How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]
2. Name your document file: "**Capstone_Stage1**"
3. Replace the text in green

GitHub Username: mahmoud1992

Akel in Arabic (eat)

Project requirements

- App is written solely in the Java Programming Language.
- App includes support for accessibility. That includes content descriptions, navigation using a D-pad, and, if applicable, non-audio versions of audio cues.
- App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts.
- App utilizes stable release versions of all libraries, Gradle, and Android Studio.
- AsyncTask is used to load the image in the detail screen.

Description

Akel 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, its rating, average cost for two people, total number of votes, popularity, top cuisines and much more You can mark some restaurants favorites, which will be shown in home screen widget and also share the details of any restaurant. It helps the users to find a way to any restaurant from their current location by directing them to Google maps.

Intended User

Intended users can be anyone who is hungry but primarily travelers 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.
- User friendly material design.
- Find a way to any restaurant from your current location.

User Interface Mocks

Screen 1

Main Screen:

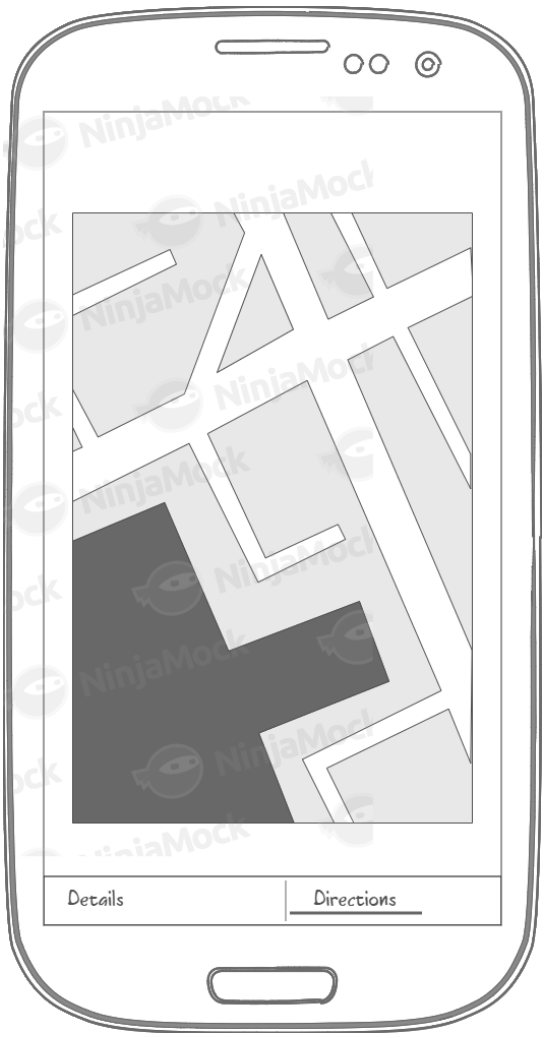
This screen is displayed when the app is started. By default, the nearby restaurants will be displayed according to the location of the user. Data in favorites tab will be displayed in the same format but will be available offline. Permission to access location will be requested as the app is started for the first time.

Screen 2

Restaurant Detail Screen:

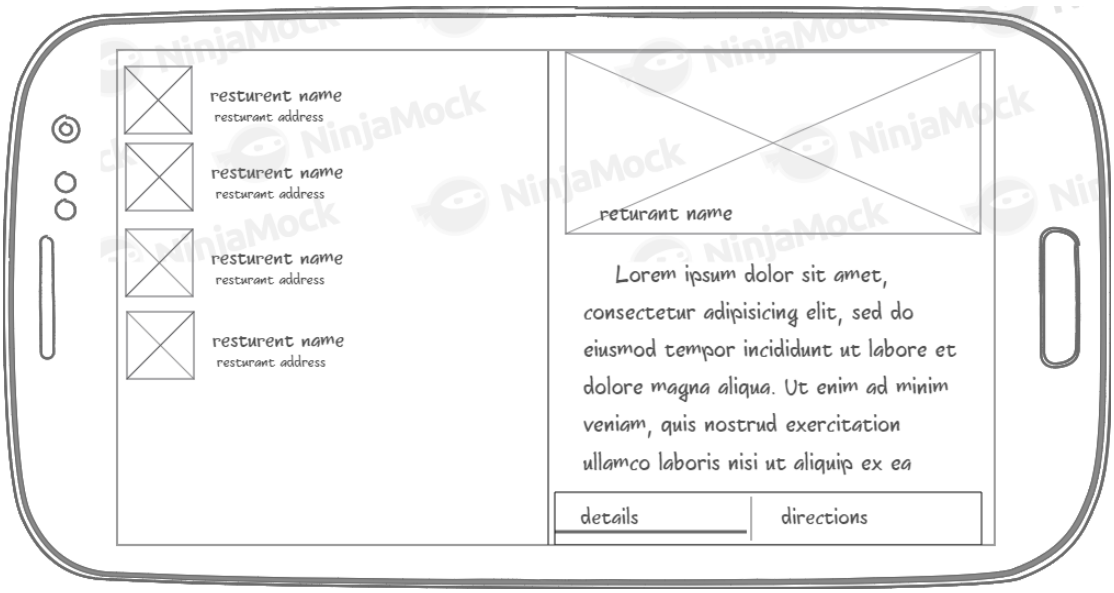
This will be displayed when we click a restaurant card on the main screen. Up navigation will be provided. Image will be loaded by a background thread implemented using AsyncTask. Browser will be opened on clicking the details button and directions button will open the Google map showing the direction to restaurant from current location.

Screen 3



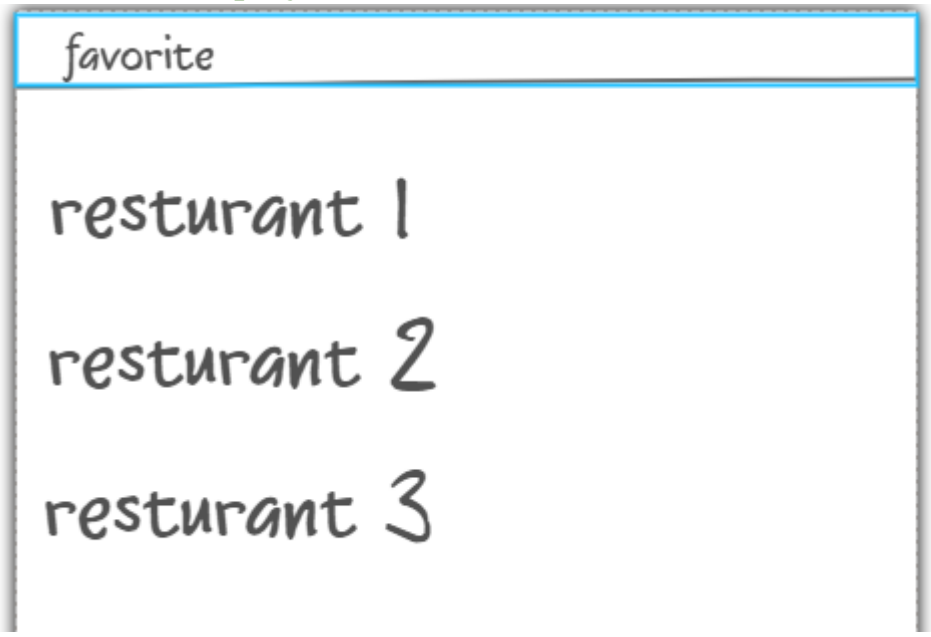
Directions:
Google map showing the direction to restaurant from current location.

Screen 4



Tablet View:

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



Widget:

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

Key Considerations

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 live data 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 search results.
Favorite restaurants will be available even when the internet connection is not available as well as will be displayed automatically in widget.

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

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-lng.

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 our 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.

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**"