Capstone_Stage1

GitHub Username: https://github.com/thai-dang

FoodQuest

Description

FoodQuest is a mobile app that allows the users to search and discover restaurants around their current location or by specific location. This app sorts and displays restaurants by cuisines (e.g. American, French, Italian, etc.), types of establishment (e.g. Bakery, Casual Dining, Fast Food, etc.), locations (e.g. nearby, zip code, city, etc.), and favorite restaurants. For each restaurant it provides address, phone number, cuisine, rating, price range, reviews by other users, etc.

Intended User

This app would satisfy the culinary needs for the following people:

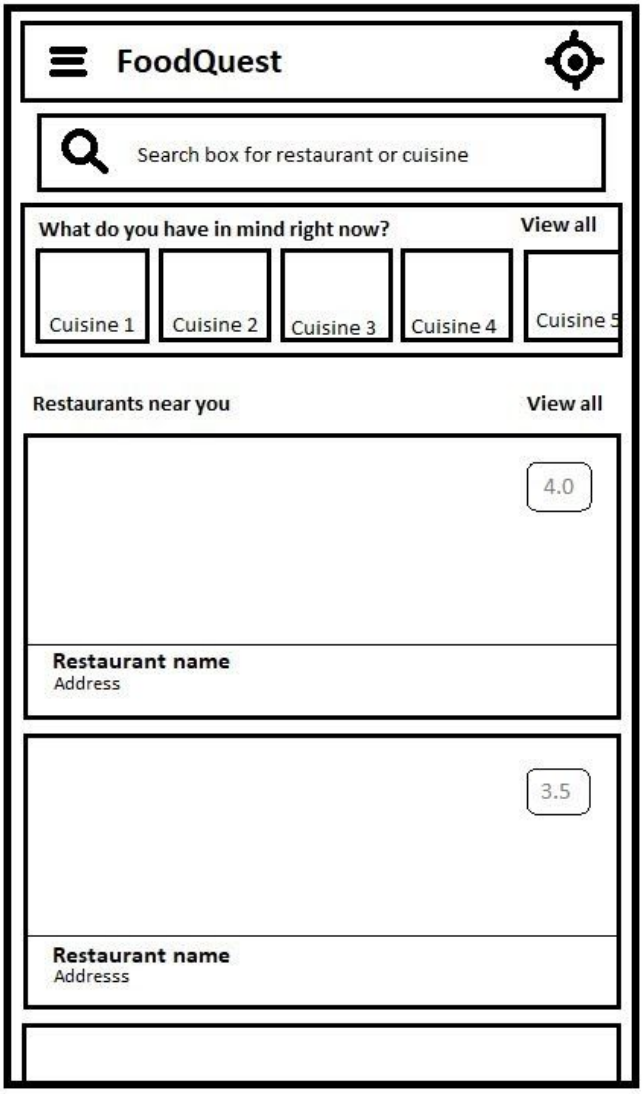
- Tourists in a strange city
- Students looking for a quick/cheap bite
- Traveling businessmen who want to impress customers
- Busy people with no time to cook
- Couples on a first (or second, or third) date
- Families on a weekly night-out
- Hungry people

Features

- Explore nearby restaurants and cuisines
- Search for restaurant by name, cuisine, location, type of establishment
- Show useful restaurant information (address, phone, price range, rating, reviews, etc.)
- Show restaurant location on map
- Show restaurant reviews shared by other users
- Add and show favorite restaurants

