

Event Map

Link zum Git Repository: <https://github.com/dlugoschvincent/Grundlagen-Mobiler-Anwendungen>

Bei der von mir programmierten App handelt es sich um eine kartenbasierte Anwendung, in welcher der Nutzer Veranstaltungen erzeugen kann, welche mit einem Namen, Datum sowie Start- und Endzeitpunkt versehen werden können.

Der Nutzer muss seinen Vor- sowie Nachnamen hinterlegen. Diese werden zusammen mit einer ID lokal gespeichert und an jede erstellte Veranstaltung angeheftet.

Als Architekturmuster verwende ich MVC.

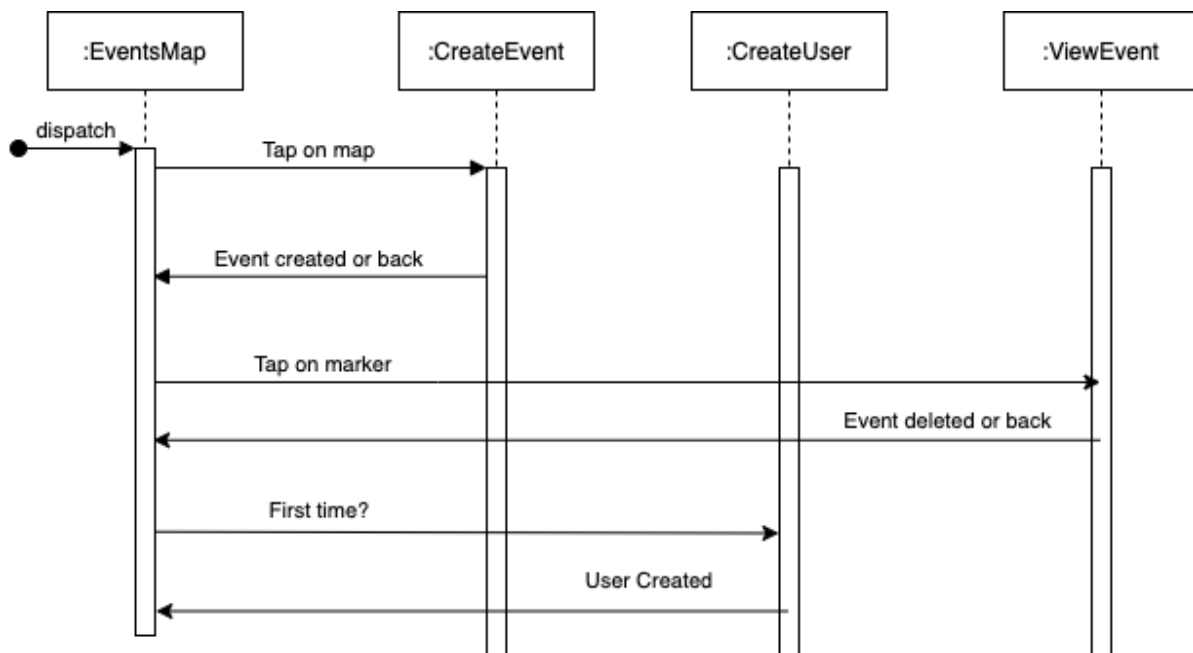
Als Kartenprovider verwende ich OpenStreetMap für Android (Osmdroid), welches mir ermöglicht eine Karte darzustellen und auf dieser Karte Markierungen zu platzieren. Außerdem bietet Osmdroid verschiedene Overlays an, welche die Interaktion mit der Karte ermöglichen.

Zu einer Synchronisation der Veranstaltungen über ein Netzwerk bin ich leider nicht gekommen, dies sollte aber aufgrund des modularisierten Aufbaus relativ einfach sein zu implementieren.

Zur Laufzeit an mehreren Stellen benötigte Objekte werden in einer Singletonstruktur gespeichert.

GUI

Das GUI besteht aus 4 Klassen. Als Hauptview fungiert die Osmdroid-Karte von hier aus werden die anderen Views zur Eingabe der Nutzerdaten, Erstellung der Veranstaltung sowie die Ansicht einer existierenden Veranstaltung gestartet.



EventsMap

Hier ist die Karte zu sehen, mit welcher interagiert werden kann.

Sollte die Anwendung zum ersten Mal geöffnet werden bzw. kein Nutzer in den “Shared Preferences” zu finden sein, so wird der Nutzer gebeten seinen Vor- sowie Nachnamen einzugeben.

