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: lamdbui

BrewMapper

Description

Beer is gaining in popularity these days and there are so many excellent choices for craft beer it is hard to keep track of. BrewMapper helps you keep up with favorite breweries and beers while helping you discover new ones.

BrewMapper simplifies how you can find, discover, and track beers and breweries from all around the world. Use BrewMapper's map function to see local breweries in your area and "favorite" ones to keep track of.

Intended User

Anyone who enjoys beer and wants to discover new beers and breweries locally and worldwide. Users will be able to track favorites and view them later on.

Features

Main features:
- Find breweries/beers local to current or specified locations
- View beer details
- View brewery details
- Show a map with brewery locations
- Add beers or breweries to a "Favorites" list

User Interface Mocks

Home Screen
This screen is the main hub to see local breweries and beer. A user can navigate to any part of the app from this screen.
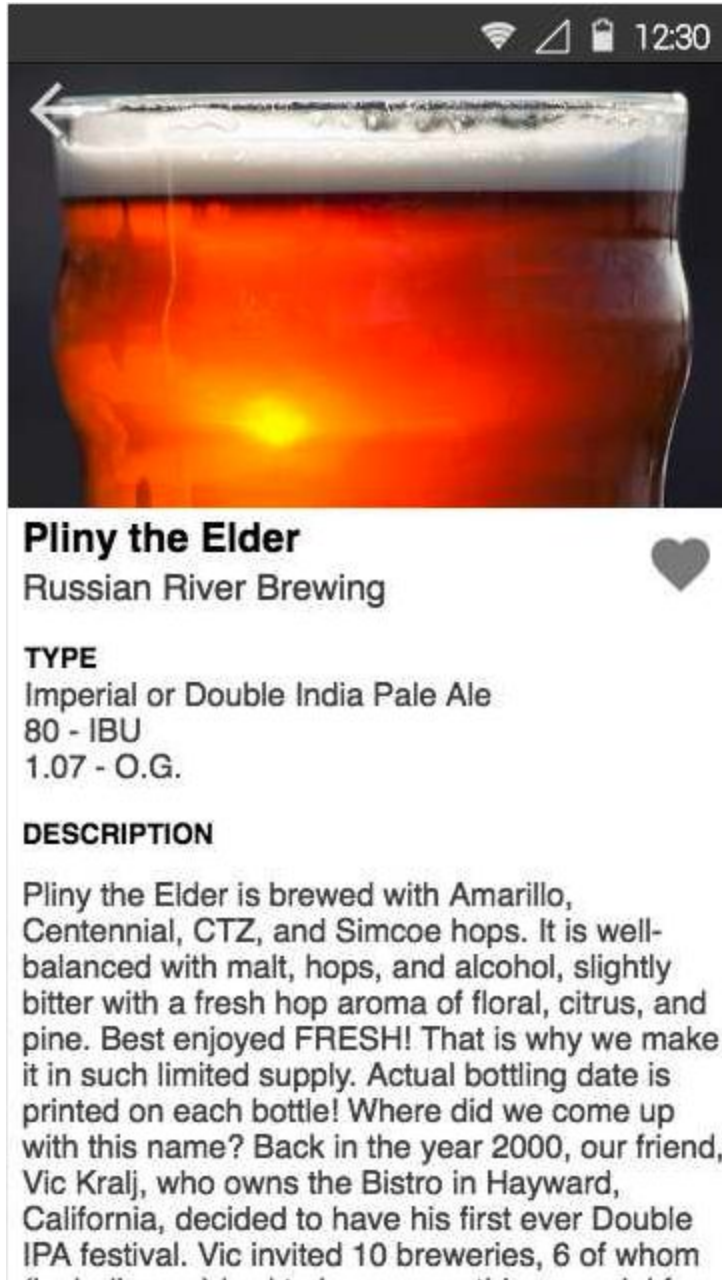
Map Screen

