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: ngallazzi

Tripbook

Description

An app to get some trip ideas for the weekend, displaying places and things to visit around you. With **Tripbook** your weekends will never be boring anymore. Get inspired by a community of travelers like you, propose new destinations and rate the ones proposed by the other users.

Intended User

Travelers, nature lovers, adventurous families

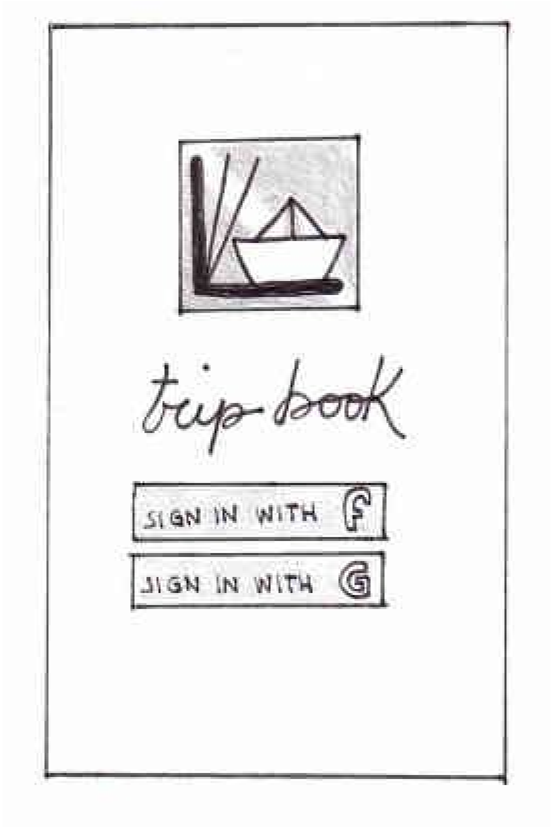
Features

The main features of **Tripbook** are:

- Provide users informations about locations and places to visit around them
- Users will have the opportunity to add new locations to the app database
- Users will be allowed to share, rate and leave comments on locations

