

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Main Screen](#)

[Cuisines List Screen](#)

[Establisments List Screen](#)

[Nearby Restaurants Screen](#)

[Favorite Restaurants Screen](#)

[Restaurant Detail Info Screen](#)

[Restaurant Detail Reviews Screen](#)

[Restaurant Detail Map Screen](#)

[Home Widget Screen](#)

[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 Data Persistence](#)

[Task 3: Implement Network Requests](#)

[Task 4: Implement Dependency Injection](#)

[Task 5: Implement Main Activity](#)

[Task 6: Implement Cuisines List Activity](#)

[Task 7: Implement Establisments List Activity and Fragment](#)

[Task 8: Implement Nearby Restaurant Activity and Fragment](#)

[Task 9: Implement Restaurant Detail Activity](#)

[Task 10: Implement App Widget](#)

[Task 11: Admob Configuration](#)

GitHub Username: [emrekose26](#)

Famula

Description

Famula is an application which allows users search and discover restaurants by location around of them. It provides restaurant adress, call number, rate, other user reviews

Intended User

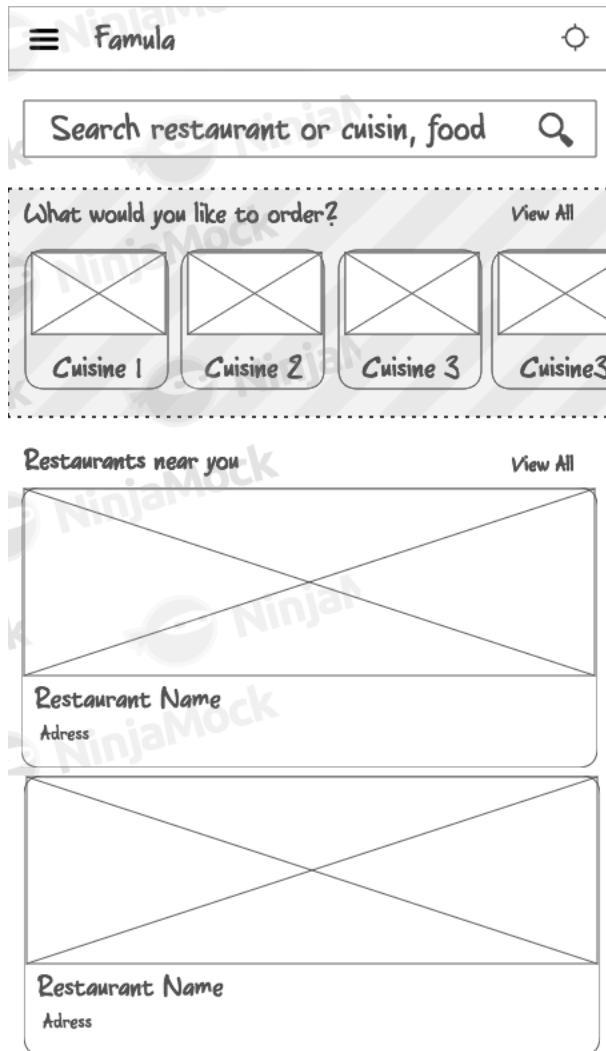
For all the users who wants to discover nearby restaurants, cuisines

Features

- Explore nearby restaurants and cuisines
- Search restaurant, cuisine, establishment by location
- Show every restaurant information(phone, adress, range, rate etc.)
- View restaurant location on map
- Show restaurant reviews that users shares
- Add to favorite restaurants

User Interface Mocks

Main Screen



Location can be selected on this screen. According to location, a limited list of cuisines and nearby restaurants can be shown. Also user can search for restaurant, cuisine, establishment and city

Cuisines List Screen

≡	Cousines	
	Cousine Type	>
	Cousine Type	>
	Cousine Type	>
	Cousine Type	>
	Cousine Type	>
	Cousine Type	>
	Cousine Type	>
	Cousine Type	>
	Cousine Type	>
	Cousine Type	>

All of the cuisine list at the location

Establisments List Screen

The mockup displays a list of establishments under the heading "Establisments". Each entry consists of a placeholder image (a rectangle with a diagonal cross), a rating in a dashed box, the restaurant name, and the address.

Rating	Restaurant Name	Adress
4.2		
4.0		
3.5		
2.8		

All of the establisment list that shows name, adress, poster and rate

Nearby Restaurants Screen

The wireframe shows a mobile application screen titled "Nearby restaurants". It contains a list of four restaurant items. Each item is represented by a rounded rectangle divided into three sections. The top section is for a poster, the middle for the restaurant name, and the bottom for the address. A rating is displayed in a dashed box in the top right corner of each item's poster section. The ratings are 4.2, 4.0, 3.5, and 2.8 for the four items respectively. The poster sections are currently empty, indicated by a large 'X' across them.

Poster	Rating	Restaurant Name	Address
[Placeholder]	4.2	[Placeholder]	[Placeholder]
[Placeholder]	4.0	[Placeholder]	[Placeholder]
[Placeholder]	3.5	[Placeholder]	[Placeholder]
[Placeholder]	2.8	[Placeholder]	[Placeholder]

All of the nearby restaurant list that shows name, adress, poster and rate

Favorite Restaurant Screen

The wireframe shows a mobile application screen titled "Favorites". It contains a list of four restaurant entries. Each entry consists of a placeholder image (a rectangle with a diagonal cross), a rating in a dashed box, the restaurant name, and the address. The ratings are 4.2, 4.0, 3.5, and 2.8 respectively. The text "NinjaMock" is visible as a watermark across the screen.

