letzteres ist hierbei auszuwählen, woraufhin im Anschluss zum Ordner navigiert werden muss, in den die in 2.1 beschriebene .zip-Datei entpackt wurde. Sobald dieser erreicht ist, wird innerhalb des geöffneten Explorers ein kleines AndroidStudio-Icon zu erkennen sein, neben dem sich der Projektname „InTime“ befindet. Eben jenes ist nun zu öffnen, sodass die IDE das öffnen des Projektes vorbereitet und im Anschluss geladen wird.

Sollte dies geschehen sein, ist es nun möglich, den Code auf der linken Seite unter mobile > java > com.example.stefangeier.intime zu betrachten.
Um jedoch den von uns entwickelten Dienstnutzer zu starten, ist es lediglich notwendig, im rechten oberen Bereich des Programmes unter *‚select Run/Debug Configuration‘ ‚mobile‘* auszuwählen und dem rechts daneben befindlichen grünen Wiedergabe-button zu betätigen. Alternativ zu letzterem ist es auch möglich, die Tastenkombination *Umschalt + F10* zu nutzen.
Das primär von uns zur Entwicklung genutzte virtuelle Gerät ist das Google Nexus 5X mit vorhandener API 27.

3.3 Verwendung des Dienstnutzers

Nachdem das Programm nun gestartet wurde, wird zu Beginn ein Login-Versuch beim Dienstgeber unternommen, der entweder erfolgreich oder gescheitert sein kann. In letzterem Fall ist eine Registrierung notwendig, in der lediglich ein Nutzername angegeben werden soll, da die Identifizierung eines Nutzers über die eindeutige Android-ID des Gerätes erfolgt.

Nach erfolgreicher Registrierung bzw. Login ist es nun möglich sich unter der Verwendung der Buttons durch die Applikation zu navigieren. Hierbei ist jedoch zu erwähnen, dass zum Stand der Entwicklung lediglich im *newSchedule*-Screen alle Interaktionselemente mit gewisser Logik und verbundenem Programmcode versehen sind. So ist es beispielsweise zwar möglich, die Nutzung des Programmes nachzuahmen, jedoch nicht tatsächlich durchzuführen.
In dem jedoch funktionierenden *newSchedule*-Screen ist es möglich, einen Namen des zu erstellenden Zeitplanes anzugeben, die aktuelle Position abzurufen und ein Ziel via GPS-Bestimmung festzulegen. In letzterem wird unter Betätigung beispielhaft die Position des Campus Gummersbach der TH Köln verwendet. Zur Angabe der aktuellen GPS-Position ist zu erwähnen, dass unter Verwendung einer virtuellen Maschine ein simulierter Wert gewählt wird. Dieser ist jedoch den Wünschen nach anzupassen, indem in unter *extended controls* des Android Emulators die entsprechenden Breitengrad- und Längengradwerte angegeben werden.

4. Abweichung von Modellen

Im Vergleich der im 2. Meilenstein eingereichten Modelle ist zu erwähnen, dass sich das Architekturmodell des Projektes „InTime“ dahingehend erweitert hat, einen Haushaltsmodus zu beinhalten.
Dieser setzt nämlich voraus, dass Daten zwischen einem Dienstnutzer und dem Dienstgeber ausgetauscht werden. Eben jener Austausch von Daten wurde von uns zur Einreichung der Modelle nicht bedacht, da zu diesem Zeitpunkt dem Haushaltsmodus und dem zugehörigen Haushaltsmanager lediglich ein geringer Wert bemessen wurde.