

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**
-

Description

Intended User

Features

User Interface Mocks

Screen 1

Screen 2

Screen 3

Screen 4

Screen 5

Screen 6

Screen 7

Screen 8

Key Considerations

How will your app handle data persistence?

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

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

Next Steps: Required Tasks

Task 1: Project Setup

Task 2: Implement UI for Each Activity and Fragment

Task 3: Add More Activity for Bottom Navigation Bar

Task 4: Define LinkedHashMap array for location name with its icon

Task 5: Add LocationListOnMap and LocationListActivity with its Adapter

Task 6: Add Location Search Option

Task 7: Implement Data Model and Content Provider

Task 8: Add Methods in UtilsMethod

Task 9: Add Location Detail Activity with its Fragments

Task 10: Add Functionality to Favorite Icon ImageView

Task 11: Add Widget

Task 12: Fix Issues and Errors

GitHub Username: [Rapter1990](#)

FindL

Description

Many people have a lot of problems to find any location in their daily life because they cannot know where they go to meet their special needs. At the same time, they cannot determine whether the location which they go is opened or not as they cannot have any detail idea about location. Moreover, they cannot know whether the distance between their position and the destination which they go is close or not. FindL provides them to find their location which they want to go in terms of location type categories, to give detail location to them in term of its detail information and users' review and to show distance to the destination with respect to how much kilometers are there between their position and the destination.

Intended User

Travelers, Students, Employees or any individual

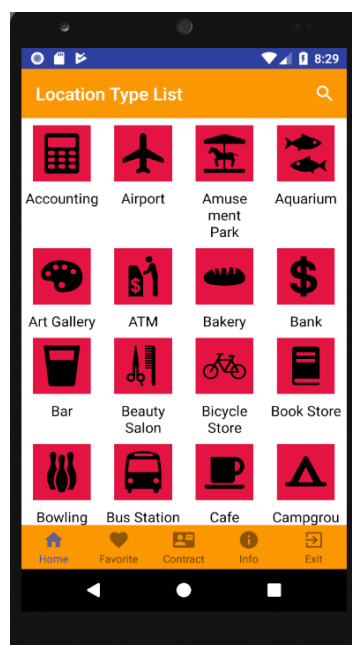
Features

List the main features of your app. For example:

- Find nearby locations with respect to their types. These types are depended on accounting, airport, bus station, hotel, car repair, mosque, restaurants and others
- Provide detail information about the location which user select. They are based on phone number for calling location, approximate distance ,website, status which is showing whether the location is opened or closed, rating for showing the popularity of location, address and users' reviews to give their opinion about locations.
- Share location which user select to friends, family and other people.
- Contract with developer for feedback process.
- Info section to show who person develops the app.
- Provide Exit to exit the app.
- Add location which users like to their favorite location app.

- Provide search option for location types for getting fast result.
- Support RTL option
- Easy to access each Activity on Bottom Navigation bar and all related activity finding and searching locations
- The app is only written in Java Programming Language.
- The app utilizes stable release versions all libraries displayed below to be implemented
 - implementation 'com.android.support:appcompat-v7:27.1.1'
 - implementation 'com.android.support:design:27.1.1'
 - implementation 'com.android.support:support-v13:27.1.1'
 - implementation 'com.android.support:animated-vector-drawable:27.1.1'
 - implementation 'com.android.support:customtabs:27.1.1'
 - implementation 'com.android.support:cardview-v7:27.1.1'
 - implementation 'com.android.support:exifinterface:27.1.1'
 - implementation 'com.android.support.constraint:constraint-layout:1.1.2'
 - implementation 'com.github.ittianyu:BottomNavigationViewEx:1.2.4'
 - implementation 'com.jakewharton:butterknife:8.8.1'
 - annotationProcessor 'com.jakewharton:butterknife-compiler:8.8.1'
 - implementation 'com.squareup.picasso:picasso:2.71828'
 - implementation 'com.android.support:recyclerview-v7:27.1.1'
 - testImplementation 'junit:junit:4.12'
 - androidTestImplementation 'com.android.support.test:runner:1.0.2'
 - androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
 - implementation 'com.android.volley:volley:1.1.0'
 - implementation 'me.zhanghai.android.materialratingbar:library:1.3.1'
 - implementation 'com.google.android.gms:play-services-maps:15.0.1'
 - implementation 'com.google.android.gms:play-services-location:15.0.1'
- The app is developed in Android Studio Version 3.1.2.

