
GitHub Username: ankan-anurag

GoLocal Kitchens

Description

Many households, especially from lower or lower-middle class families in developing countries would love to have an extra income while doing something they like, cooking. But how do they market themselves, or make themselves visible to prospective customers. This app will give them a platform to share what they can cook with people in their locale. A consumer will be able to see all the kitchens nearby to his/her location, be able to see the dishes they prepare, location on the map of the kitchen, their contact number/email and review them so other consumers like him/her can benefit.

IN FUTURE:-
In future, it should be able to do more things like request/place an order on weekly/monthly subscription to their kitchens. For kitchen owners, the app can help them with managing orders and invoicing, update their daily menu on the app so subscribers are up to date, etc.

Intended User

This is an app for local entrepreneurs who want to cook food on contract basis and consumers who are looking for good quality homemade food in their locale.

Features

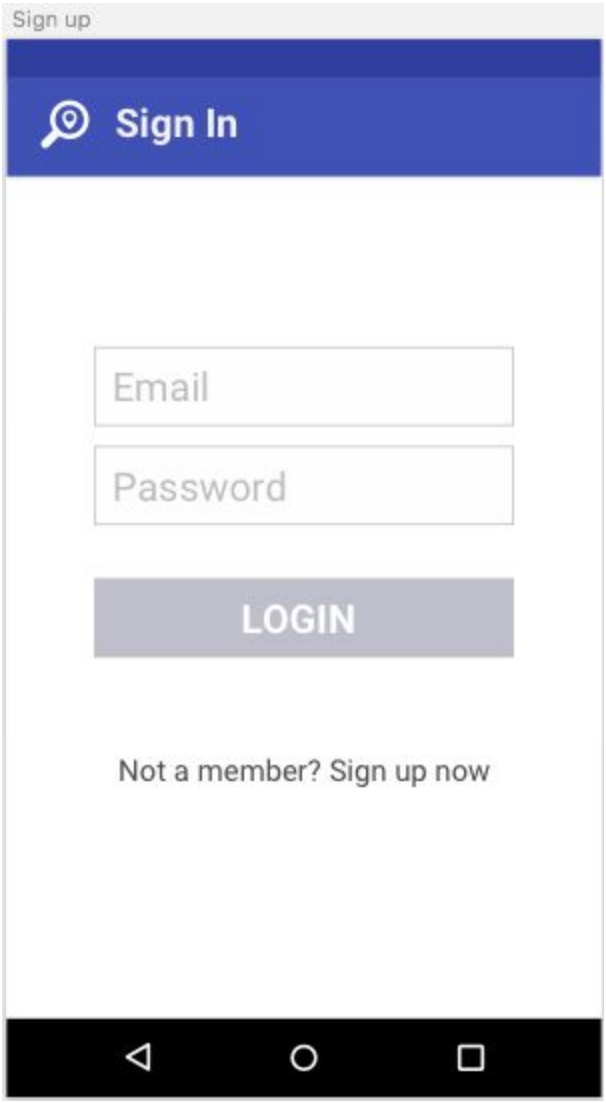
The main features of the app are:

- Saves profile and menu for kitchens to market themselves to prospective customers.
- Let consumers look for best homemade food options near them.
- Save their favourite kitchens for offline viewing.
- Being able to review kitchens, so other consumers can benefit from it.

User Interface Mocks

These are the mocks created on sketch 3. Few things may not be perfect or missing, but it will give an overall idea of the flow of the app and requirement of each screens.

