
[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any 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.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

GitHub Username: [Suparna-here](#)

Zomato

Description

zomato is an android application that suggest you nearby restaurants based on your location and cuisine preferences. You can get the location details of each restaurant, it's rating, average cost, up votes, popularity, top cuisines and much more.

User can mark some restaurants as favorite. Those restaurants will be saved in the device local database also displayed in the app widget.

Intended User

Intended users can be anyone who is hungry but primarily traveler as this app will help them to find the best restaurant nearby based on their cuisine interests.

Features

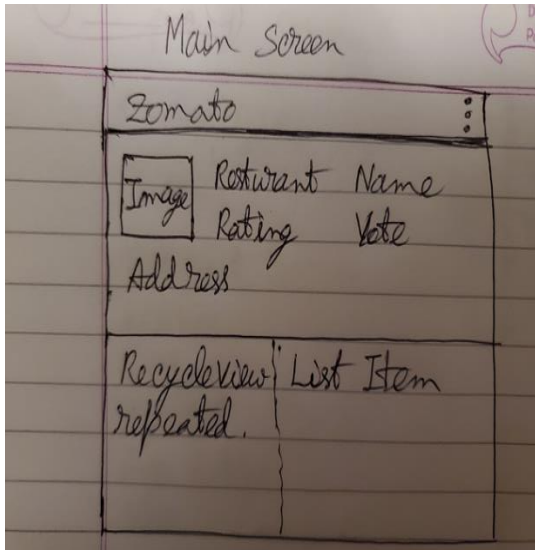
Find the best restaurants nearby.

- Bookmark your favorite restaurants.
- Set cuisine preferences to find the restaurants of your interest.
- User friendly material design.
- Find a way to any restaurant from your current location.

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.

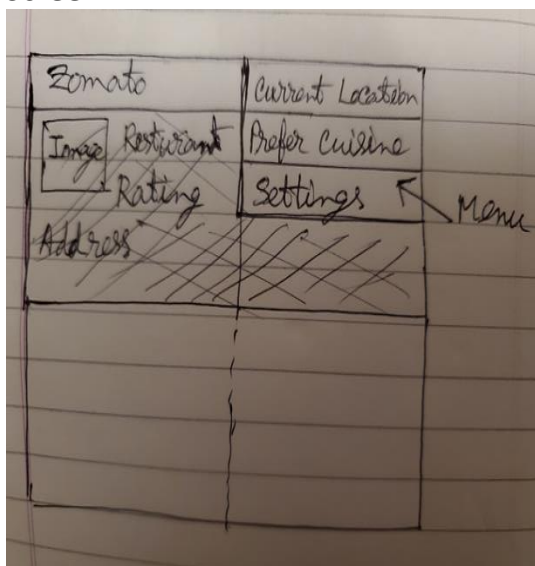
Screen 1



Replace the above image with your own mock [click on the above image, then navigate to Insert → Image...]

Main Screen - This screen is displayed when the app is started. By default, the nearby restaurants will be displayed according to the location of the user. User can move to favorite Screen by choosing from the menu. In favorite screen items will be displayed in the same format but will be available offline. Permission to access location will be requested as the app is started for the first time.

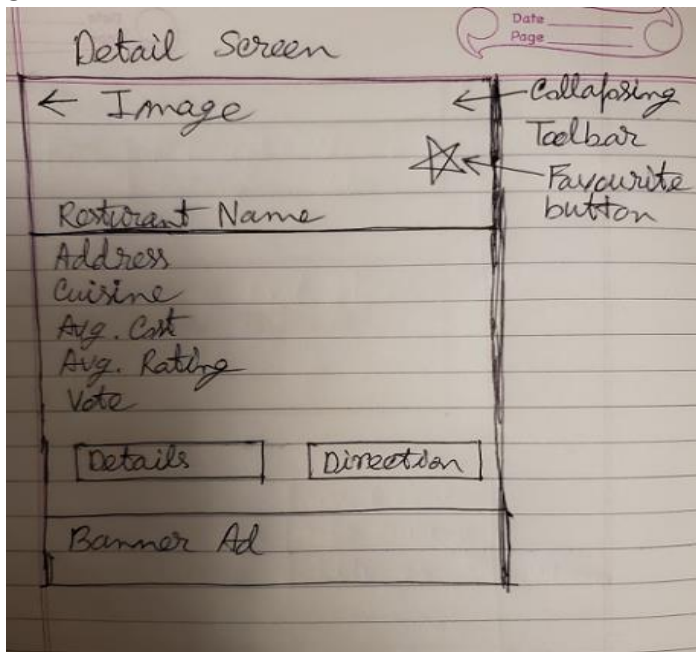
Screen 2



Replace the above image with your own mock [click on the above image, then navigate to Insert → Image...]

Settings – This will be displayed on clicking the menu button from the app bar. It'll consist of some common configuration options like open Main Screen or Favorite Screen, radius to find nearby restaurants etc.

