GitHub Username: RehamGalal

Places

Description

This App is used to save user places want to visit location with description for it in order to keep information about it

Intended User

All people

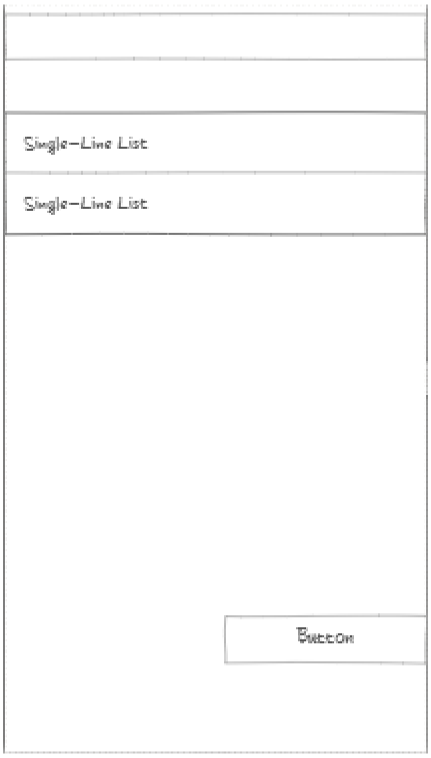
Features

- Save Location
- Show information when needed

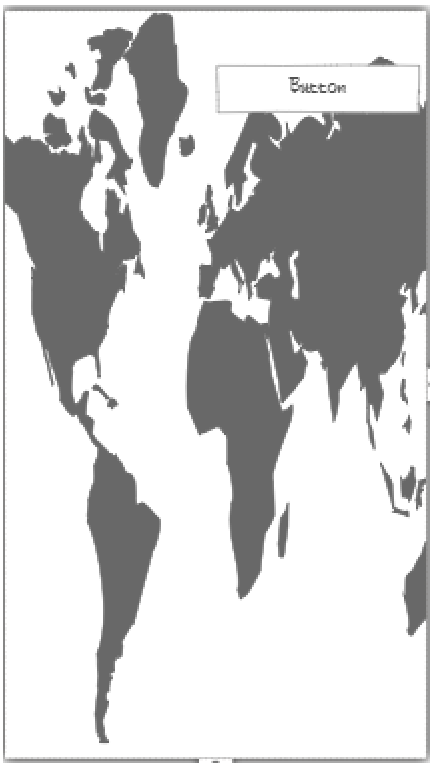
User Interface Mocks

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

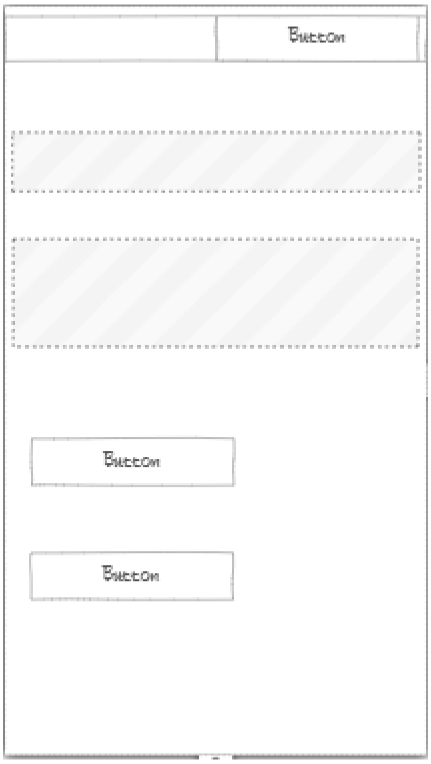


Screen 2

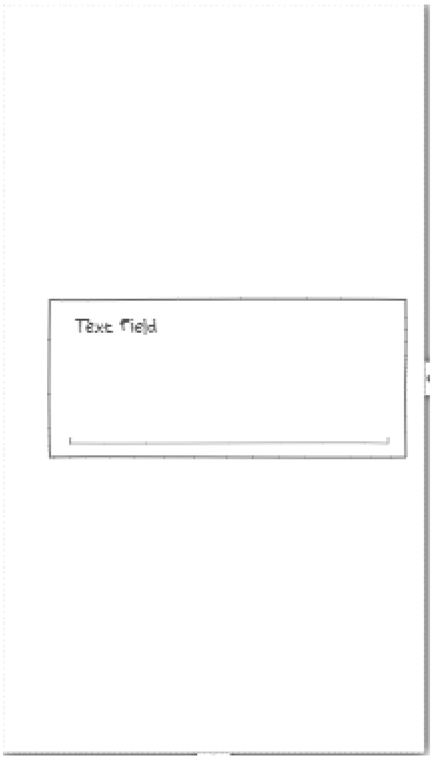


Add as many screens as you need to portray your app’s UI flow.

Screen 3



Screen 4



Key Considerations

How will your app handle data persistence?

SQLite for saving data, and Content Provider to control it

Describe any edge or corner cases in the UX.

When user press done button the exit the fragment the button also been removed from the screen.

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

ButterKnife I use it to ease the process of defining the objects on the screen in order to save time and don't make the code redundant.

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

I implemented googlemap service by adding googlemap activity and get an API to use it, and a banner to add ads to my 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

- App is written solely in the Java Programming Language
- Using content description in order to enhance app accessibility
- Use of strings.xml to store all the strings.
- Support for RTL layout switching in all layouts.
- App utilizes stable release versions of all libraries, Gradle, and Android Studio.
- Add ButterKnife library and needed google play services and recycler view in order to be ready to use them inside the app
- Follow the steps in order to get an API for google maps service

If it helps, imagine you are describing these tasks to a friend who wants to follow along and build this app with you.

Task 2: Implement UI for Each Activity and Fragment

- Adding the recycler view, floating action button, action bar.
- Adding text views and buttons to edit activity
- Adding plain texts and button to fragment
- Adding a button to the map activity to allow user to decide when to get out from the activity

Task 3: Implement Google Play Services

- Add API to google_map_api.xml
- Implement onMapReady to handle the marker and allow user to use it
- Get intent in order to decide the action depending on it

Task 4: Adding widgets

- Add widget to the app which show the name of the last showed place
- Handle the intent of the widget to open the map and add the marker on the place location

Task 5: adding google ads

- Add google adview to main activity
- Setup the ad and try it using the test add

Task 6: Build content provider

- Build content provider to handle all the insertions and querying processes
- Implement loaders in the main activity to query the database and show results

Task 7: fetching weather data from server

- To use App weather map to fetch weather data based on location we need to generate API for this task
- Implement asynctask to class in order to fetch data when running application inside it and populate the text view with the max and min temperature later

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**"