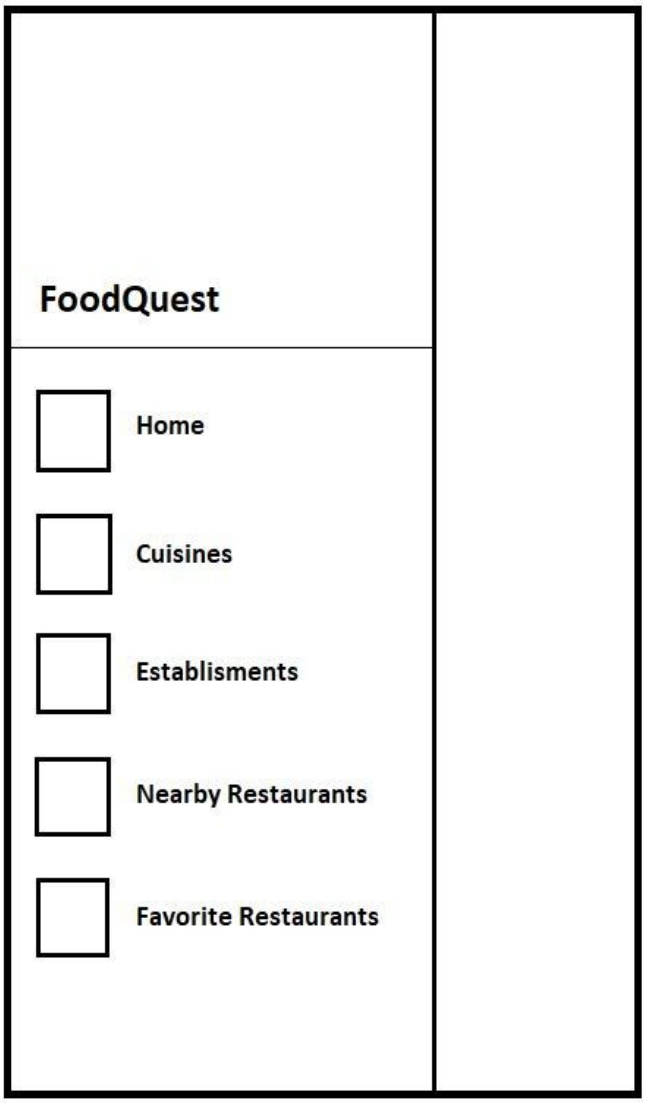
User Interface Mocks

Main (Home) Screen



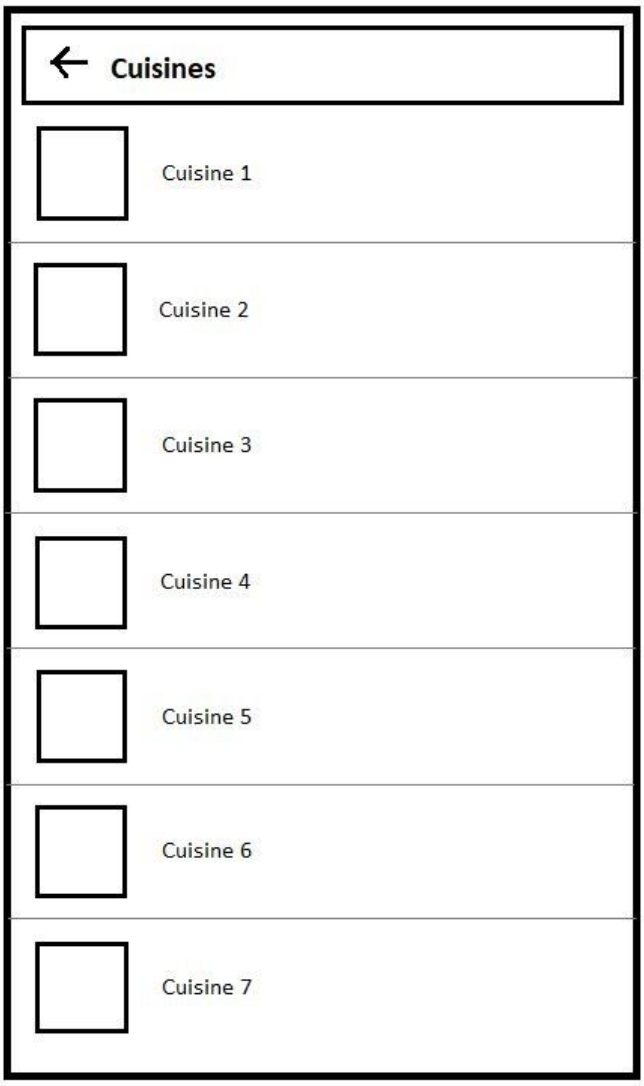
This is the main Home screen displayed when the app first started.

