How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]
2. Name your document file: "**Capstone_Stage1**"
3. Replace the text in green

GitHub Username: sabinaioana25

Blends

Description

Write a brief summary of what your app does. What problem does your app solve?
Most coffee house apps cover a very wide area of London and they don't filter through the best places, offering details on everything that serves coffee, even franchises.
Blends displays coffee houses in a map view and a list view and filters only small, independent places located in Central London.
As the user walks around in the area, they can get notified by the app about new or already-seen cafes; these notifications can also be deactivated individually.

Main highlights of the app:

- Free to use
- The guide offers details about London's best coffee shops, coffee houses and cafes, all offering locally roasted high-quality coffee beverages
- Filter results by personalised preferences
- Get quick and friendly notifications when you are near one cafe, so maybe you can pop-in or save it for later; or silence them by choice
- Save lists of cafes by two categories: by favourites and want-to-go

Intended User

This app is intended for high-quality coffee lovers, be it locals or tourists or people who just want to spice up their taste buds with locally brewed coffee by experts.

Features

- the app is used online, but saved places will be available to see at all times
- map view and/or list view is available to show nearby cafes, with all their details
- search functionality includes search by cafe name and postcode
- detailed profiles of each place includes name, address, phone number that can be directly dialled by one click, opening hours, images of the place, map location, customer reviews
- rate place by choice and share with friends

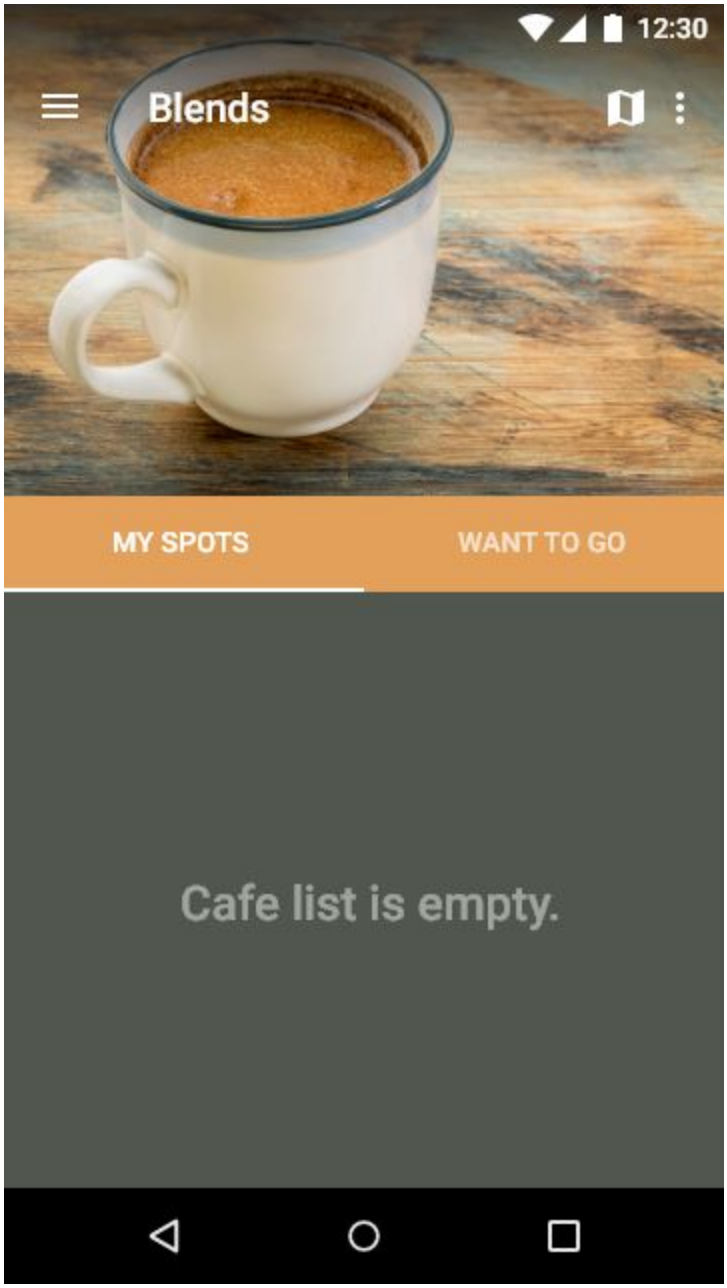
User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1 - splash screen