A user can see all the local breweries on the Map on this screen with additional details when a Marker is pressed.
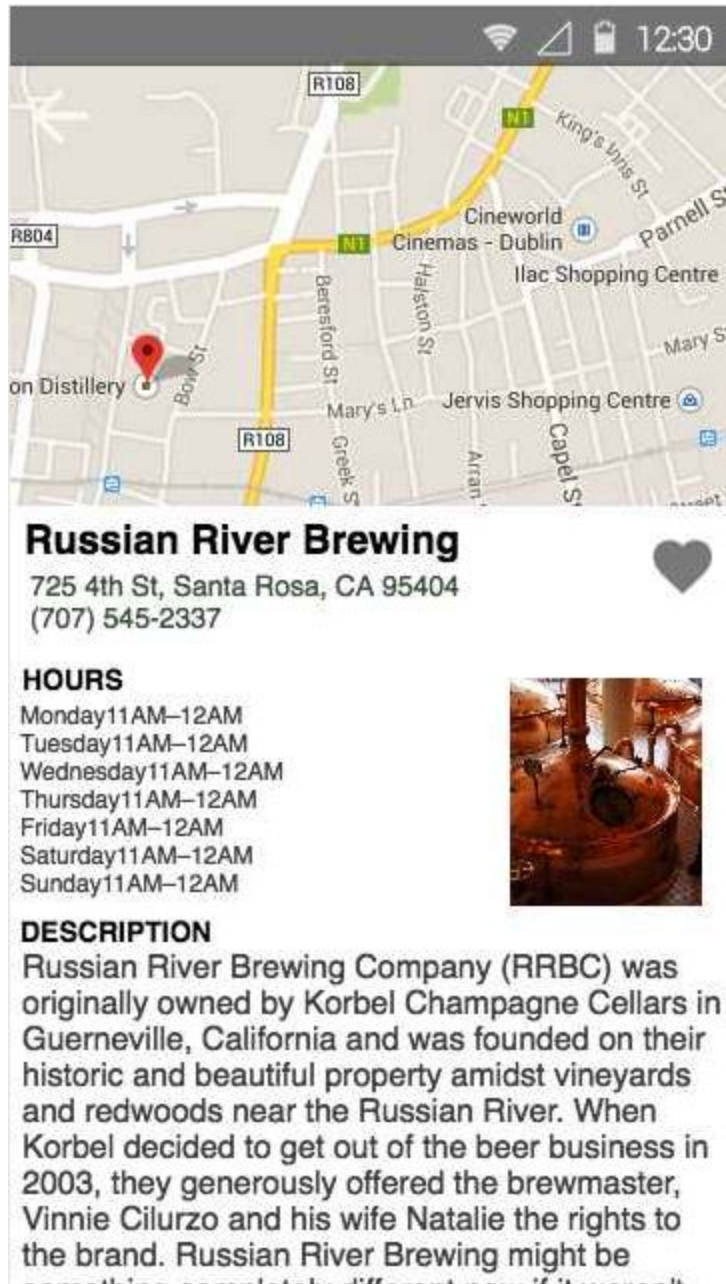
Beer Detail Screen
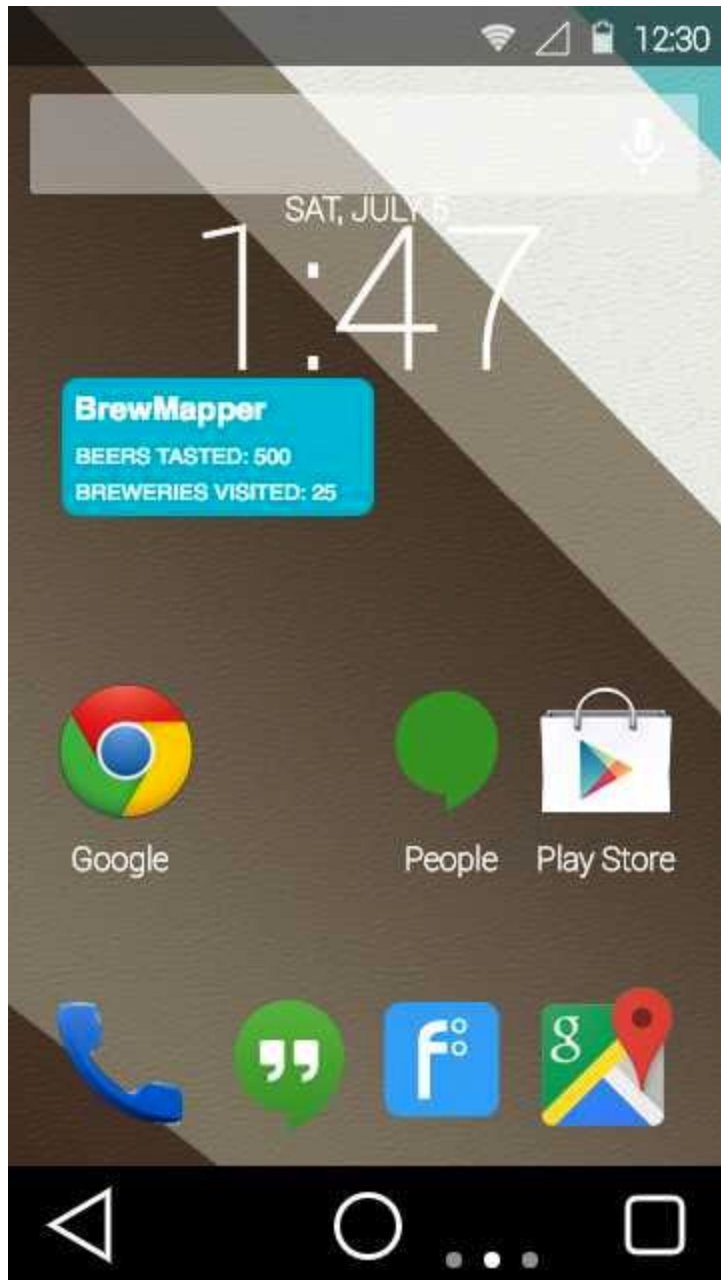Details about a specific beer can be viewed on this screen.

