**Description**

**Intended User**

**Features**

**User Interface Mocks**

    Screen 1 (Home Screen)

    Screen 2 (Navigation Drawer)

    Screen 3 (Maps Screen)

    Screen 4 (Favorite Screen)

**Key Considerations**

    How will your app handle data persistence?

    Describe any edge or corner cases in the UX.

    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 4: Mocking Data

    Task 5: Develop Adding restaurant screen

    Task 6: Configure and develop data storage and login

    Task 7: Configure and develop location service

    Task 8: Configure and develop searching functionality

**GitHub Username**: MaherSoua

# Restop (Top Restaurant)

## Description

This may help user to find the best restaurant around an address or his location, user can save his favorite and add his data with a photo that he could take himself.
The application will be solely developed in JAVA language.

# Intended User

Families Travelers

# Features

List the main features of your app. For example:
- Save favorite list
- Geofences
- Login / Logout (Firebase)
- Save data on server to allow user to restore saved data
- User can search by name or address
- Restaurant position will also be displayed on a map
- User can add Restaurant by himself using a form
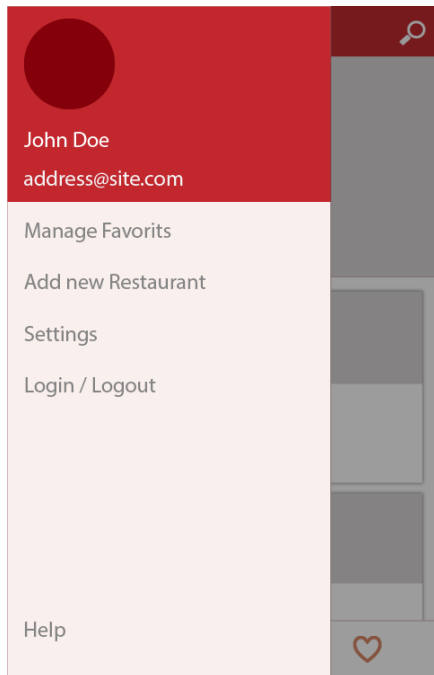- User can take a picture and add into the form