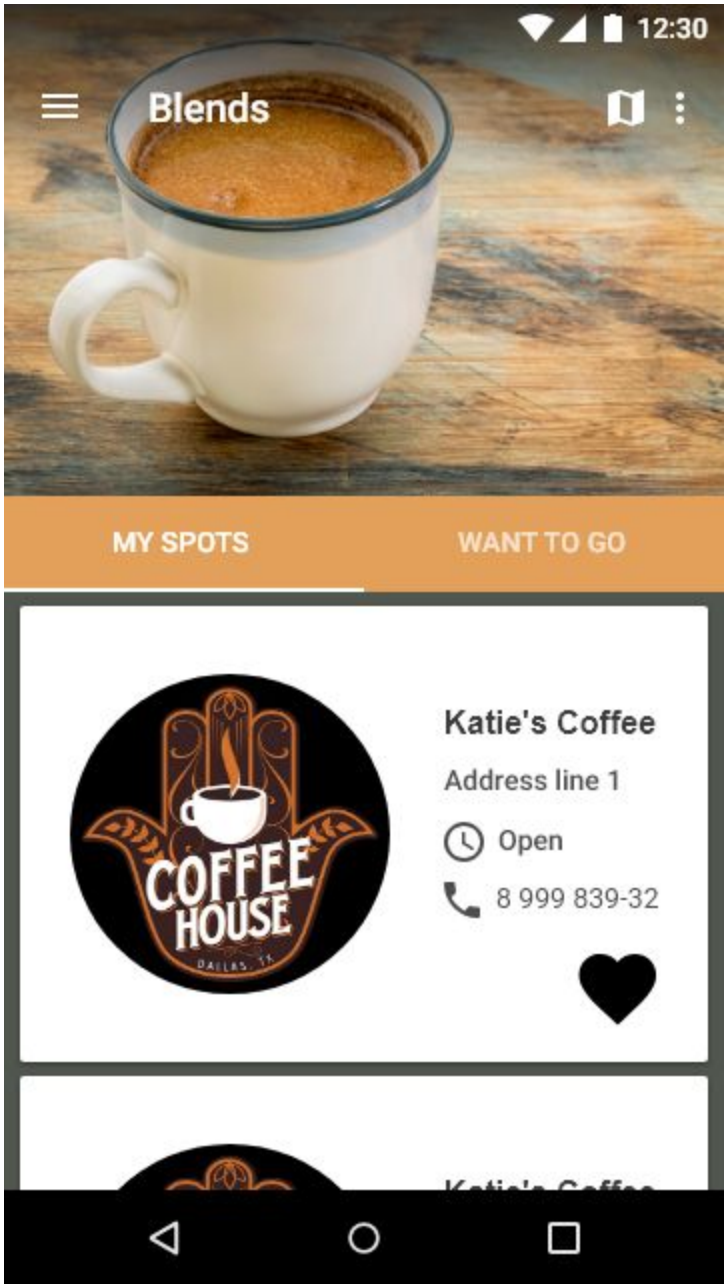
This is the first screen that opens when Blends is launched from the main screen.

