# MILESTONE 1 READ.ME

| | |
|---|---|
| **Team Name** | NullPointerException |
| **Team ID** | 5193 |
| **Project Level** | Apollo 11 |
| **Team Members** | Aarnav Srinivasan<br>Vaidyanaath Suresh |

## Motivation

Ever been confused about what to eat and where to eat inside NUS? You might be craving something or you might simply want to have a decent meal without breaking the bank. There are not many people you can ask to get to know everything about food at NUS. And it is practically impossible to visit each and every food stall/restaurant and gather the required information. Is there a way to get all this information easily?

## Aim

We hope to provide real time menu data and availability status of food from all food-courts/ stalls in NUS, with additional features and information, through **our mobile application** Eat@NUS.

## User Stories:

1. As a customer who wants to explore food options at NUS, I want to know the locations of all the food stalls/restaurants and all the available items in the menu in any stall.

2. As a customer, I want to be able to search for any food stall or dish, and filter them based on cuisine, price range, etc.

3. As a customer who has decided to buy a certain dish, I want to be notified about the availability of the dish in real time.

4. As a customer looking to try any dish, I should have the information on the ingredients and any allergens present in the dish and the calorie count, so as to make an informed decision.

5. As a customer who wants to provide feedback on food I have had, I want to be able to write a review about the quality of food for other customers to read.

6. As a food stall owner who has recently opened a stall, I should be able to create a new profile and update the display information.

7. As a food stall owner, I want to be able to update the opening and closing times of the stall so that it is reflected on the customer side of the app.

8. As a food stall owner who wishes to change menu items, I want to be able to edit, refresh, add food items and update their price and availability.

## Scope of Project:

A **mobile application** that provides real-time menu data and availability of food across all food stalls inside NUS. The app has several features for both customers and food stall owners.

### Customer features:

1. The **Search and Filter** feature enables customers to search for food stalls by name or search for food stalls that serve a particular dish. The customer can also filter the results by Cuisine, Price Range and Distance.

2. The **View Stall Info** feature displays information such as location, operating hours, menu items and rating of a particular food stall.

3. The **View Dish Info** feature displays the real-time availability, price, allergen information, calorie count and rating of a particular dish.

4. The **Write a Review** feature allows the customers to write reviews (and edit them) about the dishes.

**Stall Owner features:**

1. The **Edit Stall Info** feature enables stall owners to edit the general information about their stalls, like location, operating hours.

2. The **Edit Menu Items** feature enables stall owners to add/delete items to/from the menu.

3. The **Edit Dish Info** feature enables stall owners to edit the information of a particular dish, like real-time availability, price, allergen information and calorie count.

**Additional features:**

1. (Customer Side) A **Bookmark** feature that enables customers to bookmark dishes for easier access.

2. (Customer Side) A **Notification** feature that notifies the customer when a marked stall is closed or a marked dish becomes unavailable.

**Additional Non-Functional Requirements:**

1. **Latency:**

   Users will not like to wait to see the search results of food stall details and dishes, so the search functionality should be very fast.
2. **Availability:**

   High availability is desirable for the best customer experience.

