**Software Requirements Specification**

**for**

LetsMakanTogether

**Version 1.0 approved**

**Prepared by**

**Oh Jing Hong, Daryl**

**Kevin Tan Yong How**

**Phoon Zhen Xuan**

**Poh Jia Qi**

**Raphael Liew Jin Cheng**

**Shavonne Bay Zhirou**

**Team Stack Overflow Survivors**

**19 November 2023**

**Table of Contents**

[**1. Introduction 4**](#_1fob9te)

[**1.1 Purpose 4**](#_3znysh7)

[**1.2 Document Conventions 4**](#)

[**1.3 Intended Audience and Reading Suggestions 4**](#)

[**1.4 Product Scope 6**](#)

[**1.5 References 7**](#)

[**2. Overall Description 8**](#)

[**2.1 Product Perspective 8**](#_2s8eyo1)

[**2.2 Product Functions 8**](#)

[**2.3 User Classes and Characteristics 9**](#)

[**2.4 Operating Environment 9**](#)

[**2.5 Design and Implementation Constraints 12**](#)

[**2.6 User Documentation 14**](#)

[**2.7 Assumptions and Dependencies 14**](#)

[**3. External Interface Requirements 15**](#)

[**3.1 User Interfaces 15**](#_2jxsxqh)

[**3.2 Hardware Interfaces 30**](#)

[**3.3 Software Interfaces 30**](#)

[**3.4 Communications Interfaces 31**](#)

[**4. System Features 32**](#)

[**4.1 Landing Page 32**](#)

[**4.2 Account Registration / Registration Page 32**](#)

[**4.3 Login Page 35**](#_ol29w3bgxznr)

[**4.4 Home Page 37**](#_4bi5sal67g39)

[**4.5 Profile Page 37**](#_yde4t5io0scl)

[**4.6 Group Profile Page 39**](#_kml38554oiau)

[**4.7 Group Chat Page 40**](#_nu6uujg7vr71)

[**4.8 View Meeting Page 41**](#_b3wg0rs3irrp)

[**4.9 Group Finding Page 43**](#_bzhrgekzujmu)

[**4.10 Food Location Page 44**](#_tzkitu4w3jlb)

[**5. Other Nonfunctional Requirements 45**](#_3whwml4)

[**5.1 Usability 45**](#_2bn6wsx)

[**5.2 Reliability 46**](#)

[**5.3 Performance 46**](#)

[**5.4 Supportability 46**](#)

[**5.5 Security 46**](#)

[**6. Testing 48**](#)

[**6.1 Black Box Testing 48**](#_2cz11v73yn6y)

[**6.2 White Box Testing 86**](#_3l1qfru7smsi)

[**Appendix A: Data Dictionary 87**](#_ll34x3ezgrja)

[**Appendix B: Analysis Models 89**](#_i6g2k55h8f99)

[**Appendix C: Design Pattern Writeup 89**](#_kb1227ut4xdj)

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| Phoon Zhen Xuan | 19/11/23 | Initial Document | 1.0 |
|  |  |  |  |

# **Introduction**

## **Purpose**

This Software Requirement Specification (SRS) document is intended for the LetsMakanTogether web application, build version 1.0. The purpose of this SRS document is to describe the requirement specifications for the LetsMakanTogether web application to facilitate the development and production process for all stakeholders. All aspects of the web application, including but not limited to, the system features, the limitations, the non-functional, and interface requirements, are documented within this SRS document. The LetsMakanTogether web application aims to let users search for people with similar food interests and preferences, and arrange for a meeting to enjoy food together.

## **Document Conventions**

This section describes the conventional standards used throughout this document.

|  | Description |
| --- | --- |
| Heading (Font, Face, Size) | Times New Roman, Bold, 18 |
| Subheading (Font, Face, Size) | Times New Roman, Bold, 14 |
| Content (Font, Face, Size) | Times New Roman, Normal, 12 |
| Align & Indent, Line Spacing | Justify, 1.5 |
| Technical Standards | IEEE 830-1998 |

Refer *Appendix A: Data Dictionary* for the definitions of special terms used throughout this documentation.

## **Intended Audience and Reading Suggestions**

This document is intended for all LetsMakanTogether’s stakeholders, which includes the users, the development team, the testing team, the project manager, and the marketing team.

The following is a summary of each section of the document:

| Part 1 (Introduction) | This section provides an overview of the LetsMakanTogether project, outlining its goals, objectives, product scope, and general system details. |
| --- | --- |
| Part 2 (Overall Description) | This section provides the main functions, user classes, operating environment, design and implementation constraints of the LetsMakanTogether web application. |
| Part 3 (External Interface Requirements) | This section covers the user interfaces, hardware interfaces, software interfaces, and communication interfaces of the LetsMakanTogether web application. |
| Part 4 (System Features) | This section covers the key system features and functional requirements of the LetsMakanTogether web application. |
| Part 5 (Other Non-Functional Requirements) | This section covers the non-functional requirements, such as the usability, reliability, performance, supportability, and security requirements of the LetsMakanTogether web application. |
| Part 6 (Testing) | Black Box Testing & White Box Testing |
| Part 7 (Appendix) | Appendix A: Glossary (Data Dictionary)  Appendix B: Analysis Models & Others   1. Use Case Diagram 2. Use Case Description 3. Sequence Diagram 4. Class Diagram 5. State Machine Diagram / Initial Dialog Map 6. Key Boundary and Control Classes   Appendix C: Design Pattern Writeup |

All stakeholders are advised to begin by reading Part 1: Introduction and Appendix A: Data Dictionary to be familiarized with the purpose of the web application, as well as the documentation standards and technical terms definition used throughout this document.

The LetsMakanTogether development team is strongly encourage to proceed with Part 2: Overall Description, to have a high-level understanding of the application functionalities, design and implementation constraints, Part 4: System Features, to gain a low-level understanding of each system features to be included in the application, Part 3: External Interface Requirements, and finally Part 5: Other Non-Functional Requirements, to understand the requirements specified for the application to function as desired.

On the other hand, the users of the LetsMakanTogether web application, the LetsMakanTogether testing team, the project managers and marketing team are encouraged to proceed reading this document in sequential order.

## **Product Scope**

**Description:**

LetsMakanTogether is a social networking platform designed to connect individuals with similar food interests and preferences, facilitating meet-ups, and shared dining experiences. The name “LetsMakanTogether” is a fusion of “Let’s eat together” and “Makan”, which means “to eat” in Malay language. The web application is dedicated to foster real-life connections through the love of food, and to promote commensalism.

**Purpose:**

The primary purpose of LetsMakanTogether is to enable users to discover like-minded individuals, who share their passion for food and dining experiences. It offers a platform where people can connect, and organize food-related events. By bridging the gap between virtual connections and in-person interactions, LetsMakanTogether aims to enhance the joy of dining and build meaningful relationships around a shared interest.

**Benefits:**

1. Connectivity: Users can find and connect with people who share their food preferences, leading to enriching social interactions and potential friendships.
2. Local Insights: Gain insights into the local food scene, discovering hidden gems and off-the beaten-path eateries.
3. Networking: Expand social networks with people who have a shared passion for food, leading to more opportunities for dining and social events.

**Objectives / Goals:**

1. User Engagement: Encourage users to actively participate in the platform, creating profiles, and sharing their food preferences.
2. Connection Building: Facilitate meaningful connections among users based on common food interests and preferences.
3. Event Organization: Empower users to plan and host food-related meet-ups and events.
4. User Base Growth: Attract and retain a diverse user base, fostering a vibrant community of food enthusiasts.

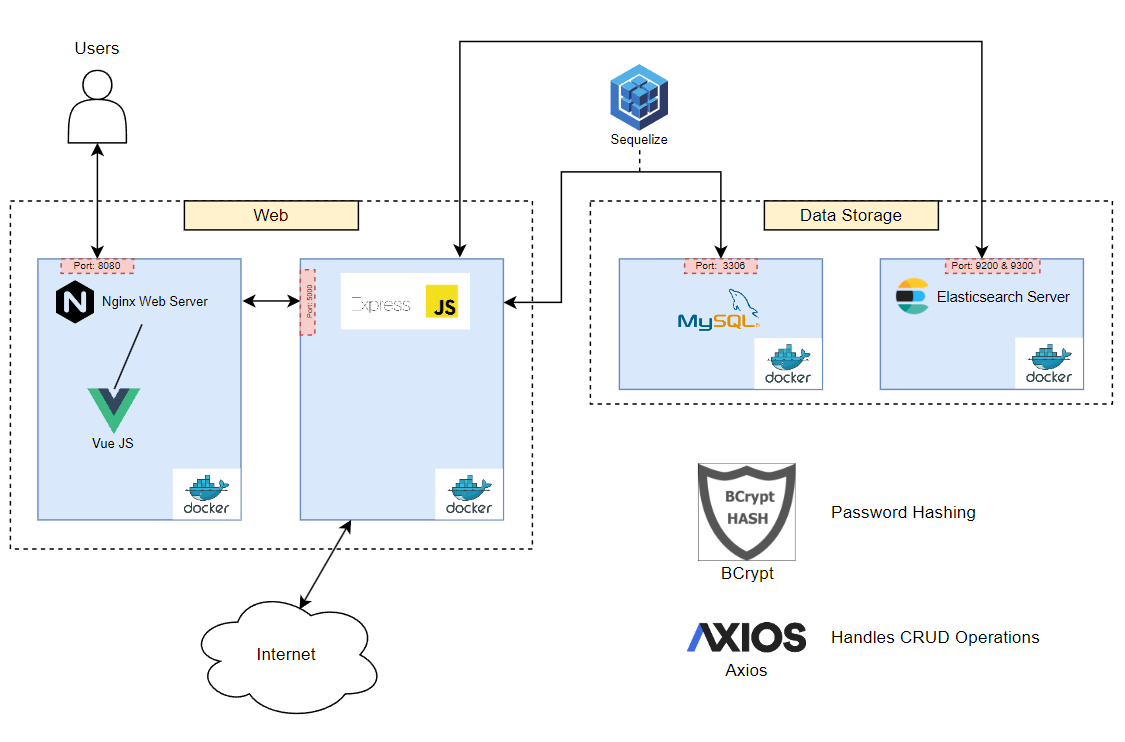
## **References**

1. 830-1998 - IEEE recommended practice for software requirements ... (n.d.). https://ieeexplore.ieee.org/document/720574
2. *Documentation: Node.js*. Documentation | Node.js. (n.d.). https://nodejs.org/en/docs
3. *Home*. Docker Documentation. (2023, August 22). https://docs.docker.com/
4. *The progressivejavascript framework*. Vue.js - The Progressive JavaScript Framework | Vue.js. (n.d.). https://vuejs.org/
5. *Sequence diagrams*. IBM. (n.d.). https://www.ibm.com/docs/en/rsm/7.5.0?topic=uml-sequence-diagrams

# **Overall Description**

## **Product Perspective**

The LetsMakanTogether web application is a new, standalone, and self-contained web application. An overall system diagram depicting the operation of LetsMakanTogether web application is as follows:

**

***Figure 2.1.1: System Architecture Diagram***

## **Product Functions**

The LetsMakanTogether web application’s system features can be broken down into three main sub-categories – Accounts, Groups, and Find Food.

**2.2.1 Accounts**

1. Account Registration
2. Account Login
3. Create and edit user’s profile page

**2.2.2 Groups**

1. Search for groups based on food and regional preferences
2. Create and edit group’s profile page
3. View and edit group’s meeting details

**2.2.3 Find Food**

1. Get user’s current location
2. Search for food locations
3. Display recommended food options

## **User Classes and Characteristics**

**2.3.1 Users looking for fellow food lovers to dine together**

| **Attributes** | **Description** |
| --- | --- |
| Frequency of Use | High |
| Subset of Product Functions Used | All |
| Technical Expertise | Low |
| Characteristics | The LetsMakanTogether marketing team predicts that the user of this demographic will use the web application to find people with similar food preferences easily by searching for groups to join based on filtered food and regional preferences. |

## **Operating Environment**

This section breaks down the operating environment into two sub-categories:

1. Production Environment
2. Development Environment

### 2.4.1 Production Environment of LetsMakanTogether

This subsection describes the settings which the web application is using.

| **Setting** | **Description** |
| --- | --- |
| HTML Support | The web application requires a browser which supports at least HTML5 or above to run without issues. Additionally, the application can be deployed within a Docker environment and accessed through a designated domain name (www.letsmakantogether.com). |
| CSS Support | The web application requires a browser which supports at least CSS3 to run without displaying any missing elements on the pages |
| JavaScript Support | The web application runs mostly on JavaScript and therefore would need a browser which supports JavaScript. |

### 2.4.2 Development Environment of LetsMakanTogether

This subsection describes the setting of which the web application is built and tested during the development phase.

| **Setting** | **Description** |
| --- | --- |
| Front-end development using VueJS | VueJS is an open-source JavaScript framework used to build the user interface (UI) of LetsMakanTogether. The development team uses VueJS to build all of its UI of the web application. VueJS allows the team to define structure and behavior of the application’s components and views using JavaScript. The added benefit of using VueJS is that it encourages separation of concerns. The development team is able to define the functionality of the JavaScript components as well as the styling and appearance of those components using CSS which keeps the project’s codebase organized. |
| Back-end Development using NodeExpress & Axios | NodeExpress and Axios are both open source frameworks used for back-end development. Both of these frameworks are used to handle CRUD operations by user as well as the routing logic of the system. (i.e from one page to another). Axios follows the Model, View, Controller architectural pattern. |
| Database using MySQL | MySQL is an open-source SQL data management system. The LetsMakanTogether team also deploys the web application’s database on Docker which hosts the database server. |
| Application Hosting using Microsoft Azure | Microsoft Azure is a cloud service provider that is maintained by Microsoft. The development team opts for Microsoft Azure over alternative cloud platforms because of its seamless integration with Visual Studio which helps to accelerate the web application’s deployment procedure. |

## **Design and Implementation Constraints**

This section of the report covers constraints which have limited or will be limiting the project during the development stage and post-production maintenance stage. Additionally, this section also covers the design standards followed by the development team.

### 2.5.1. Limitations

This subsection covers all the limitations which may or may not affect certain functionalities and future functionalities to be developed.

| **Limitations** | **Details** |
| --- | --- |
| Google Maps API | The Google Maps API used by LetsMakanTogether web application adopts a freemium business model which gives the user $500 in credits to query for API requests. Each API request incurs a small cost to use and eventually the credits will run out. Any subsequent request after the free credits have run out will not display the Google Maps result on the web application. |
| Tourism API - Food & Beverages | The Tourism API integrated onto the web application limits the result in each API request to only 50 per requests. This means that more API request calls have to be made to retrieve more results which may incur overheads in displaying all results at once.  Additionally, the API does not have all the food places data. Food businesses have to apply to Singapore Tourism Board to have their food places listed in the API and hence not all food places are listed in the API. |
| Cloud Service Provider Limitation | The development team implements the web application through the App Service provided by Microsoft Azure. However, the App Service has a free trial of only 12 months before a subscription fee is charged on per usage basis.  The App Service will also force the deployed web application into deep sleep mode after 20 minutes of inactivity monitored by Microsoft Azure. To fix this issue, the development team has to regularly communicate to the web application by pinging to the servers. |

### 2.5.2 Design Standards

This subsection covers all the design standards adopted by LetsMakanTogether web application.

| Design Standards | Details |
| --- | --- |
| Programming Standards | All non-class variables in the project adopts the camelCase naming convention  All class variables in the project adopt the Pascal Case naming convention. |
| User Interface Standards | All UI designs should adhere to the stakeholder’s requirement of color scheme and UI design. |

## **User Documentation**

### 2.6.1 LetsMakanTogether User Manual

The LetsMakanTogether team has made a user manual on how to set up and run the web application locally for their future Quality Assurance Team & Testing Team.

* Jia, Q. (n.d.). Letsmakantogether User Manual. LetsMakanTogether User Manual. [*https://online.fliphtml5.com/slbkl/ieya/#p=1*](https://online.fliphtml5.com/slbkl/ieya/#p=1)

### **2.6.2 LetsMakanTogether Demo Video**

A tutorial video to demonstrate how to navigate through the LetsMakanTogether web application.

* Link to the demo video: [*https://youtu.be/yR5GD1u\_z-w?si=n8SeD-6ZnB4y73bH*](https://youtu.be/yR5GD1u_z-w?si=n8SeD-6ZnB4y73bH)

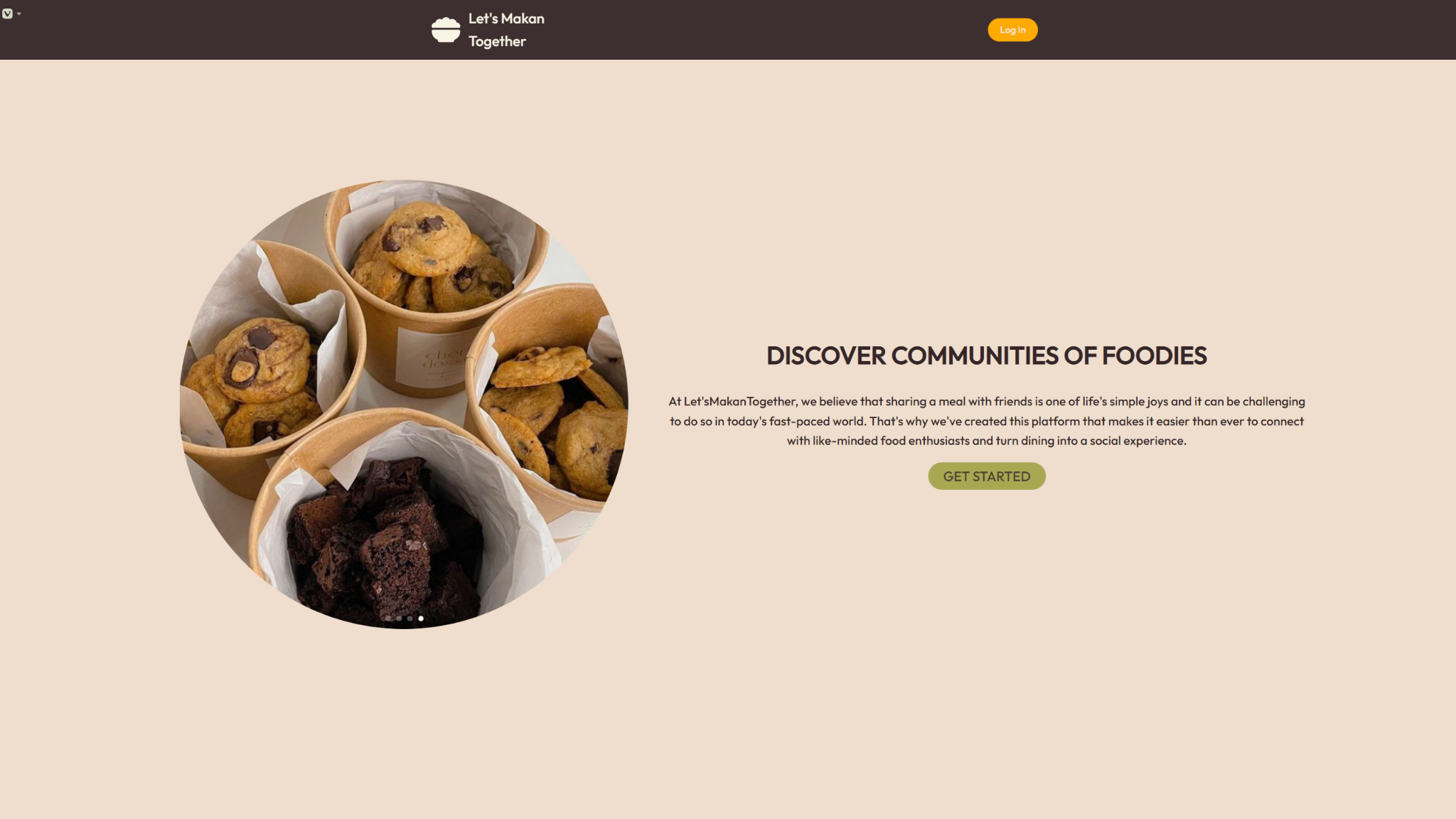
## **Assumptions and Dependencies**

The LetsMakanTogether development team developed the web application with the following assumptions:

