GitHub Username: santbob

LetMeEat

Description

What and where do we eat today is a question everybody have to answer and decide everyday and its involves preferences and agreement when in group. We will help you answer that by providing 5 best places you or your group can eat from, at that time of the day. The recommendations will be based on the following factors

1. You or Your Friends social profile and preferences
2. Restaurant Ratings
3. Past Week Choices

Intended User

Everybody - Groups and Individuals who eats out regularly.

Features

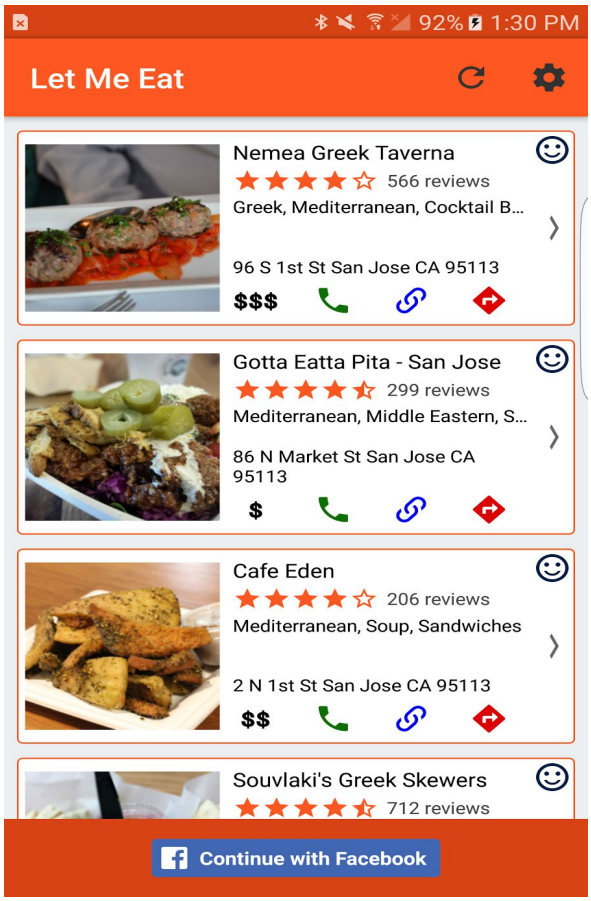
Here are some of the Main features of the app:

- Get Recommendations based on the preferences
- Ability to filter out restaurants which have been visited.
- Ability to Persist using a Facebook account so the preferences are applied when a logged from a new phone.
- Preferences Settings

User Interface Mocks

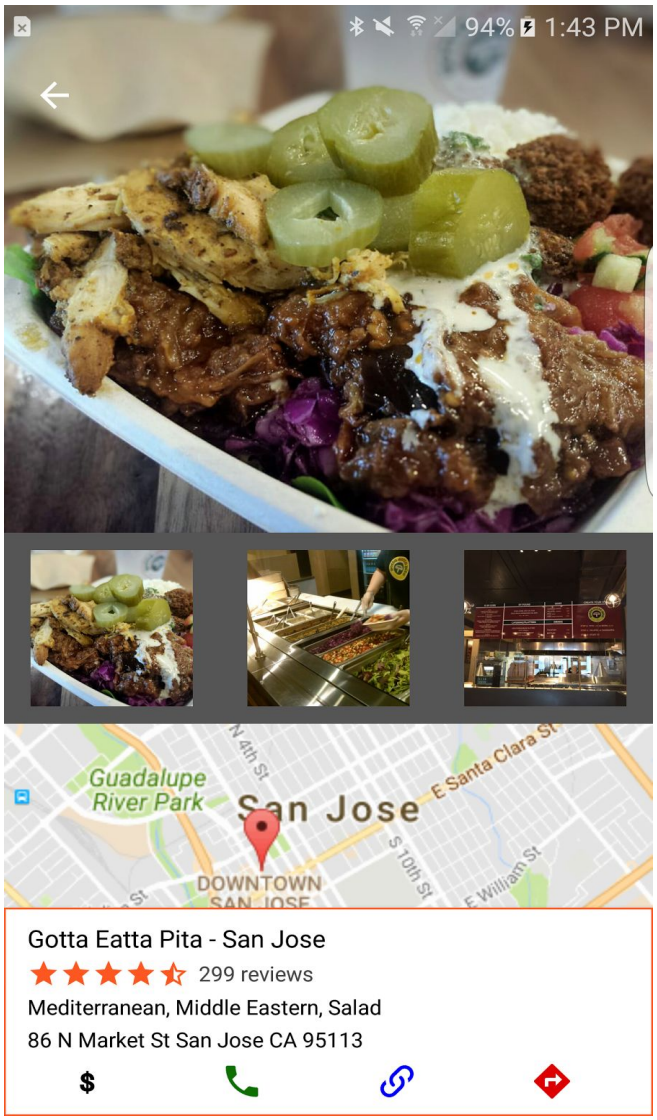
Screen 1 - Main Screen - Recommendations