User Interface Mocks

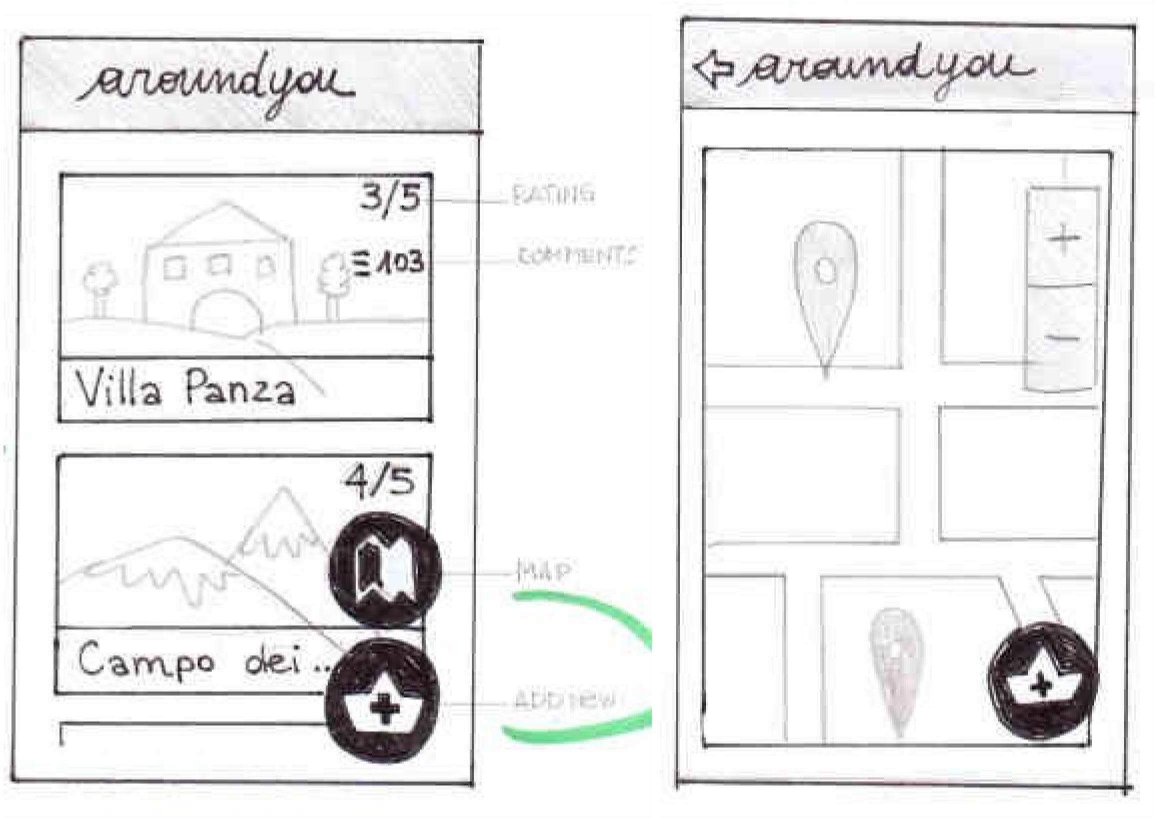
Authentication



First activity of the app. It implements the authentication/authorization flow. Users will have two different choices to signup (Facebook or Google).

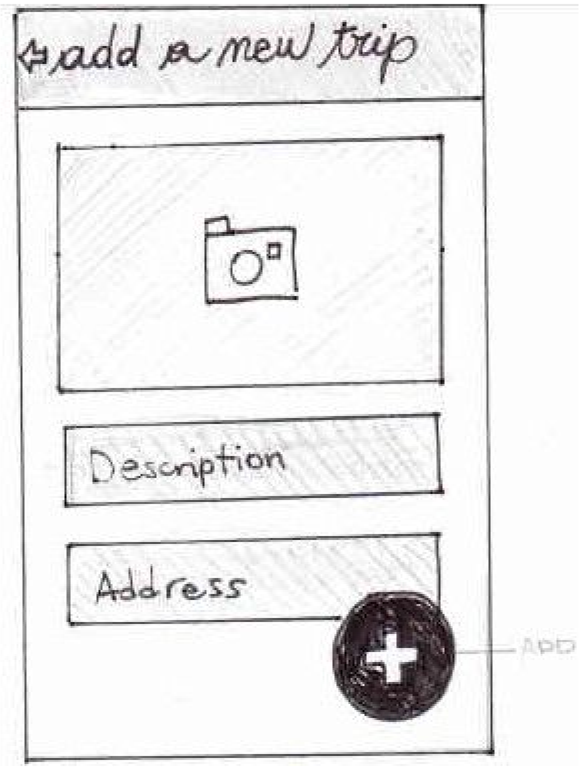
After the signup, this activity should be never displayed to the user anymore, unless he decides to logout or clear user data.

“Around you” activity



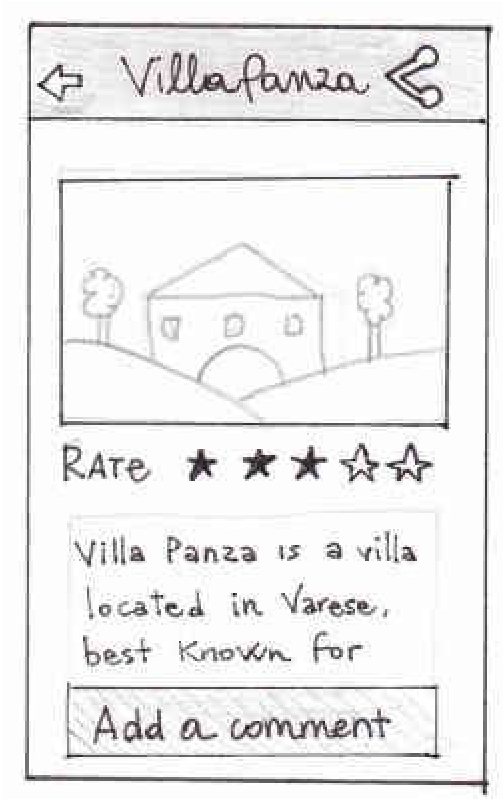
It’s the main activity of the app. It displays the list of locations near to the user. The default distance range is 50 km, but the user can edit this value in the app preferences (access from the action bar)
The “item location” displays a picture, the title of the location, the users’ rate and the number of comments posted for the location.
The user is also allowed to add a new location by tapping on the floating action button “add location”.
In the action bar there is a “map button”, useful switch to a map-mode layout.

Adding locations



This activity gives to the user the opportunity to add a location he knows, by inserting its address (or his own position, in case the user wants to add a location he is currently visiting). The user is required to add a picture(s) and a short description for the location.