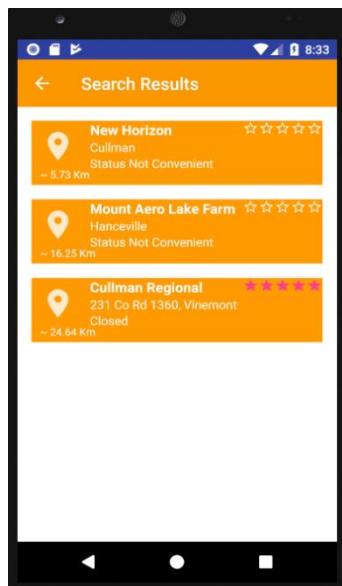
User Interface Mocks



Screen 1

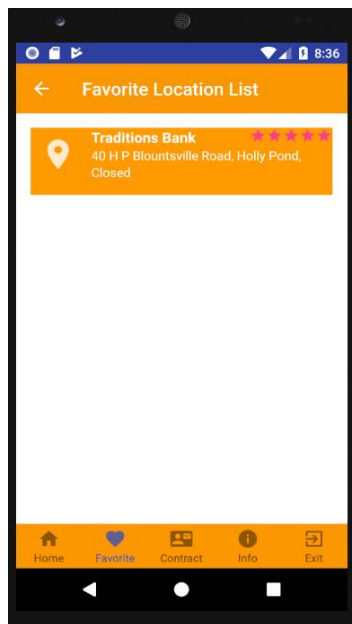
Main Screen provides users to select location type nearby them and navigate transition between icons in the bottom navigation bar. Moreover, there is an search icon on the toolbar to search any location type in the list.

Screen 2



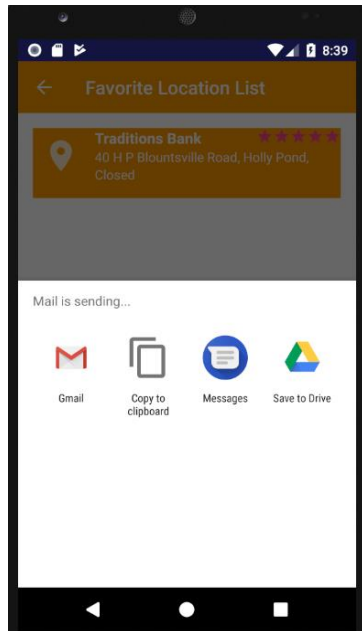
Search Option provides users to show location result list in the screen according to user's preference.

Screen 3



Favorite Location list provides users to list their favorite location stored in the database.