This the main screen where 5 best recommendations will be shown based on the preferences. It will contain all the needed information to decide where to eat, name of the Restaurant, Ratings, Cuisine Type, Price Range and where it is.



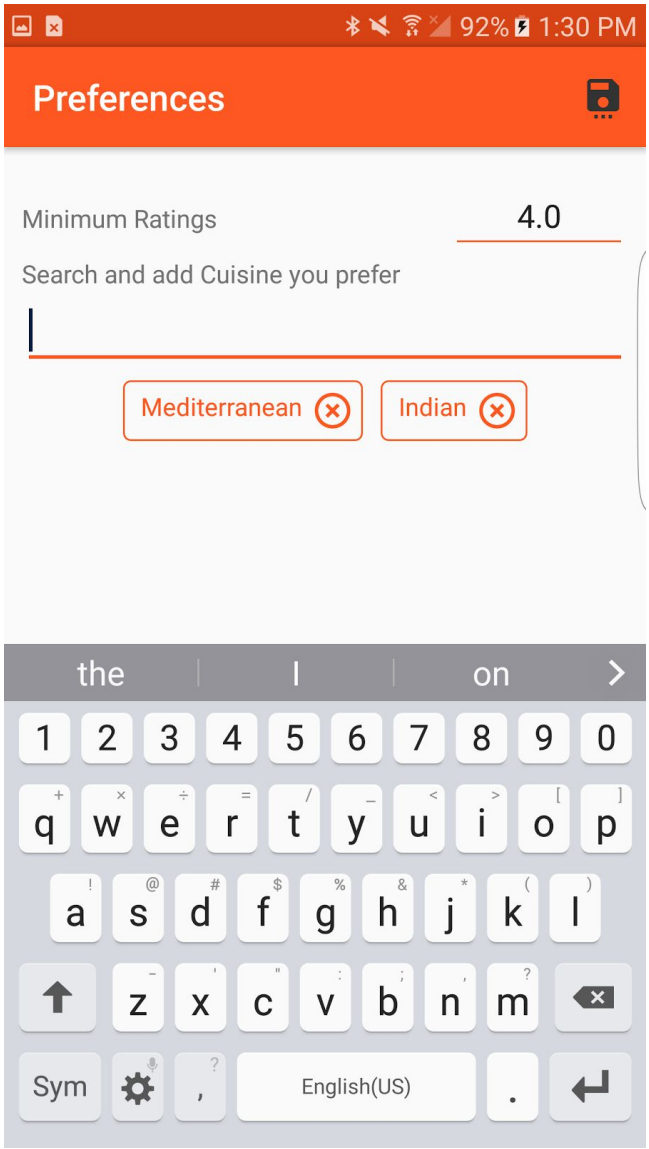
Screen 2 - Details Screen

Back of the card will contain more pics from the restaurants about food and users can see bigger image from the picture strip.