Screen 2(A) - main screen with empty lists



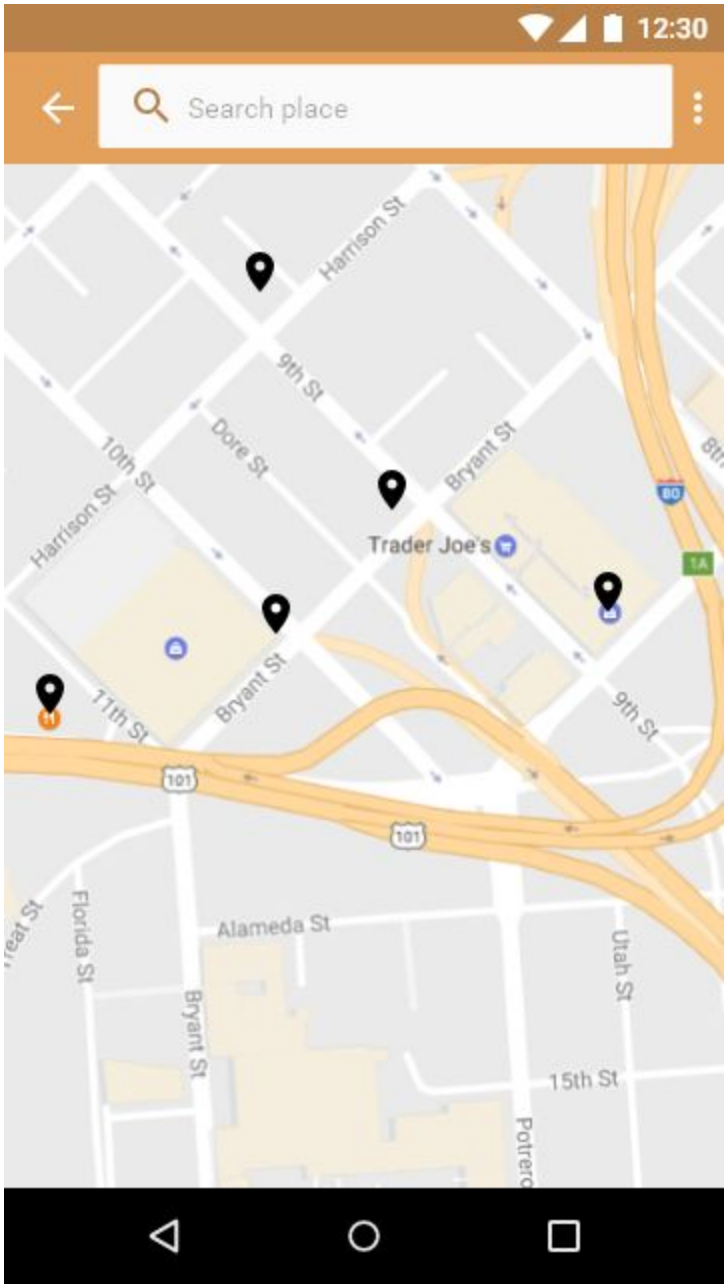
Within the first screen - if the user has only just begun to use the app or if he just hadn't added any items to his their list.
As a coffee passionate, I also want to be able to add a cafe/coffee place manually. To add these, users can click the FAB button.