Screen 4



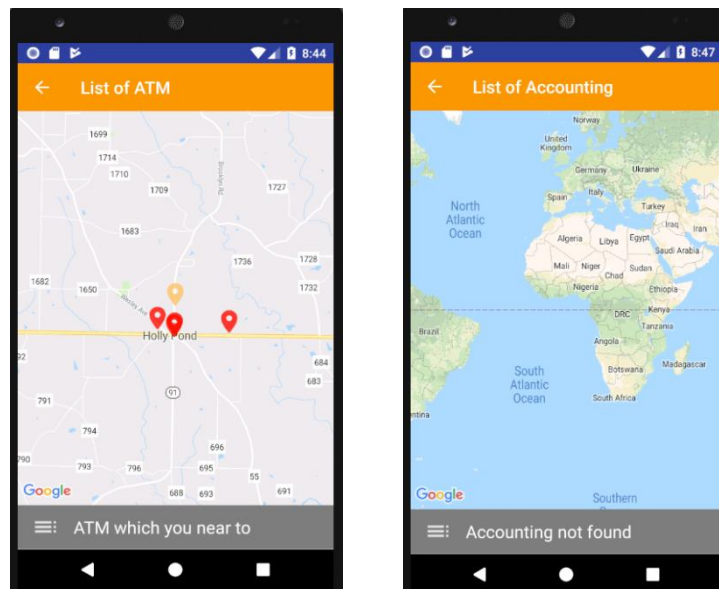
Contract Icon in the bottom navigation bar provides users to make a communication with developer to give a feedback about the app .

Screen 5



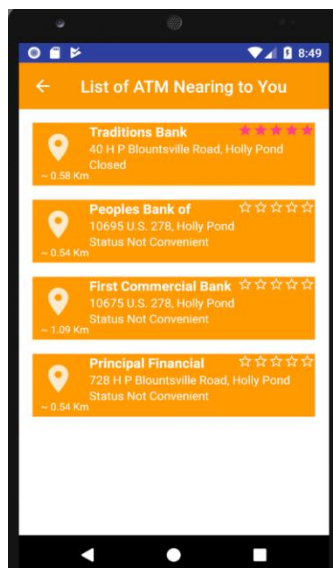
Info Icon in the bottom navigation bar provides users to get information about the app.

Screen 6



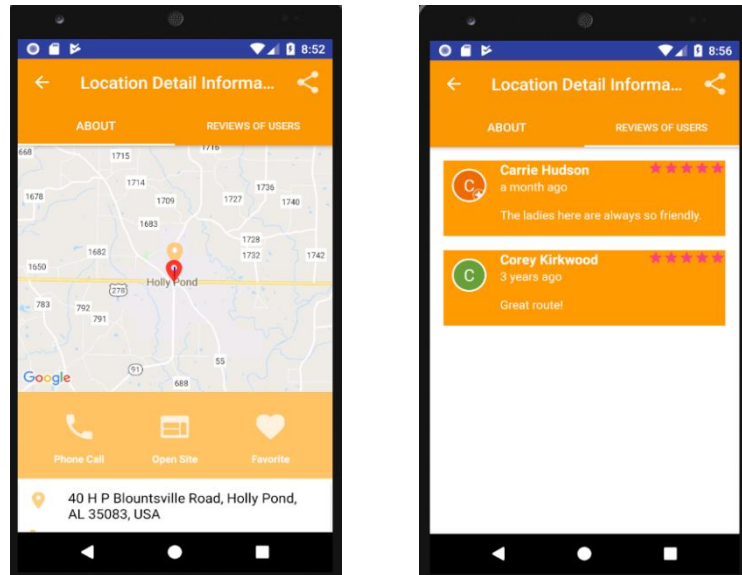
After user select any location type such as “ATM”, a new activity is opened to show its location nearby user and user’s current position. Moreover, a bottom divider side show a about location nearing to user. If there is no location nearby user, it shows that location is not found.

Screen 7



After clicking the bottom side, it shows location list nearby user with detail information and rating.

Screen 8



After clicking any location in the list, there are two parts to be shown. At the first side, it shows detail information about it. These are dependent on location's address, location's phone number and location status in terms of determining whether it is opened, closed or no status. Moreover, calling phone number, open website and adding it to favorites are added to this part. At the same time, user share this location to anyone. At another side, it shows users' reviews of location. This activity is also called from favorite and search side.

Add as many screens as you need to portray your app's UI flow.

Key Considerations

How will your app handle data persistence?

The app will be used for SQLite database to store location which users select in their favorite part.

