GitHub Username: emadabel

My Dishes

Description

My Dishes is a cooking recipes App, you can search with some keywords to get the full recipe for your delicious meal, And you can save at favorites and explorer it offline.

Intended User

This app for cooking lovers.

Features

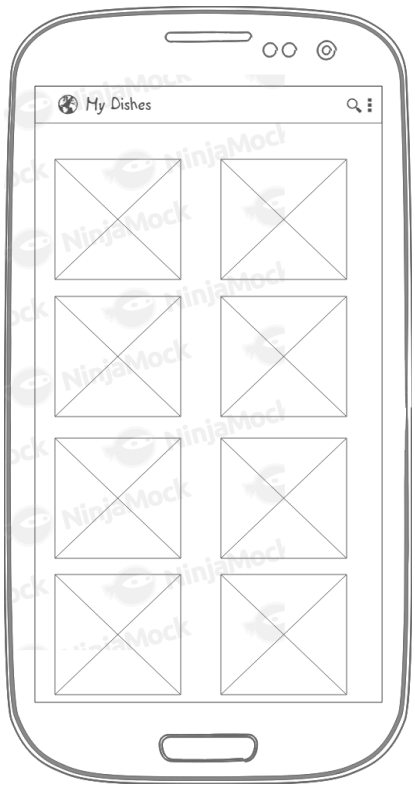
List the main features of your app. For example:

- Getting daily updated list of top recipes
- Search for a recipe
- Getting recipes by ingredients

- Save recipes to favorite list
- Share recipes with friends

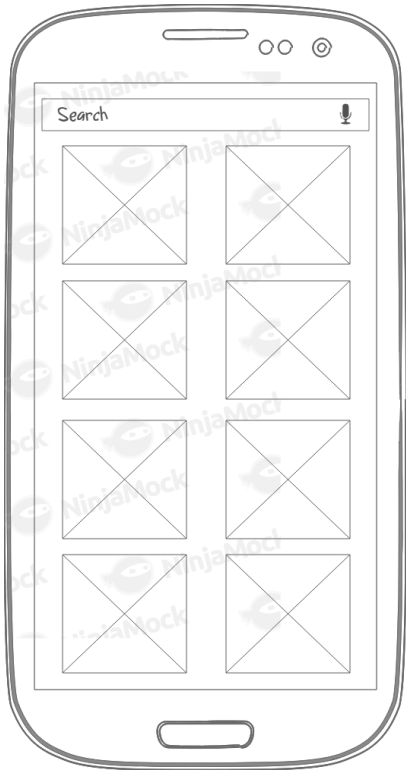
User Interface Mocks

Screen 1



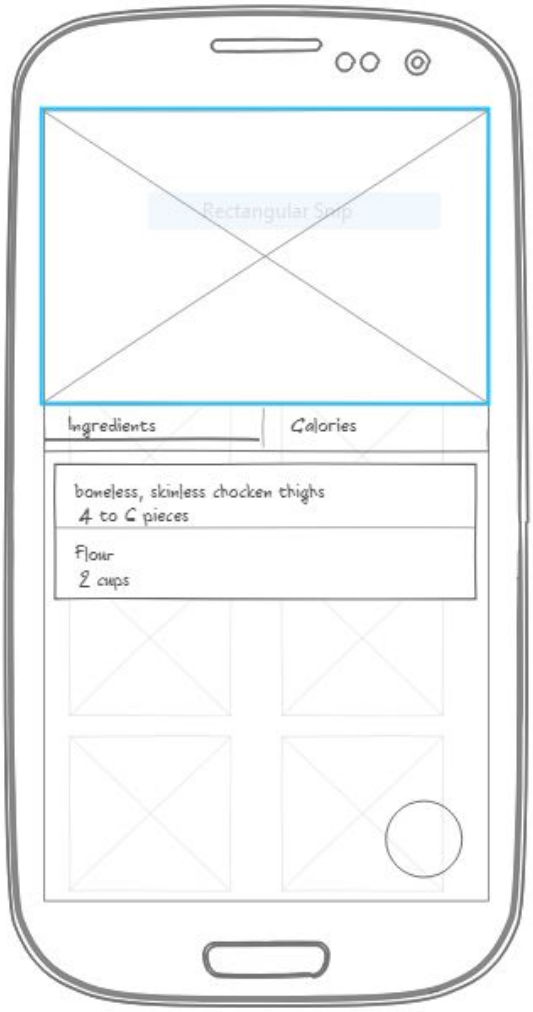
This is the main screen, it shows a grid of random recipes with image and title of the recipe