Screen 2(B) - main screen with filled lists



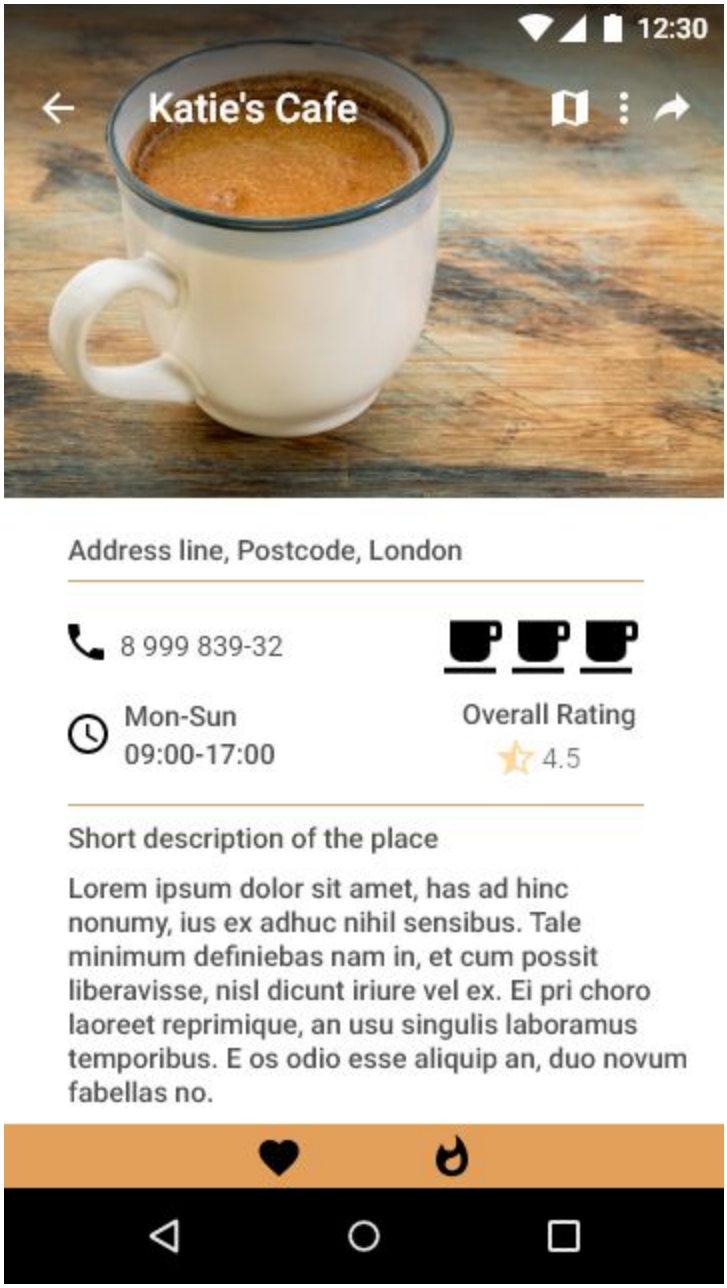
Favourite or ‘want-to-see’ places are added either manually or by adding them to the favourites list using the ‘heart’ icon, in the detail view.

Screen 3 - map view of coffee shops or cafes



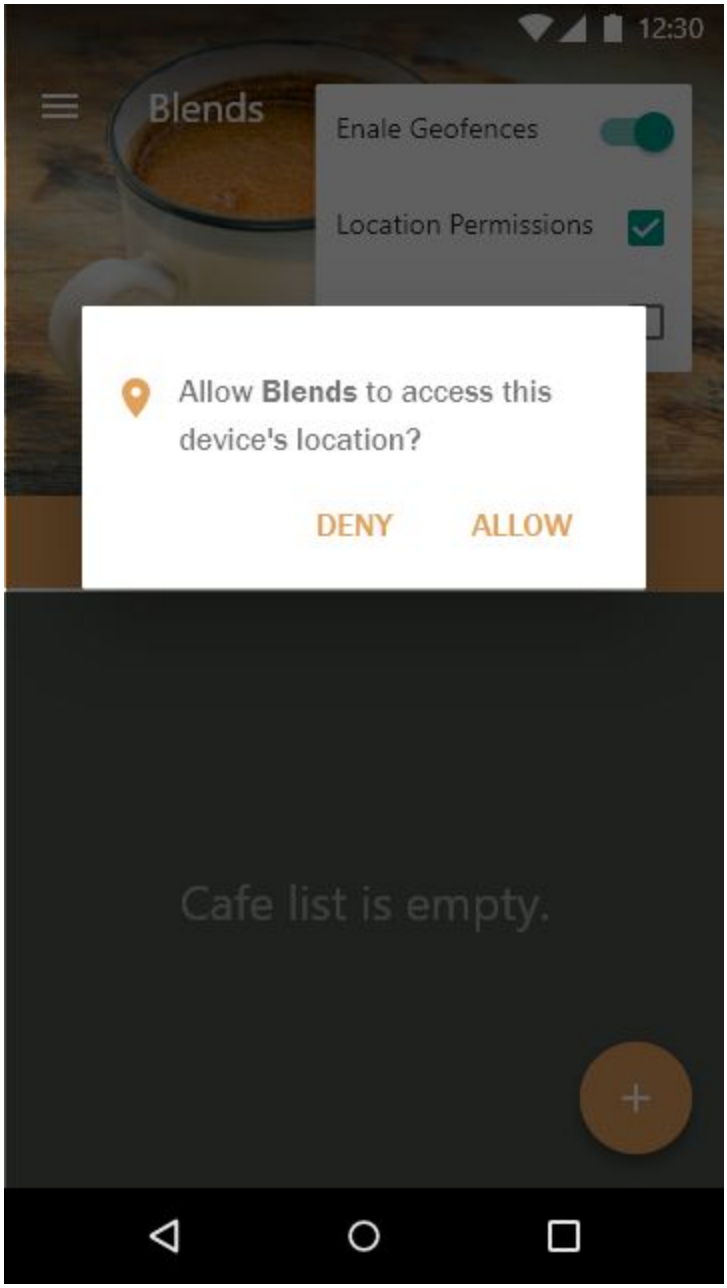
If the user wants to see where to find coffee places depending on their location, they can press the map icon located at the top right of the screen and a full map appears. They can navigate back to the list by using the back arrow on the left.
Clicking any pin will take the user to the detail page of that cafe.