**Tech Stack:**

1. React Native
2. Firebase
3. Redux

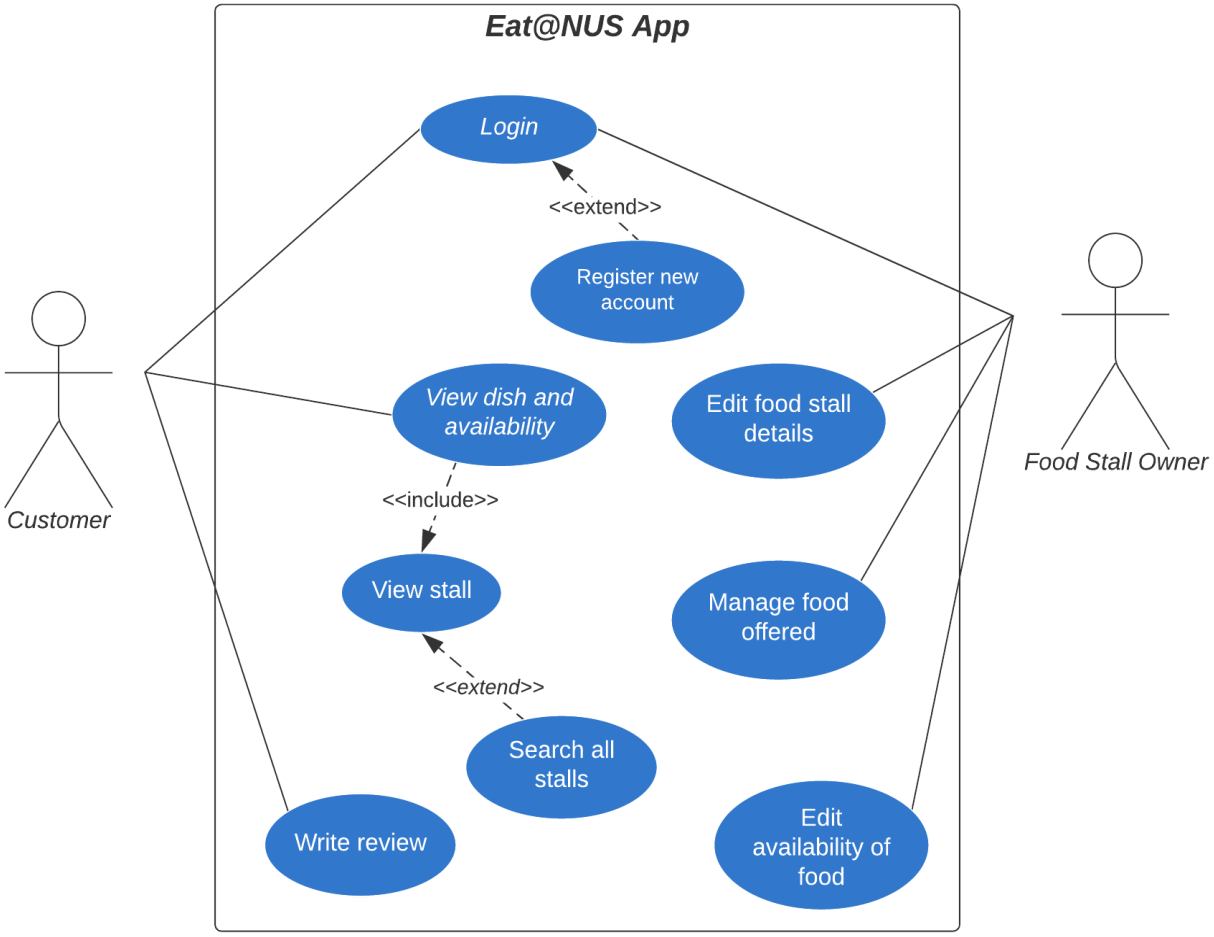
**Timeline:**

| Features to be completed by the mid of June: | Features to be completed by the mid of July: |
|---|---|
| 1. View Stall Info<br>2. View Dish Info<br>3. Edit Stall Info<br>4. Edit Dish Info<br>5. Edit Menu Items | 1. Search and Filter<br>2. Write a Review<br>3. Bookmark feature<br>4. Notification feature |

