GitHub Username: alvarosantisteban

Berlin Market Finder

Description

Berlin Market Finder allows you to easily find Flea- and Second-hand- markets in Berlin by using the open API from Berlin.de (the official site of Berlin).

Intended User

Berlin Market Finder aims to help both residents and tourists alike, and therefore will be available in German, English and Spanish.

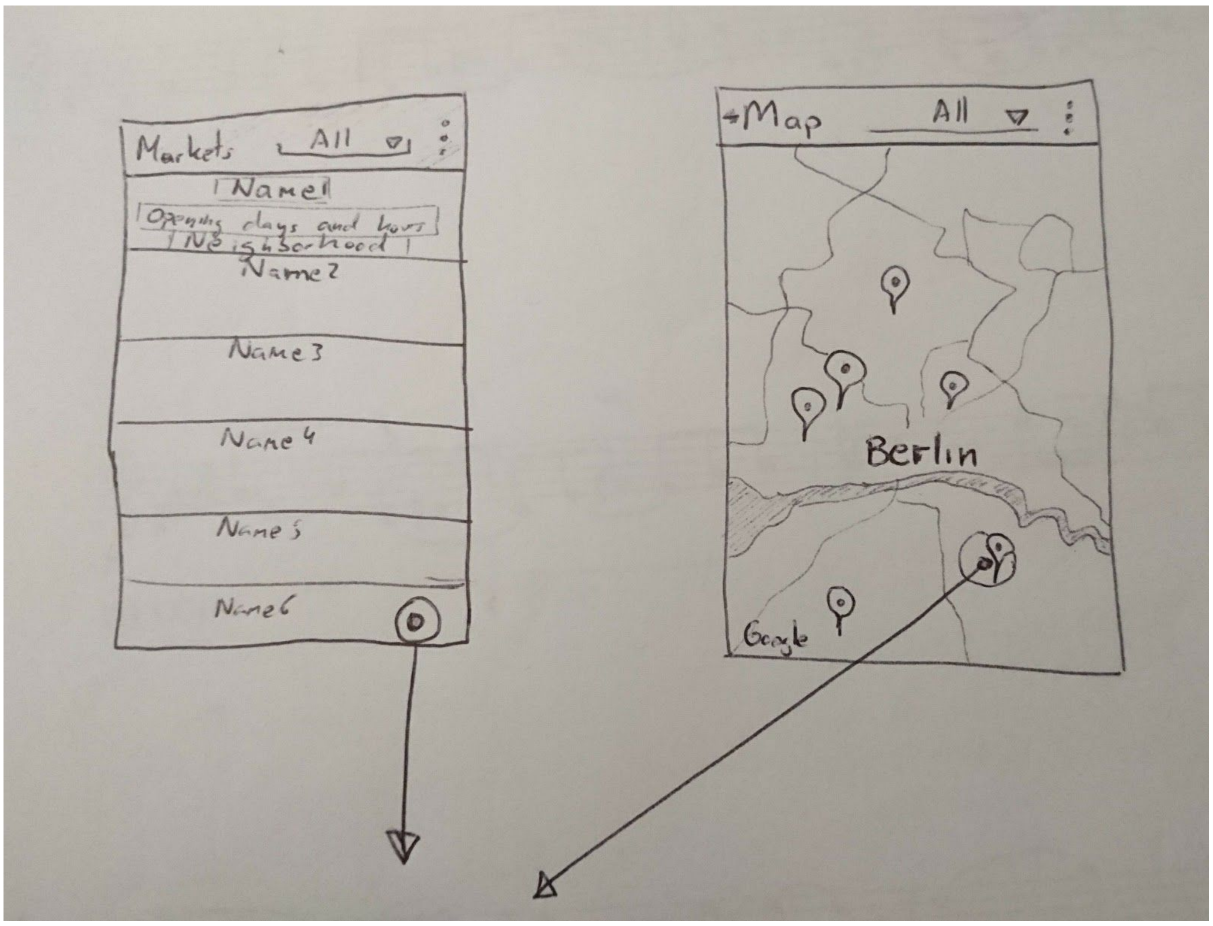
Features

The main features of the app:

- Display markets in a list or in a map
- Filter markets by neighborhood and/or by the current day
- Available offline
- Share the details of a market with friends and family