Image Placeholder	Rating	Restaurant Name	Address
	4.2	Restaurant Name	Address
	4.0	Restaurant Name	Address
	3.5	Restaurant Name	Address
	2.8	Restaurant Name	Address

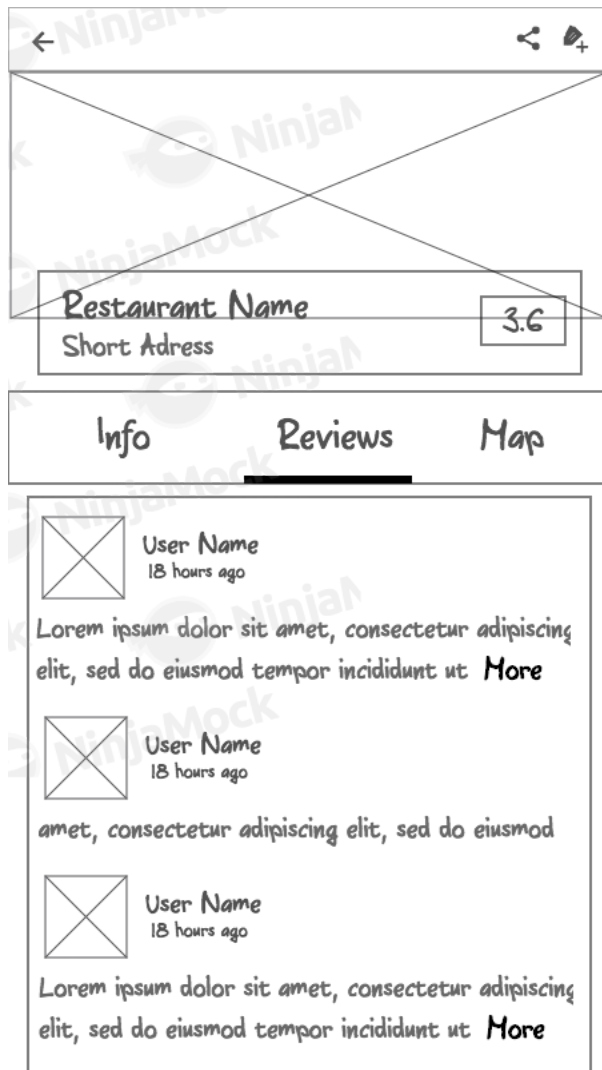
Favorite restaurant list that shows name, adress, poster and rate

Restaurant Detail Info Screen



This screen shows restaurant name, short address and rate on the top of the viewpager. Also user can add to favorites restaurants and share address. Restaurant address, call number, price range etc shows in to info tab

Restaurant Detail Reviews Screen



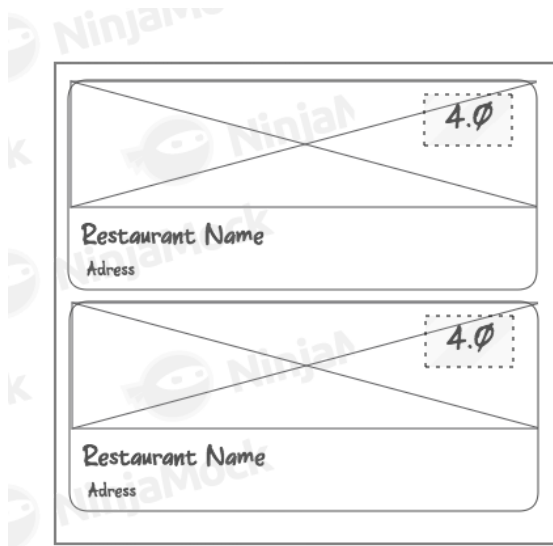
This screen shows other user reviews about this restaurant in to reviews tab

Restaurant Detail Map Screen



This screen shows restaurant address on map in to map tab

Home Widget Screen



Home screen widget that contains favorite restaurants

Key Considerations

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 shows on the home screen widget and detail screen if connection does not available.

Describe any edge or corner cases in the UX.

This app will be using Material design principles.

NavigationDrawer is using for quick and easy navigation of app's features

When user click the back button app will return the proper list activity

When user click the app widget it will open favorite restaurant activity

Search filter and get the user location features using BottomSheet

User can add to favorites with click the bookmark icon

User can share to restaurant address with click the share icon

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

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

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

Admob : This app will be display test banner ads.

Google Maps: This app uses Google Play Services for Google Maps API that display restaurants adress on map.

Next Steps: Required Tasks

Task 1: Project Setup

- Creation of the project
- Adding the project dependencies
- Obtain API key from [Zomato](#)

Task 2: Implement Data Persistence

- Create Room Database
- Create entities for cuisines, restaurants and establishments
- Crete dao 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 Cuisines List Activity

- Build UI for Cuisines List Activity
- Implement RecyclerView item for cuisines list
- Create ViewModel and Repository classes

Task 7: Implement Establishments List Activity and Fragment

- Build UI for Cuisines List Activity and Fragment
- Implement RecyclerView item for establishments list
- Create ViewModel and Repository classes

Task 8: Implement Nearby Restaurants Activity and Fragment

- Build UI for Cuisines List Activity and Fragment
- Implement RecyclerView item for establishments list
- Create ViewModel and Repository classes

Task 9: Implement Restaurant Detail Activity

- Build UI for Restaurant Detail Activity
- 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

Task 10: Implement App Widget

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

Task 11: Admob Configuration

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