
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. Time of the day
2. You or Your Friends social profile and preferences
3. Restaurant Ratings
4. Past Week Choices

Intended User

Everybody - Groups and Individuals who eats out regularly.

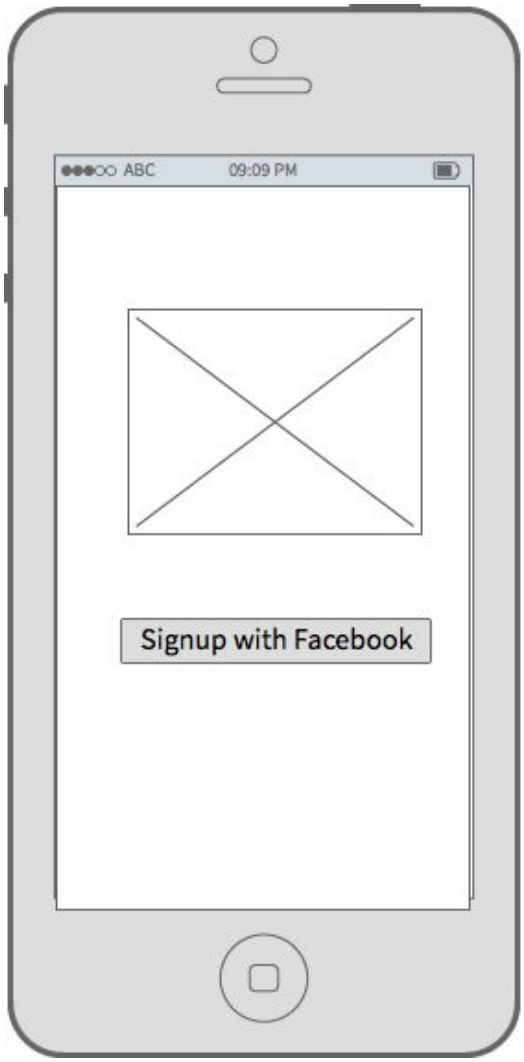
Features

Here are some of the Main features of the app:

- Account Creation.
- Talk to Server to get the recommendations
- Groups Creation
- Notifications when it's time to eat and notify group about restaurant selection
- Favourite or UnFavourite the restaurants
- Preferences Settings