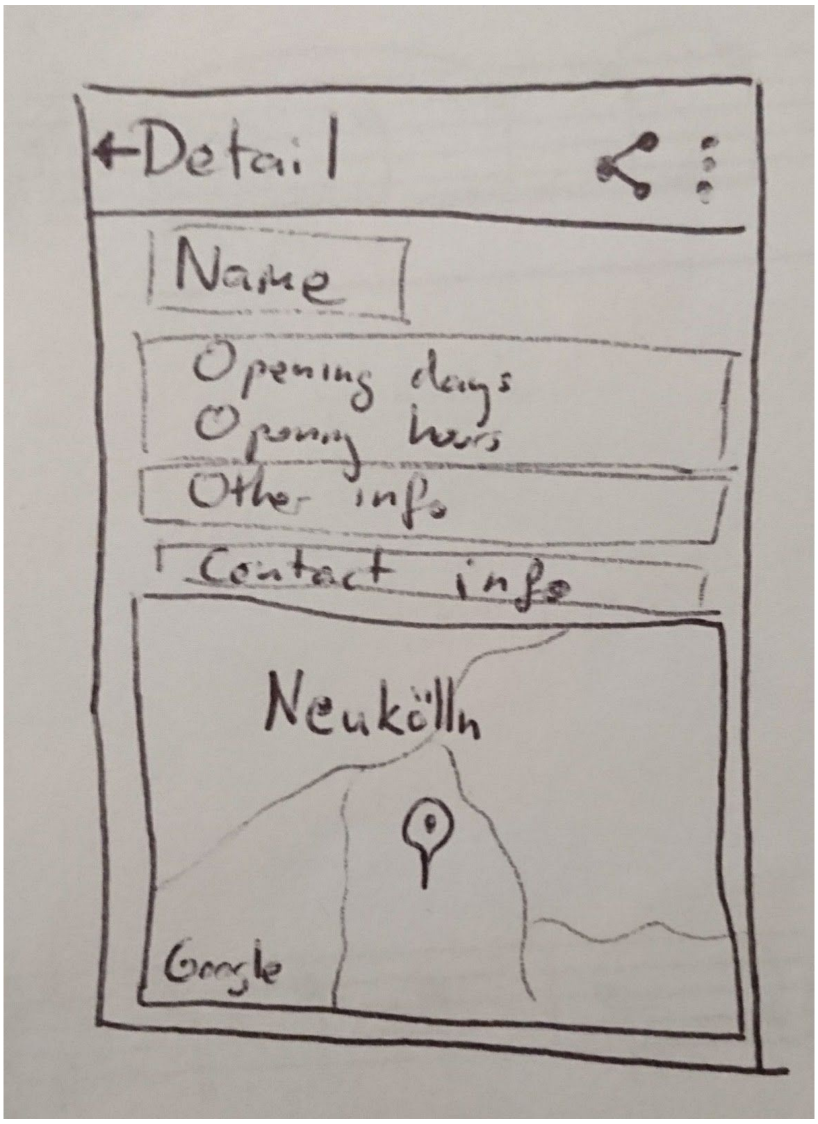
User Interface Mocks

Screen 1



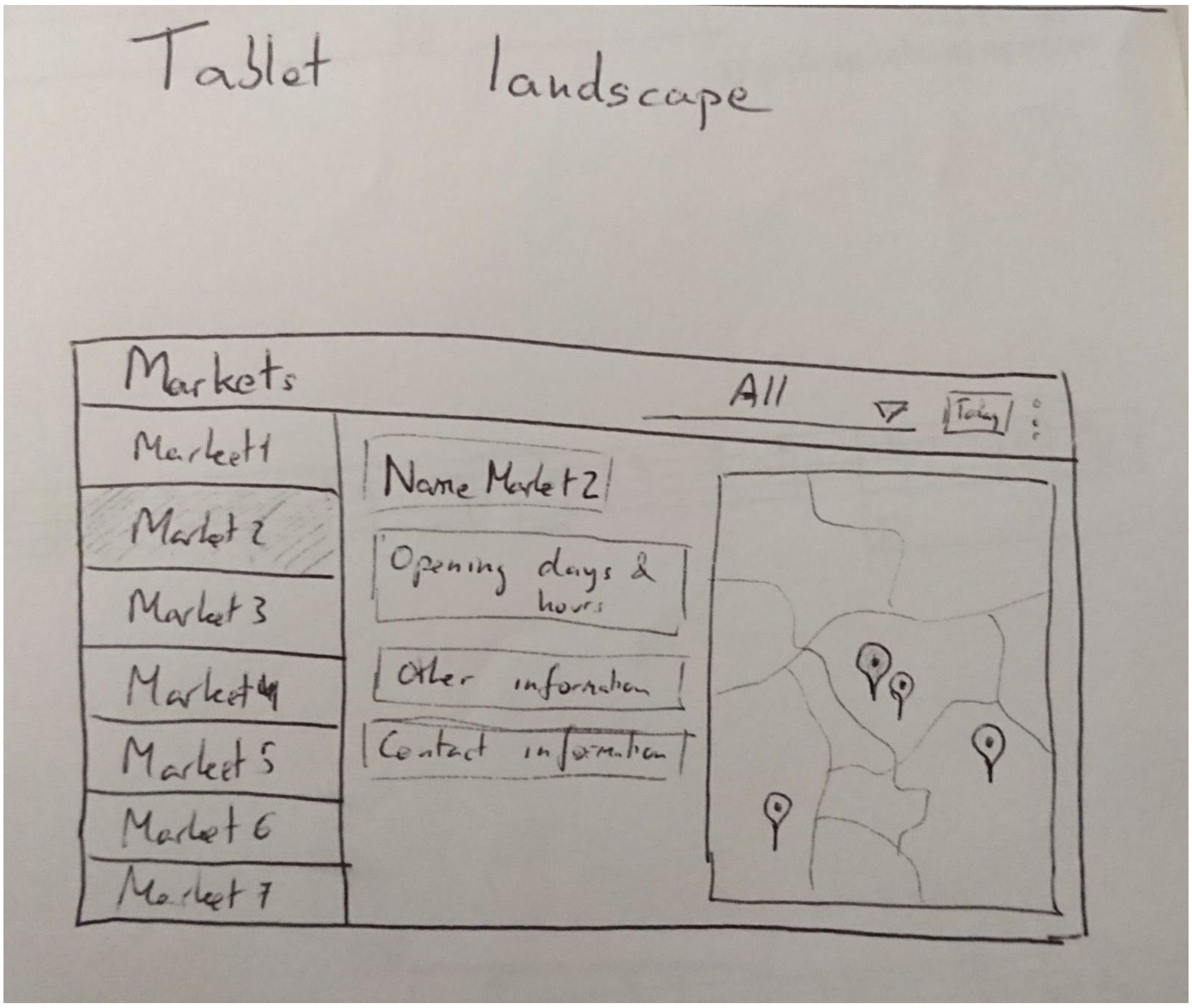
Screens for displaying the markets in a list and in a map.

Screen 2



Detail site of a market, accessed after clicking on an item of the list or a marker in the map.

Screen 3



Mockup for the tablet version in landscape mode.

Key Considerations

How will your app handle data persistence?

In order to allow accessing the markets offline, the app will have a database reachable through a content provider.

Describe any edge or corner cases in the UX.

The data returned from the API is not very consistent, so I will be very careful when dealing with it.
The idea is to handle the data before inserting it into the DB, so we can be sure that once the data is accessed using the content provider, there are no unpleasant surprises.

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

Gson to parse the JSON.
Butterknife to avoid boilerplate code.
Stetho to check the state of the database easily.

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

I will use Maps and Location. On the main activity I will allow the user to change from viewing a list to a map with all markets. Also in the detail site of each market, a small map will be displayed.

Next Steps: Required Tasks

Task 1: Project Setup

The setup consists of the following main steps:

- Create project and git repository
- Set up gradle file (minSdkVersion, configure libraries, naming of files, etc.)

Task 2: Create domain class

Create the Market class and use GSON to do the parsing.

Task 3: Implement Content Provider

Implement a content provider to store the markets' list.

Task 4: Implement UI for list of Markets

This task consists of the following subtasks:

- Create layout for the activity and for the items of the recycler view
- Download json and store information in the DB using the content provider
- Display list of markets extracting the information from the DB
- Allow filtering by neighborhood and/or current day

Task 5: Implement UI for map of Markets

This task consists of the following subtasks:

- Create activity and layout for map
- Set up key for displaying the map
- Display markers for the markets
- Connect this activity with the main activity

Task 6: Implement UI for detail site of a Market

This task consists of the following subtasks:

- Create activity and layout for detail site
- Connect this activity with the main and map activity
- Allow sharing the market information

Task 7: Implement UI for tablets

Create a layout specific to tablets in landscape mode.

Task 8: Visual fine tuning

Improve the visual look of the app, test with different devices, correct bugs.

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
    - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"