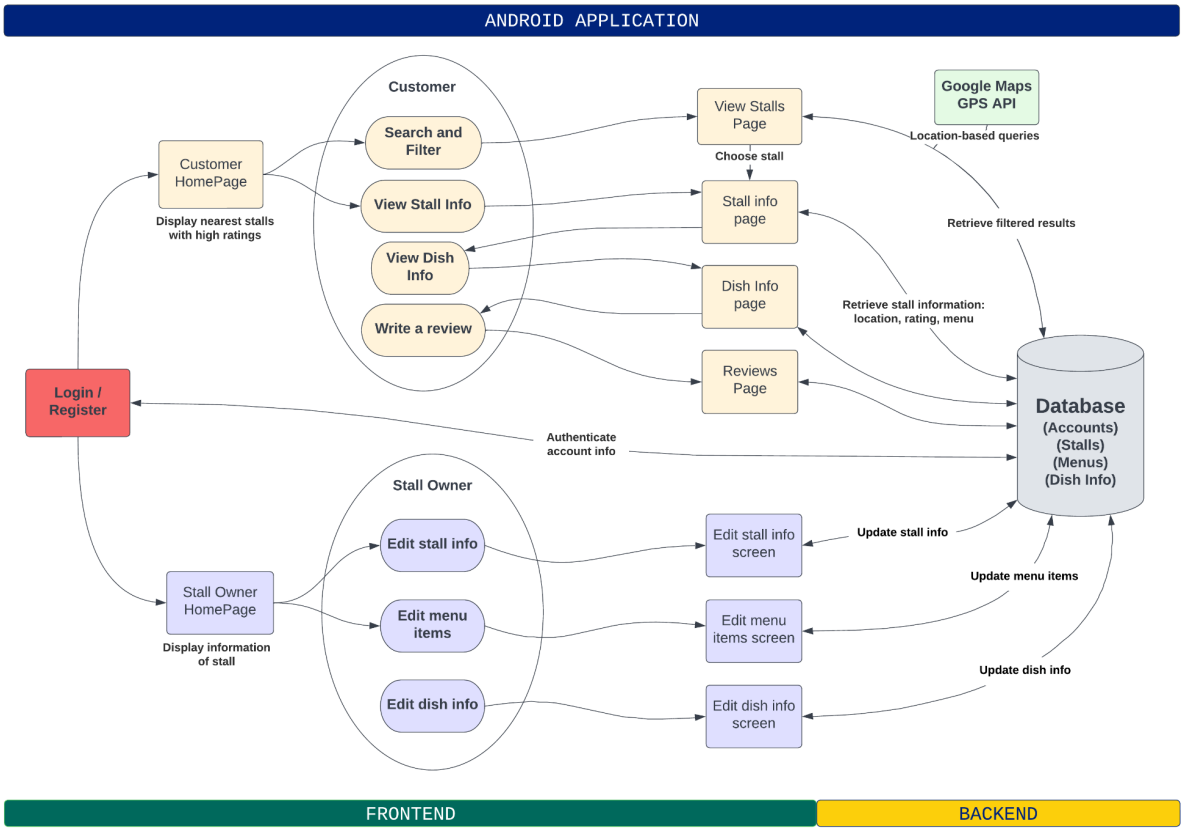
## Design:

### Use Case diagrams:

Based on the user stories, we created a use-case diagram below that illustrates how the users of the Eat@NUS app might interact with the system.



## Technical Proof of Concept:

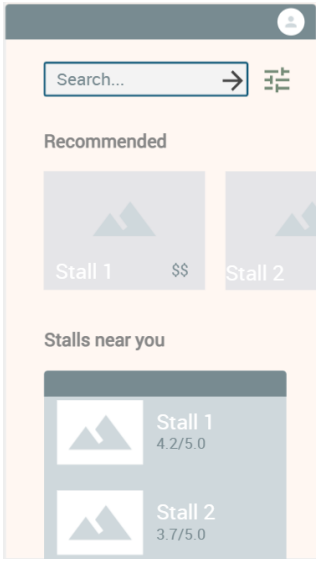



## Explanation:

Eat@NUS is a mobile application built using React Native. The app has different features and interfaces, for customers and food stall owners.