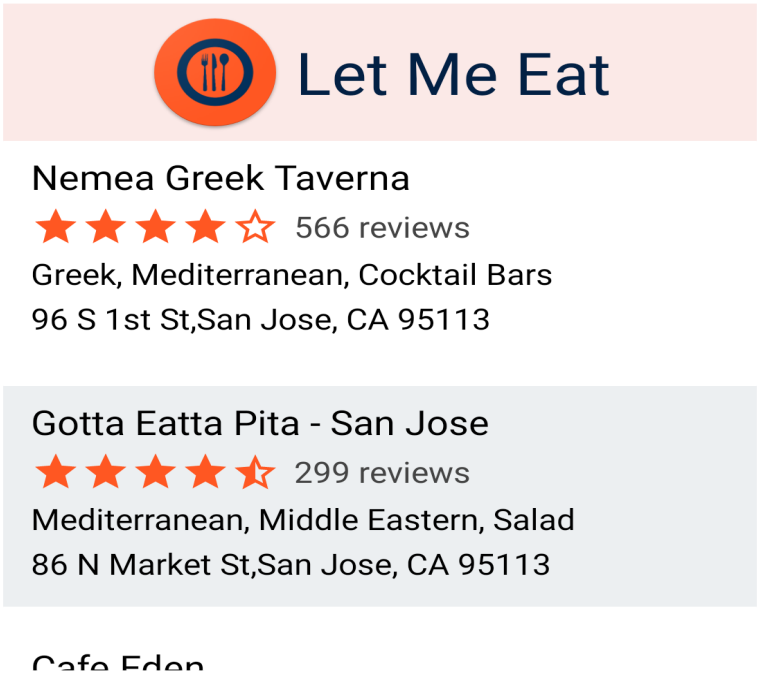
Screen 3 - Preferences

All the Personalization that user can do will be here in the preferences, currently it will have Cuisine and Minimum Ratings required.



Screen 4 - Widget

App will be released with a widget, where users can put it in home screen and it will be showing the top 5 recommendations.



Key Considerations

How will your app handle data persistence?

All the data will come from server as the Recommendations are dynamic, but the Preferences and Recommendations after receiving from the api will be stored locally for few hours as the recommendations won't change for the meal, unless a new user is added to the group.

Describe any corner cases in the UX.

Nothing, i could think as of now.

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

Retrofit - for Making Http calls
Picasa - for downloading images and caching
Moshi - JSON parsing library to parse JSON to Java and vice versa.
Design - Support Library for smooth Animation and design.
Crashlytics - To record app crashes with details.
FacebookSDK - to integrate facebook login.

Describe how you will implement Google Play Services.

Maps - Each card will show a snapshot of map for the recommendation.
Firebase - This acts a backend to the app to store user session and preferences
Gson - for parsing json data used in the app.

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

Create a new Android application with Navigation

Here are the list of things to configure:

- Libraries - Retrofit, Picasa, Moshi, Design and Facebook Libraries
- Decide on the Color Palette
- Build the launcher Icon and drop it in the app
- Define the common dimensions to use.
- Push to GitHub

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build the Main Screen, using a RecyclerView to show list of recommendations
- Build the List Item View to show the recommendations
- Build the Details Screen to show pictures, maps and other details.
- Build the Preferences Screen
- Build the Widget to Show, top 5 recommendations for Current Meal

Task 3: Integrate the API

Add the Data fetching and storing functionality

Here is the list of the steps to integrate the data:

- Create a Retrofit API Service Interface to start fetching data.
- Create bunch of Java Object equivalent to the JSON that we are going to receive from server.
- Create DataBaseHelper, which can create db, tables and the queries.
- Create Content Provider to expose the data.
- Create a Loader to get the data using the api.
- Integrate the UI with their corresponding API calls

Task 4: Permissions & Notifications

Add the Permissions and handle the scenarios of getting the location permission and check permission before it.

Location Permission is needed to find the nearest restaurants and sort the recommendations by best_sorting algorithm.

Task 5: Register for Google Services

Here are the Google Services we will be using:

- Create a Google Play App Id
- Register for Maps
- Register an app with facebook for Facebook Login

Task 6: Set Up Firebase Server

Firebase will act as backend:

- Create a firebase account and project
- Integrate third party apis to get ratings and details about restaurants
- Build the api required for the api.
