GitHub Username: TomMichalkevic

See Vilnius

Description

Everyone needs a small app they can refer to that does one thing and one thing only good. This app shall allow the users who are in Lithuania's capital, Vilnius to see what kind of landmarks are in their chosen radius. It will provide them with POIs (Points of Interest) along with their descriptions, ratings and, of course, the distance. The app will developed with material design best practices in mind, so a beautiful user interface is to be expected. Alongside that, the app will provide up-to-date information, however limited, about how to get to the specified destination.

Intended User

The intended user is any traveller visiting Vilnius or a local looking to learn more about their beloved city.
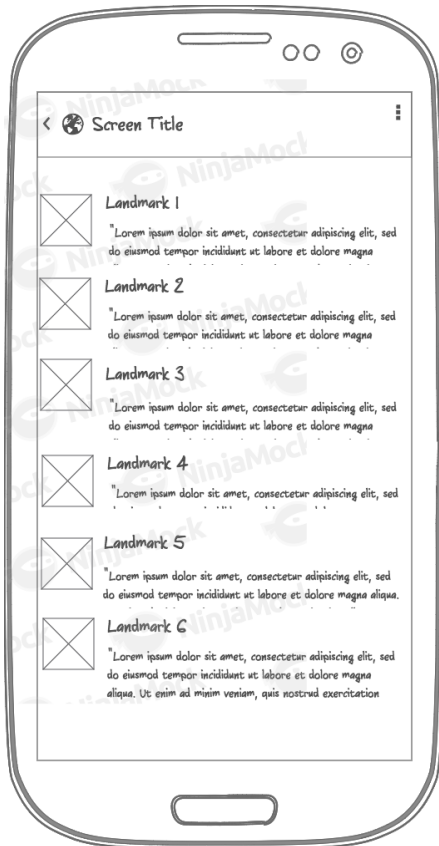
Features

The main features of your app::
- Shows POIs in the Vilnius area given the radius

- Shows information for the chosen places
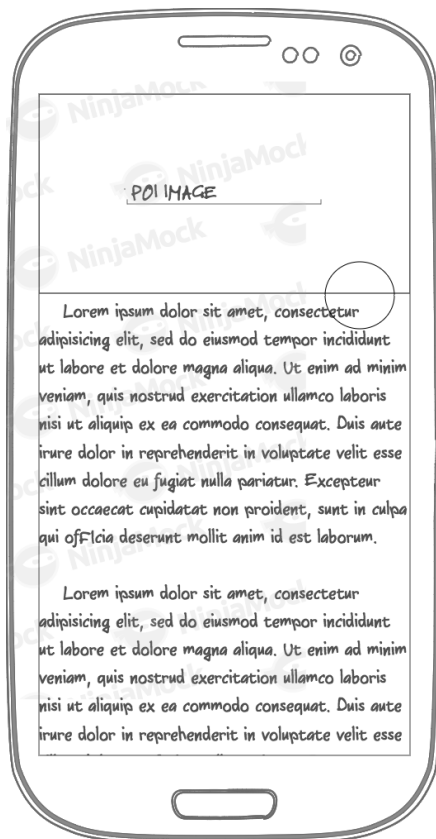- Shows nearby stations

User Interface Mocks

Screen 1

This screen mock shows how a list of POIs will be presented. They will be managed using a RecyclerView and each item in the list will contain an image and a short description.

Screen 2

On this mock we can see how an image will be shown along with its long description. This will be achieved using a CoordinatorLayout and the user will also be able to tap on a FAB button to mark the POI as visited.

Key Considerations

How will your app handle data persistence?

The app will utilize Android Architecture Components and therefore for data persistence Android Rooms shall be used. Location data shall be stored only for a certain duration as allowed by the Places API and also some user data will be stored, such as which places have been visited already.

Describe any edge or corner cases in the UX.

Usually it is difficult to preserve data on phone rotation. In such cases we would normally use onSavedInstanceState methods. However, due to the potential amount of data, this might not be ideal and therefore ViewModels will be used for better saving of viewable data on the screen after rotation.

Describe any libraries you'll be using and share your reasoning for including them.

I will utilize Picasso for image loading as all the necessary code optimizations are already present in the library so I will not have to implement anything on my side which will improve code readability.
I will also use ButterKnife, which will improve code readability since it safes the developer from writing a lot of code boilerplate to bind views.

Describe how you will implement Google Play Services or other external services.

Places API shall be used from the Google's Cloud Engine. This will allow me to gather information about various points of interest in the Vilnius city area. Functionality for working with the specified API will be split into their own dedicated methods based on the use case. Direct calls will not be done as a wrapper called the Places SDK for Android will be used.

Next Steps: Required Tasks

Task 1: Project Setup

- Configure libraries in the gradle file. Make sure stable versions are used
- Configure versions for the app, make sure it compiles on the newest Android
- Create a way to read API keys from outside resource to keep secrets safe

Task 2: Implement UI for Each Activity

- Build UI for MainActivity, which will display POIs based on the default distance (30km)
- Build UI for the DetailsActivity for the clicked locations
- Build SettingsActivity for modifying variables such as the search radius

Task 3: Create Android Architecture Components to save data and a way to call the endpoint for stations

- Create Entities
- Create DAO
- Create Executors
- Create a way of calling the station database endpoint asynchronously

Task 4: Make sure there is data persistence

- Make sure the data is preserved on rotation or if there is no internet

Task 5: Create a widget

- Create the view for a simple widget
- Create a way to display the last visited place with a picture and a short description