How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

GitHub Username: marekhakala

MyNomadLife App

Description

The application will provides guide for digital nomads. The app brings public information from website www.nomadlist.com that are provided by my own RubyOnRails API application from http://mnl.showmelink.co.uk/api/v1/ to native android app. Imagine that you are digital nomad, which want work and travel around the world. Then this app is right for you, because the app offer you score for cities. Further will be the ability to store information for offline mode and function for mark your favorite cities.

Intended User

It is an application for people that love traveling and work during travel. It is simply a group of people who are interested in digital technologies and traveling.

Features

- The app gets list of cities from website over API
- You can search city with parameters
- You can mark city as favorite
- You can mark city for offline mode
- You can display score of city
- You can display places to work for selected city

User Interface Mocks

Screen 1,2 - Main screen & Filter screen

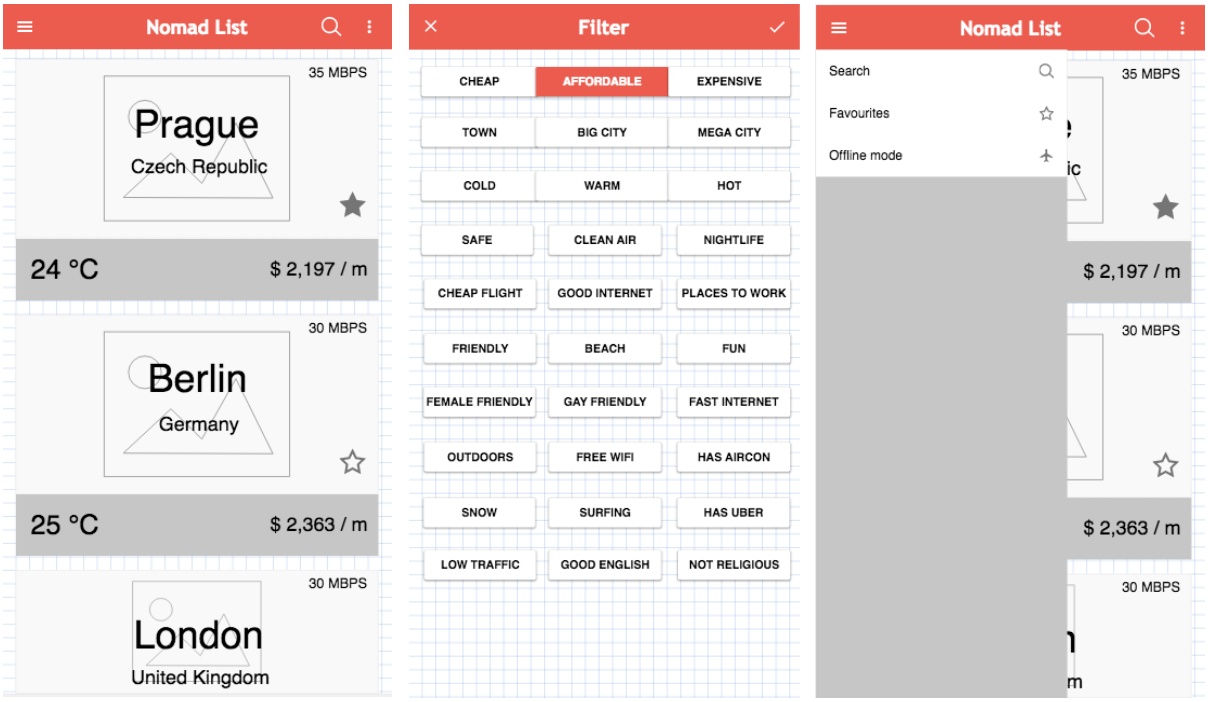

Figure #1 – Main screen, Figure #2 – Main screen with side menu, Figure #3 – Filter screen

The main screen will provides the preview of cities from data API. On the left side will be hamburger menu with other views and settings. If user want search the city or use the filter for the list of cities that can set over upper menu.