Brewery Detail Screen

Details about a specific brewery can be viewed on this screen.

Home Screen with App Widget

App Widget displaying statistics about beers and breweries a user has tried.

Key Considerations

How will your app handle data persistence?

"Favorites" and tracked beers/breweries will be local to a SQLite database and accessed via a Content Provider. Search queries will be pulled dynamically from BreweryDB and loaded with a CursorLoader into the UI.

Images will be fetched with an AsyncTask for display.

"Favorites" should be available offline.

Describe any corner cases in the UX.

- Offline functionality should still have "Favorite" data accessible
- Placeholder images should be used if no source image is available

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

- Retrofit for handling the HTTP requests
- Butterknife for less UI boilerplate

Describe how you will implement Google Play Services.

- Google Maps for showing brewery locations
- Firebase Analytics to track app usage and behavior

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

Subtasks:
- Configure libraries
- Setup BreweryDB API keys
- Setup Google Maps integration with Activity
- Setup Firebase Analytics

Task 2: Connect to BreweryDB and verify data retrieval is successful

Subtasks:
- Connect to BreweryDB API to pull down Brewery and Beer data
- Verify data is correct after fetching

Task 3: Build Data Model for Brewery and Beer

We want to abstract the data model from the app and the database so that we can easily swap it out the backend for a different solution, if desired. The current backend will use a SQLite database for local storage.

Subtasks:
- Create Brewery and Beer data models (POJO and SQLite database)
- Create Utility functions to handle conversions between POJO and database
- Create Content Provider to access the database

Task 4: Implement UI for Each Activity and Fragment

Implement all main UI components.

Subtasks:
- Build and connect Google Maps Activity
- Build UI for Brewery Screen
- Build UI for Beer Screen
- Build UI for Home Screen
- Build UI for Map Screen

Task 5: Create App Widget

Implement an App Widget that tracks a user's BrewMapper history.

Subtasks:
- Build and connect App Widget through a AppWidgetProvider

Task 5: Verify application is accessible for all

Subtasks:
- Verify contentDescriptions are available
- Enable RTL layout switching

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**"