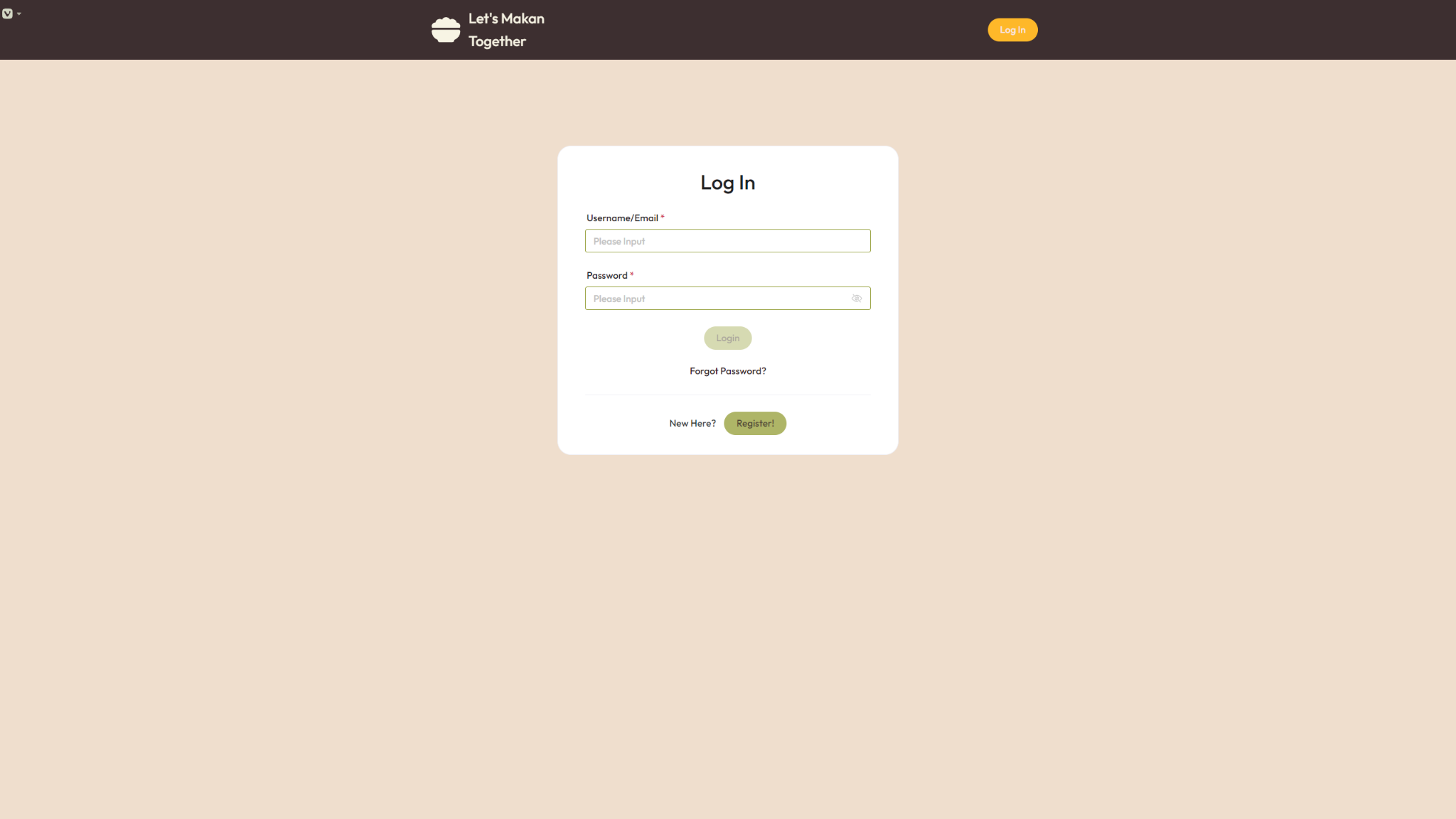
* The full-scale developed LetsMakanTogether web application will receive full funding for full subscription towards all APIs required as per documented in section *2.5.1 Limitations* once the prototype application is approved by all stakeholders.
* The full-scale developed LetsMakanTogether web application will be sponsored with free access to unrestricted usage of database and cloud hosting service as per documented in section 2.5.1 Limitations once the prototype application is approved by all stakeholders
* The LetsMakanTogether web application is designed to be best viewed with desktop browsers.
* The LetsMakanTogether web application did not incorporate OAuth implementation. Thus, the back-end server will not log out any user until 24 hours has passed since login. Hence, the web application is built with the assumption that users will log out of their account once they are done.

# **External Interface Requirements**

## **User Interfaces**

***Figure 3.1.1: Landing Page***

*Users are presented with a slideshow of food pictures alongside a short description about us. Users can click on “Get Started” or “Sign In” button to visit the login page.*

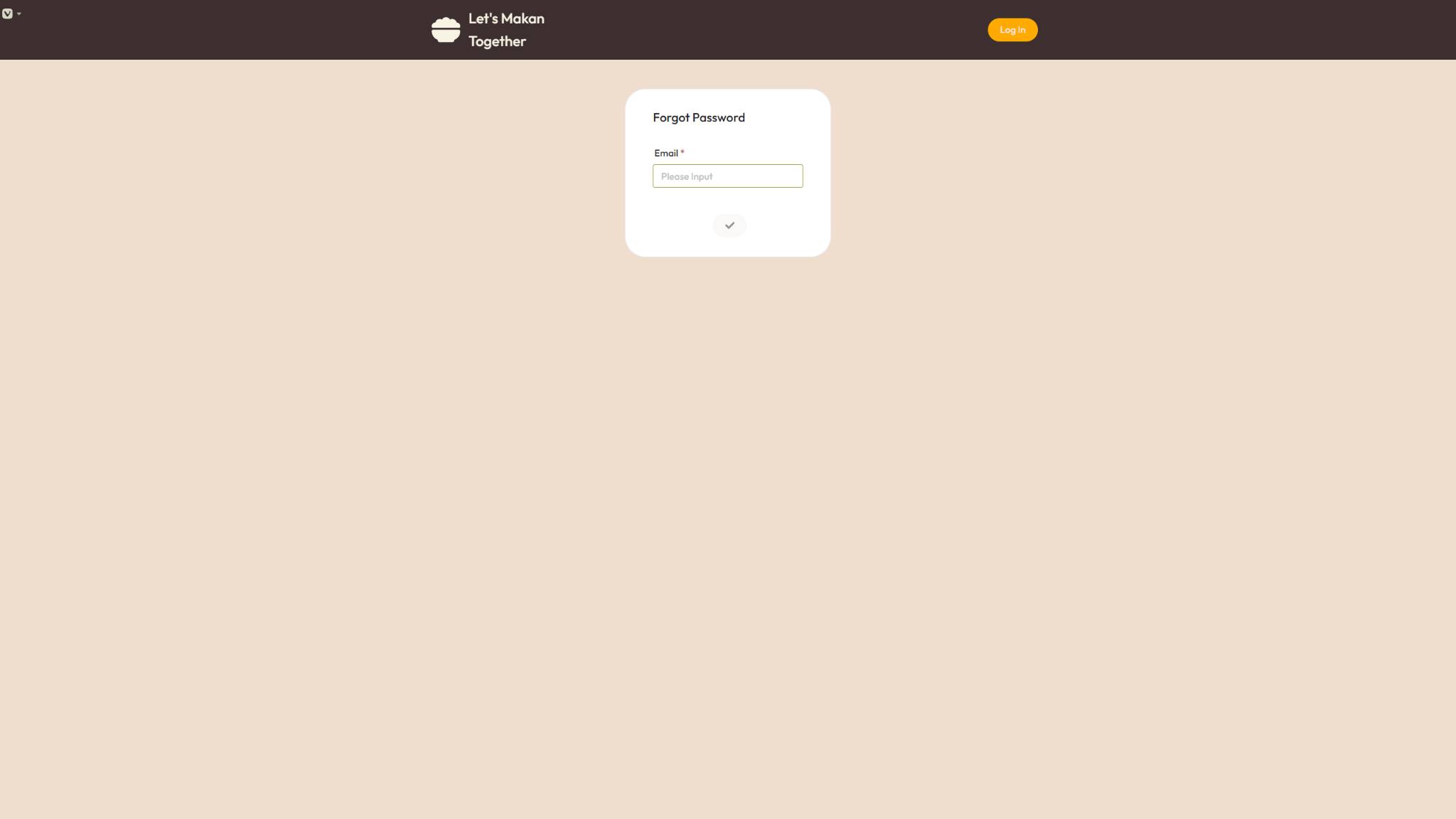
****

***Figure 3.1.2: Login Page***

*Users are presented with two text boxes.*

*Users have the choice to sign in using Username or Email, and they must input the correct password to sign in. A mask icon can be clicked to unmask the password.*

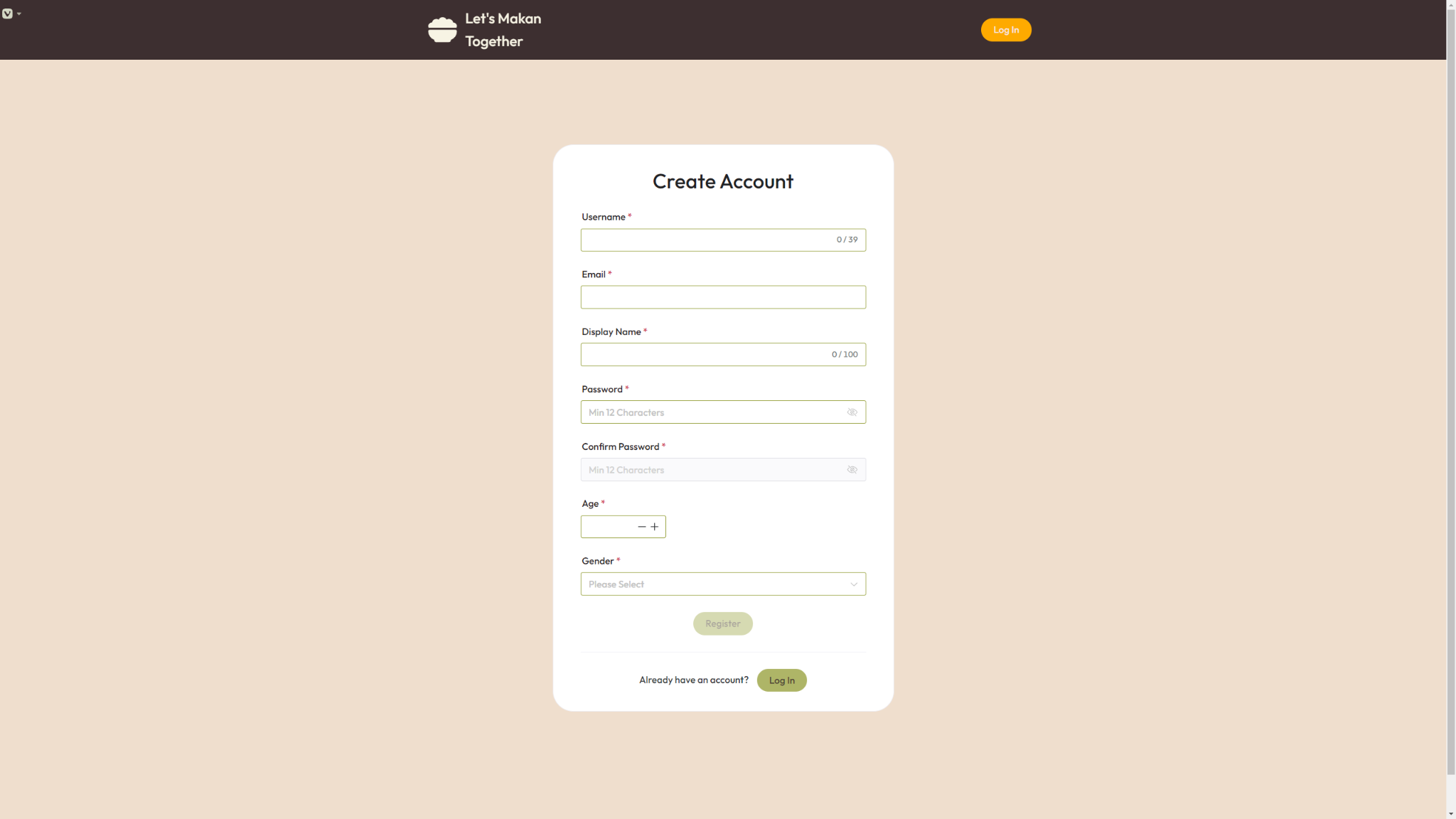
*Click on “Login” to sign in after filling up the form, “Forgot Password” to reset password, or “Sign Up” to register a new account.*

******

***Figure 3.1.3: Reset Password***

*Users are prompted to enter their email to receive an email to reset their password.*

*Users can then click on the “Confirm” button to proceed or the ‘“Let’s Makan Together” icon to return to landing page.*

******

***Figure 3.1.4: Create Account Page***

*Users must fill up every field present in the form.*

*Username will be each user’s unique identifier, which cannot exceed 39 characters.*

*Email must be in a valid format.*

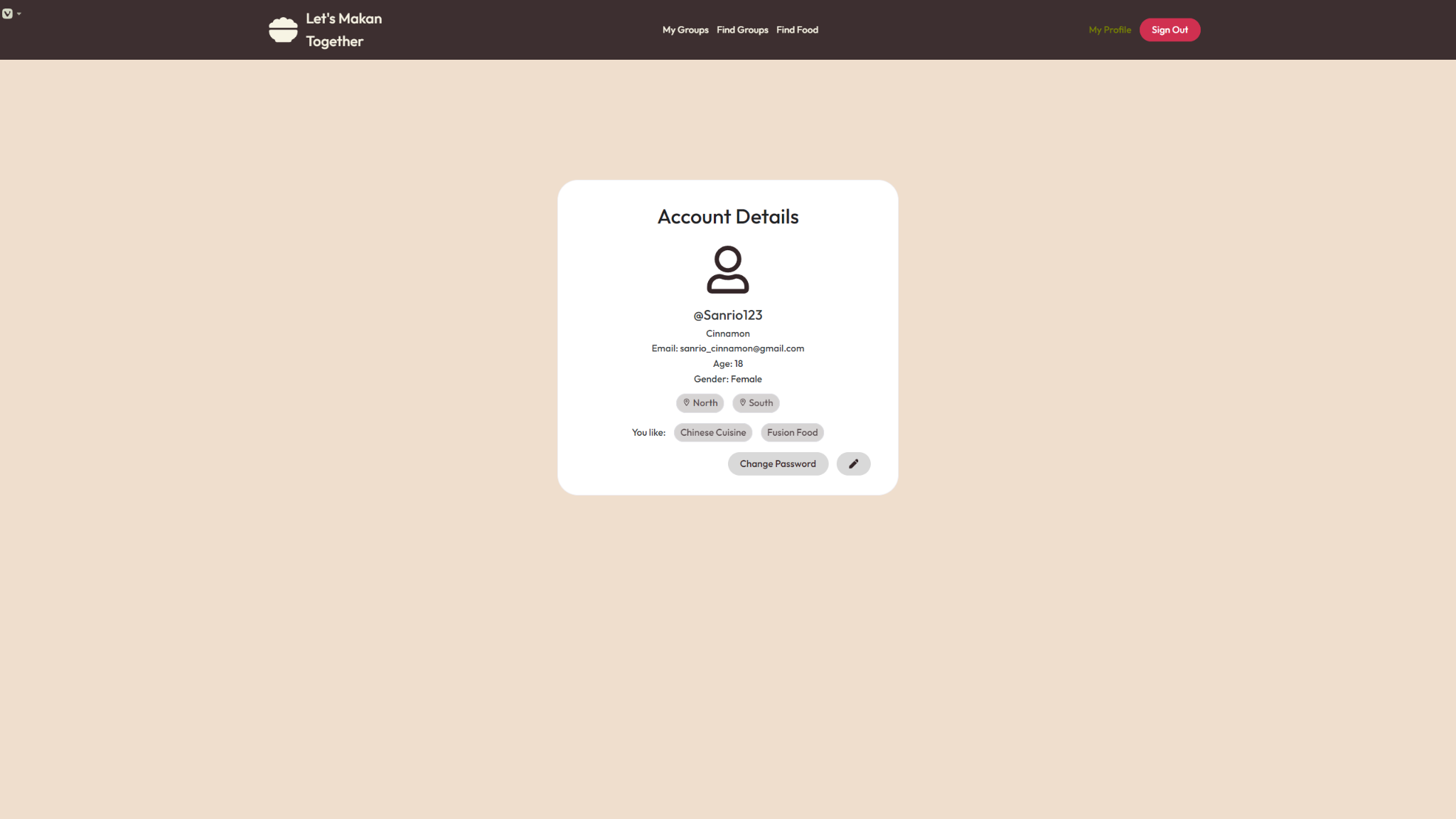
*Display name is a customisable handle, which cannot exceed 100 characters.*

*Passwords must contain at least 12 alphanumeric characters.*

*Users must be at least 18 to sign up.*

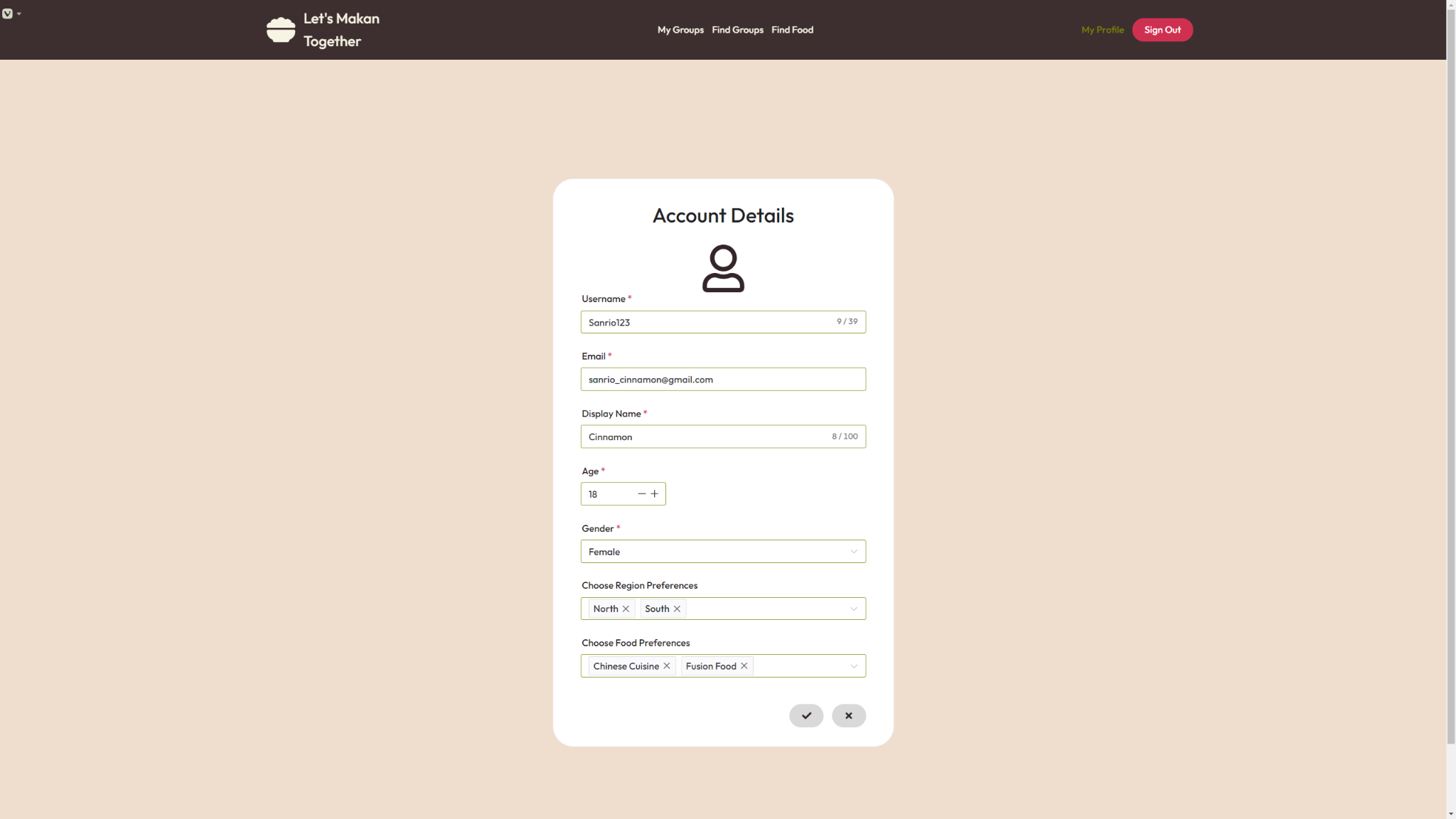
*Users can choose “Male”, “Female” or “Prefer not to say” as their gender.*