Screen 3, 4 - Search screen & Offline mode screen

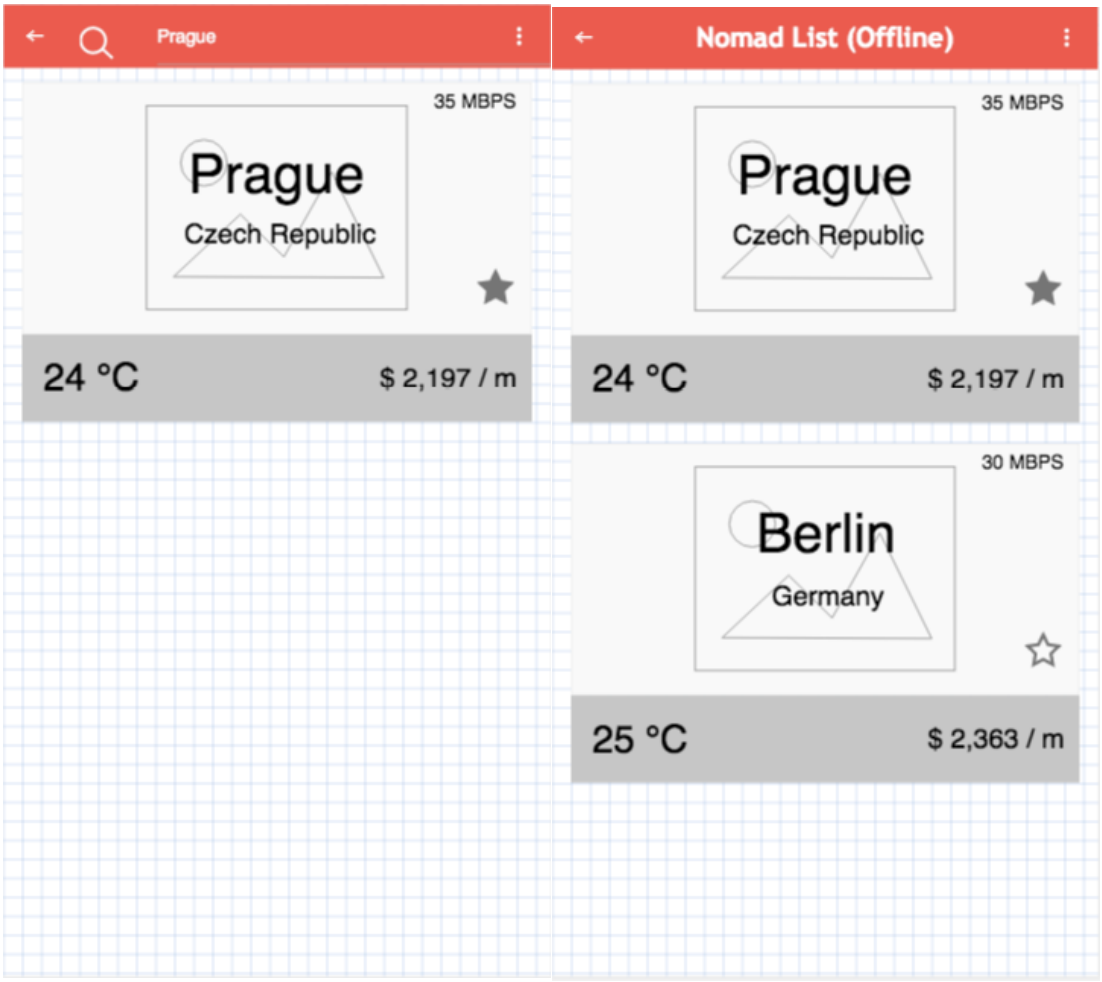

Figure #4 – Search screen, Figure #5 – Offline mode screen

The search screen will provide same functionality like main screen, but there will be available filters. The offline mode will be like a wallet for marked cities, where you will be able to show all information about city without an internet connection.