Screen 2(C) - detail screen containing more information about a specific place



This page contains the details of a cafe that the user has clicked on. Additional to the name and contact details, it also shows the user a brief presentation of the place in order to get a sense/vibe of the cafe.
The icons located at the bottom of the screen are for adding the place to the user's favourite list or to the 'want-to-go' lists.

Screen 4 - the notification preferences menu



The notification preferences menu can be seen, allowing the user to mute/unmute notifications, enable geofences and enable location permission.

Screen 5 - the app widget that will appear on the main screen

The user will be able to add an app widget on their main screen, which will be updated daily with information on a different or new cafe.

Key Considerations

How will your app handle data persistence?

The app will have SQLite Database implemented in it, which will cache the locations chosen by the user.
Also, I will be using a Content Provider to share data between relevant apps.
With SharedPreferences, the app will be able to save all setting chosen by the user (e.g. notifications would be on for some coffee places).

Describe any edge or corner cases in the UX.

An possible edge case for the app would be the unexpected loss of wifi or internet data that allows the user to see cafes on the map and in the list view. If that would happen while the user is exploring these features, they would immediately receive a notification on the wifi loss issue, but the results (either in list view or map view) would not be lost.

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

Library that I will be using are:

- Picasso to handle the loading and caching of images
- Butterknife to simplify common tasks
- ShapeOfView to give a custom shape to android views

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

In my capstone app, I will implement Google Maps Services, to enable quick and efficient localization of cafes within the specified area.

Steps to implement Google Maps Services:

- Set my project up Google Play Services SDK
- Add Google Play Services to my project
- Add Google Places to my project
- Ensure that Google Play delivers service updates for user Android 4.0 and higher through the Google Play Store app

Next Steps: Required Tasks

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

Task 1: Project Setup

- Implement Google Places API as per Google Play Services instructions
- Setup the foundation of the project containing all necessary libraries
- The application is solely written in the Java Programming language
- The application makes use of stable release versions of all libraries, Gradle and Android Studio
- Application keeps all strings in a strings.xml file and enables RTL layout switching on all layouts

Task 2: Implement UI for Each Activity and Fragment

- Build UI for splash screen
- Build UI for MainActivity
- Create Fragment subclasses
- Build UI for DetailActivity
- Build UI for MapViewActivity
- The application includes support for accessibility (content descriptions, navigation using a D-pad, and, if applicable, non-audio versions of audio cues)

Task 3: Initial API Integration Database testing and storage

- Obtain key for Places API
- Create POJO classes for the API model response
- Test Places API accuracy
- Create all helper classes after setting up main classes
- Test the API response with JSON
- The application performs short duration, on-demand requests using an AsyncTask.

Task 4: Implement Persistence

- Set up SharedPreferences
- Create SQLite Database for data storage and helper classes

- App stores data locally by implementing a ContentProvider; the app uses a Loader to move its data to its views

Task 5: Implement notification service

- Display a notification when app detects a stored coffee place / ask user for location access

Task 6: Implement error handling

- Display a notification when connectivity to the API is lost

Task 7: Implement error handling

- Build and submit project for review

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**"