*Click on “Register” to complete sign up process, or “Log In” to visit the login page for users with existing account.*

******

***Figure 3.1.5: Edit Profile Page***

*Users can view their account details at this page. Users can click on “Change Password” to display Change Password Form (Figure 3.1.7) or “Pencil” button to display Edit Profile Form (Figure 3.1.6).*

******

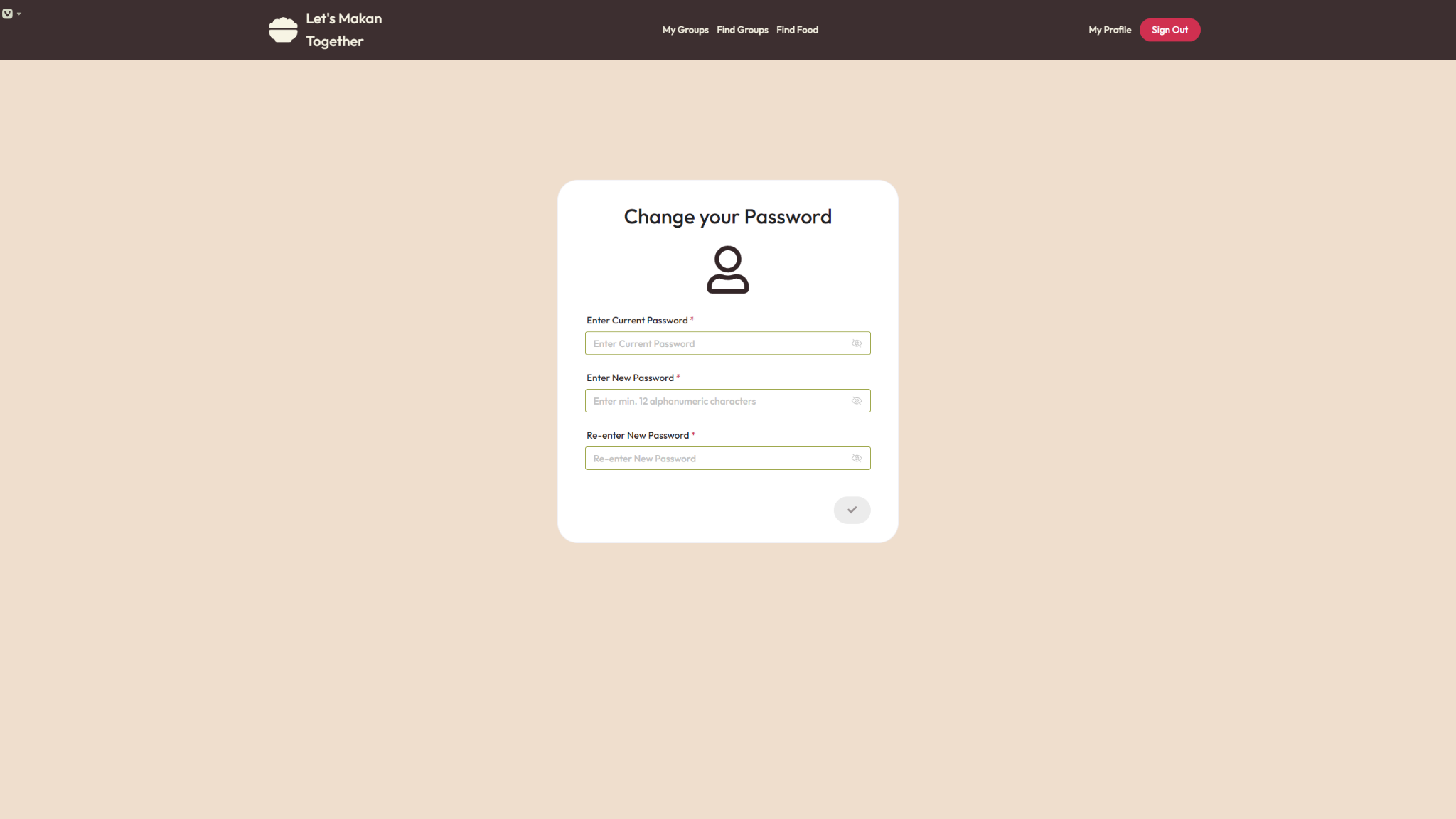
***Figure 3.1.6: Edit Profile Form***

*Users are presented with editable fields to change their account details.*

*Fields with the asterisk cannot be empty.*

*Users can edit their Regional and food preferences to filter groups by default at the ‘Find Groups’ page (Figure 3.1.13.1).*

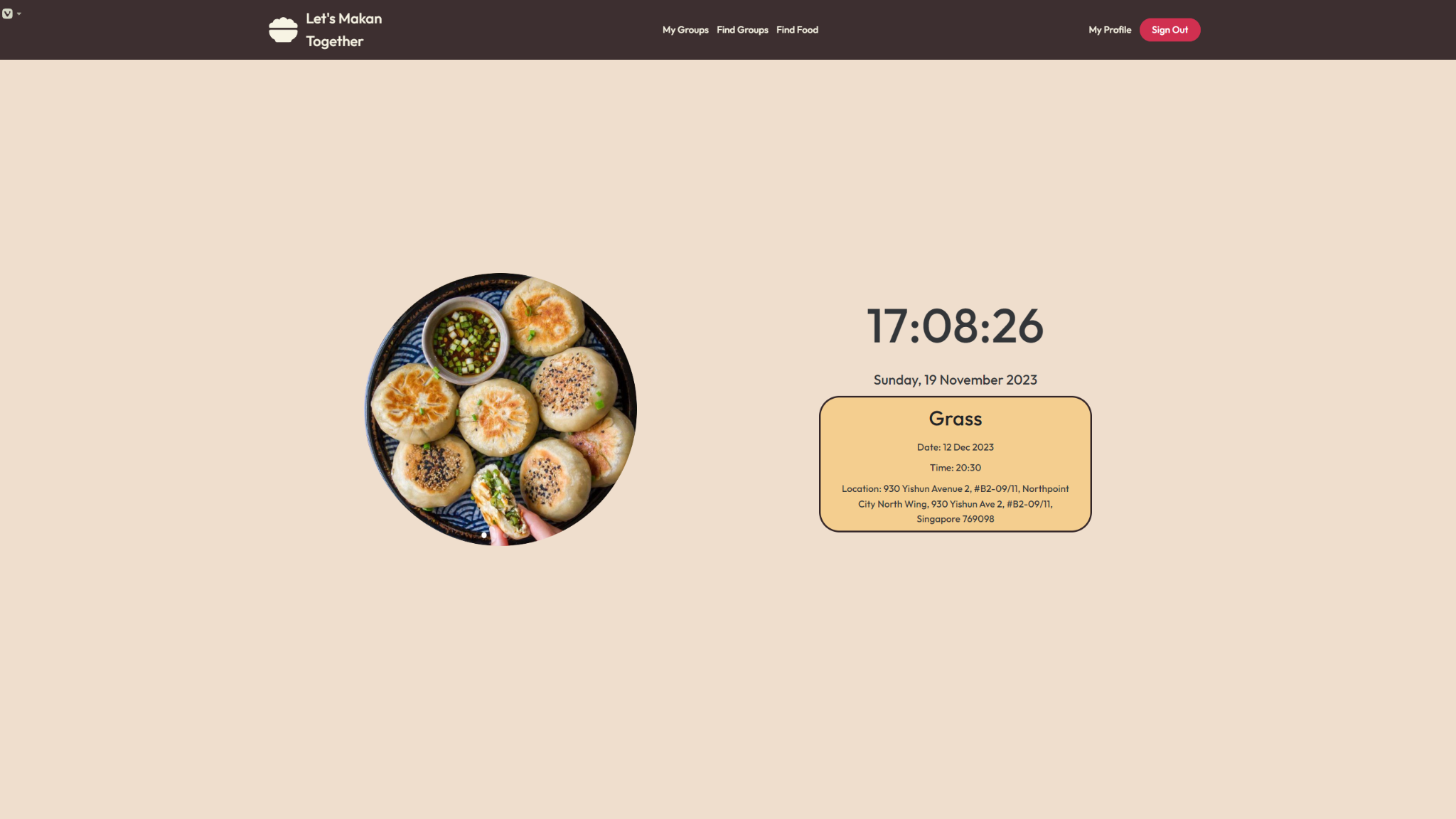
*Users can click “Tick” button to save edit or “x” button to cancel their edits which would not be reflected.*

******

***Figure 3.1.7: Change Password Form***

*Users are prompted to enter compulsory fields such as current password, new password and confirm new password. Password requirements remain the same at least 12 alphanumeric characters.*

*Users can then click “Tick’” button to confirm changing their password.*

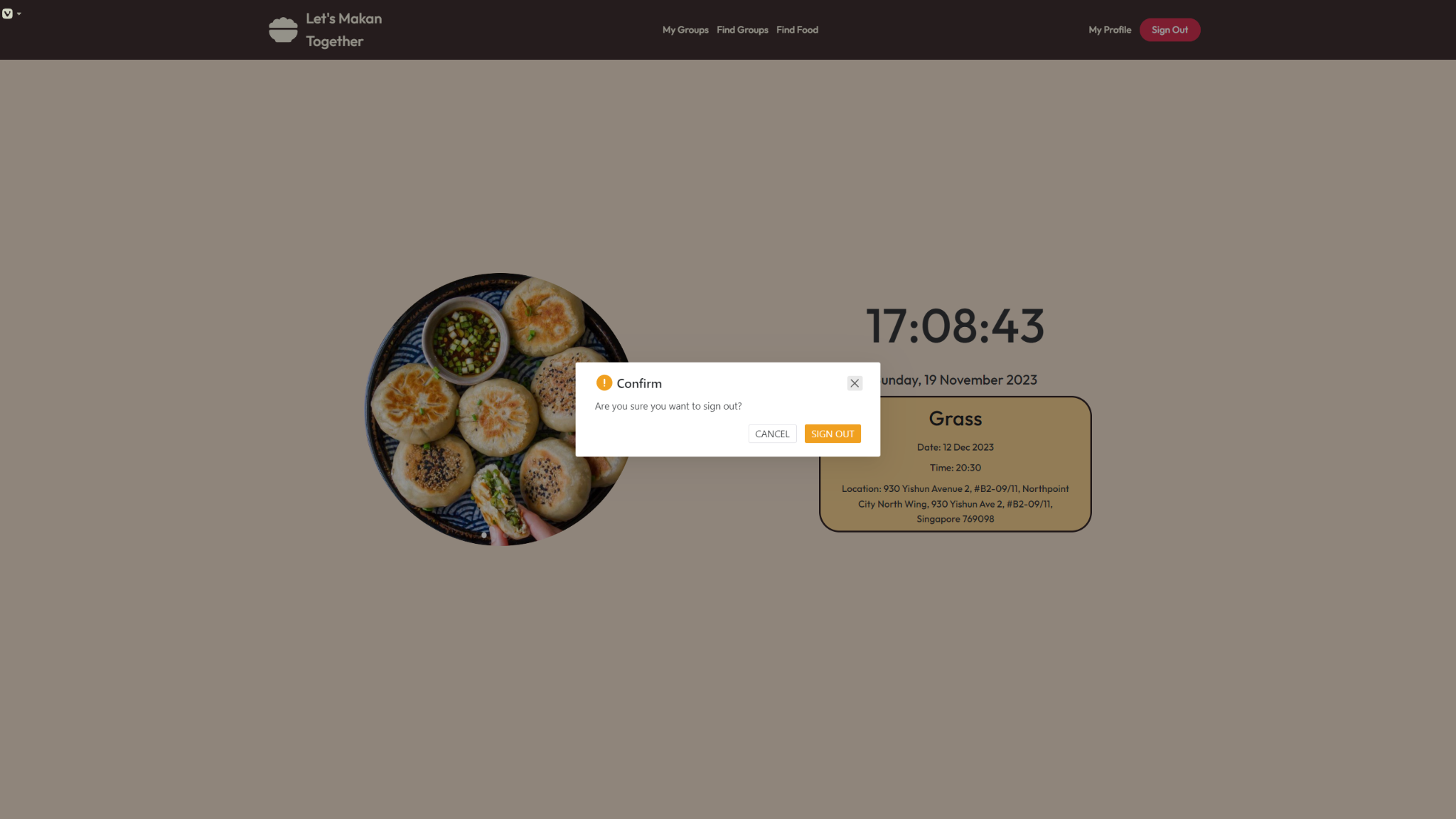
******

***Figure 3.1.6.1: Home Page***

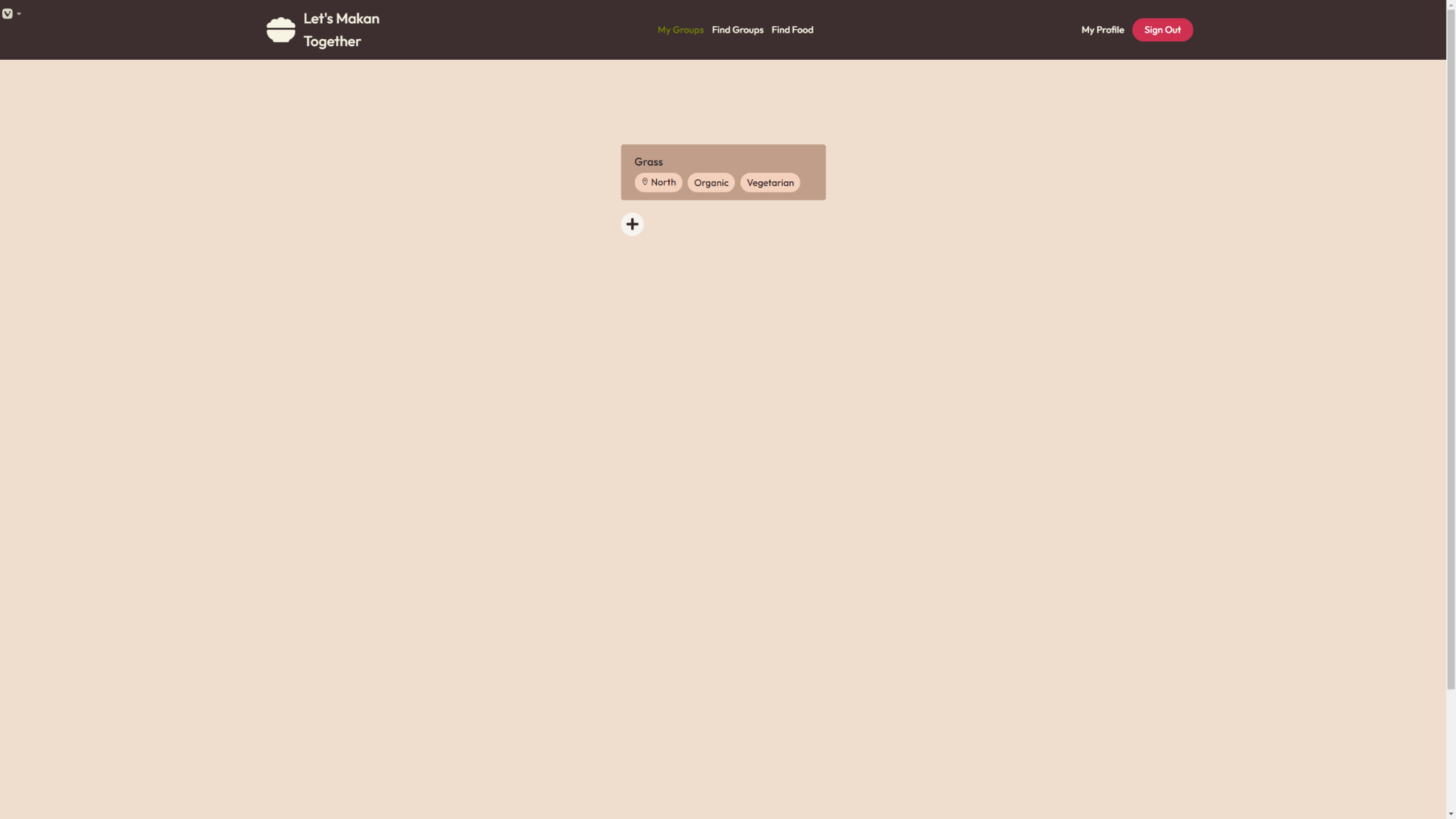
*Clicking on any one of the tabs in the navigation bar consisting of “My Groups”, “Find Groups”, “Find Food”, and “My Profile” to proceed to the corresponding pages.*

*This page is also accessible by clicking on the “Let’s Makan Together” icon on the top left.*

*Users will see a live clock and the details of any upcoming meetings with the groups they have joined.*

***Figure 3.1.6.2: Home Page (Sign Out Prompt)***

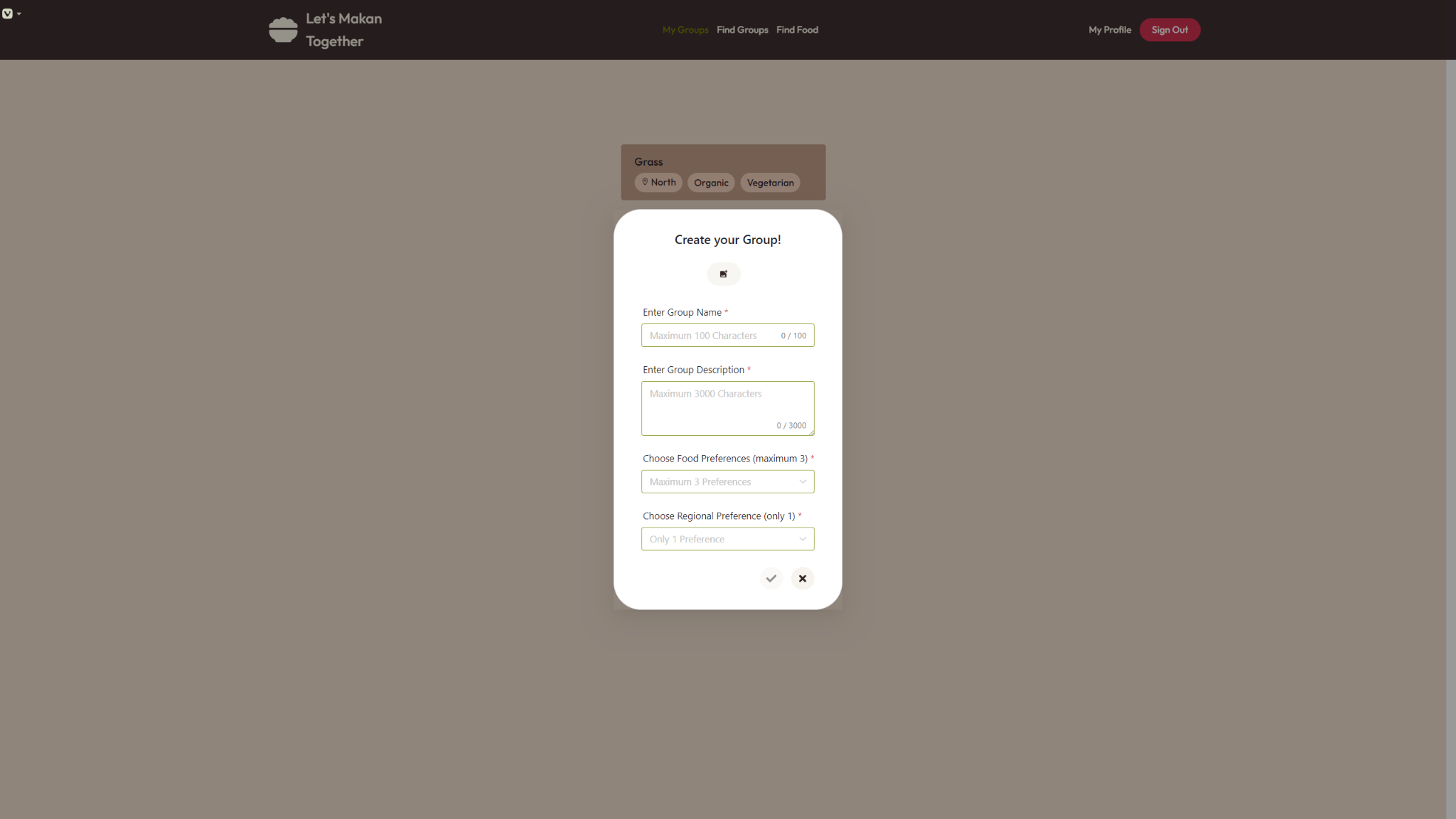
*Clicking the “Sign Out” button, will result in a dialog message prompting the user for confirmation to sign out. Clicking on “SIGN OUT” will have the user logged out, while clicking on “CANCEL’” will have the user return to the home page.*