Main Menu Screen



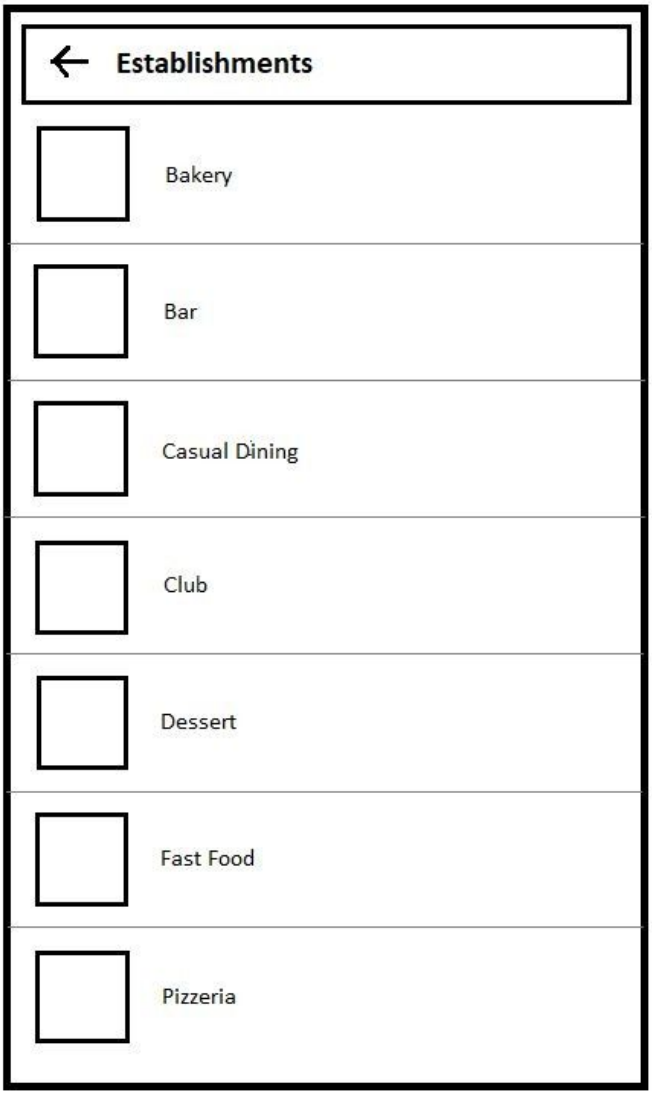
When the user clicks on the “hamburger” icon on the top left corner of the menu bar, this Main Menu popup screen will be displayed.

Cuisine List Screen



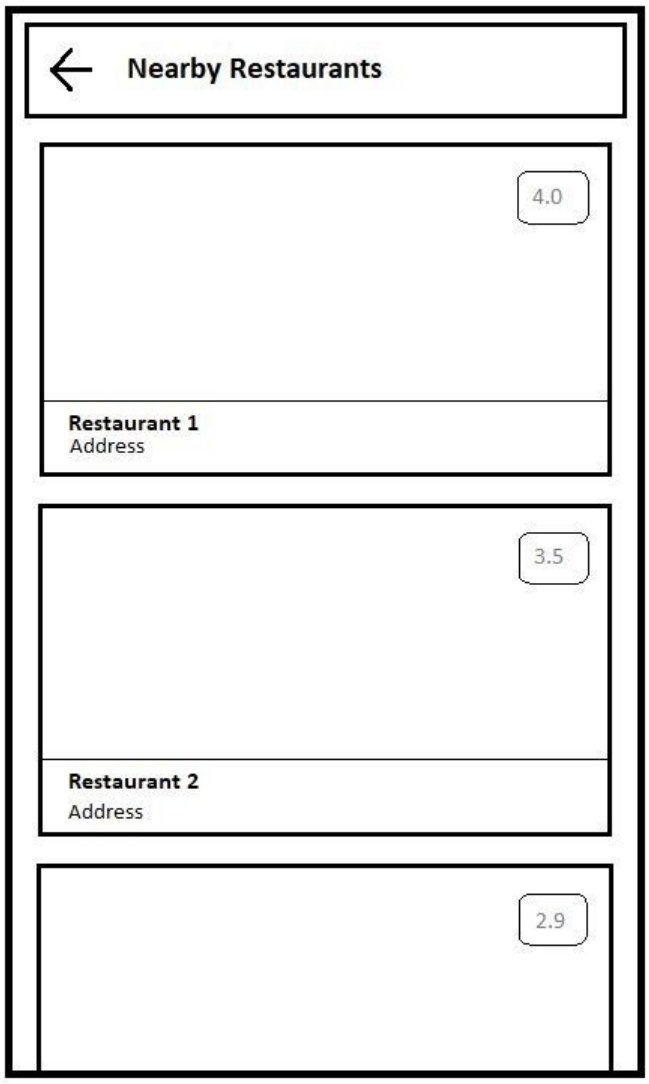
This is the screen to be displayed when the “Cuisines” button on the Main Menu is selected. This screen lists all available cuisines, alphabetically sorted.