Screen 2



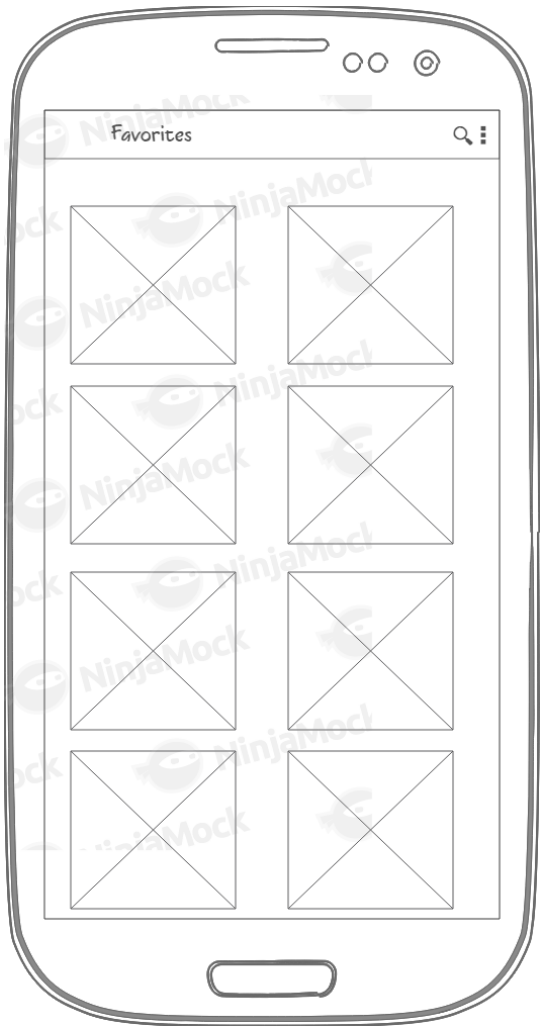
This is the search screen, when the user click the search button, the search box appear. And when the search finished the result shown as a grid for returned recipes.

Screen 3



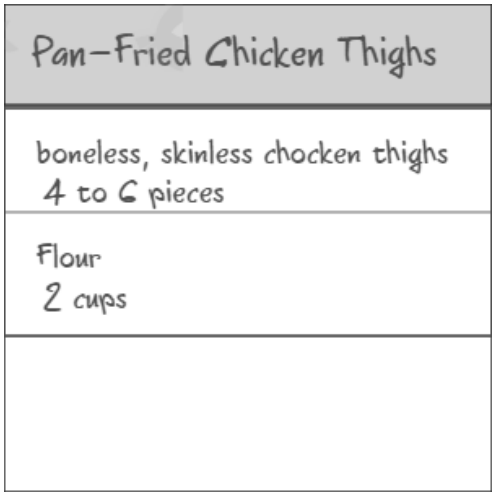
This is the details screen, it shown when the user click on the recipe icon from (main, search, favorites).

Screen 4



This is the favorites screen, you can navigate to favorites from the menu, it shows the saved recipes and you can browse its details in offline mode.

Widget



Widget shows the ingredients of the selected recipe.

Key Considerations

App is written in Java language, and using stable release of all libraries, Gradle, Android Studio

How will your app handle data persistence?

I will use Room to save favorites in the database and using LiveData and ViewModel where there is no necessary calls on the database.

Describe any edge or corner cases in the UX.

When the app start it start fetching data from the internet, Users can open Favorites or Search from menu, And can using the navigation bar or back button to back to previous screen.

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

Retrofit 2.4.0 : to handle network calls and convert response to object by the internal gson converter.
Butterknife 8.8.1 : to bind views easily and reduce boilerplate codes.
Glide 4.7.1: is awesome library to loading images from the network and have caching capability and good performance.

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

I will use Admob to show ads and Firebase Analytics.

Next Steps: Required Tasks

Task 1: Project Setup

- Find good Api to fetching the data from
- Register and acquire the API key
- Testing the API and get Json data from it to figure out how POJO will be created.
- Design the mockups
- Creating Android Studio Project
- Configure the libraries

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build RecyclerView Adapter
- Build UI for SearchActivity
- Build UI for DetailsActivity
- Build UI for FavoritesActivity
- Build menus

Task 3: Implement Networking

Implement retrofit interfaces and instance, and creating models that will be using to carry the response data

- Add internet permission in the manifest file
- Create models aka POJO
- Create interface for retrofit calls
- Creating Singleton class to handle retrofit instance
- Making all network calls from AsyncTask

Task 4: Implement Data Persistence

- Add room dependencies

- Creating Entity object
- Creating Dao object
- Creating Database class
- Add room insert logic to FavoritesActivity
- Add room delete logic
- Add room query logic for all items
- Add room query by id logic

Task 5: Implement Google Play Services

- Create Firebase project from console
- Add Analytics dependencies
- Create Analytics instance to log some events
- Add Admob dependencies
- Add Admob view

Task 6: Testing and handling error cases

Testing the app, and checking all logics, Log outputs, Log Exceptions and handle null response

---

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**"