******

***Figure 3.1.7: My Groups***

*Under the tab “My Groups”, users are presented with groups that they have currently joined.*

*Users can click the “+” button to display the create group form (Figure 3.1.8) or click on a specific group to view group details (Figure 3.1.10).*

******

***Figure 3.1.8: Create Group Form***

*Users must fill up every field present in the form to create a new group.*

*Group name must be unique and not exceed 100 characters.*

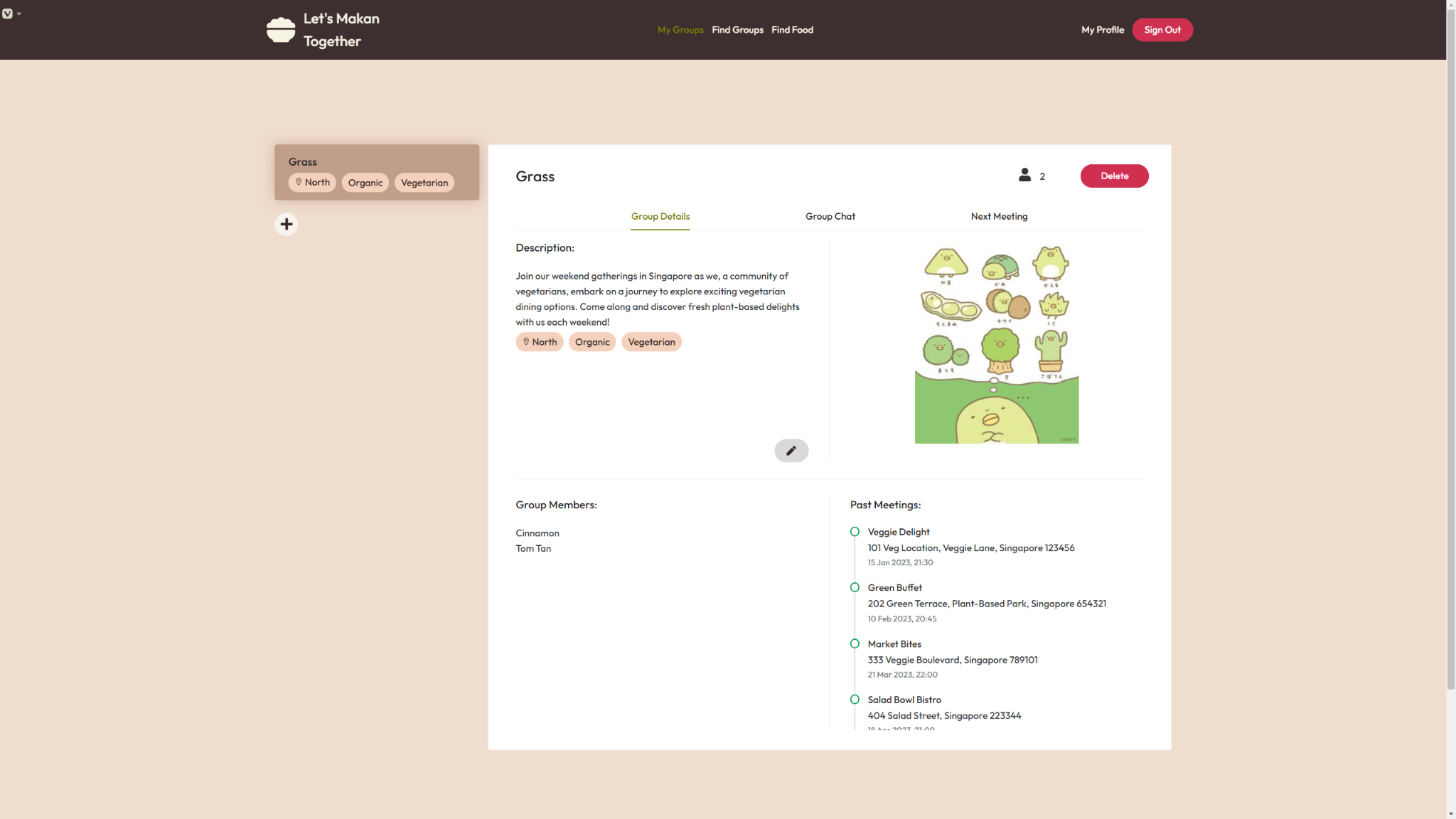
*Group description must not exceed 3000 characters.*

*“Food Preferences” has a maximum tag count of 3.*

*“Regional Preference” only allows a single selection.*

*Users have the option to upload a group display picture.*

*Users click the “Create” button to confirm group creation or “Cancel” button to cancel group creation.*

******

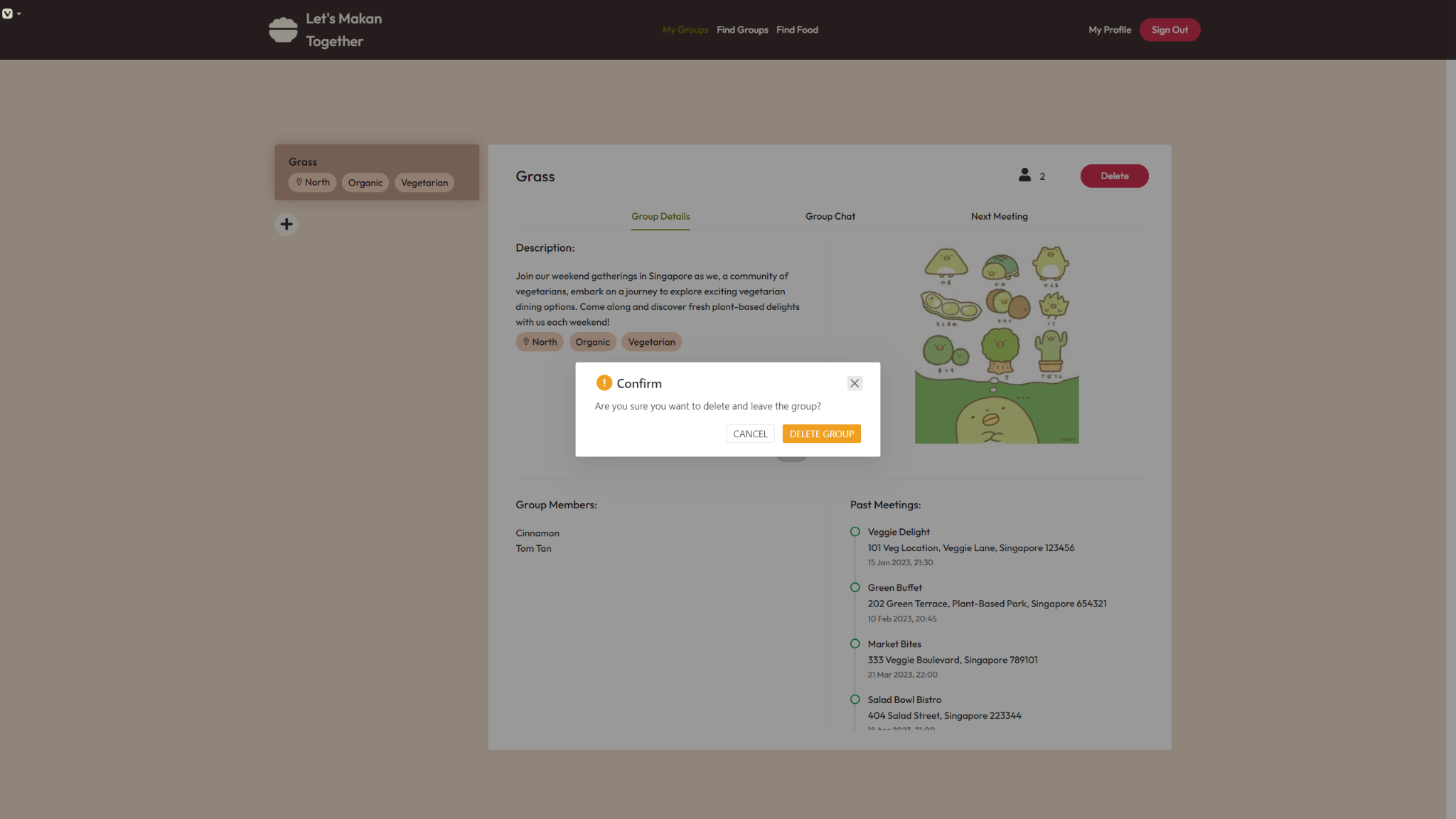
***Figure 3.1.9.1: View Group Details***

*Users can view group details such as description, preference tags, group display picture, current group members and past meetings.*

*Only group owners can see and click the “Pencil” button to display the edit group form (Figure 3.1.10).*

*Only group owners can see and click the “Delete” button to display the delete group prompt (Figure 3.1.9.2).*

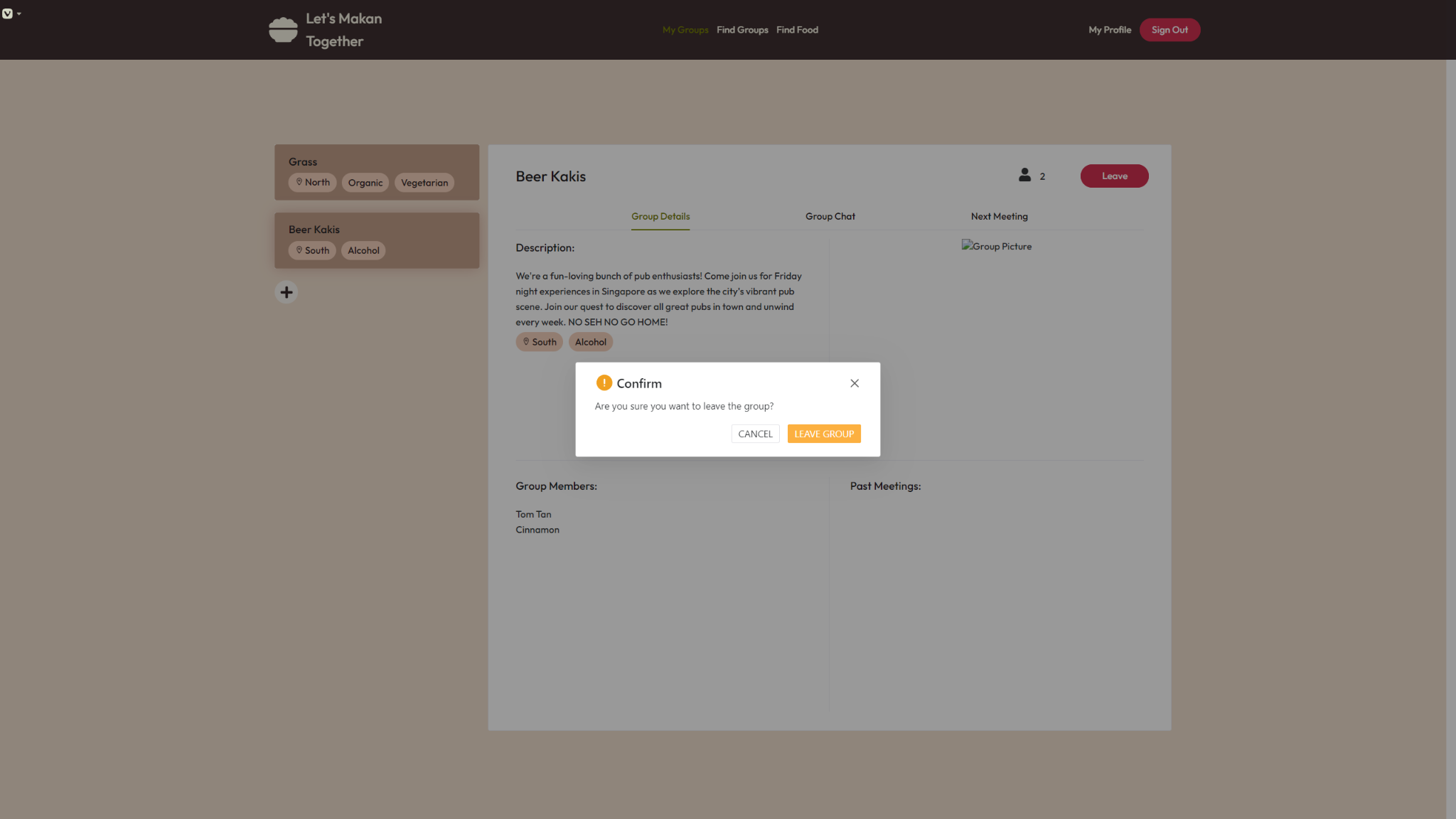
*Group members can only see and click the ‘Leave’ button to display the leave group prompt (Figure 3.1.9.3).*

***Figure 3.1.9.2: Delete Group Prompt (for group owners)***

*Clicking the “Delete” button will result in a dialog message prompting group owners for confirmation to delete the group.*

*Clicking “DELETE GROUP’”will confirm the deletion of the group along with the removal of all group members from the group.*

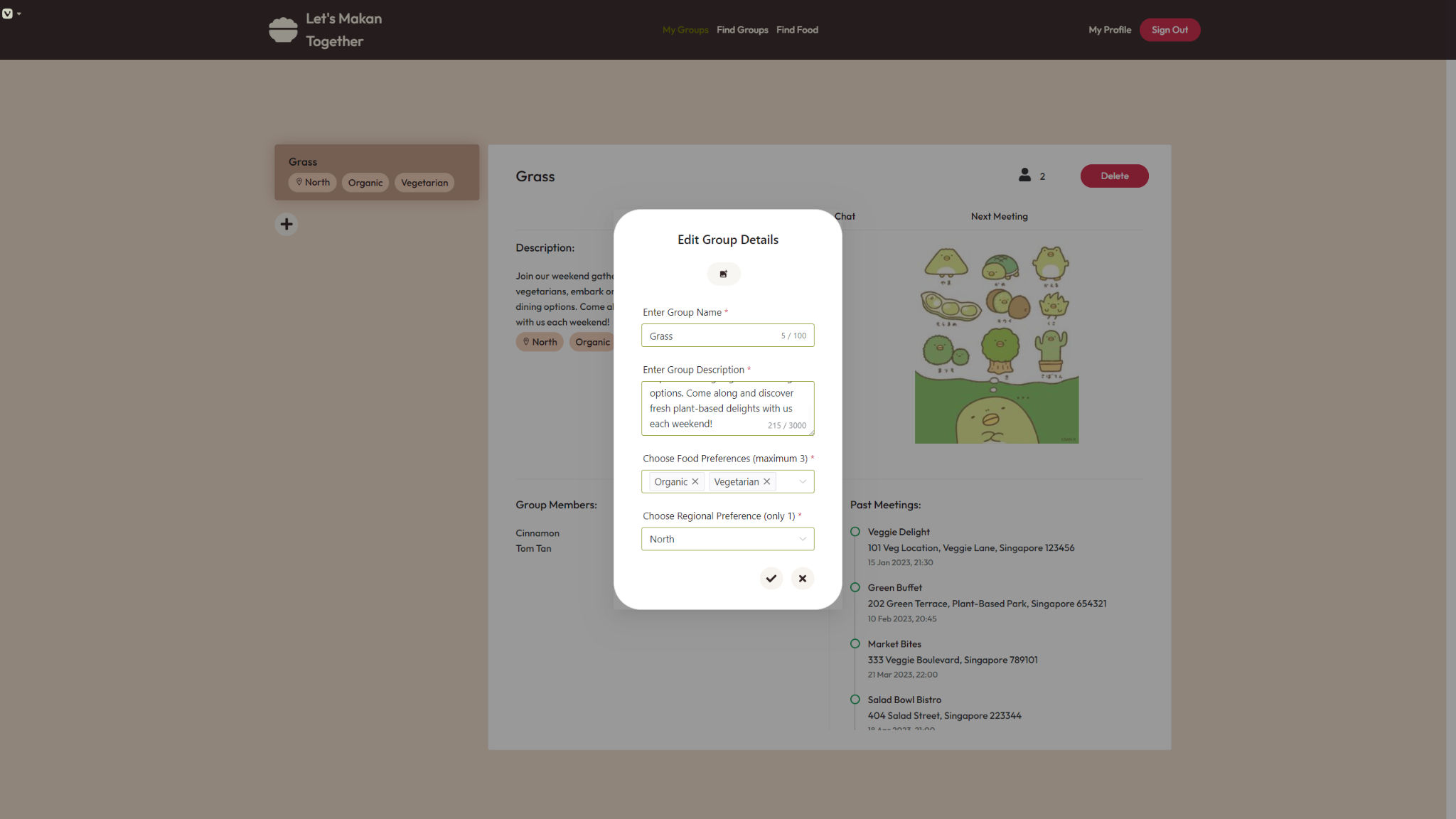
*Clicking “CANCEL” will have the group owners return to the current page.*

**

***Figure 3.1.9.3: Leave Group Prompt (for group members)***

*Clicking the “Leave” button will result in a dialog message to prompt users for confirmation to leave the group.*

*Clicking “LEAVE GROUP” will confirm leaving the group, while clicking “CANCEL” will have the users return to the current page.*