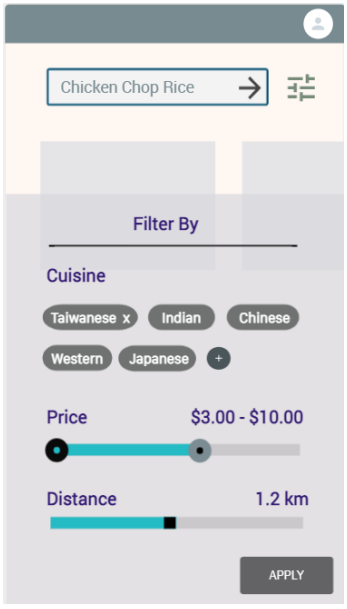
**Customer-side features and interfaces:**

The **Customer Homepage** displays a list of recommended stalls (stalls with the highest ratings, pulled from the database, and a list of the stalls closest to the user, obtained using data from the Google Maps API.




- A horizontally scrollable display of the recommended stalls.
- A vertically scrollable display of the stalls nearest to the user.
- A textbox which when clicked upon, triggers the Search and Filter feature.

The **Search and Filter** feature allows the user to search for a stall by its name or the list of stalls that serve a particular dish. The user can also filter the search results by Cuisine, Price Range and Distance.

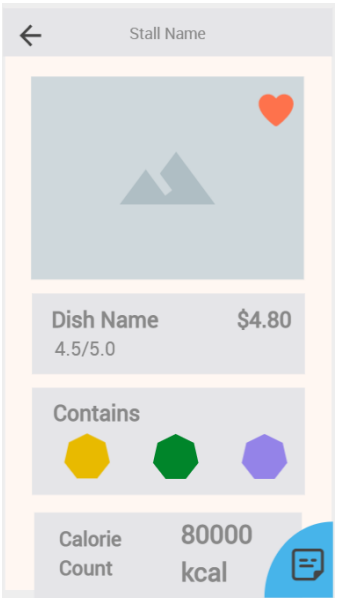



- A textbox to allow user input.
- A button which when pressed, opens a small window to apply filters to the search. The filter window contains tags for Cuisine and two sliders, one each for Price Range and Distance.
- A button, which when pressed, takes the user input and the selected filters to obtain desired results using Firebase.
- A vertically scrollable display of the list of desired search results.

The **View Stall Info** feature allows the customer to view information about a stall, like the location, operating hours, customer rating and menu items.


- Textview to display the name of the stall.
- Display of pictures of the stall.
- Textviews to display the general information of the stall.
- Textviews to display names and prices of dishes in the menu, which when clicked upon, triggers the View Dish Info feature

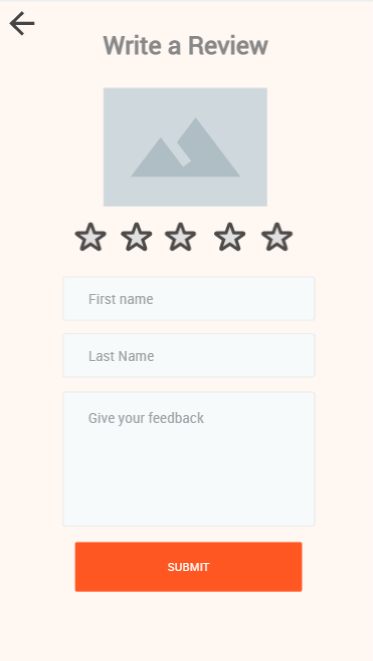
The **View Dish Info** feature allows the user to view information about a particular dish served in a particular stall. The user can view a picture of the dish, the real-time availability of the dish, and in addition to it, get to know of any possible allergens present in that dish and the approximate calorie count. The user can also view the rating and read reviews of the dish written by others.