Establishment List Screen



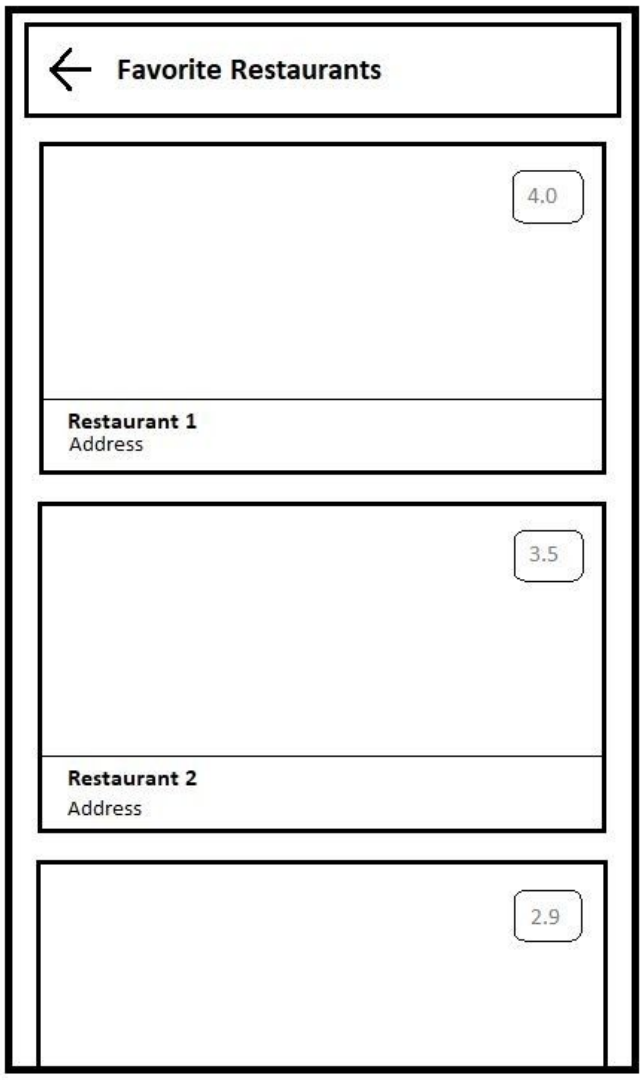
When the “Establishments” button on the Main Menu is selected, this screen will be displayed to show all available types of establishment.