******

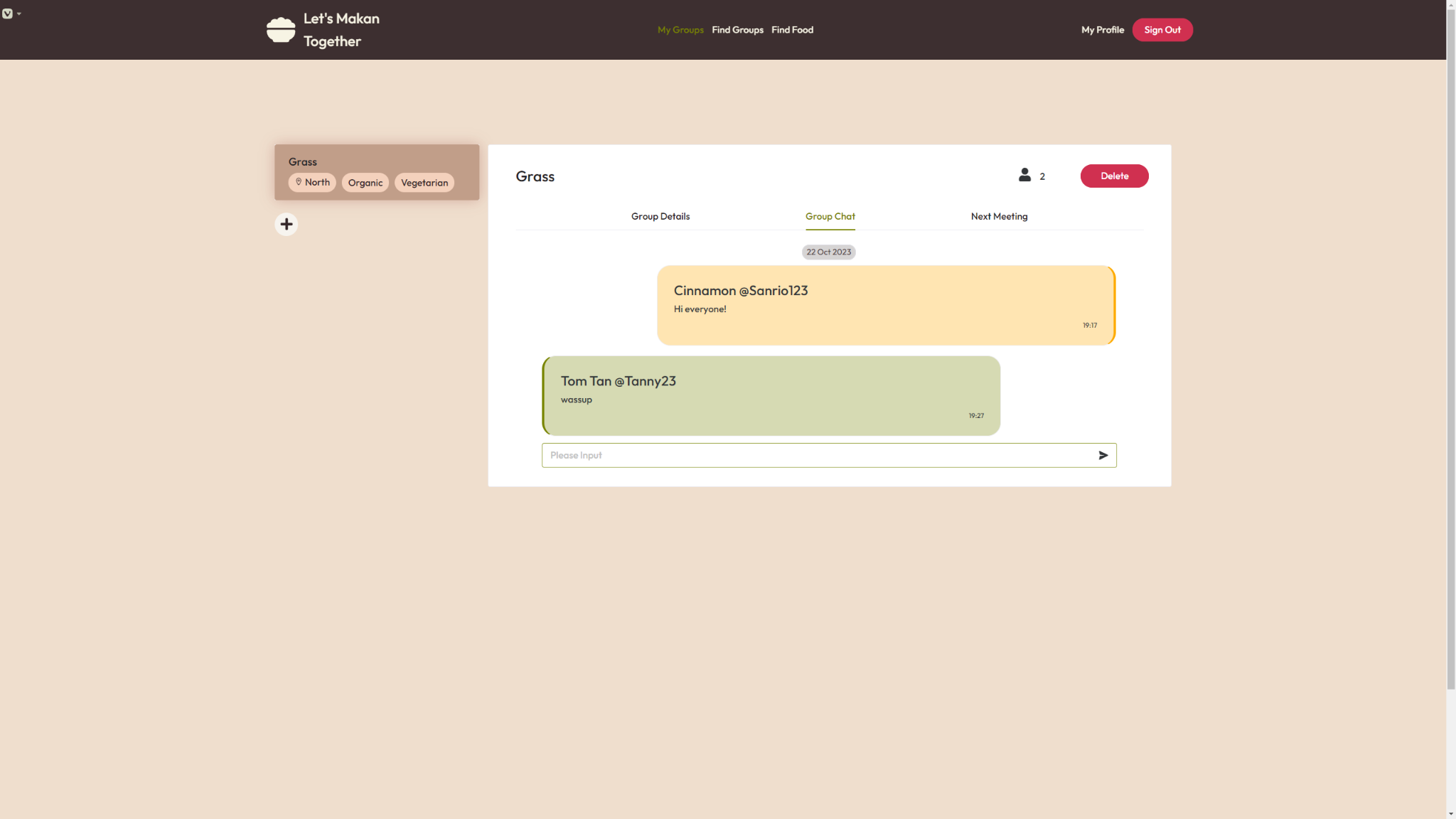
***Figure 3.1.10: Edit Group Form (for group owner)***

*Group owners can edit a group’s details when they click the “Pencil” button from View Group Details, where they will be presented with the above Edit Group Details form.*

*Group Owners must fill up every field of the form.*

*Group owners can click the “x” button to cancel their edits where the group details would remain unchanged in View Group Details (Figure 3.1.9).*

*Group Owners can click the “Tick” button to save their edits where the new group details would be reflected in View Group Details (Figure 3.1.9).*

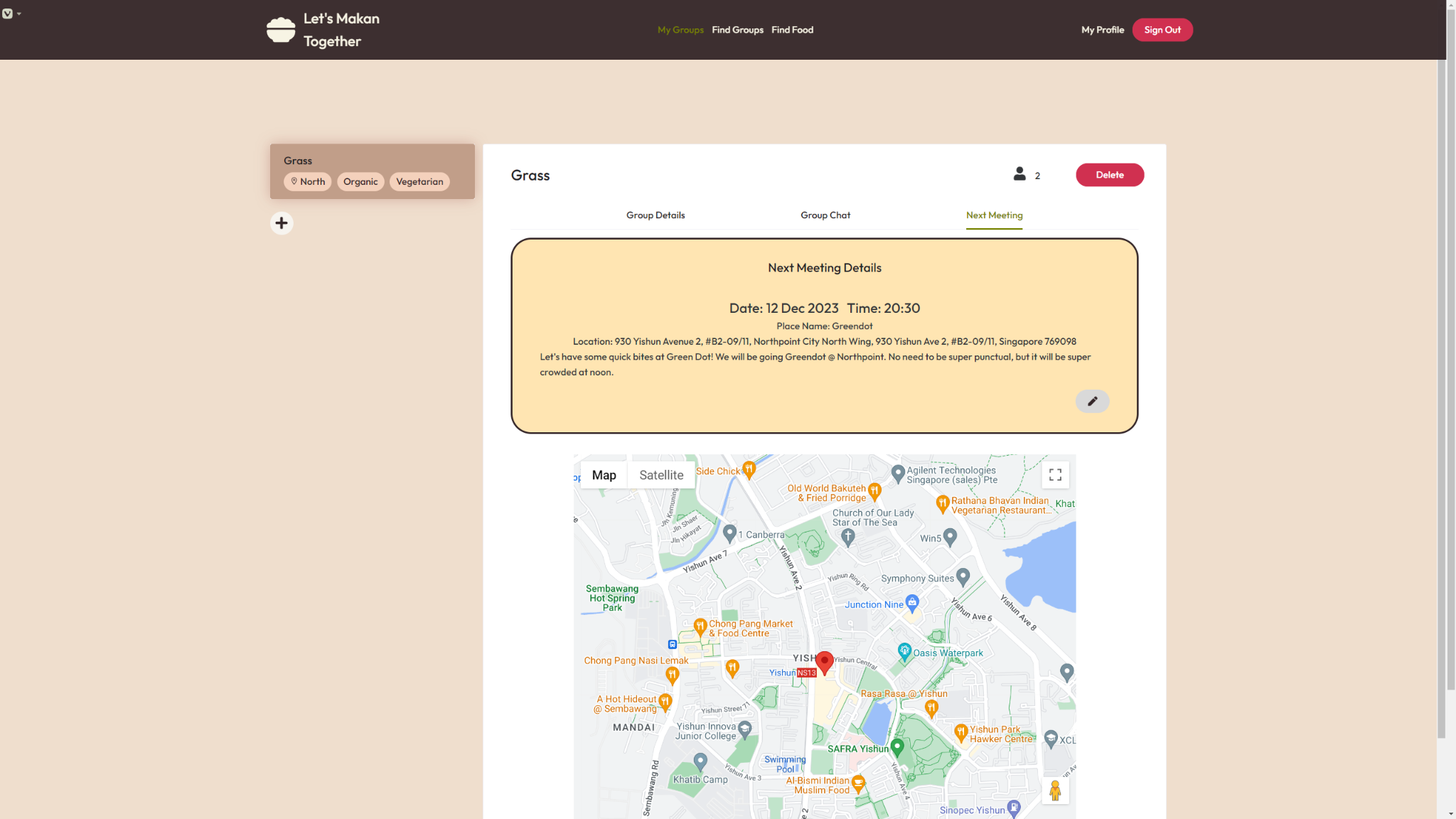
******

***Figure 3.1.11: View Group Chat***

*Users can view the group chats of any group they have joined.*

*Users can see the past message history within the group chat, and can send messages into the group chat.*

*Each group chat message has the sender’s username and display name attached, as well as the date and time that the message was sent.*

******

***Figure 3.1.12: View Next Meeting***

*Users can view the details of the group’s Next Meeting, including the meeting Date, meeting Time, meeting Place, meeting Location, and meeting Description.*

*Users can also view the meeting Location on the map below the Next Meeting card.*

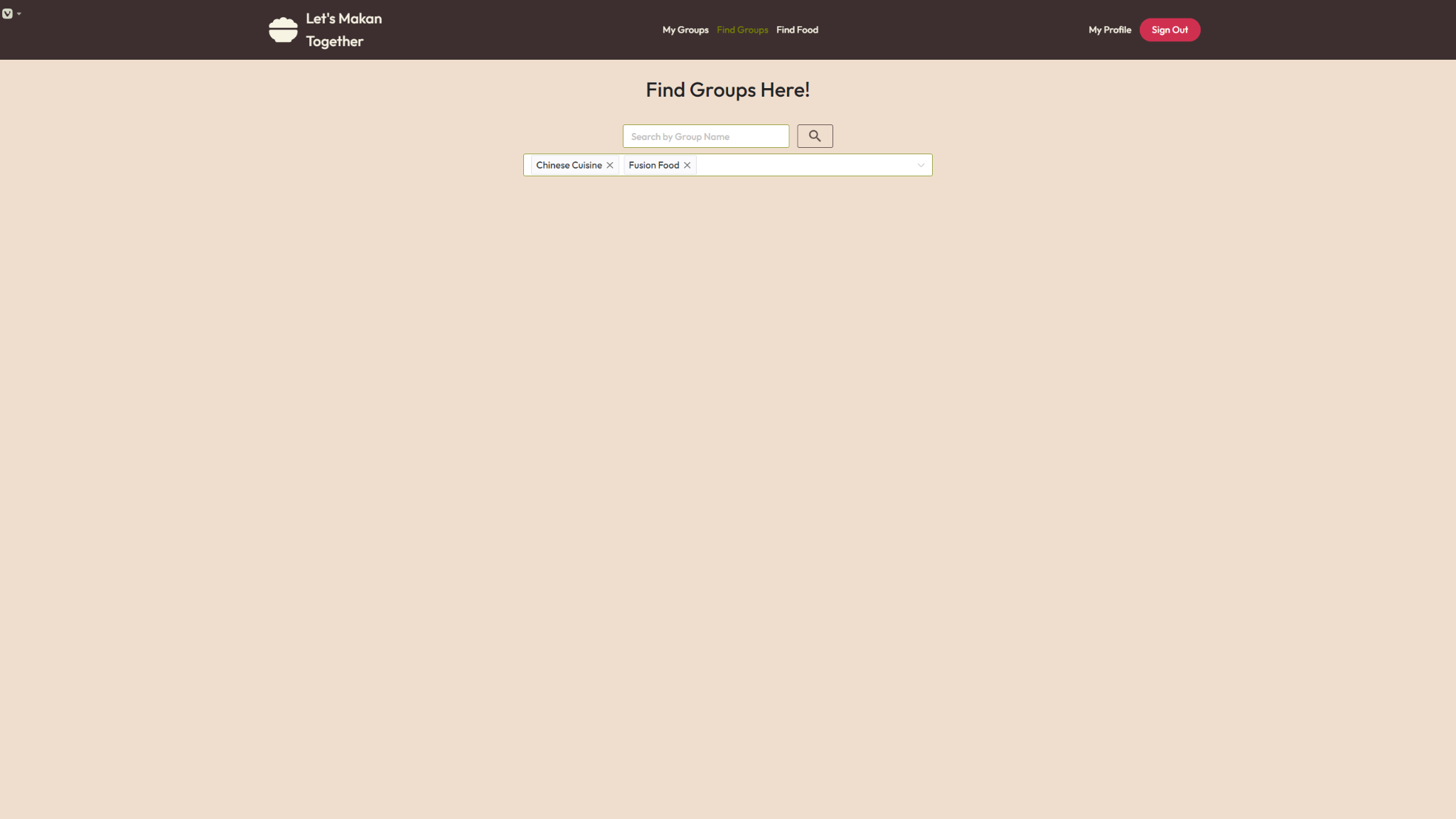
*Only group owners can view and click the ‘Pencil’ icon to display Edit Next Meeting page (Figure 3.1.13).*

******

***Figure 3.1.13: Edit Next Meeting (for group owner)***

*Group Owners will be able to edit Next Meeting when they click on the pencil button under View Next Meeting. Group owners will be presented with the above Next Meeting form and they must fill up every field.*

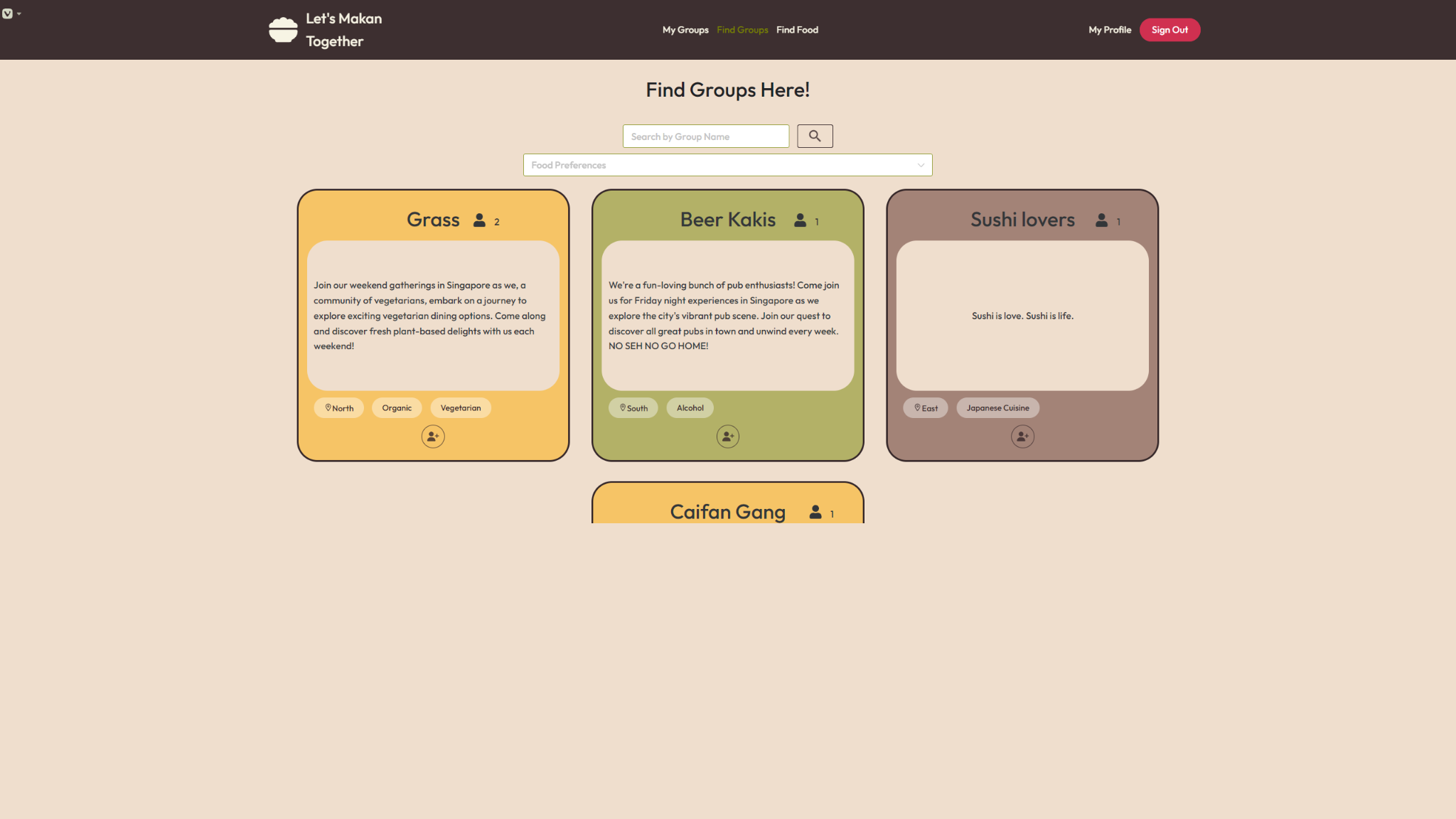
*To fill in the fields Place and Location, Group Owners must search a food location under the search bar below the form, before selecting a landmark from the search result. Both fields will then be automatically populated correspondingly when the group owner clicks the “Confirm” button.*

******

***Figure 3.1.14.1: Find Groups***

*Users can search for existing groups through 2 methods of search:*

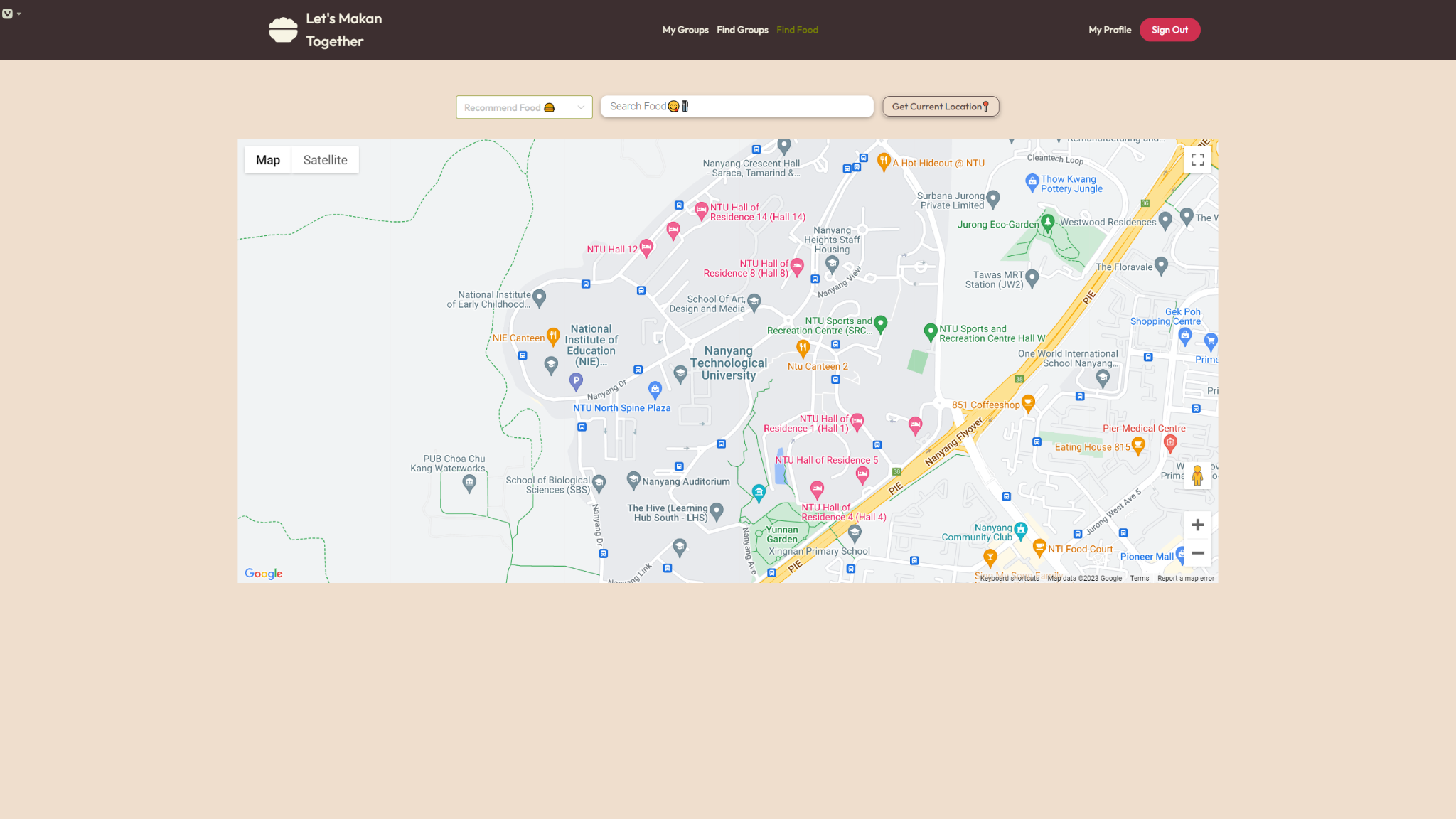
1. *Searching a specific group name under the Search by Group Name search bar.*
2. *Selecting an option under the Food Preferences filter. Should the user have any existing food preferences saved, those food preferences would appear by default as pre-selected Food Preferences.*