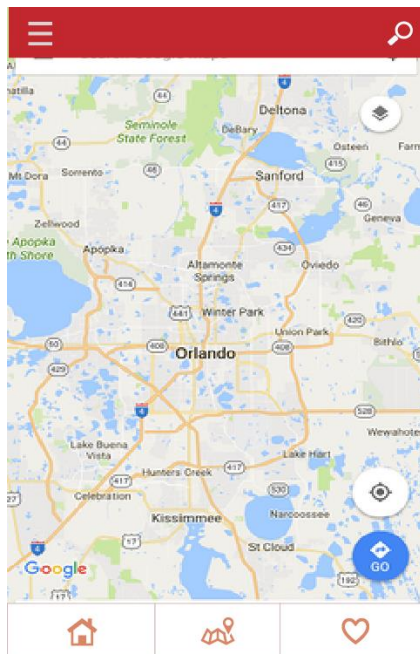# User Interface Mocks

## Screen 1 (Home Screen)
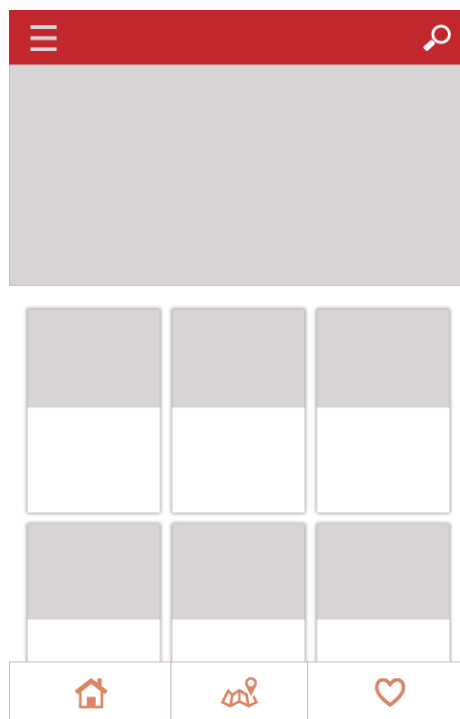


This is the home page, it will display a slide show and most known/rated restaurant.
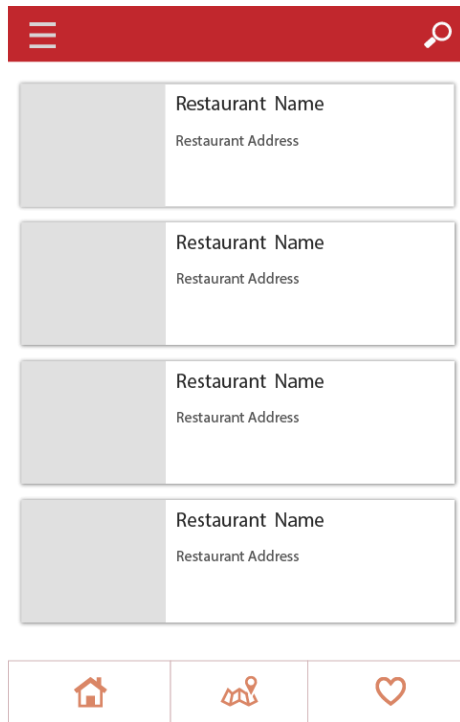
## Screen 2 (Navigation Drawer)



Menu drawer, as listed it will help to access to those different section.

## Screen 3 (Maps Screen)



This screen will display the restaurant that are near to user position

## Screen 4 (Favorite Screen)



This screen will display the user favorite list

**Widget Screen**



# Key Considerations

## How will your app handle data persistence?

The app should store data locally and in server, in this case we have to use content provider and also, he can save his data in server using Firebase Realtime database.

All colors value will be saved in colors.xml, all strings will be stored in strings.xml, all dimensions will be stored in dimens.xml, all style will be stored in styles.xml

## Describe any edge or corner cases in the UX.

In the app I will use Drawer navigation and bottom tab bar navigation, user can use tab to navigation between functionality (Map, Favorite, Home, Search)
Can use FAB to add new Restaurant
Navigation Drawer To edit profile Rapid navigation to different section
Top menu login /logout.
All Design will respect Material design recommendation, that will let user have good experience in the app relating to ergonomic layout and color.
App will use savedInstanceState to store temporarily data, in case of pausing app and resume, or when app change configuration.

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

Android Studio (3.2)
Gradle Version (4.1.0)
Picasso (2.71828) to load image, I like it because it is too easy to use and many great app use it
Retrofit (2.4.0) for Google api
Geofences to locate restaurant around
Firebase storage (16.0.1), messaging (17.3.0), core (16.0.3) and auth
Espresso (3.0.1) for test.
ContraintLayout (1.1.2)
Api Level 27.1.1

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

Place service (15.0.1) add it in gradle module
Google Services (4.1.0) add it in gradle project

# 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

- Configure libraries
- Organize gradle variable for lib version
- Create Build Variant

## Task 2: Implement UI for Each Activity and Fragment

- Build UI for Home Screen
- Build UI for Map Screen
- Build UI for Search Screen
- Build UI for Favorite Screen
- Build UI for adding restaurant
- Build UI for all navigation menu (NavDrawer, bottom tab nav)
- Build UI for Details screen

## Task 4: Mocking Data

- Create mock data
- Disable login

## Task 5: Develop Adding restaurant screen

- Implementing Form
- Implementing Saving data added by user
- Adding camera to take photo
- Adding zone to store location in the map

## Task 6: Configure and develop data storage and login

- Configure Firebase server side
- Implementing Firebase configuration
- Implementing Firebase Auth for login
- Add Content provider
- Save favorite restaurant with fake data on local machine and remote server

## Task 7: Configure and develop location service
- Implementing map
- Implementing location pin on map

## Task 8: Configure and develop searching functionality

- Implementing Searching
- Displaying searching result