Die Karte zentriert automatisch nach einer bestimmten Zeit auf die aktuelle Position, insofern die benötigten Berechtigungen erteilt wurden (Standortfreigabe). Ist dies nicht der Fall, so schließt sich die App automatisch.

Es besteht die Möglichkeit das Veranstaltungserstellungsoverlay zu aktivieren oder zu deaktivieren. Die Voreinstellung macht eine Interaktion mit der Karte nicht möglich. Erst durch Betätigen der Schaltfläche “Switch to create mode” besteht die Möglichkeit auf die Karte zu tippen und somit das Fenster für die Erstellung einer Veranstaltung zu öffnen.

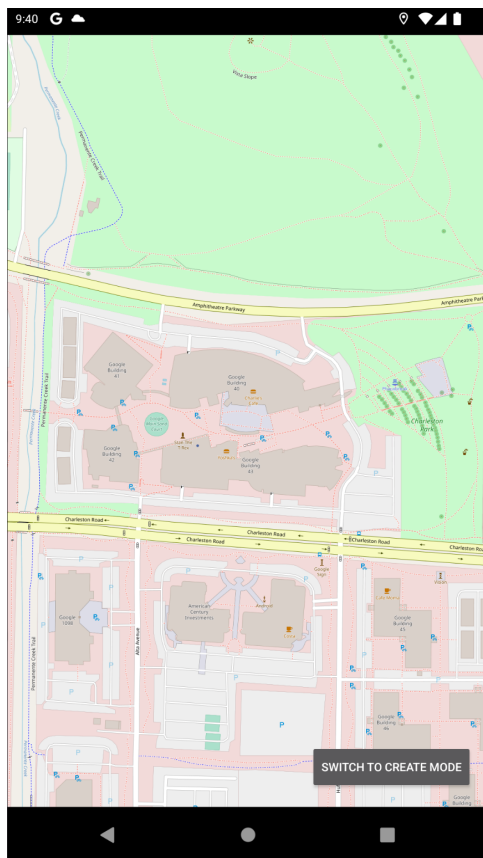
Sollten Veranstaltungen bereits existieren, so werden diese als Pins auf der Karte dargestellt. Ein Klick auf einen dieser Pins öffnet ein Fenster, in dem Informationen zu dieser Veranstaltung bereitstehen.

In dieser Klasse läuft auch ein Thread, welcher die Pins, auf der Karte aktuell hält.

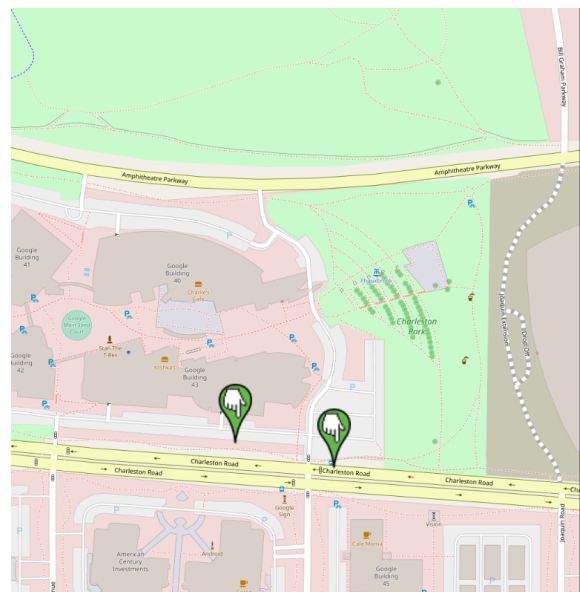
Hier habe ich mit Espresso getestet, ob die verschiedenen Karten Modi funktionieren und ob sich die “CreateEvent” Activity öffnet bei Klick auf die Karten.

Espresso tut in dem Fall auch was es soll und die richtige Activity geht auf, nur leider erkennt Espresso nicht, dass der Intent ausgelöst wurde und auch keines der Felder

in der neuen Activity. Bin hier mit meinem Latein am Ende.



Event Map ohne Events zentriert auf Standort



Karte mit 2 Veranstaltungen

CreateUser

Dieser View wird gestartet, sollte noch kein Nutzer in den “Shared Preferences” zu finden sein. Der Nutzer ist aufgefordert Vor- sowie Nachnamen einzugeben Diese Eingabe wird dann An den “UserManager” weitergegeben, welche den User initialisiert und schließlich in den “Shared Preferences” ablegt um ihn zu persistieren.