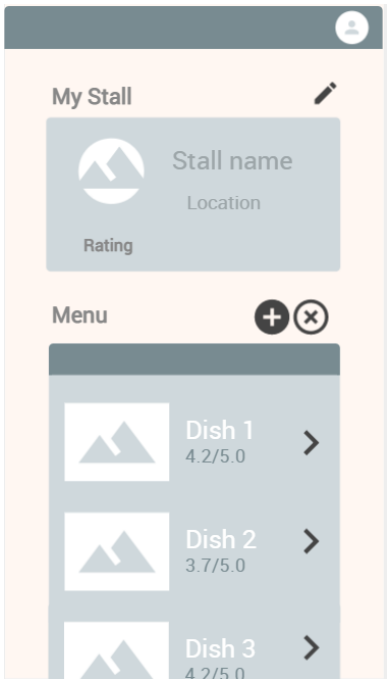
- Textview to display the name of the stall.
- Display of a picture of the dish.
- Textviews to display name, price, customer rating, allergen information and calorie count of a dish.
- Textviews to display reviews written by other customers.
- A button which triggers the Write a Review feature.

The **Write a Review** feature allows the user to share his/her thoughts about the dish.

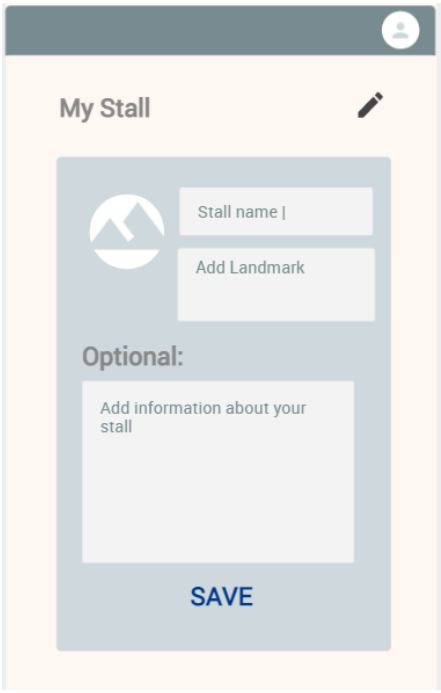
| | |
|---|---|
|  | • Textbox to allow user input.<br>• A button which when pressed, stores the user input to the database.<br>• A button which enables editing of the user's previously inputted text. |

**<u>Stall Owner-side features and interfaces:</u>**

The **Stall Owner Homepage** displays all the information about the food stall associated with the account used to login, pulled from the database. The displayed content includes general information about the stall (like location, customer rating, operating hours, etc.), and the menu.

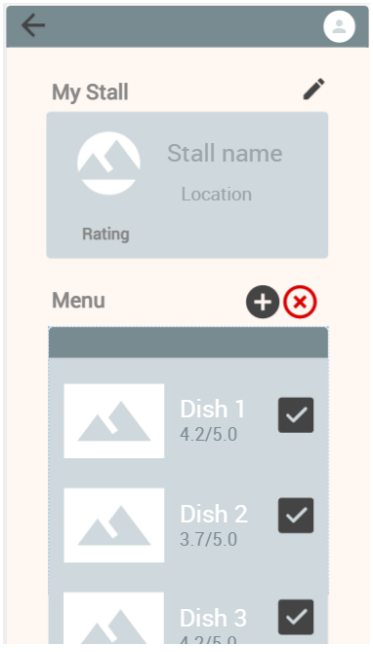
| | |
|---|---|
|  | • Textviews to display stall name, customer rating of the stall, stall location, operating hours.<br>• A button to trigger the Edit Stall Info feature.<br>• Display of pictures of the stall.<br>• Textviews to display the dishes in the menu of the stall, which when tapped, triggers the Edit Dish Info feature.<br>• A button which triggers the Add Items part of the Edit Menu Items feature.<br>• A button which triggers the Delete Items part of the Edit Menu Items feature. |

The **Edit Stall Info** feature enables the stall owner to edit the general information of the stall. Any changes made will be reflected in the Customer-side interface of the app.