Nearby Restaurant List Screen



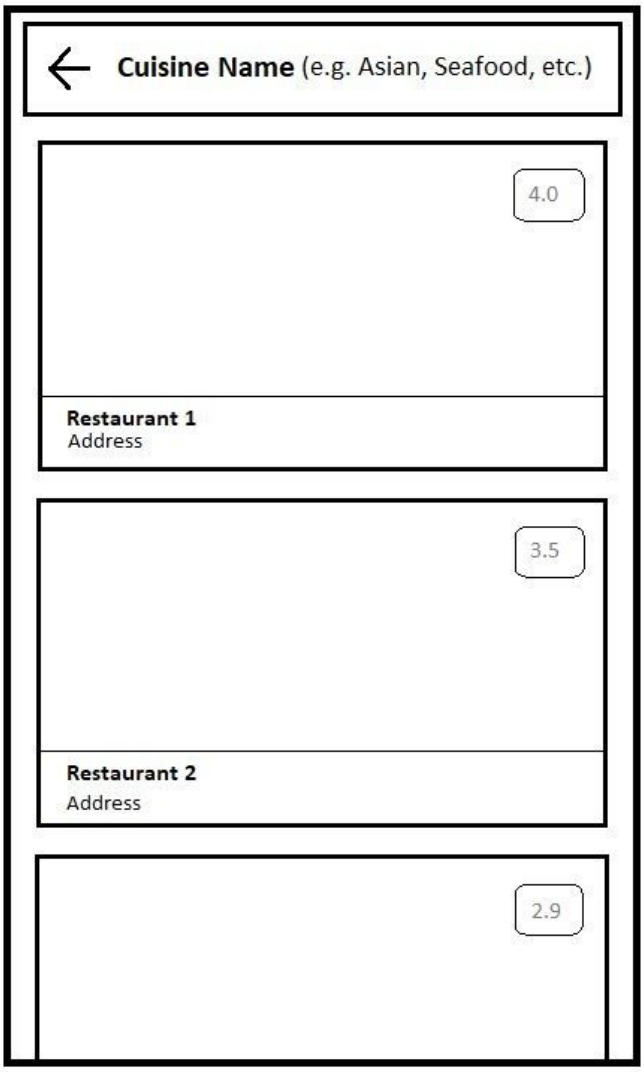
When the “Nearby Restaurants” button on the Main Menu is selected, this screen will be displayed to show all available restaurants near the current location (sorted alphabetically).

Favorite Restaurant List Screen



When the “Favorite Restaurants” button on the Main Menu is selected, this screen will be displayed to show all of the user’s favorite restaurants (sorted alphabetically).

Cuisine Restaurant List Screen



When the user selects a particular cuisine (e.g. American, French, Asian, etc.) from the Cuisine List screen, this screen will be displayed to show all available restaurants for that cuisine (sorted alphabetically).

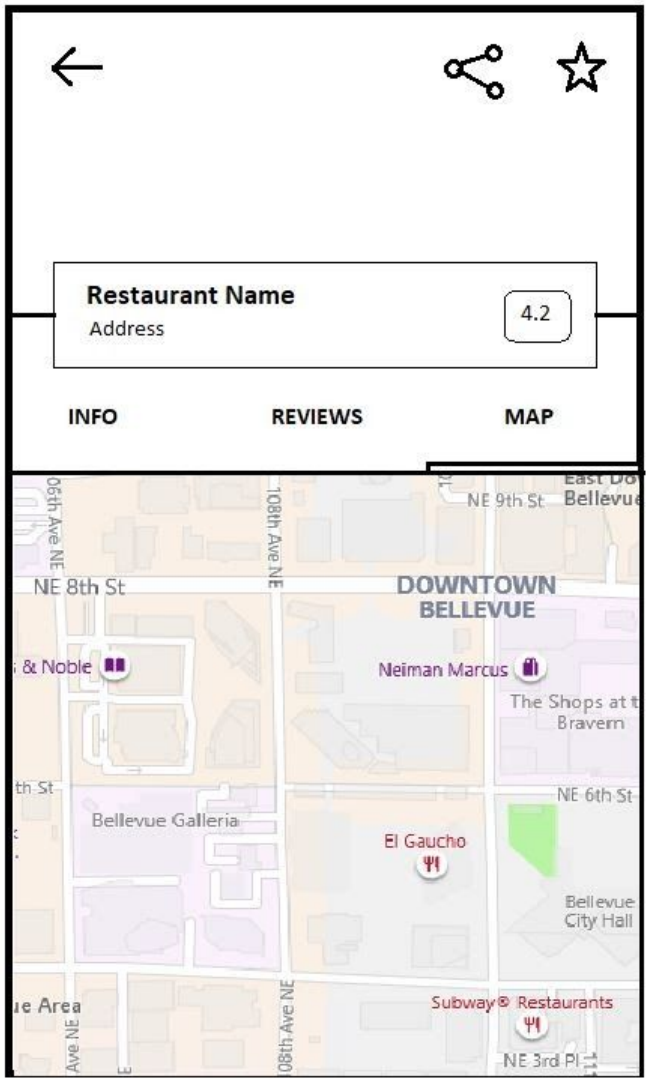
Restaurant Detail Info Screen

This is the screen that shows all the info about a restaurant (e.g. name, address, phone, rating, price range, etc.) when that restaurant was selected from a Restaurant List screen. The “Info” tab is by default the selected one.