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

- Use Picasso library to show location icon.
- Use Volley library to get location information from query based on JSON structure via GET method.
- Use Butterknife library to find and automatically cast the corresponding view in your layout in termd of data binding.
- Use BottomNavigationViewEx library to add bottom navigation bar with providing transition between activity easily.
- Use CircleImageView library to display circularly user's profile photo in Review Part.
- Use MaterialRatingBar library to display the rating of location.
- Use Android Material Design library to benefit from RecyclerView.

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

- Using Google Map Sdk for Android
 - Locate user's current position and nearby location around the user.
 - Calculate the distance between user's current position and the location which user selects.
- Using Google Firebase
 - Determine how much users are actively used this 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

- Learing Google Map Service to add it in the app.
- Search libraries on Internet for developing and using how to implement them in the app.
- Register Google Play Service to get API Key for query.
- Implement libraries used for the app in build.gradle file.
- Design the flow of the app.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Splash Activity to open its layout before opening Main Activity.
- Build UI for Main Activity where all location types shown in Google Map Sdk for Android are listed thanks to LocationMainListItemAdapter
- Build UI for Main Activity where BottomNavigationViewEx is implemented.
- Use BroadCast Receiver as network connection to determine whether the location is convenient or not.

Task 3: Add More Activity for Bottom Navigation Bar

- Add Info Activity to Bottom Navigation Bar to show who is developed the app.
- Add Favorite Activity to Bottom Navigation Bar to list location added from users in favorite part.
- Add Exit to Bottom Navigation Bar to exit the app.
- Add Contract to Bottom Navigation Bar to provide a connection between users and develop as giving a feedback about the app.

Task 4: Define LinkedHashMap array for location name with its icon

- Create LinkedHashMap for lcoation to get its information to be listed in MainAcitvty in terms of key value shown as locations' names and its value shown as drawable images files of location stored in drawable folder.

Task 5: Add LocationListOnMap and LocationListActivity with its Adapter

- Add LocationListOnMap Activity to add map in the screen and tab bar in the bottom of screen to open LocationListActivity.
- Add LocationList Activity to list location in certain style defined in LocationItemListAdapter.
- Use BroadCast Receiver as network connection to determine whether the location is convenient or not.

Task 6: Add Location Search Option

- Add LocationSearchResult Activity to show location which users find easily.
- Add Adapter to list location with adapter's style
- Use BroadCast Receiver as network connection to determine whether the location is convenient or not.

Task 7: Implement Data Model and Content Provider

- Create UserLocationRating class for showing users' reviews and Location class for giving detail information about location
- Create LocationDetailHelper class to determine the structure of database.
- Create LocationDetailContract class to the column of database with content authority url of the app.
- Create LocationDetailContentProvider class to determine CRUD operation of SQLite database.
- Create CursorRecyclerViewAdapter class to load information from database for favorite location list.

Task 8: Add Methods in UtilsMethod

- Create UtilsMethod class to add network connection method as a BroadCast Receiver, calculation of two location points and getting method for permission defined in Android Manifest file.

Task 9: Add Location Detail Activity with its Fragments

- Create Location Detail Activity to show the detail information of location and users' reviews about location by showing them in fragments named for LocationInformationAbout and LocationUserReview.

Task 10: Add Functionality to Favorite Icon ImageView

- Add a functionality to Favorite Icon ImageView. If user presses the icon at first and the icon is based on not filling heart, the selected location will be stored in the database with respect to its location id and icon converts to filling heart. If user presses this filling heart icon, the selected location will be deleted in the database with respect to its location id and icon converts to not filling heart.
- Use Loader to list locations related with user's favorite ones.

Task 11: Add Widget

- Add Favorite Location List Widget on the home Screen of the phone
- See the details of the selected location in the list by clicking it in the list of widget.
- Use Intent Service as a RemoveViewsService to define an FavoriteWidgetAdapter listing location with respect to adapter's style.

Task 12: Fix Issues and Errors

- Analyze the bugs thrown from the app.
- Fix some errors in the app.

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

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**"