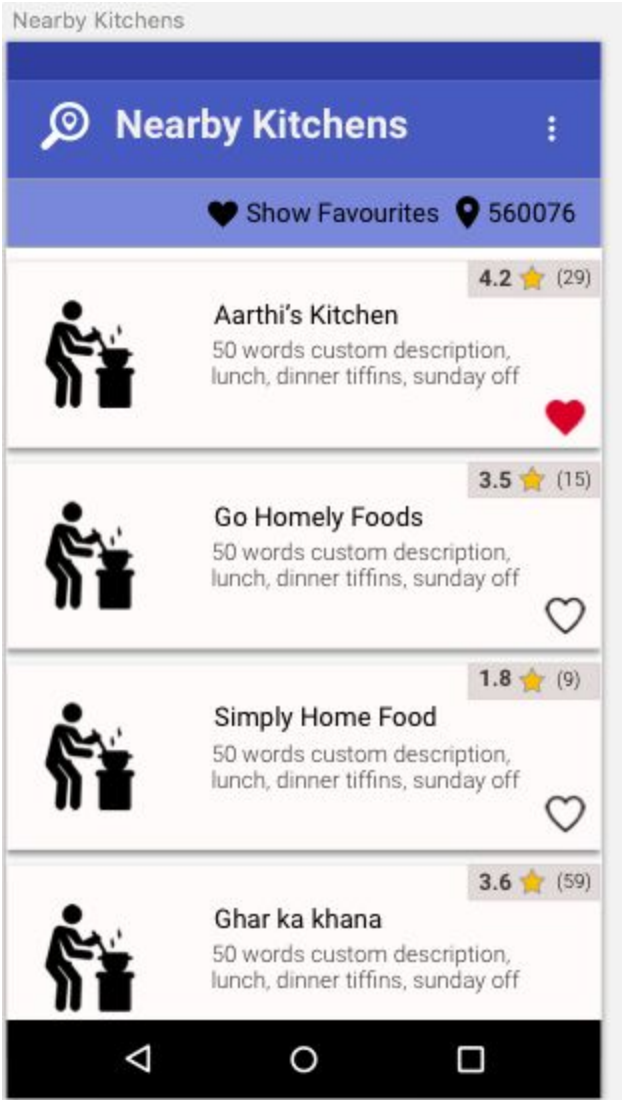
1. Sign up Screen



Sign up screen for users. Use firebaseUI to implement this. Firebase will take care of authentication/authorization of users, in addition to handling other cases like email reset mail when you forgot password, google smart lock feature, etc.

Reference: https://github.com/firebase/FirebaseUI-Android

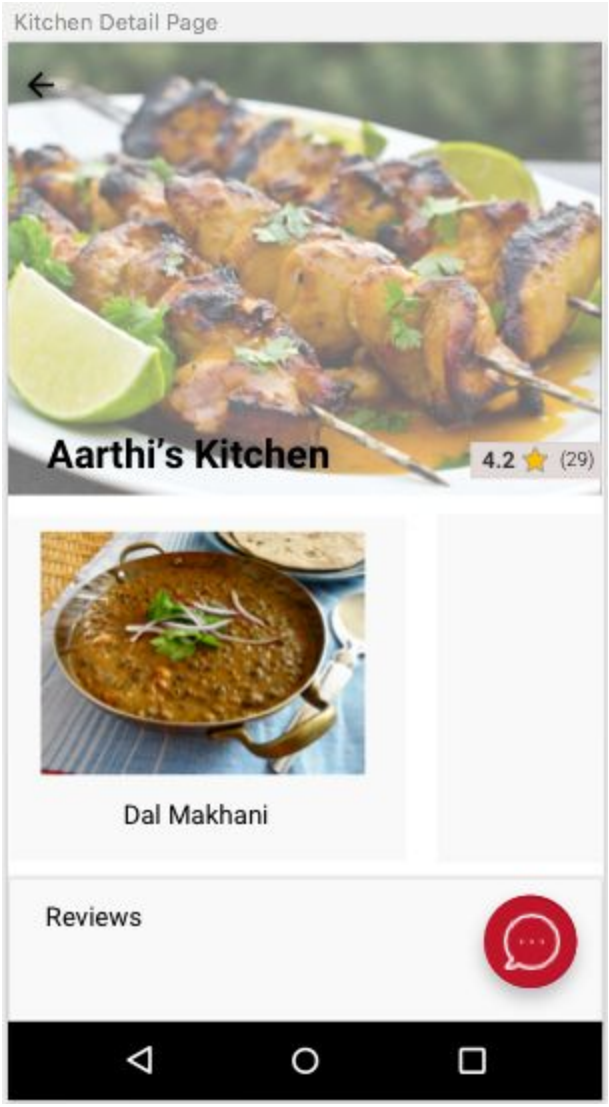
2. Nearby Kitchens



This is the landing screen for users once they have logged in but do not have their own kitchen. They can select “”show favourites” to show only their favourite restaurants. Also, clicking on the heart icon in the list item, will add the kitchen in their favourite list.

This screen shows kitchens nearest to them according to their current gps location. They also have the option to manually enter their pin code.
There will be more options in the overflow menu (can make visible in app bar itself), like chat lists, and to start your own kitchen.

3. Kitchen Detail Screen



When you click on a list item in screen 2, you will see a detail view of the kitchen. Need to be fully material design, with full bleed images at the top and show all the sections in their own surfaces. Floating button lets you connect to the kitchen owner through chat.