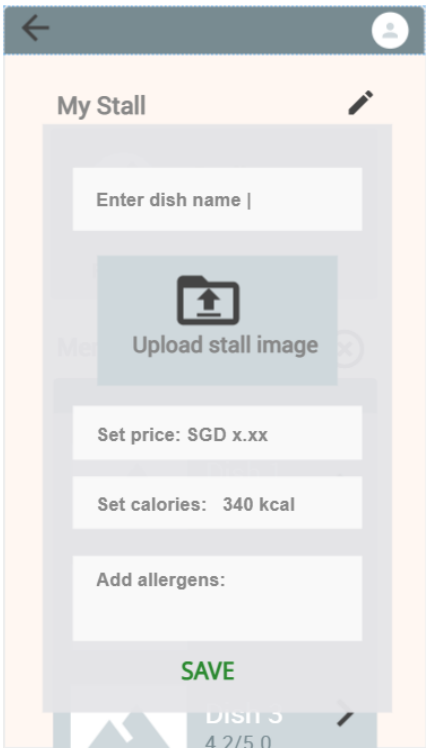
| | |
|---|---|
|  | • Textboxes with the pre-existing data, to allow user input.<br>• A button, which when pressed, saves the changes to the database, through Firebase. |

The **Edit Menu Items** feature enables the stall owner to add/delete items to.from the menu. Any changes made will be reflected in the Customer-side interface of the app.

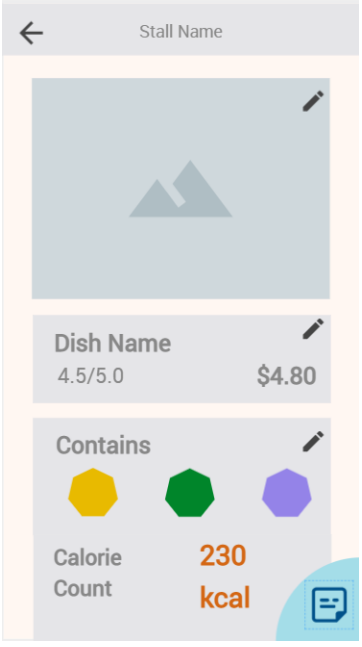
- ***Delete Items***

| | |
|---|---|
|  | • Checkboxes beside each textview displaying dish names, to select the dishes to be deleted from the stall menu.<br>• A button which when pressed, removes the textviews and erases the data from the database, through Firebase. |

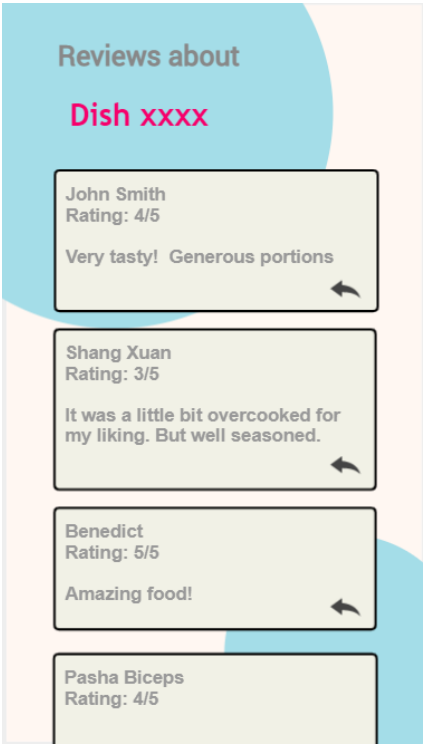
- ***Add Items***

| | |
|---|---|
|  | • Textboxes to allow user input for dish name, price, allergen info and calorie count.<br>• Picture input for the stall owner to add a picture of the dish.<br>• A button which when pressed, saves the data to the database, through Firebase. |

The **Edit Dish Info** feature enables the stall owner to edit the information about a particular dish, such as the real-time availability, allergen information and calorie count. Any changes made will be reflected in the Customer-side interface of the app.

| | |
|---|---|
|  | • Textboxes with the pre-existing data, to allow user input.<br>• A button which when pressed, saves the changes to the database, through Firebase. |

The stall owner will also be able to read the reviews for the dishes written by customers in the View Dish page.

| | |
|---|---|
|  | • Textviews to display customer reviews, pulled from the database, through Firebase. |