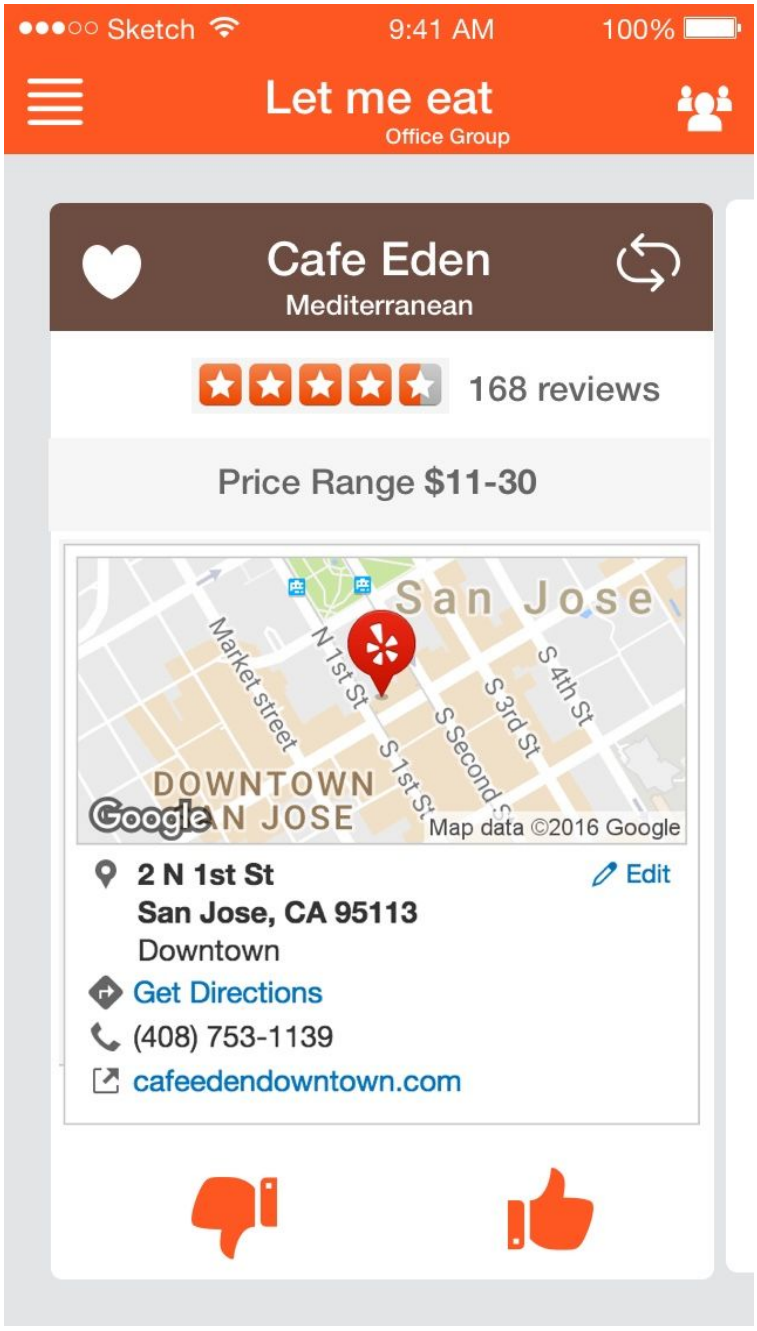
User Interface Mocks

Screen 1 - Login Screen



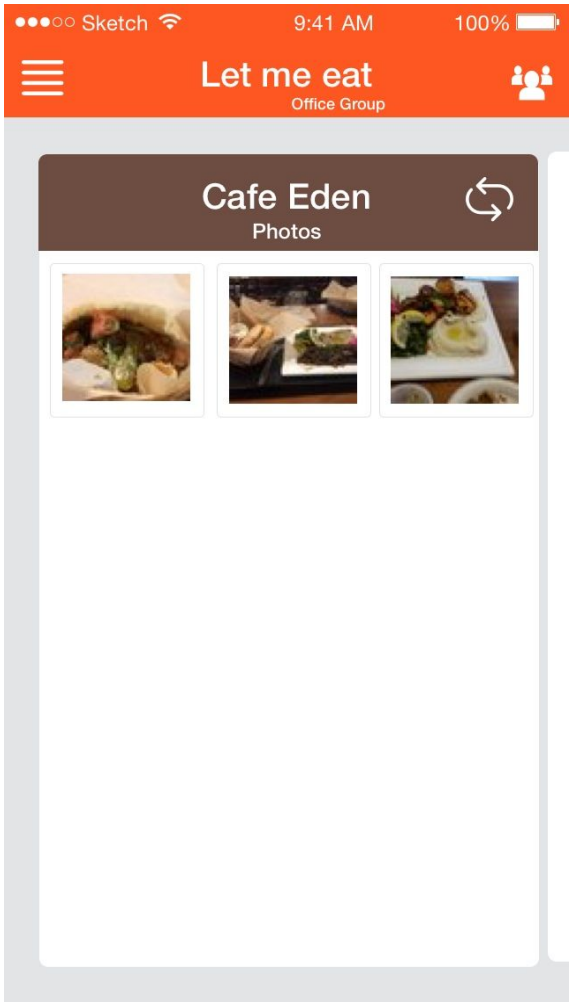
We are going to support only Facebook login to start with.

Screen 2 - Main Screen - Recommendations (Card Front View)



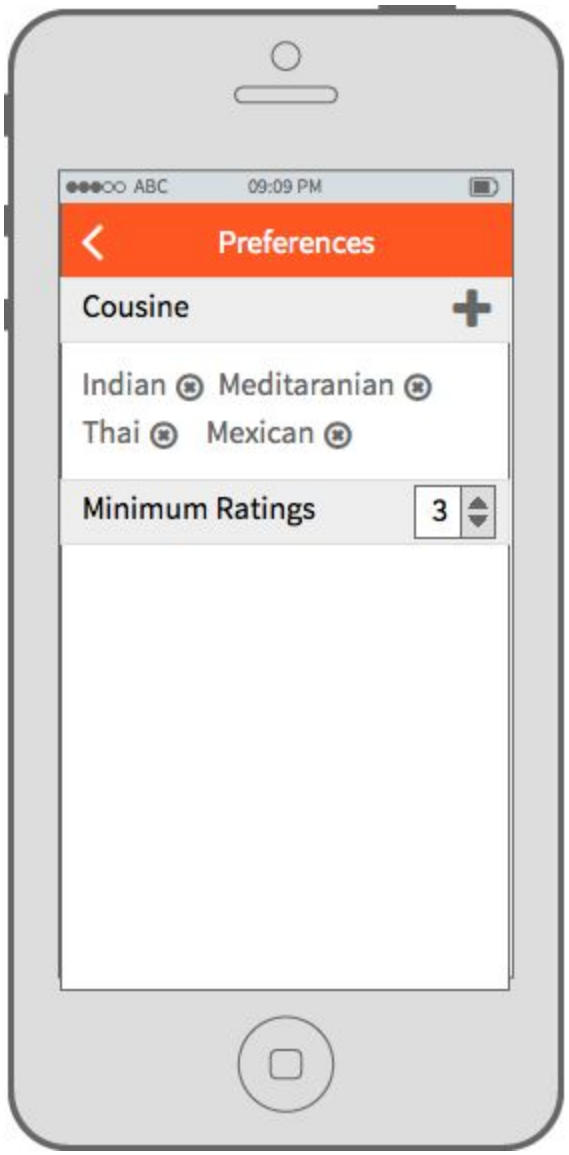
This the main screen where 5 best recommendations will be shown based on the preferences. It will contain all the needed information for the individual or group to decide, name of the Restaurant, Ratings, Cuisine Type, Price Range and where it is.

Screen 3 - Main Screen - Recommendations (Card Back View)



Back of the card will contain Pics from the restaurants about food and upon clicking the picture, it animates and opens the pic in full view.

Screen 4 - Preferences



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

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.
CardView Support Library for drawing the cards.

Describe how you will implement Google Play Services.

Ads - Ads will show up for the free version at the bottom of the app.
Maps - Each card will show a snapshot of map for the recommendation.
Firebase - This acts a backend to 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 CardView Support 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 Login Screen
- Build the Main Screen, this will be Pager with 5 cards.
- Build the Custom View for the Card - Front Side
- Build the Custom View for the Card - Back Side
- Build the View for the Full View of the Image.
- Build the Navigation Adapter for other pages in the left nav
- Build the Preferences Screen
- Build the Screen to Create groups and Invite people.

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 Sync adapter to keep the recommendations up to date.
- Integrate the UI with their corresponding API calls

Task 4: Permissions & Notifications

Add the Permissions and handle the scenarios of getting the permission and checking permission before using them:

Here are the permission we need:

- Location Permission - to find the nearest restaurants and sort the recommendations by distance.
- Request Push Notifications Access - to notify users its time to eat and also notify group when invited and when a restaurant is selected by the group.

Task 5: Register for Google Services

Here are the Google Services we will be using:

- Create a Google Play App Id
- Register for Admob
- 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.