Location details



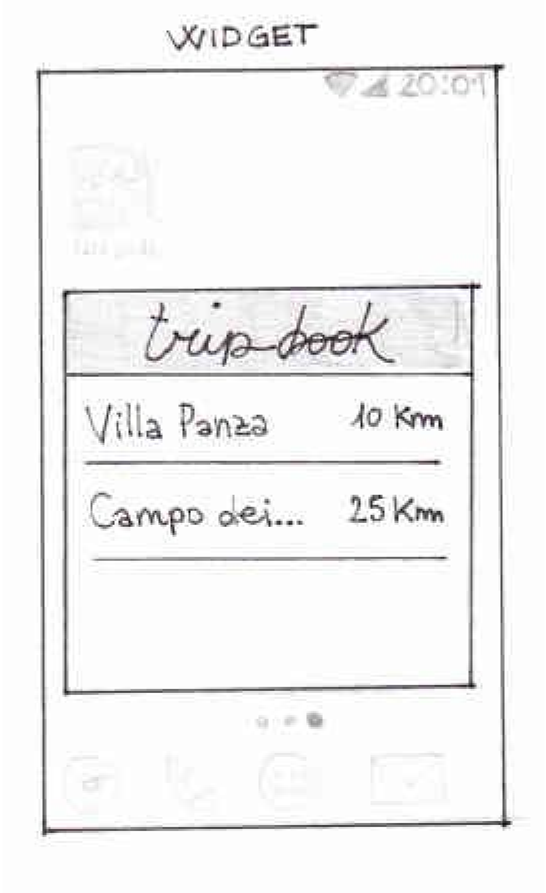
In this activity the user can see the details of a chosen location. In particular, the activity shows:

- One or more pictures of the location
- A rating bar to rate the location and to display the current average rating
- The description of the location
- The users comments about the location

In addition, the user can leave his own comment about his visit on the location.

Locations Widget

The app will provide to the user a locations widget, updated with the “in range” locations, in accordance with user’s settings



Key Considerations

How will your app handle data persistence?

The data persistence will be guaranteed by the content provider, fully synchronized with the firebase db. App also uses preferences to store the preferred range for user locations.
The pictures will be stored thanks to the firebase storage for pictures.
Sync adapter will be used to provide synchronization between content provider and the remote data.

Describe any corner cases in the UX.

The back button will always return to the previous activity, as stated in the master-detail activity pattern. In the “add a destination” activity case, if the back button is pressed before submitting, the location will not be saved.
The same logic will be adopted for the comments (not saved until submitted)

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

- Picasso for image loading and caching
- Butterknife for view bindings
- Firebase database for realtime database
- Firebase storage for images
- Firebase auth for authentication
- Facebook android sdk for authentication
- Firebase ui to manage authentication flow
- Android support libraries for retro compatibility (min version Api level 19 - Android 4.4)

Describe how you will implement Google Play Services.

The app will use Google Maps api to get user’s current location, in order to improve user experience and to show the nearest locations to the user’s position.
Furthermore, the Google Maps api will be used for the “map-mode” layout.

Next Steps: Required Tasks

Task 1: Project Setup

- Define Firebase data model for Locations, Users, Rates and Comments
- Populate the database with some dummy data in order to be able to generate a non-empty UI
- Include Firebase in the project and implement authentication, use FirebaseUI library to dominate the authentication flow

Task 2: Implement UI for Each Activity and Fragment

- Build UI for “authentication” activity
- Build UI for “around you” activity (both list and map layouts)
- Build UI for “location details” activity
- Build UI for “add location” activity
- Build UI for widget

Task 3: Include Firebase, define ContentProvider

Include Firebase into the project. Write classes for content provider and synchronization:

- Define SqlLiteOpenHelper and Contract class
    - Define “Locations” entry
    - Define “Users” entry
    - Define “Rates” entry
    - Define “Comments” entry
- Define ContentProvider
    - Build uriMatcher for data access paths
    - Build methods for queries, inserts and updates

Task 4: Implement firebase authentication

- Implement Google authentication
- Implement Facebook authentication

Task 5: Implement “around you” activity

- Use current location position to show localized content to the user, implement data items, adapters. Require data remotely only if necessary and show the list of locations to the user.
- Implement list view item click listener to allow the user to go to “Location details” activity
- Implement both layouts (list and map view)
- Implement click listener on floating action button in order to go to drive user to “add location” activity.
- Use parcelable class for locations to share information seamlessly between activities

Task 6: Implement Preferences

- Implement preferences change activity, useful to edit “around you” range

Task 7: Implement “add location” activity

- Allow user to select a picture for the location or to take a new one
- Provide an EditText to add description
- Get user coordinates before submit or ask an address to the user
- Write location remotely

Task 8: Implement “location details” activity

- Get location infos from content provider
- Implement rate functionality
- Implement “comment” functionality
- Implement a basic sharing functionality

Task 9: Implement “Locations Widget”

- Implement a widget which displays the nearest locations to the user. The list in the widget will display location’s picture, location’s name and location’s distance from the user position.

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**”