

Probleme bei der Entwicklung

Zeitmessung

Problem

Es gab Probleme beim Senden des aktuellen Zeitpunktes von der App zum Server. Der Android Emulator und der Java Server hatten verschiedene Zeitzonen eingestellt, weswegen der gesendete Timestamp am Server anders interpretiert wurde als am Emulator.

Lösung

In der App wird, vor dem Senden von Zeitdaten, die Zeitzone auf die Zeitzone des Servers gesetzt („Atlantic/Canary“).

Push-Notifications nicht angezeigt

Problem

Die gesendeten Push-Notifications an den Android Emulator wurden nicht angezeigt.

Lösung

Die Internetverbindung des Emulators aus- und wieder einschalten.

Keine Serververbindung auf echtem Gerät

Problem

Beim Testen der App auf einem echtem Android Gerät konnte keine Verbindung zum Server aufgebaut werden.

Lösung

Es wurde ein Ping Paket von dem PC, auf dem der Server läuft, an die IP Adresse des Android Gerätes gesendet. Damit wurde eine Route zwischen den beiden Geräten aufgebaut, womit das Problem gelöst wurde.

Probleme beim Senden von Serveranfragen

Problem

Das Android Gerät konnte keine Requests zum Server aufbauen, da das Android Betriebssystem dies nicht zugelassen hat.

Lösung

Das Senden von Server-Requests musste in einem anderen Thread realisiert werden, da Netzwerkaktivitäten nicht im gleichen Thread wie GUI-Aktivitäten ausgeführt werden dürfen/sollen.

Dafür wurde die abstrakte Klasse „ServletTask“ geschrieben, die von einer Background-Worker Klasse erbt. In der „ServletTask“ Klasse werden die allgemeinen Einstellungen für Server-Requests festgelegt. Die davon erbenden Sub-Klassen (für jede Art von Request) implementieren noch weitere Parameter und setzen den Namen der aufzurufenden URL.

Erhalten des FCM-Tokens

Problem

Anfangs wurde der FCM-Token (welcher pro App pro Gerät eindeutig ist) nur beim Registrieren eines neuen Benutzers gespeichert. Da sich dieser Token jedoch plötzlich geändert hat, konnten die Push-Notifications nicht mehr an das Gerät gesendet werden.

Lösung

Auf den FCM-Token wurde ein Listener gesetzt, der bei dessen Änderung den FCM Token aktualisiert.

Laden der Google Map

Problem

Es traten Fehler auf, wenn die Google Map Karte ein zweites Mal geladen wird. Dies passiert, wenn der **dritte** Tab nach dem Google-Maps-Tab angezeigt wird. Denn dann wird der Google-Maps-Tab automatisch entladen, um Arbeitsspeicher zu sparen. Wird jedoch wieder auf dem **zweiten** Tab nach dem Google-Maps-Tab gewechselt, wird der Google-Maps-Tab im Vorhinein schon geladen. Der Fehler tritt auf, nachdem die Karte geladen, entladen und wieder geladen wird.

Lösung

Anpassung der Gradle Dependencies in der Manifest Datei und Imports in den Tab-Klassen. Dort waren zuerst die falschen Klassen eingestellt, da meine Informationsquelle diesbezüglich (Google Maps API Dokumentation) nicht auf dem neuesten Stand war.

Ausführen der App auf echtem Gerät

Problem

Weil die Datenbank und der Server lokal am Rechner laufen, musste es irgendwie ermöglicht werden vom mobilen Android Gerät auf den Server und von dort auf die Datenbank zuzugreifen.

Lösung

Für die Lösung des Problems wurde ein Einstellungstab erstellt, bei dem man vor dem Einloggen bzw. Registrieren die IP Adresse des Servers, den Port des Servers sowie den Servernamen eingeben muss, damit man innerhalb des selben Netzwerkes auf dem Server zugreifen kann.

Mögliche Erweiterungen

Chat

Nutzer können mit anderen Personen (Freunde) chatten.

Zusatzfeatures

- vorgefertigte Nachrichten (z.B. „Gehe mit meinem Hund gassi“)
- automatisierte Nachrichten nach bestimmten Aktivitäten (z.B. „Bin gerade Gassi gegangen. Die Route war xxkm lang und hat yy lang gedauert.“)
- Pushmitteilungen per E-Mail
- Ratings, wie gut die Person gerade Zeit hat gassi zu gehen
 - z.B. 3/5 Sterne

Diese Zusatzfeatures könnten direkt in den Chat integriert werden und in Form von anders gefärbten Text-Nachrichten realisiert werden.

Profilbilder

Ein Nutzer kann ein Profilbild einstellen, welches für seine/ihre Kontakte sichtbar ist.

Erweiterungen

- Zugriff auf Bitmoji API
 - Bitmoji wird von anderen Apps (z.B. Snapchat) verwendet. Dort kann ein Avatar von sich selbst erstellt werden.
 - Dieser Avatar könnte als alternatives Profilbild eingestellt werden.
 - Wenn für Bitmoji und die HundeGassiApp die gleiche E-Mail Adresse gewählt wurde, könnte eventuell der Bitmoji Avatar automatisch als default Profilbild eingestellt werden.

Social Media Erweiterungen

Automatisches Posten von Posts auf anderen Social Media Seiten (z.B. Facebook, Twitter, usw.)

Beispiele

- Post direkt nach gegangener Route: „Ich bin gerade xxkm lang Gassi gegangen! #doglove“
- Monatlicher Post: „Ich bin diesen Monat xxStunden lang mit meinem Hund Gassi gegangen! #hundegassiapp“

UI Verbesserung

Verbesserung der Übersichtlichkeit der gegangenen Routen (z.B. Kalenderansicht, Protokoll, Einsicht von Gassi-Daten von Freunden, etc).