Restaurant Detail Review Screen

When the “Reviews” tab is selected, this is the screen that will show all the user reviews for this restaurant.

Restaurant Detail Map Screen



When the “Map” tab is selected, this is the screen that will show the restaurant on a map.

Home Widget Screen

The home screen widget which contains the first couple favorite restaurants.

Key Considerations

This app would be implemented with the following requirements:

- Use Java as the sole programming language.
- Use only stable versions of libraries, Gradle and Android Studio.
- Use one or more of the following mechanisms: SyncAdapter/JobDispatcher or Async Task or Intent Service.
- All string and dimension resources will be defined in the appropriate value XML files.

How will your app handle data persistence?

This app uses Room data persistence to save favorite restaurants and uses SharedPreferences to save location. Favorite restaurants would be shown on the home screen and detail screens if internet connection is not available.

Describe any edge or corner cases in the UX.

- User can add to favorites by clicking on the Star icon on the main menu bar
- User can share restaurant address by clicking on the Share icon on the main menu bar
- When the user clicks on the Back button, the app will return to the immediate previous screen

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

- **Architecture Components (LiveData, ViewModel, Room, Paging)** for creating a robust, testable, and maintainable app
- **DataBinding** for binding UI components in layout and prevent boilerplate code
- **Dagger** for dependency injection
- **Retrofit** for a type safe HTTP client
- **OkHttp** for HTTP and HTTP/2 client
- **GSON** for JSON serialization
- **RxJava** & **RxAndroid** for composing asynchronous and event-based programming
- **Glide** for image loading and caching
- **Stetho** for network inspection
- **EasyPermission** for easy way runtime permission
- **Timber** for logging

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

- **Admob** to display test banner ads.
- **Google Maps** to display map and restaurant address on the map.

Next Steps: Required Tasks

Task 1: Project Setup

- Create a project in Android Studio
- Add project dependencies
- Obtain required API keys

Task 2: Implement Data Persistence

- Create Room Database
- Create entities for cuisines, restaurants and establishments
- Create DAO (Data Access Object) classes

Task 3: Implement Network Requests

- Create models classes
- Create Retrofit API service

Task 4: Implement Dependency Injections

- Create module classes and components

Task 5: Implement MainActivity

- Build UI for MainActivity
- Implement RecyclerView item for cuisines (show only 10 cuisine)
- Implement RecyclerView item for nearby restaurants (show only 5 nearby restaurants)
- Create ViewModel and Repository classes
- Save user location to SharedPreferences (current location or any location that user wants)

Task 6: Implement CuisineListActivity

- Build UI for CuisineListActivity
- Implement RecyclerView item for cuisine list
- Create ViewModel and Repository classes

Task 7: Implement EstablismentListActivity and Fragment

- Build UI for EstablishmentListActivity and Fragment
- Implement RecyclerView item for establishments list
- Create ViewModel and Repository classes

Task 8: Implement NearbyRestaurantActivity and Fragment

- Build UI for NearbyRestaurantListActivity and Fragment
- Implement RecyclerView item for nearby restaurant list
- Create ViewModel and Repository classes

Task 9: Implement RestaurantDetailActivity

- Build UI for RestaurantDetailActivity
- Create ViewPager for Restaurant Info, Reviews and Map
    - Build UI for Restaurant Info Fragment
        - Create ViewModel and Repository classes
    - Build UI for Restaurant Reviews
        - Implement RecyclerView items for reviews
        - Create ViewModel and Repository classes
    - Build UI for Restaurant Maps
        - Obtain Google Maps API Key
        - Implement RecyclerView items for map
        - Create ViewModel and Repository classes

Task 10: Implement App Widget

- Build UI for home widget
- Implement AppWidgetProvider class
- Create RemoteViewService to update widget

Task 11: Configure Google AdMob

- Import Mobile Ads SDK
- Add AdView to layout
- Initialize Mobile Ads in *onCreate()* method