******

***Figure 3.1.14.2: Search Result for Find Group***

*After a successful search query, users will be able to see the list of corresponding groups if any.*

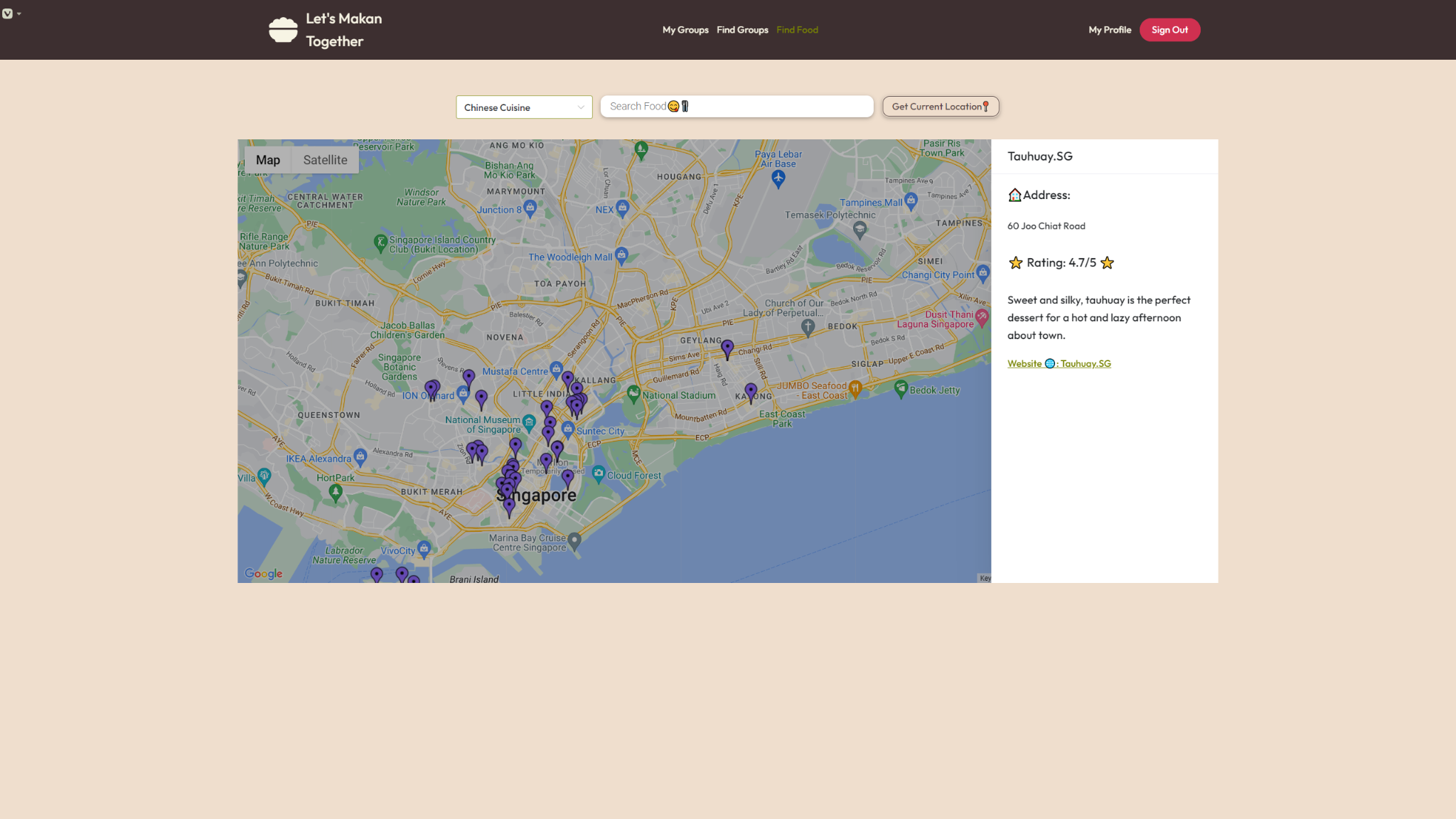
*Users can click the “+” button at the bottom of each group to join the respective group.*

******

***Figure 3.1.15.1: Find Food***

*Users can search for food locations through 2 methods of search:*

1. *Selecting an option under the Recommended Food filter.*
2. *Entering a specific search query of a type of food, a food location name, or an address of a food location under the Search Food search bar.   
   Users can also get their current location by clicking the “Get Current Location” button, which will prompt the user for access to their current location if it has not been enabled already.*

******

***Figure 3.1.15.2: Search Result for Find Food***

*Users can view the search results from either the selection from the Recommended Food filter, or their search query. The search results are indicated by the purple landmarks.*

*Users can click on any of the purple landmarks to view the side panel containing details of the food location, such as the name, address, ratings and website.*

## **Hardware Interfaces**

This section covers all the hardware interface requirements for LetsMakanTogether web application to achieve desired functionalities. The requirements are divided into client-side requirements and server-side requirements to facilitate easier reading for stakeholders.

### 3.2.1 Client-side Requirements

The web application supports all desktops and laptops setup. The device used to access the web application must support the usage of a desktop browser requirement indicated in section *2.4.1 Production Environment of LetsMakanTogether.*

### 3.2.2. Server-side Requirements

The web application back-end server must be hosted and run on a server-computer. The back-end server will handle all of the Create, Read, Update, Delete (CRUD) operations done by users on the front-end to the back-end database. The database must be hosted and run alongside on the server-computer.

## **Software Interfaces**

### 3.3.1 Software Components and Versions

LetsMakanTogether web application utilizes a back-end server and a database to handle all the systems’ features implementation. The back-end server is built and implemented using ExpressJS and Axios as mentioned earlier in *2.4.2 Development Environment of LetsMakanTogether.* The development team adopts MySQL as the database for the web application.

### 3.3.2 Software Architecture

LetsMakanTogether web application software architecture must follow the Model-View-Controller design pattern. The interface must be able to connect to the database to store persistent data in a format where SQL is able to read.

## **Communications Interfaces**

The LetsMakanTogether web application communication architecture must follow a client-server model. Each communication must go through a REST-styled Application Programming Interface (API) provided by the back-end server. Each request sent by the clients must be served over HyperText Transfer Protocol Secure (HTTPS).

Clients must invoke GET and POST requests to communicate with the server. Furthermore, communication from server to client must serve data standardization in JavaScript Object Notation (JSON) format.

# **System Features**

## Landing Page

**4.1.1 Description and Priority**

The Landing Page is the first point of contact that greets users when they load the website with new cookies. Its primary function is to provide an engaging and user-friendly gateway to the website. It allows new users to get started by signing up for an account, and existing users to sign in to their account.

| **Overall Priority** | High |
| --- | --- |

**4.1.2 Stimulus/Response Sequences**

| **Stimulus** | **Response** |
| --- | --- |
| 1. Load website at https://letsmakantogether.com with new cookies | Shows Landing Page |
| 1. Click “Sign In” button | Shows Login Page |
| 1. Click “Get Started” button | Shows Login Page |

**4.1.3 Functional Requirements**

1.1 When the user loads the website with new cookies, the website logo and name

must be shown.

1.2 The website must display an option that allows the user to sign in.

1.2.1 The system must redirect the user to the Login Page.

## Account Registration / Registration Page

**4.2.1 Description and Priority**

New users of LetsMakanTogether can register for an account. The registration process involves inputting a valid email address and password, which is at least 12 characters long. The web application must validate all inputs and display appropriate error messages if the inputs are invalid or incomplete. The user’s email address has to be verified through a verification email link. Upon registration, a record of the user’s email address, username and password is stored in the database. Any subsequent updates made by the user, such updating user profile, joining groups, searching for food location, will be using this registered account as reference.

| **Overall Priority** | High |
| --- | --- |

**4.2.2 Stimulus/Response Sequences**

| **Stimulus** | **Response** |
| --- | --- |
| 1. User clicks ‘Register’ button | The system displays a register form |
| 1. User input valid details for all fields | Register button turns clickable |
| 1. User clicks register button | The system shows “Successfully registered” Message and redirects user to Login page |

**4.2.3 Functional Requirements**

2.1 The user must input their email address into an input text box.

2.1.1 The system must check whether the input is a valid email address.

2.1.1.1 If the email address is not valid, the system must display an error

message “Email is invalid!”.

2.1.1.2 If the email has already been taken, the system must display an error

message “A user with the given username or email already exists.”

that prompts the user to input a different email.

2.2 The user must input an alphanumeric password into an input text box.

2.2.1 The system must display an icon inside the input text box that allows the user

to mask their password input.

2.2.1.1 By default, the icon is unchecked, where the system must mask the

password when typed by the user.

2.2.1.2 If the icon is pressed by the user, the system must not mask the

password input.

2.2.2 The system must ensure that the password length is 12 characters or more.

2.2.2.1 If the password length is not met, the system must display an error

message that prompts the user to input a new password.

2.2.3 The system must ensure that the password is alphanumeric.

2.2.3.1 If the password input is not alphanumeric, the system must display an

error message that prompts the user to input a new password.

2.2.4 The user must input the aforementioned password again into an input text box.

2.2.4.1 The system must ensure that both passwords match.

2.2.4.1.1 If the passwords do not match, the system must display an

error message that prompts the user to check their password.

2.3 The user must input a username into an input text box.

2.3.1 The system must ensure that the length of the username is 39 characters or less.

2.3.1.1 If the username is 39 characters long, the system must not allow the

user to input more characters.

2.3.1.2 If the username has already been taken, the system must display an

error message “A user with the given username or email already

exists.” that prompts the user to input a different username.

2.4 The user must input a name into an input text box.

2.4.1 The system must ensure that the length of the name is 100 Unicode characters

or less.

2.4.1.1 If the name is 100 characters long, the system must not allow the user

to input more characters.

2.5 The user must input their age into an input text box.

2.5.1 The system must ensure that the age must be 18 or more.

2.5.1.1 If the age is 17 or less, the system must display an error message

“Age should be above 18” that prompts the user to input a different

age.

2.6 The user must select their gender from a drop-down selection.

## Login Page

**4.3.1 Description and Priority**

The login page allows users with existing accounts to access their accounts. Users are required to input their username/ email address and password as verification during the login process. Users must be signed in prior to accessing all the features provided by LetsMakanTogether web application.

| **Overall Priority** | High |
| --- | --- |

**4.3.2 Stimulus/Response Sequences**

| **Stimulus** | **Response** |
| --- | --- |
| 1. User clicks ‘Sign In’ Button | The system will display a login form |
| 1. User inputs login credentials accordingly and clicks ‘Login’ Button | The website sends a request to the  system to authenticate the user |
| 1. The system authenticates the user | User can view the Home Page |

**4.3.3 Functional Requirements**

3.1 If the user does not have an account, the system must display an option that leads to the

Registration Page where they will create an account.

3.2 The system must allow the user to sign in using their existing account.

3.2.1 The user must click on the input text box.

3.2.2 The user must input their email address or username in the top input text box

that was used in registration.

3.2.3 The user must input their password in the bottom input text box that was used

in registration.

3.2.3.1 The system must display an icon inside the input text box that allows

the user to mask their password input.

3.2.3.1.1 By default, the icon is unchecked, where the system must

mask the password when typed by the user.

3.2.3.1.2 If the icon is pressed by the user, the system must not mask

the password input.

3.3 The system must authenticate the email and password entered by the User.

3.3.1 If the login information cannot be ascertained, the system must display an error

message to the user that prompts them to input their login credentials again.

3.3.2 If the login information has been ascertained, the system must allow the user to

log in to the website.

3.4 The system must display the option “Forgot Password?” for the users who have forgotten

their password.

3.4.1 Upon clicking on the option, the user must input the email address used for the

account in the input text box.

3.4.2 The system must verify the authenticity of the email address.

3.4.2.1 If the email address is not found in the account database, the system

must display an error message “Invalid User Email!” that prompts the

user to input a different address.

3.4.2.2 If the email address is found in the account database, the user will

receive an email from the system allowing them to reset their

password.

## Home Page

**4.4.1 Description and Priority**

The Home Page shows all the upcoming meeting details from the groups that the user has joined.

| **Overall Priority** | Medium |
| --- | --- |

**4.4.2 Stimulus/Response Sequences**

| **Stimulus** | **Response** |
| --- | --- |
| 1. User signs in successfully | User can view Home Page |
| 1. User clicks on “Let’s Makan Together’ Logo | User can view Home Page |

**4.4.3 Functional Requirements**

4.1 Upon signing in, the user must be directed to the home page.

4.2 The system must display the details of upcoming meetings from the groups that the user

has joined.

4.2.1 If the user did not join any groups, the system must display “There are no

upcoming meetings”.

4.2.2 If none of the groups have upcoming meetings, the system must display “There

are no upcoming meetings”.

4.3 The system must display an option that allows the user to sign out.

## Profile Page

**4.5.1 Description and Priority**

Users can view and edit their own profile page. The editable fields include Name, Username, Email, Age, Gender, Food Preferences, and Regional Preferences.

| **Overall Priority** | Medium |
| --- | --- |

**4.5.2 Stimulus/Response Sequences**

| **Stimulus** | **Response** |
| --- | --- |
| 1. User clicks “My Profile” Button | The system will display all the user’s  profile details |
| 1. User clicks “Edit” Button | The system will display a form for the  user to change any user’s details. |
| 1. User clicks “Change Password” Button | The system will display a form for the  user to change password. |

**4.5.3 Functional Requirements**

5.1 The system must display the user’s

5.1.1 name

5.1.2 username

5.1.3 email

5.1.4 age

5.1.5 gender

5.2 The system shall display the user’s

5.2.1 food preference(s)

5.2.2 regional preference(s)

5.3 The system must allow the user to edit the aforementioned items.

5.3.1 The system must display an option to confirm changes.

5.3.2 The system must display the saved edits of the profile correctly.

5.4 The system must display an option for the user to change password.

5.4.1 The system must display three input text boxes:

5.4.1.1 one for the user to input their current password

5.4.1.2 two for the user to enter a new password

5.4.2 The system must display an option to confirm changes.

5.4.2.1 The system must verify that the new password in both text boxes

match.

5.4.2.2 The system must ensure that the password length is 12 characters or

more.

5.4.2.2.1 If the password length is not met, the system must display an

error message that prompts the user to input a new password.

5.4.2.3 The system must ensure that the password is alphanumeric.

5.4.2.3.1 If the password input is not alphanumeric, the system must

display an error message that prompts the user to input a

new password.

## Group Profile Page

**4.6.1 Description and Priority**

Users can create groups and edit the group profile page that they are the owner of. The editable fields include Group Name, Food Preferences, Regional Preferences, and Group Description. Users can view the groups that they have joined, and also choose to leave a group.

| **Overall Priority** | Medium |
| --- | --- |

**4.6.2 Stimulus/Response Sequences**

| **Stimulus** | **Response** |
| --- | --- |
| 1. User accesses the view to see all the groups they are a part of | The system displays a list of all  groups that the user has currently  joined |
| 1. User clicks “+” button | Create group form will be displayed |
| 1. User inputs create group fields accordingly and clicks ‘tick’ button to confirm creation | System creates the group and displays  the newly created group in “View My  Groups” |
| 1. User selects a specific group from the list | The system displays all the group  details with panes that lead to “Group  Chat” and “Next Meeting” |
| 1. Group Owner clicks “Edit” button | Edit group form will be displayed |
| 1. Group owner selects the option to delete the group | The system prompts for confirmation  and upon receiving it, deletes the  group and notifies the owner of the  successful deletion |
| 1. Non-owner user selects the option to leave the group | The system prompts for confirmation  and upon receiving it, removes the  user from the group and updates the  member count accordingly |

**4.6.3 Functional Requirements**

6.1 The system must display all the groups that the user has currently joined.

6.2 In each group page, the system must display the group’s

6.2.1 name

6.2.2 picture

6.2.3 member count

6.2.4 description

6.2.5 food preference(s)

6.2.6 regional preference(s)

6.2.7 current group members’ name

6.2.8 past meetings

6.3 The system must allow the user to create a new group.

6.3.1 The user must enter a name for the group.

6.3.1.1 The system must ensure that the name of the group is 100 Unicode

characters long or less.

6.3.2 The user must enter the description of the group.

6.3.2.1 The system must ensure that the description of the group is 3000

Unicode characters long or less.

6.3.3 The user must select one or more food preferences of the group.

6.3.4 The user must select one or more regional preferences of the group.

6.3.5 The user must select a picture for the group’s profile picture.

6.4 The system must allow only the owner of the group to edit the aforementioned details.

6.5 If the user is the group owner, the system must allow the user to delete the group.

6.6 If the user is not the group owner, the system must allow the user to leave the group.

## Group Chat Page

**4.7.1 Description and Priority**

Users are allowed to chat in groups that they have joined. Users can discuss and decide on the upcoming meeting details, so that they are able to meet-up and enjoy food together.

| **Overall Priority** | Medium |
| --- | --- |

**4.7.2 Stimulus/Response Sequences**

| **Stimulus** | **Response** |
| --- | --- |
| 1. User clicks “Group Chat” pane | Users can view Group Chat with all the  messages |
| 1. User inputs text messages and press enter | The system receives a new message  from the user and displays the new  message |

**4.7.3 Functional Requirements**

7.1 All groups will have a group chat automatically created for all group members.

7.2 The system must display all the messages sent by members of the group.

7.3 The system must display the member’s name and username for each message.

7.4 The system must display the date in “dd LLL yyyy” format for messages sent on a

particular day.

7.5 The system must display the time in HH:MM 24-hour format for each message.

7.6 The system must allow the user to send a text message.

7.6.1 The system must ensure that the text message contains 3000 Unicode

characters or less.

## View Meeting Page

**4.8.1 Description and Priority**

Users can view the meeting details, which include the date, time and location of the upcoming meeting. Group owners can edit the meeting details. The editable fields include date, time and location.

| **Overall Priority** | High |
| --- | --- |

**4.8.2 Stimulus/Response Sequences**

| **Stimulus** | **Response** |
| --- | --- |
| 1. Group owner attempts to set the next meeting for the group | The system allows the group owner to  access the next meeting setup  functionalities |
| 1. User enters a search query for a food location | The system searches the directory and  displays a list of matching food  locations |
| 1. User selects a food location from the search results | The system displays the selected  location on Google Maps using the  Google Maps API |
| 1. User clicks on the marker representing a food location on Google Maps | The system highlights the marker and  prepares to display detailed information  about the location |
| 1. Confirmation button is clicked by the user | The system saves the food location's  name and address for meeting setup |
| 1. User accesses the date selection functionality | The system presents an interface to  select a date for the meeting |
| 1. User clicks “Save” Button | The system will save the meeting details  in the database and display the meeting  details to every user in the group. |

**4.8.3 Functional Requirements**

8.1 The system must allow only the owner of the group to set a next meeting for the group.

8.1.1 The user must find the directory of the food location through a search query.

8.1.1.1 The system must display the queried location correctly on Google

Maps using Google Maps API.

8.1.1.2 The system must allow the user to click on the marker on the map.

8.1.1.2.1 Upon clicking on the marker, the system must display the

location’s

8.1.1.2.1.1 name

8.1.1.2.1.2 address

8.1.1.2.1.3 ratings

8.1.1.2.1.4 contact

8.1.1.2.1.5 website

8.1.1.2.2 The system must display a confirmation button.

8.1.1.2.2.1 Upon clicking on the button, the system must save

the location’s name and address.

8.1.2 The user must select a date for the meeting.

8.1.3 The user must select a time for the meeting.

8.1.3.1 The system must ensure that the date and time selected must be in the

future.

8.2 The system must display the meeting’s

8.2.1 date

8.2.2 time

8.2.3 location name

8.2.4 location on Google Maps

8.2.4.1 The system must display the food location on Google Maps correctly.

8.2.4.1.1 The system must allow the user to click on the marker on the

map.