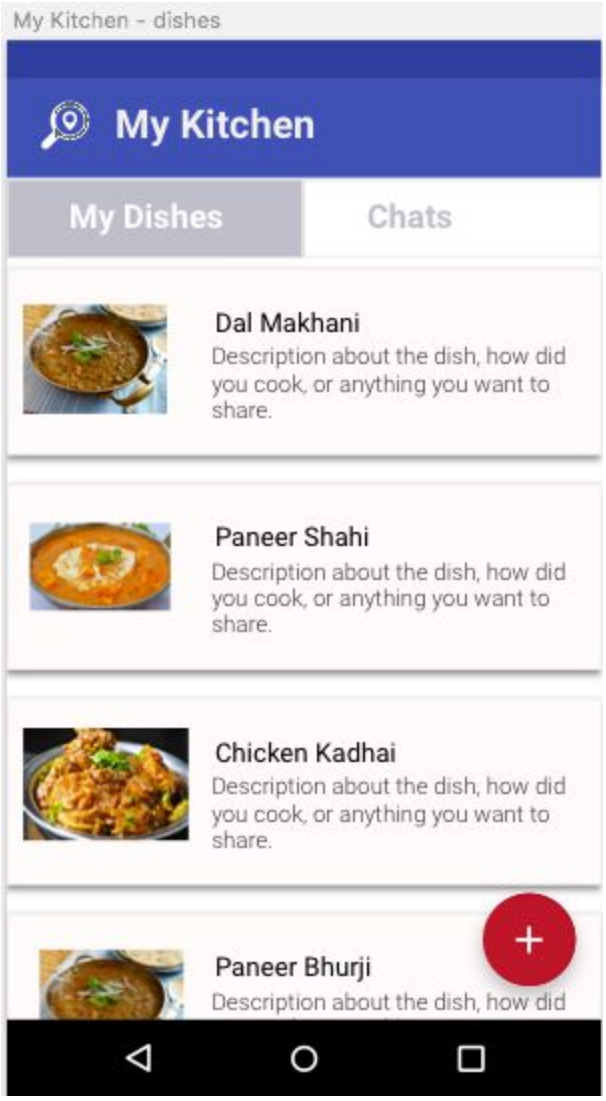
Sections:-

- Kitchen Description
- Dishes (in horizontal recycler view)
- Add a review
- User Reviews
- Address
- Location on Map

4. Setup Kitchen Screen

If a user chooses to publish his/her kitchen, this is a one-time screen to create the kitchen and add the required values. Once the user has completed this step, he can start adding dishes.

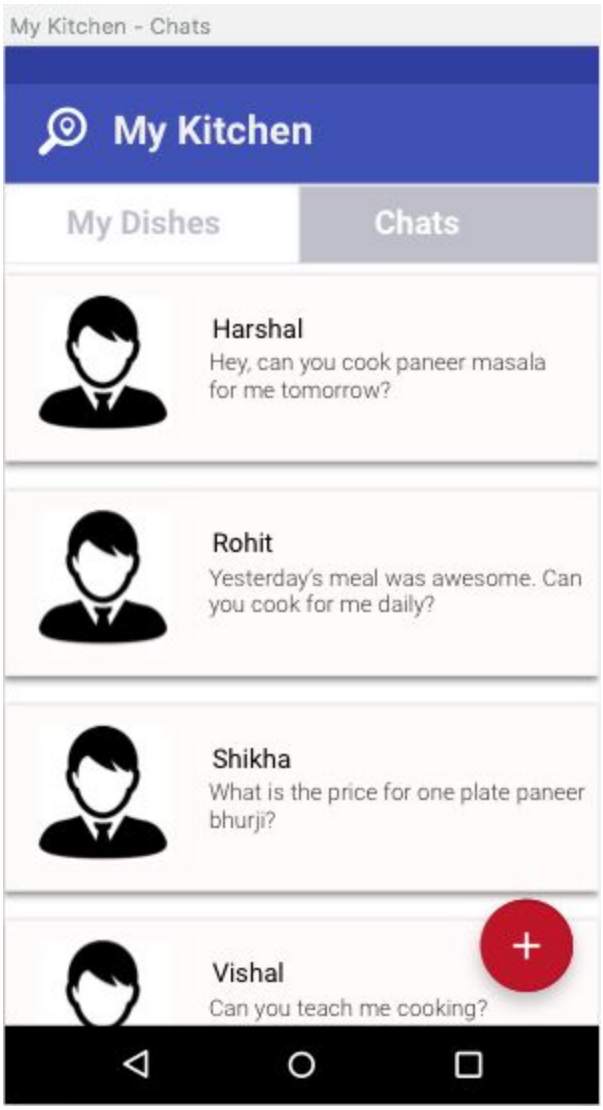
5. My Kitchen - Dishes Screen



This is the default landing screen for kitchen owners. Here they can keep adding their dishes for other users nearby to see. There will be an overflow menu to check out nearby kitchens also. Clicking on floating button will add a dish.