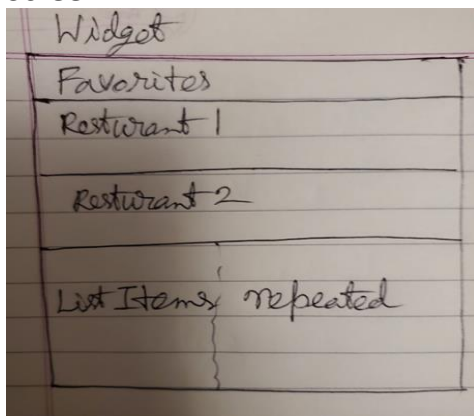
Screen 3



Replace the above image with your own mock [click on the above image, then navigate to Insert → Image...]

Detail Screen - This will be displayed when we click a restaurant card on the main screen. Up navigation will be provided. Image will be loaded by a background thread implemented using AsyncTask. Zomato website will be opened on browser by clicking the details button. Directions button will open the google map showing the direction to restaurant from current location.

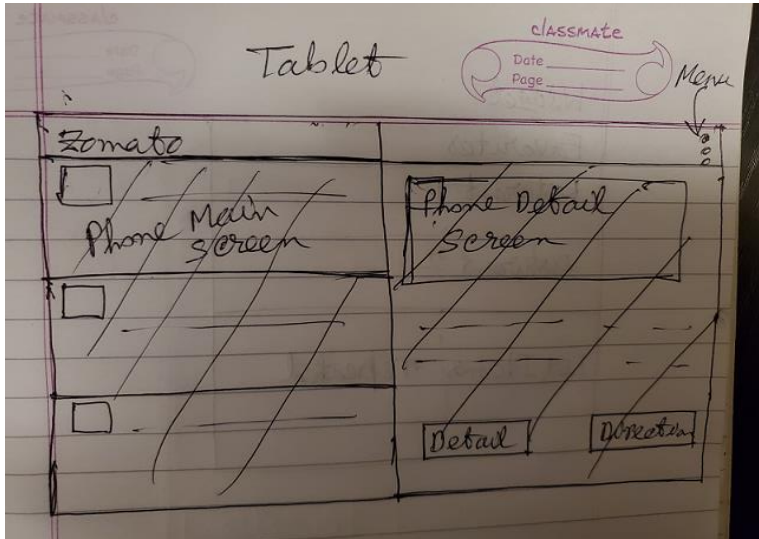
Screen 4



Replace the above image with your own mock [click on the above image, then navigate to Insert → Image...]

Widget – This will be displayed on the home screen showing the list of restaurants marked as favorite.

Screen 5



Replace the above image with your own mock [click on the above image, then navigate to Insert → Image...]

Tablet View – This will be displayed when the app is opened in tablet.

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

Key Considerations

What are the Apl versions and programming Languages will be used?

- Programming Language: Java
- Android Sdk version: Android 9(Pie), API 28,
- Google Places Sdk version: 16.0.0

How will your app handle data persistence?

For any restaurants that'll be marked as favorite, the data will be stored in android SQLite database. Room Library will be used to have the advantage of executors and livedata for a better implementation and user experience.

Describe any edge or corner cases in the UX.

- If internet is not available, the app will display a blank screen with text - No Internet available.
- If the user has no cuisine preferences, by default all cuisines available will be used to display in search results.
- Favorite restaurants will be available even when the internet connection is not available as well as will be displayed automatically in widget.
- App will either use AsyncTask for loading image or use JobDispatcher to update favorite restaurant data in device periodically.

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

- Picasso or Glide to handle the loading and caching of images.
- ButterKnife to avoid using multiple findViewById() calls.
- Room to have the features like:- livedata and easily connect with android SQLite database.
- Retrofit to connect with external api along with gson.
- Zomato API - Connection will be made manually using an API key
- Material Design Components like:- constraint layout, floating action bar etc.
- Google Play Services.
- AdMob for showing Banner Ad in detail Screen.

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

Google Play Services will mainly be used to track the current location of user and using places api to get the address based on lat-long.

AdMob API will be used to show Banner Ad.

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

Setup all the libraries that are mentioned in this document as well as configure the external api which will be used by this application to find the data about nearby restaurants based on the location.

Task 2: Implement UI for Each Activity and Fragment

- Setup basic UI for all the activities, fragments and widget.
- Implement the flow from one activity to another.
- Setup UI for phone and tablet layouts and make sure they're working as expected.

Task 3: Setting up Zomato API

Setting up the zomato api in our android application. Implementing connections using retrofit and gson library.

- Defining constants for base url and api key.
- Defining different methods, each for a specific purpose to interact with the api.

Task 4: Setting up location services and google places API

Setting up location tracking in our android application and google places api to fetch the location details based on coordinates.

Task 5: Setting up AdMob API

Setting up AdMob API to show banner Ad in detail screen.

Task 6: Testing the functionality and improve UI

Connecting different functionalities to work together in synchronization. Handling errors and showing user friendly messages. Using material design to build a better UI and make sure that all the specifications are properly followed and implemented.

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**"