8.2.4.1.1.1 Upon clicking on the marker, the system must

display the location’s

8.2.4.1.1.1.1 name

8.2.4.1.1.1.2 address

8.2.4.1.1.1.3 ratings

8.2.4.1.1.1.4 contact

8.2.4.1.1.1.5 website

8.2.4.2 The user shall select click on the website.

8.2.4.2.1 The system must redirect the user to Google Maps which

allows the user to find more information, e.g. directions to

the location.

8.3 If the group does not have any upcoming meetings, the system must display “There are

no upcoming meetings”.

## Group Finding Page

**4.9.1 Description and Priority**

Users can search for groups based on their food and regional preferences or by group name. Users can choose to join a group that aligns with their food interests.

| **Overall Priority** | High |
| --- | --- |

**4.9.2 Stimulus/Response Sequences**

| **Stimulus** | **Response** |
| --- | --- |
| 1. User clicks “Find group” tab in the navigation bar | System displays “Group Finding Page” |
| 1. User clicks on ‘Search’ icon | System displays joinable groups  relevant to the preference filters. |

**4.9.3 Functional Requirements**

9.1 The system must allow the user to search for groups through a search query.

9.2 The system must allow the user to filter the search results.

9.2.1 The system must filter with food preferences.

9.2.2 By default, the system must use the user’s current food preferences from their

profile to filter.

9.3 The system must display the groups correctly after a search has been queried by a user.

9.3.1 The system must ensure that the following group details of each group are

displayed correctly:

9.3.1.1 Name

9.3.1.2 Description

9.3.1.3 Region Preference

9.3.1.4 Food preference(s)

## Food Location Page

**4.10.1 Description and Priority**

Users can search for food locations based on their food preferences. The LetsMakanTogether web application displays a map with food places that users can select to view details of the food directory. It also provides users with food recommendations based on what the user has searched for.

| **Overall Priority** | High |
| --- | --- |

**4.10.2 Stimulus/Response Sequences**

| **Stimulus** | **Response** |
| --- | --- |
| 1. Click “Find Food” button in navigation bar | View Find Food Page |
| 1. Click “Recommended Food” button | The website displays Food Preference  Options. |
| 1. Enter search query into “Search Food” search bar | The website displays different  landmarks corresponding to the search  query on Google Maps. |
| 1. Click “Get Current Location” button | The website displays user’s current  location on Google Maps. |
| 1. Click a Food Location Marker | The website displays a side panel  consisting the food location’s name,  address, ratings, contact and website. |

**4.10.3 Functional Requirements**

10.1 The system must display a button “Get Current Location”.

10.1.1 Upon clicking on the button, the system must check if the user has given

Global Positioning System (GPS) permission.

10.1.1.1 If GPS permission is not given yet, the system must prompt the user

to allow GPS permission to be given.

10.1.1.2 If the user disallows GPS permission once, the website must not

prompt the user again.

10.1.2 The system must display the Google Maps with the user’s current location

using Google Maps API.

10.2 The system must allow the user to search for a location through a search query.

10.2.1 The system must display the Google Maps with the queried location using

Google Maps API.

10.3 The system must allow the user to select recommended food options.

10.3.1 The system must display all the recommended food locations using Tourism

Info Hub API.

10.4 Upon clicking on a marker, the system must display the location’s

10.4.1 name

10.4.2 address

10.4.3 ratings

10.4.4 contact

10.4.5 website

# **Other Nonfunctional Requirements**

## Usability

5.1.1 The website and UI must have features to help users navigate for the first time.

5.1.1.1 The system must have a guided walkthrough tutorial.

5.1.2 The UI shall be organized with no visual clutters.

5.1.3 Features such as finding individuals, selecting food preferences, and finding food areas must

be easily accessible.

5.1.4 The text in the website and UI must be easy to read at 100% zoom.

5.1.5 The system must provide a simple and intuitive registration and login process.

## Reliability

5.2.1 The system must perform without failure 95% of the time.

5.2.2.1 The system must recover from any failures within 3 minutes.

5.2.2 The system must backup all the data stored in the cloud when it receives new data.

5.2.3 The system must provide reliable location data using Google Maps API.

5.2.4 The recommendation system’s suggestions must be accurate, reliable and consistent.

5.2.5 The system must ensure reliable group chat interactions, with maximum uptime and minimal

lag.

## Performance

5.3.1 The website must load completely in 5 seconds or less.

5.3.2 The website must load the location map in 10 seconds or less.

5.3.2.1 The user’s location should be accurate within a maximum error of 50m.

5.3.2.2 Real-time update of information, such as food areas, groups and directions must be

ensured.

5.3.3 The system must be able to respond to actions within 1s.

5.3.4 The system must be able to support up to 10,000 users simultaneously without slowing down

or compromising the user experience.

## Supportability

5.4.1 The website must be compatible across multiple web browsers (Chromium-based, Firefox,

Edge, Safari)

5.4.2 The website must provide information and helpful error messages for users to diagnose the

problem for technical support.

5.4.3 The website should include a FAQ or help portal to handle usual queries of users quickly.

## Security

5.5.1 The website must use secure HTTPS protocols and encrypted databases.

5.5.2 The Login and Account system should have added features like 2-factor authentication for

enhanced safety.

5.5.3 The system must protect user’s personal data from unauthorized access during their use of

group chat and public groups.

# Testing

## Black Box Testing

***Test Case 1: Login Function***

| **Test Scenario** | Login into the system | **Test Case ID** | Login |
| --- | --- | --- | --- |
| **Test case description** | Invalid inputs results in unsuccessful login | **Test Priority** | High |
| **Prerequisite** | Email & Username should have been a registered entry in the system | | |

| **#** | **Description** | **Action** | **Inputs** | **Expected Output** | **Actual Output** | **Test Result** |
| --- | --- | --- | --- | --- | --- | --- |
| 1. | User manage to log in successfully using email | 1. On any page, users press on “Sign In”. 2. In the Sign In Page, users type in his/her email and password. 3. Users press the “Login” button. | **email:** sanrio\_cinnamon@gmail.com  **password:**  correct\_password123 | Users will be redirected to the “Dashboard”Page | Users are redirected to the “Dashboard” Page | Pass |
| 2. | User typed in the wrong password with correct email | 1. On any page, users press on “Sign In”. 2. In the Sign In Page, users type in his/her email and password. 3. Users press the “Login” button. | **email:** sanrio\_cinnamon@gmail.com  **password:**  incorrect\_password123 | An error message will be displayed that says “Wrong Username or Email!” | An error message is displayed that says “Wrong Username or Email!” | Pass |
| 3. | Users manage to log in successfully using username | 1. On any page, users press on “Sign In”. 2. In the Sign In Page, users type in his/her username and password. 3. Users press the “Login” button. | **username:** sanrio123  **password:**  correct\_password123 | Users will be redirected to the “Dashboard”Page | Users are redirected to the “Dashboard” Page | Pass |
| 4. | User typed in the wrong password with correct username | 1. On any page, users press on “Sign In”. 2. In the Sign In Page, users type in his/her username and password. 3. Users press the “Login” button. | **email:** sanrio123  **password:**  incorrect\_password123 | An error message will be displayed that says “Wrong Username or Email!” | An error message is displayed that says “Wrong Username or Email!” | Pass |
| 5. | Users typed in the wrong email address with correct password | 1. On any page, users press on “Sign In”. 2. In the Sign In Page, users type in his/her email and password. 3. Users press the “Login” button. | **email:** incorrect\_email@gmail.com  **password:**  correct\_password123 | An error message will be displayed that says “Wrong Username or Email!” | An error message is displayed that says “Wrong Username or Email!” | Pass |
| 6. | Users typed in the wrong username with correct password | 1. On any page, users press on “Sign In”. 2. In the Sign In Page, users type in his/her username and password. 3. Users press the “Login” button. | **username:**  sanrio231  **password:**  correct\_password123 | An error message will be displayed that says “Wrong Username or Email!” | An error message is displayed that says “Wrong Username or Email!” | Pass |
| 7. | User typed in the wrong username/email and password | 1. On any page, users press on “Sign In”. 2. In the Sign In Page, users type in his/her username and password. 3. Users press the “Login” button. | **username:**  wrong\_name  **password:** wrong\_password | An error message will be displayed that says “Wrong Username or Email!” | An error message is displayed that says “Wrong Username or Email!” |  |
| 8. | There is missing field for any of the inputs | 1. On any page, users press on “Sign In”. 2. Users press the “Login” button. | - | Users are not allowed to press “Login” | User is not allowed to press “Login” | Pass |

***Test Case 2: Register Function***

| **Test Scenario** | Register for an account | **Test Case ID** | register |
| --- | --- | --- | --- |
| **Test case description** | Invalid inputs results in unsuccessful registration | **Test Priority** | High |
| **Prerequisite** | Email & Username should not have been a registered entry in the system | | |

| **#** | **Description** | **Action** | **Inputs** | **Expected Output** | **Actual Output** | **Test Result** |
| --- | --- | --- | --- | --- | --- | --- |
| 1. | Users manage to register successfully | 1. On the login page, Users press on “Register”. 2. In the Register Page, users fill up the inputs accordingly - display name, email, username, password, confirm password, age and gender 3. Users press the “Register” button | **display name:** tester  **email:** stump@gmail.com  **username:** stump  **password:** correct\_password123  **confirm password:** correct\_password123  **age:** 19  **gender:** Male | A message popup will be shown to the user that says “Registration Successful” and redirects user to login page | A message popup is shown to the user that says “Registration Successful” and it correctly redirects user to login page | Pass |
| 2. | Users password does not match | 1. On the login page, Users press on “Register”. 2. In the Register Page, users fill up the inputs accordingly - display name, email, username, password, confirm password, age and gender 3. Users press the “Register” button | **display name:** tester  **email:** stump@gmail.com  **username:** stump  **password:** correct\_password123  **confirm password:** different\_password123  **age:** 19  **gender:** Male | An error message will show below the confirm password text field that says “Password is not the same as re-entered password!”. The register button is also not clickable if password do not match | An error message showed below the confirm password text field that says: “Password is not the same as re-entered password!”  The register button is not clickable when passwords do not match. | Pass |
| 3. | User type in an invalid email | 1. On the login page, Users press on “Register”. 2. In the Register Page, users fill up the inputs accordingly - display name, email, username, password, confirm password, age and gender 3. Users press the “Register” button | **display name:** tester  **email:** stump  **username:** stump  **password:** correct\_password123  **confirm password:** correct\_password123  **age:** 19  **gender:** Male | An error message will popup that says “Invalid email format entered” | An error message popped up that says “Invalid email format” entered | Pass |
| 4. | Email has been registered before | 1. On the login page, Users press on “Register”. 2. In the Register Page, users fill up the inputs accordingly - display name, email, username, password, confirm password, age and gender 3. Users press the “Register” button | **display name:** tester  **email:** sanrio\_cinnamon@gmail.com  **username:** stump  **password:** correct\_password123  **confirm password:** correct\_password123  **age:** 19  **gender:** Male | An error message will popup that says “Email has already been registered before” | An error message shows up to the user that says “Email has already been registered before” | Pass |
| 5. | Username has been registered before | 1. On the login page, Users press on “Register”. 2. In the Register Page, users fill up the inputs accordingly - display name, email, username, password, confirm password, age and gender 3. Users press the “Register” button | **display name:** tester  **email:** stump@gmail.com  **username:** sanrio123  **password:** correct\_password123  **confirm password:** correct\_password123  **age:** 19  **gender:** Male | An error message will popup that says “Username has already been registered before” | An error message shows up to the user that says “Username has already been registered before” | Pass |
| 6. | Password length is less than 12 characters | 1. On the login page, Users press on “Register”. 2. In the Register Page, users fill up the inputs accordingly - display name, email, username, password, confirm password, age and gender 3. Users press the “Register” button | **display name:** tester  **email:** stump@gmail.com  **username:** stump  **password:** 123  **confirm password:** 123  **age:** 19  **gender:** Male | An error message will show below password text field that says “Password must be at least 12 characters long” | An error message shows up at the password text field that says “Password must be at least 12 characters long” | Pass |
| 7. | Password does not contain alphanumeric characters | 1. On the login page, Users press on “Register”. 2. In the Register Page, users fill up the inputs accordingly - display name, email, username, password, confirm password, age and gender 3. Users press the “Register” button | **display name:** tester  **email:** stump@gmail.com  **username:** stump  **password:** prostumperasdefg  **confirm password:** prostumperasdefg  **age:** 19  **gender:** Male | An error message will show below password text field that says “Password must contain at least one letter and one number” | An error message shows below the password text field that says “Password must contain at least one letter and one number” | Pass |
| 8. | Age is less than 18 | 1. On the login page, Users press on “Register”. 2. In the Register Page, users fill up the inputs accordingly - display name, email, username, password, confirm password, age and gender 3. Users press the “Register” button | **display name:** tester  **email:** stump@gmail.com  **username:** stump  **password:** correct\_password123  **confirm password:** correct\_password123  **age:** 17  **gender:** Male | User will not be allowed to register | User cannot register to with current fields entered | Pass |
| 9. | User does not fill in any inputs (i.e did not enter anything in any fields) | 1. On the login page, Users press on “Register”. 2. Users press the “Register” button | - | Users will not be allowed to click on the register button | User cannot press the register button | Pass |

***Test Case 3: Reset Password Function***

| **Test Scenario** | Reset Password | **Test Case ID** | forget\_password |
| --- | --- | --- | --- |
| **Test case description** | Invalid inputs results in not being able to reset password | **Test Priority** | Medium |
| **Prerequisite** | Email should be a registered entry in the system | | |

| **#** | **Description** | **Action** | **Inputs** | **Expected Output** | **Actual Output** | **Test Result** |
| --- | --- | --- | --- | --- | --- | --- |
| 1. | User managed to successfully send a request password change token to their email | 1. On the login page, the user clicks on “Forget Password” option 2. In the Forget Password Page, User enters their email they registered with 3. User press the tick to send a password change | **email:** sanrio\_cinnamon@gmail.com | A popup message will show “Reset password email sent.” to the user | The popup message that says “Reset password email sent.” is shown to the user | Pass |
| 2. | User enters an invalid email format | 1. On the login page, the user clicks on “Forget Password” option 2. In the Forget Password Page, User enters their email they registered with 3. User press the tick to send a password change | **email:** stump | A popup message will show that says “Invalid user email” | The popup message showed to the user that says “Invalid User Email” | Pass |
| 3. | User enters an email that is not registered before | 1. On the login page, the user clicks on “Forget Password” option 2. In the Forget Password Page, User enters their email they registered with 3. User press the tick to send a password change | **email:**  abc@gmail.com | A popup message will show that says “Invalid user email” | The popup message showed to the user that says “Invalid User Email” | Pass |
| 4. | User does not enter anything on the email text field | 1. On the login page, the user clicks on “Forget Password” option 2. User tries to press the tick to send a password change | - | The user will not be allowed to send a password change request | The user is not allowed to send a password change request | Pass |

***Test Case 4: Find Food Function***

| **Test Scenario** | Find Food | **Test Case ID** | find\_food |
| --- | --- | --- | --- |
| **Test case description** | Invalid inputs results in unsuccessful search | **Test Priority** | High |
| **Prerequisite** | Location should be valid and within Singapore | | |

| **#** | **Description** | **Action** | **Inputs** | **Expected Output** | **Actual Output** | **Test Result** |
| --- | --- | --- | --- | --- | --- | --- |
| 1. | Users manage to search and display markers on the map successfully | 1. In the Find Food page, users are able to enter food places to search up on the embedded Google Maps API. 2. Users fills up the place of interest in “Search Food” text field and press “Enter” | GPS Permission: Allowed/Denied  Search Food input:  Chicken Rice | User will be shown nearby markers of food places that sells “Chicken Rice” based on the position centered on the map | User is able to see the markers on the map that are marked nearby of the current position centered on the map | Pass |
| 2. | User successfully gets current location on Google Maps API | 1. In Find Food page, users are able to get its current location to display on the map 2. Users clicks on “Get Current Location📍” 3. Users press “Allow” on prompt of website reading current location | GPS Permission: Allowed | User’s current location will be shown on the Google Maps API with a blue marker | User’s current location is shown on the Google Maps API with a blue marker | Pass |
| 3. | User unsuccessfully gets current location on Google Maps API | 1. In the Find Food page, users are able to get its current location to display on the map 2. Users clicks on “Get Current Location📍” 3. Users press “Block” on prompt of website reading current location | GPS Permission:  Denied | User’s current location is not shown on the map. An error message popup saying “Location Permission Denied” | User’s current location is not shown on the map. The error message popup saying “Location Permission Denied” is shown | Pass |
| 4. | User does not input an alphanumeric address in the search text box | 1. In the Find Food page, users are able to fill up the 2. Users fills up the place of interest with non-alphanumeric address and press “Enter” | GPS Permission: Allowed/Denied  Search Food Input: 🍕🍔🍟🌭🍿🧂🥓🥚 | Users will not be able to find results in the Google Maps API.  No markers will be displayed. | Users do not find any results in the Google Maps API.  No markers is displayed | Pass |
| 5. | User select one of the option for “Recommend Food 🍔” | 1. In the Find Food page, users are able to select one of the drop down options in “Recommend Food 🍔” 2. Users selects “Indian Cuisine” | GPS Permission:  Allowed/Denied | Users will be able to see purple markers marked on the map | Users are able to see the purple markers on the map | Pass |
| 6. | User selects on “Select Cuisine” option for “Recommend Food 🍔” | 1. In Find Food page, users are able to select one of the drop down options in “Recommend Food 🍔” 2. Users selects “Select Cuisine” | GPS Permission:  Allowed/Denied | Markers previously made on the map using “Recommend Food 🍔” will be cleared | Markers that is made previously on the previous option of “Recommend Food 🍔” are cleared | Pass |

***Test Case 5: Change Password Function***

| **Test Scenario** | Change Password | **Test Case ID** | change\_password |
| --- | --- | --- | --- |
| **Test case description** | Invalid inputs result in unsuccessful changing of password. | **Test Priority** | Medium |
| **Prerequisite** | Email or Username should have been registered before | | |

| **#** | **Description** | **Action** | **Inputs** | **Expected Output** | **Actual Output** | **Test Result** |
| --- | --- | --- | --- | --- | --- | --- |
| 1. | Users manage to update password successfully | 1. In My Profile Page, users clicks on “Change Password” 2. In Change Password page, users fill up the inputs accordingly - Current Password, New Password, Confirm New Password 3. User clicks on the “tick” button to change password | current password: 1234567891011a  new password:  correct\_password123  confirm password:  correct\_password123 | User will be shown a popup message that says that “Password Successfully Updated” and redirected to Edit Profile Page | User is shown a popup message that says “Password Successfully Updated” and is redirected to Edit Profile Page | Pass |
| 2. | User’s current password input does not match password in the database | 1. In My Profile Page, users clicks on “Change Password” 2. In Change Password page, users fill up the inputs accordingly - Current Password, New Password, Confirm New Password 3. User clicks on the “tick” button to change password | current password: wrong\_password123  new password: new\_password123  confirm password:  new\_password123 | User will be shown a popup message that says “Please enter the correct current password” | User is shown a popup message that says “Please enter the correct current password” | Pass |
| 3. | User’s new password input is the same as old password | 1. In My Profile Page, users clicks on “Change Password” 2. In Change Password page, users fill up the inputs accordingly - Current Password, New Password, Confirm New Password 3. User clicks on the “tick” button to change password | current password: 1234567891011a  new password: 1234567891011a  confirm password:1234567891011a | User will be shown a popup that says “New Password cannot be the same as Current Password” | User is shown a popup message that says “New Password cannot be the same as Current Password” | Pass |
| 4. | User’s new password is less than 12 characters long | 1. In My Profile Page, users clicks on “Change Password” 2. In Change Password page, users fill up the inputs accordingly - Current Password, New Password, Confirm New Password 3. User clicks on the “tick” button to change password | current password: 1234567891011a  new password: abc123  confirm password:  abc123 | User will be shown a popup text message below the textbox that says “Password has to be at least 12 characters long” | User is shown a popup message that says “Password has to be at least 12 characters long” | Pass |
| 5. | User’s new password does not contain alphanumeric characters | 1. In My Profile Page, users clicks on “Change Password” 2. In Change Password page, users fill up the inputs accordingly - Current Password, New Password, Confirm New Password 3. User clicks on the “tick” button to change password | current password: 1234567891011a  new password: abcdefghijklmnop  confirm password:  abcdefghijklmnop | User will be shown a popup message that says “Password must contain at least one letter and one number” | User is shown a popup message that says “Password must contain at least one letter and one number” | Pass |
| 6. | User have missing inputs (i.e did not fill in any of the fields) | 1. In My Profile Page, users clicks on “Change Password” 2. User tries to clicks on the “tick” button to change password | - | User will not be able to click on the “tick” button to submit the form | User is not able to click on the “tick” button to submit the form | Pass |