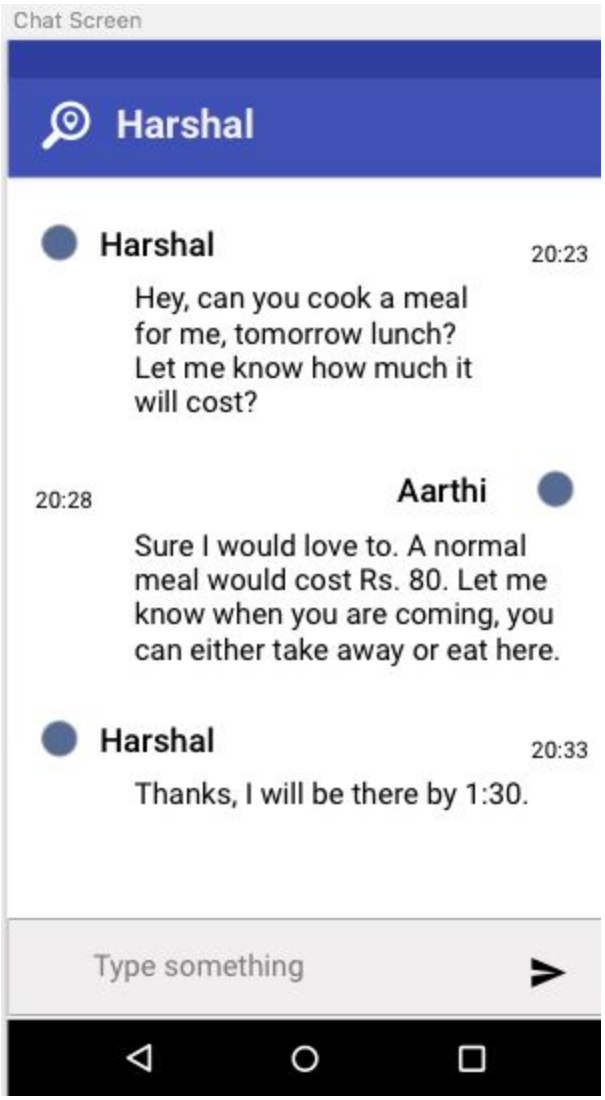
6. My Kitchen - chat list

This is the second tab in landing screen for kitchen owners. Here people who have messaged with their queries or requests to the kitchen owner will show up.



7. Chat Screen

This is the chat screen where kitchen owner and a prospective customer can talk to each other.



Key Considerations

How will your app handle data persistence?

The app will use firebase database to store kitchens data in cloud. App will store kitchens marked as favourite by the consumer locally using sqlite, and build a content provider on top of it. It will use loaders to load them in favourite kitchen's view.

Describe any corner cases in the UX.

For new customers, app will take them to nearby kitchens page. If they want to publish their own kitchen, they will have to select the start my kitchen option in the landing screen.
Once a kitchen owner has set up his.her kitchen, they will be sent to “My Kitchen Page” by default.

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

Picasso: to handle the loading and caching of images.
Firebase: for real time cloud database, managing the signup flow (firebase ui), authenticating users, storage for image of dishes, and geo-fire to get nearby kitchens.
Butterknife: to inject views which helps in less and cleaner code.

Describe how you will implement Google Play Services.

Location Services: will use location services by getting user’s accurate position coordinates using gps to find nearby kitchens. The app will use getLastLocation api as a one time call to manage power, as user’s position is not likely to change.
Map Services: to show the location of kitchen on the map in kitchen detail page

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 project for the app, structure your package by features, make a BaseActivity which extends AppCompatActivity that all your user activities will extend. Make the BaseActivity abstract, and use this to add common functionalities you need across the whole app or to get instances of helper classes. Also make it implement an AppConstants interface which will have all the constants we will be using in the app.

- Configure libraries: Add library dependencies in build.gradle with latest versions for google play services, firebase core, firebase auth, firebase db, geofire, picasso and butterknife. We may need to add few more libraries as needed. Try using wrapper classes for all library dependencies.

Task 2: Integrate ability to log in

- Let users be able to log in using email/password or using google
- Do this using firebaseui and firebase auth

Task 3: Implement UI for Each Activity and Fragment

Build UI for all the mock screens above.

Task 4: Structure data for firebase db and Integrate with cloud

- Structure data as we will need for the firebase realtime db. Try to keep the structure flat
- Integrate with firebase to be able to create and fetch database in real time. Make sure it works offline also.
- Add dummy data and test to make sure all the fetching, saving works fine

Task 5: Integrate google location services

- Use google location services to get users coordinate using fine access permission
- Save kitchens location with the kitchen owner's coordinates in firebase database using geofire
- Query firebase db with customer's coordinates using geofire to get nearby kitchens in order of distance, nearest first.

Task 6: Add other features

- Add a map view to the kitchen detail page
- Add ability for customers to rate kitchen and review them
- Add a floating button to connect/chat with kitchen owner

Task 7: Persist favourite kitchens

- Add ability for users to mark a kitchen as favourite. Persist it in local sqlite database
- Build a content provider to query database
- Use loader to load favourite kitchens when user selects favorite kitchen in kitchen list screen.