Screen 5, 6 - Detail view - Scores & Detail view - Info

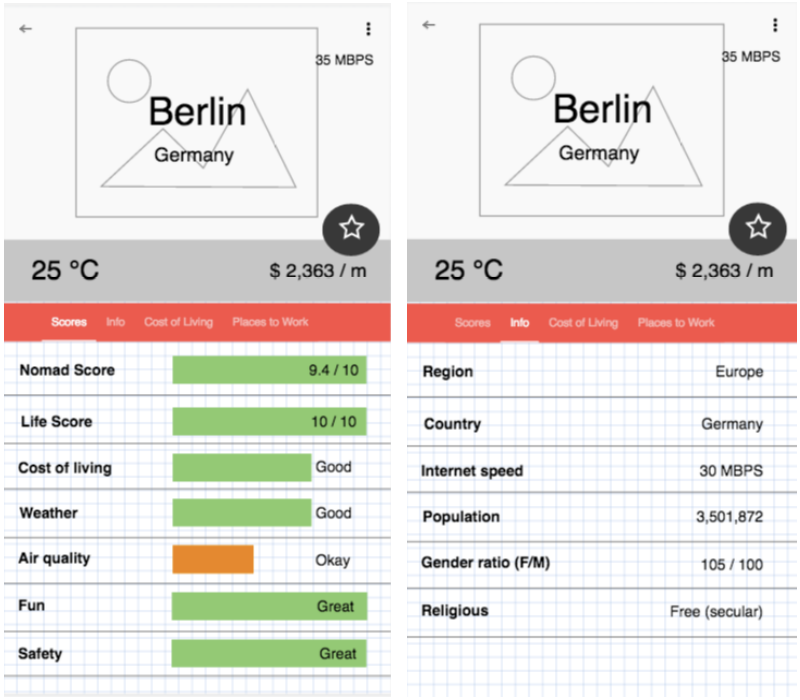

Figure #6 – Detail view – Scores, *Figure #7 – Detail view – Info*

Screen 7, 8 - Detail view - Cost of Living & Detail view - Places to Work

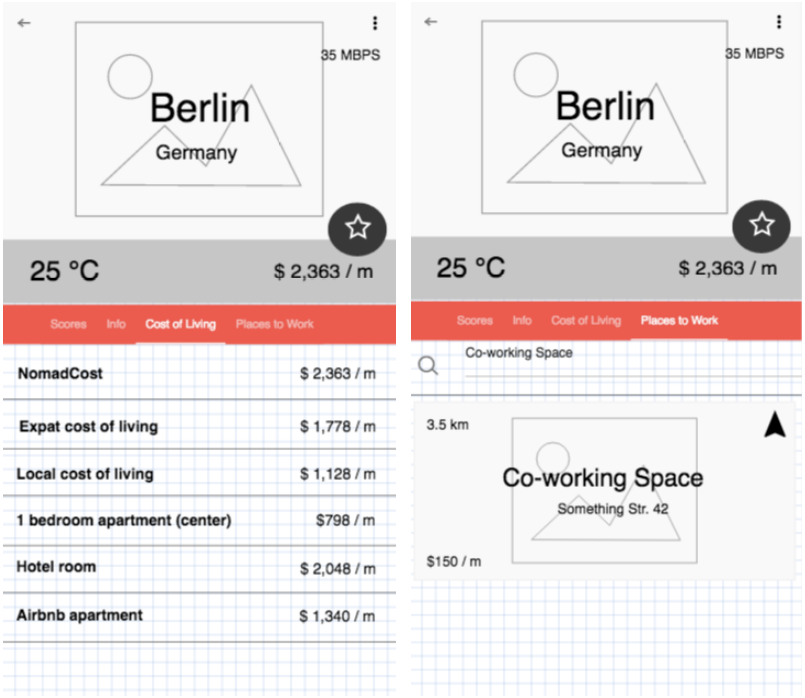

Figure #8 – Detail view – Cost of Living, *Figure #9 – Detail view – Places to Work*

This is the detail view screen where user can found detail information about chosen city. The information about the city will be included in four tabs with the labels: Scores, Info, Cost of Living and Places to Work. Further in section Places to Work will have two buttons: Website link and navigation button for Google maps.

Key Considerations

How will your app handle data persistence?

Application will store data in a SQLite database using a SqlBrite wrapper (https://realm.io/).

Describe any corner cases in the UX.

In the filter screen will be possible apply new settings or cancel changes.

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

- Support Design libraries (Material design library)
  - https://developer.android.com/topic/libraries/support-library/features.html
- Dagger 2 (Dependency Injection library)
  - http://google.github.io/dagger/
- Butter Knife (UI binding library)
  - http://jakewharton.github.io/butterknife/
- Retrofit 2 (Web API library)
  - http://square.github.io/retrofit/
- Picasso (Image handling library)
  - http://square.github.io/picasso/
- RxJava (Asynchronous and event-based programs library)
  - https://github.com/ReactiveX/RxJava
- SQLBrite (A lightweight wrapper around `SQLiteOpenHelper` and `ContentResolver`)
  - https://github.com/square/sqlbrite
- Timber (Advanced logging library)
  - https://github.com/JakeWharton/timber
- LeakCanary (A memory leak detection library)
  - https://github.com/square/leakcanary

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

- Create a new project
- Add libraries
- Configure libraries

Task 2: Implement UI for Each Activity and Fragment

- Create Dagger module for UI
- Main view
    - Build Main Activity
    - Build Search Activity
    - Build OfflineMode Activity
        - Build CityList Fragment
- Detail view
    - Build DetailView Activity
        - Build CityScores Fragment
        - Build CityInfo Fragment
        - Build CityCostOfLiving Fragment
        - Build CityPlacesToWork Fragment
- Build the XML resources for all UI components

Task 3: Create Data Layer

- Create Dagger module for data Layer
- Create Parcelable classes for application data

Task 4: Create API Sync Layer

- Create Dagger module for API Layer
- Create Retrofit API interface for NomadList API
- Create Retrofit data classes
- Implement RxJava methods subscribeOn and observeOn

Task 5: Implement logic for offline mode

- Add support for automatic sync of offline data when is available an internet connection
- Implement support methods for data synchronization

Add as many tasks as you need to complete your app.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"