Bei erstmaliger Benutzung wird Nutzer aufgefordert Name einzugeben

CreateEvent

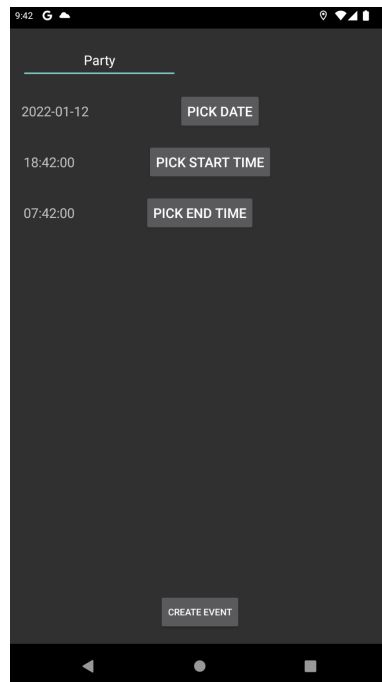
Diese Klasse Bietet die Möglichkeit Über ein Normales Texteingabefeld, sowie einem "DatePickerDialog" und zwei "TimePickerDialog", den Namen, das Datum sowie die Start- und Endzeit der Veranstaltung festzulegen.

Ist die Eingabe erfolgreich, so wird ein neues Event von dem "EventManager" erstellt und im "EventStorage" abgelegt. Nach erfolgreicher eingabe wird dieser View geschlossen.

Hier teste ich mit Espresso, ob die Nutzereingaben möglich sind. Die Eingaben funktionieren auch nur leider kann ich nicht die Zeiten bei dem TimePicker auswählen, was mich daran hindert, automatisiert eine komplette Veranstaltung zu erstellen.

Zusätzlich würde die Erstellung sowieso fehlschlagen, da die wichtige Information über die Koordinaten auf der Karte bei der isolierten Testung dieses Views nicht übergeben werden fehlen.

Espresso von der Karte diese Activity startet, kann ich bedauerlicherweise keines ihrer Elemente ansteuern, wie bereits oben erklärt.



Event Erstellung, nachdem auf Karte getippt wurde

ViewEvent

Dieser View wird durch einen Klick auf einen der Marker geöffnet. Hier werden die Veranstaltungsdetails dargestellt. Außerdem kann der Nutzer, dem die Veranstaltung gehört, diese löschen. Hierfür wird eine Methode aus dem “EventManager” aufgerufen, welche den Nutzer autorisiert und schließlich löscht, sollte diese erfolgreich sein. In beiden Fällen wird ein entsprechender Toast dargestellt.



Veranstaltungsansicht, wenn auf Marker getippt wird

Logic

Hier befinden sich die Manager, die die Interaktion zwischen Nutzereingaben und den produzierten Daten ermöglichen. So werden hier auch die Nutzerdaten, in die “Shared Preferences” geschrieben von dem “UserManager” um möglichst Android spezifischen code aus dem Model fernzuhalten.

UserManager

Der “userManager” verwaltet den Nutzer und implementiert das Interface “userManagerInterFace”, welches Methoden zur Speicherung/Ladung der Nutzerdaten und zur Abfrage, ob ein Nutzer bereits registriert ist, bereitstellt.

Außerdem wird hier ein zufälliger positiver Integer für die NutzerID generiert, sollte ein neuer Nutzer erstellt werden.

EventManager

Diese Klasse stellt durch Ihr Interface “EventManagerInterface” Methoden zur Verwaltung von Veranstaltungen bereit. So können anfragen an das Model gestellt werden eine Veranstaltung hinzuzufügen oder zu entfernen oder zu modifizieren(Im GUI nicht implementiert). Außerdem kann der “EventManager” einen Nutzer

autorisieren Änderungen an einer Veranstaltung vorzunehmen oder diese gar zu löschen.

Zusätzlich kann der Manager alle Veranstaltungen als Liste zurückgeben, welche im Model in einer HashMap gespeichert sind. Dies ist wichtig, um beispielsweise alle Veranstaltung auf der Karte darstellen zu können.

Hier kommen einige Unit-Tests zum Einsatz, welche die Funktionalitäten der Methoden für die Event-Verwaltung testen.

Model

Hier werden Daten gespeichert. Sowohl Event als auch die User Klasse finden hier ihren Platz. Die Singleton Klasse "Component Manager", welche Instanzen vom "UserManager" und "EventManager" auf Anfrage bereitstellt. Die Veranstaltungen werden in einer Hashmap gespeichert, welche in der Klasse "EventStorage" liegt.

Das Interface "EventStorageInterface" stellt Methoden zum Hinzufügen, Entfernen, Ersetzen und Holen von Events aus der HashMap zur Verfügung.