***Test Case 6: Edit Profile Function***

| **Test Scenario** | Edit Profile | **Test Case ID** | edit\_profile |
| --- | --- | --- | --- |
| **Test case description** | Invalid inputs result in unsuccessful changing of profile details | **Test Priority** | Medium |
| **Prerequisite** | User needs to be logged in to edit profile | | |

| **#** | **Description** | **Action** | **Inputs** | **Expected Output** | **Actual Output** | **Test Result** |
| --- | --- | --- | --- | --- | --- | --- |
| 1. | User manage to edit their profile successfully | 1. On any page upon logging in, user clicks on My Profile and then clicks on the “pencil” icon 2. User fill in the necessary details to edit their profile 3. User clicks on “tick” to confirm their changes | **Username:** FreddyTam  **Email:**  sanrio\_cinnamon@gmail.com  **Display Name:**  Freddy  **Age:**  18  **Gender:**  Female  **Region Preferences:** North  **Food Preferences:**  Chinese Cuisine | Users new profile details will be reflected on the Edit Profile page | Users new profile details is reflected on the Edit Profile page | Pass |
| 2. | Users new username is an existing username in the system | 1. On any page upon logging in, user clicks on My Profile and then clicks on the “pencil” icon 2. User fill in the necessary details to edit their profile 3. User clicks on “tick” to confirm their changes | **Username:** existinguser  **Email:**  sanrio\_cinnamon@gmail.com  **Display Name:**  Cinnamon  **Age:**  18  **Gender:**  Female  **Region Preferences:** North  **Food Preferences:**  Chinese Cuisine | Users will be shown a popup error message that says “A user with the given username or email already exists!” | User is shown a popup error message that says “A user with the given username or email already exists!” | Pass |
| 3. | Users new email is an existing email in the system | 1. On any page upon logging in, user clicks on My Profile and then clicks on the “pencil” icon 2. User fill in the necessary details to edit their profile 3. User clicks on “tick” to confirm their changes | **Username:** Sanrio123  **Email:**  existing\_email@gmail.com  **Display Name:**  Cinnamon  **Age:**  18  **Gender:**  Female  **Region Preferences:** North  **Food Preferences:**  Chinese Cuisine | Users will be shown a popup error message that says “A user with the given username or email already exists!” | User is shown a popup error message that says “A user with the given username or email already exists!” | Pass |
| 4. | Users new email entered is in an invalid format | 1. On any page upon logging in, user clicks on My Profile and then clicks on the “pencil” icon 2. User fill in the necessary details to edit their profile 3. User clicks on “tick” to confirm their changes | **Username:** Sanrio123  **Email:**  new\_email  **Display Name:**  Cinnamon  **Age:**  18  **Gender:**  Female  **Region Preferences:** North  **Food Preferences:**  Chinese Cuisine | Users will be shown a popup error message that says “Email is invalid!” | User is shown a popup error message that says “Email is invalid!” | Pass |
| 5. | Users age updated their age to 18 | 1. On any page upon logging in, user clicks on My Profile and then clicks on the “pencil” icon 2. User fill in the necessary details to edit their profile 3. User clicks on “tick” to confirm their changes | **Username:** Sanrio123  **Email:**  sanrio\_cinnamon@gmail.com  **Display Name:**  Cinnamon  **Age:**  15  **Gender:**  Female  **Region Preferences:** North  **Food Preferences:**  Chinese Cuisine | Users will be shown a popup error message that says “Age should be above 18” | User is shown a popup error message that says “Age should be above 18” | Pass |
| 6. | User did not enter any new details to submit | 1. On any page upon logging in, user clicks on My Profile and then clicks on the “pencil” icon 2. User clicks on “tick” to confirm their changes | - | User will be redirect back to the edit profile page with no changes made | User is redirected back to the edit profile page with no changes made | Pass |

***Test Case 7: Create Group Function***

| **Test Scenario** | Create Group | **Test Case ID** | create\_group |
| --- | --- | --- | --- |
| **Test case description** | Invalid inputs result in unsuccessful creation of groups | **Test Priority** | High |
| **Prerequisite** | User needs to be logged in to create a group | | |

| **#** | **Description** | **Action** | **Inputs** | **Expected Output** | **Actual Output** | **Test Result** |
| --- | --- | --- | --- | --- | --- | --- |
| 1. | User manage to create a group successfully | 1. In the My Groups Page, users press on the “+” button. 2. A popup will show that details a form for user to fill in the details to create a group 3. User fills in the form 4. User press “tick” button to create a group | **Group Name:** Chinese Food Lovers  **Group Description:**  Join if you love chinese food!  **Food Preference:**  Chinese Cuisine  **Regional Preference:**  West | Users will be shown a popup message that says “Group successfully created!” | User is shown a popup message that says “Group successfully created!” | Pass |
| 2. | User did not enter the group name | 1. In the My Groups Page, users press on the “+” button. 2. A popup will show that details a form for user to fill in the details to create a group 3. User fills in the form without the group name 4. User press “tick” button to create a group | **Group Name:**  **Group Description:**  Join if you love chinese food!  **Food Preference:**  Chinese Cuisine  **Regional Preference:**  West | Users will be shown a popup text message below the Group Name field saying “Group Name is required” | User is shown a popup message that says “Group Name is required” below the Group Name Field | Pass |
| 3. | User did not enter the group description | 1. In the My Groups Page, users press on the “+” button. 2. A popup will show that details a form for user to fill in the details to create a group 3. User fills in the form without the group description 4. User press “tick” button to create a group | **Group Name:** Chinese Food Lovers  **Group Description:**  **Food Preference:**  Chinese Cuisine  **Regional Preference:**  West | Users will be shown a popup text message below the Group Description field saying “Description is required” | User is shown a popup message that says “Description is required” below the text field of Group Description | Pass |
| 4. | User did not choose a food preference | 1. In the My Groups Page, users press on the “+” button. 2. A popup will show that details a form for user to fill in the details to create a group 3. User fills in the form without selecting a food preference 4. User press “tick” button to create a group | **Group Name:** Chinese Food Lovers  **Group Description:**  Join if you love chinese food!  **Food Preference:**  **Regional Preference:**  West | Users will not be allowed to click on the “tick” button as food preferences is not selected yet | Users is not allowed to click on the “tick” button to create the group | Pass |
| 5. | User did not choose a regional preference | 1. In the My Groups Page, users press on the “+” button. 2. A popup will show that details a form for user to fill in the details to create a group 3. User fills in the form without selecting a regional preference 4. User press “tick” button to create a group | **Group Name:** Chinese Food Lovers  **Group Description:**  Join if you love chinese food!  **Food Preference:**  Chinese Cuisine  **Regional Preference:** | Users will not be allowed to click on the “tick” button as regional preferences is not selected yet | Users is not allowed to click on the “tick” button to create the group | Pass |
| 6. | User did not fill in any fields in the form | 1. In the My Groups Page, users press on the “+” button. 2. A popup will show that details a form for user to fill in the details to create a group 3. User press “tick” button to create a group | - | Users will not be allowed to click on the “tick” button as none of the fields are filled up yet | Users is not allowed to click on the “tick” button to create the group | Pass |

***Test Case 8: Create Next Meeting Function***

| **Test Scenario** | Create Next Meeting | **Test Case ID** | create\_next\_meeting |
| --- | --- | --- | --- |
| **Test case description** | Invalid inputs result in unsuccessful creation of next meeting | **Test Priority** | High |
| **Prerequisite** | User needs to be an owner of a group to create a meeting | | |

| **#** | **Description** | **Action** | **Inputs** | **Expected Output** | **Actual Output** | **Test Result** |
| --- | --- | --- | --- | --- | --- | --- |
| 1. | Users are able to create a next meeting for a group they have created | 1. In the My Groups Page, Users selects on a group that they have created 2. Users click Next Meeting to go to the View Next Meeting Page 3. Users click on the “Create Meeting” button to create meeting 4. User fills in the necessary fields needed to create next meeting | **Date and Time:**  2023-11-17 23:30:00  **Place:**  Popeyes  **Location:**  1 Jurong West Central 2, #03-48 Jurong Point, Singapore 648886  **Description:** I want to eat chicken | Users will be able to see a popup message that says “Successfully saved” | Users is shown a popup message that says “Successfully saved” | Pass |
| 2. | Users did not fill in the date and time for the meeting | 1. In the My Groups Page, Users selects on a group that they have created 2. Users click Next Meeting to go to the View Next Meeting Page 3. Users click on the “Create Meeting” button to create meeting 4. User fills in the necessary fields without the date to create next meeting | **Date and Time:**  **Place:**  Popeyes  **Location:**  1 Jurong West Central 2, #03-48 Jurong Point, Singapore 648886  **Description:** I want to eat chicken | Users will not be able to save the next meeting details | User is not able to save the next meeting details | Pass |
| 3. | Users did not fill in the description for the meeting | 1. In the My Groups Page, Users selects on a group that they have created 2. Users click Next Meeting to go to the View Next Meeting Page 3. Users click on the “Create Meeting” button to create meeting 4. User fills in the necessary fields without the description to create next meeting | **Date and Time:** 11 November 2024  **Place:**  Popeyes  **Location:**  1 Jurong West Central 2, #03-48 Jurong Point, Singapore 648886  **Description:** | Users will not be able to save the next meeting details | Users is not able to save the next meeting details | Pass |
| 4. | Users did not search a location for the next meeting | 1. In the My Groups Page, Users selects on a group that they have created 2. Users click Next Meeting to go to the View Next Meeting Page 3. Users click on the “Create Meeting” button to create meeting 4. User fills in the necessary fields without the description to create next meeting | **Date and Time:** 11 November 2024  **Place:**  **Location:**  **Description:**  I love chicken | Users will not be able to save the next meeting details | Users is not able to save the next meeting details | Pass |
| 5. | Users did not fill in any of the fields for the next meeting | 1. In the My Groups Page, Users selects on a group that they have created 2. Users click Next Meeting to go to the View Next Meeting Page 3. Users click on the “Create Meeting” button to create meeting | - | Users will not be able to save the next meeting details | User is not able to save the next meeting details | Pass |

***Test Case 9: Edit Next Meeting Function***

| **Test Scenario** | Edit Next Meeting | **Test Case ID** | edit\_next\_meeting |
| --- | --- | --- | --- |
| **Test case description** | Invalid inputs result in unsuccessful edits of next meeting | **Test Priority** | Medium |
| **Prerequisite** | User needs to be an owner of a group to edit a meeting | | |

| **#** | **Description** | **Action** | **Inputs** | **Expected Output** | **Actual Output** | **Test Result** |
| --- | --- | --- | --- | --- | --- | --- |
| 1. | Users are able to create a next meeting for a group they have created | 1. In the My Groups Page, Users selects on a group that they have created 2. Users click Next Meeting to go to the View Next Meeting Page 3. Users click on the “Create Meeting” button to create meeting 4. User fills in the necessary fields needed to create next meeting | **Date and Time:**  2023-11-17 23:30:00  **Place:**  Popeyes  **Location:**  1 Jurong West Central 2, #03-48 Jurong Point, Singapore 648886  **Description:** I want to eat chicken | Users will be able to see a popup message that says “Successfully saved” | Users is shown a popup message that says “Successfully saved” | Pass |
| 2. | Users did not fill in the date and time for the meeting | 1. In the My Groups Page, Users selects on a group that they have created 2. Users click Next Meeting to go to the View Next Meeting Page 3. Users click on the “Create Meeting” button to create meeting 4. User fills in the necessary fields without the date to create next meeting | **Date and Time:**  **Place:**  Popeyes  **Location:**  1 Jurong West Central 2, #03-48 Jurong Point, Singapore 648886  **Description:** I want to eat chicken | Users will not be able to save the next meeting details | User is not able to save the next meeting details | Pass |
| 3. | Users did not fill in the description for the meeting | 1. In the My Groups Page, Users selects on a group that they have created 2. Users click Next Meeting to go to the View Next Meeting Page 3. Users click on the “Create Meeting” button to create meeting 4. User fills in the necessary fields without the description to create next meeting | **Date and Time:** 11 November 2024  **Place:**  Popeyes  **Location:**  1 Jurong West Central 2, #03-48 Jurong Point, Singapore 648886  **Description:** | Users will not be able to save the next meeting details | Users is not able to save the next meeting details | Pass |
| 4. | Users did not fill in any of the fields for the next meeting | 1. In the My Groups Page, Users selects on a group that they have created  2. Users click Next Meeting to go to the View Next Meeting Page  3. Users click on the “Create Meeting” button to create meeting | - | Users will not be able to save the next meeting details | User is not able to save the next meeting details | Pass |

***Test Case 10: Join Group Function***

| **Test Scenario** | Join Group | **Test Case ID** | join\_group |
| --- | --- | --- | --- |
| **Test case description** | Joining the group you are already in results in error | **Test Priority** | High |
| **Prerequisite** | User needs to be logged in to join a group | | |

| **#** | **Description** | **Action** | **Inputs** | **Expected Output** | **Actual Output** | **Test Result** |
| --- | --- | --- | --- | --- | --- | --- |
| 1. | Users are able to join a group successfully | 1. In the Find Groups page, user is able to see a list of groups available 2. Users selects the join group icon in one of the group | - | Users will see a popup message that says “Successfully joined the group” | Users is shown a popup message that says “Successfully joined the group” | Pass |
| 2. | Users are not able to join a group they are already in | 1. In the Find Groups page, user is able to see a list of groups available 2. Users selects the join group icon in one of the group | - | Users will see a popup message that says “You are already in this group” | Users is shown a popup message that says “You are already in this group” | Pass |

***Test Case 11: Find Groups Function***

| **Test Scenario** | Find Groups | **Test Case ID** | find\_group |
| --- | --- | --- | --- |
| **Test case description** | Finding a group name that does not exists does not show the groups | **Test Priority** | Medium |
| **Prerequisite** | User needs to be logged in search for groups | | |

| **#** | **Description** | **Action** | **Inputs** | **Expected Output** | **Actual Output** | **Test Result** |
| --- | --- | --- | --- | --- | --- | --- |
| 1. | Users are able to find groups successfully | 1. In the Find Groups page, user is able to search for groups 2. Users search for a group based on the food preferences | **Group Name:**  **Food Preferences:** Organic | Users will be able to see a list of groups available based on the food preferences of the groups | Users is shown a list of groups that has Organic in its food preferences | Pass |
| 2. | Users search for groups without entering anything in the search box | 1. In the Find Groups page, user is able to search for groups 2. Users search for groups by not entering any fields | **Group Name:**  **Food Preferences:** | Users will be able to see all the list of available groups | User is shown all the available groups to join | Pass |
| 3. | Users search groups by searching for a specific group name | 1. In the Find Groups page, user is able to search for groups 2. Users search for groups by not entering any fields | **Group Name:** Grass  **Food Preferences:** | Users will be able to see the list of groups that has the group name that contains “Grass” | User is shown a list of groups that has the group name that contains “Grass” | Pass |
| 4. | Users enters a group name that does not exist | 1. In the Find Groups page, user is able to search for groups 2. Users search for groups by entering a group name | **Group Name:**  does not exist  **Food Preferences:** | Users will not be able to see any groups in the list | User is not shown any groups | Pass |
| 5. | Users selects a preference that any group is not in | 1. In the Find Groups page, user is able to search for groups 2. Users search for groups by selecting a preference that no group has | **Group Name:**  **Food Preferences:**  Fine Dining | Users will not be able to see any groups in the list | User is not shown any groups | Pass |

***Test Case 12: Edit Group Function***

| **Test Scenario** | Edit Group | **Test Case ID** | edit\_group |
| --- | --- | --- | --- |
| **Test case description** | Editing a group with invalid inputs results in unsuccessful editing | **Test Priority** | Medium |
| **Prerequisite** | User needs to be an owner of the group | | |

| **#** | **Description** | **Action** | **Inputs** | **Expected Output** | **Actual Output** | **Test Result** |
| --- | --- | --- | --- | --- | --- | --- |
| 1. | User manage to edit a group successfully | 1. In the My Groups Page, users press on the “pencil” button to edit a group. 2. A popup will show that details a form for user to edit the fields they want to make changes to 3. User fills in the form 4. User press “tick” button to edit the group | **Group Name:** Chinese Food  **Group Description:**  Join if you love chinese food!  **Food Preference:**  Chinese Cuisine, Vegan  **Regional Preference:**  East | Users will be shown a popup message that says “Group successfully edited!” | User is shown a popup message that says “Group successfully edited!” | Pass |
| 2. | User removed the group name and did not enter any group name | 1. In the My Groups Page, users press on the “pencil” button to edit a group. 2. A popup will show that details a form for user to edit the fields they want to make changes to 3. User deletes the group name in the form 4. User tries to press “tick” button to edit the group | **Group Name:**  **Group Description:**  Join if you love chinese food!  **Food Preference:**  Chinese Cuisine, Vegan  **Regional Preference:**  East | Users will not be allowed to saved the changes made.  An error message will show below the Group Name text field that says “Group name is empty!” | Users are not allowed to save the changes made to the group. An error message is shown to the user that “Group name is empty!” | Pass |
| 3. | User removed the group description and did not enter any group description | 1. In the My Groups Page, users press on the “pencil” button to edit a group. 2. A popup will show that details a form for user to edit the fields they want to make changes to 3. User deletes the group description in the form 4. User tries to press “tick” button to edit the group | **Group Name:**  Chinese Food  **Group Description:**  **Food Preference:**  Chinese Cuisine, Vegan  **Regional Preference:**  East | Users will not be allowed to saved the changes made  An error message will show below the Group Name text field that says “Group description is empty!” | Users are not allowed to save the changes made to the group. An error message is shown to the user that “Group description is empty!” | Pass |
| 4. | User removed all food preferences and did not select a food preferences | 1. In the My Groups Page, users press on the “pencil” button to edit a group. 2. A popup will show that details a form for user to edit the fields they want to make changes to 3. User deletes the group preferences in the form 4. User tries to press “tick” button to edit the group | **Group Name:**  Chinese Food  **Group Description:**  Join if you love Chinese Food  **Food Preference:**  **Regional Preference:**  East | Users will not be allowed to saved the changes made | User is not allowed to save the changes made to the group | Pass |

***Test Case 13: Chat in Group Function***

| **Test Scenario** | Chat in Group | **Test Case ID** | chat\_in\_group |
| --- | --- | --- | --- |
| **Test case description** | Chatting in a group with no valid input strings does not allow user to send the messages | **Test Priority** | Medium |
| **Prerequisite** | User needs to be a member of a group | | |

| **#** | **Description** | **Action** | **Inputs** | **Expected Output** | **Actual Output** | **Test Result** |
| --- | --- | --- | --- | --- | --- | --- |
| 1. | Users are able to send a message successfully | 1. In the My Groups Page, users press on any group that they have joined. 2. The user then clicks on “Group Chat” 3. User enters a message in the chat box 4. User then press enter to send the message to the group | Message: “Hello!” | The message “Hello” will be sent to the group and the message can be seen | The message “Hello” is sent to the group and the message can be seen by the user | Pass |
| 2. | Users are not able to send a message | 1. In the My Groups Page, users press on any group that they have joined. 2. The user then clicks on “Group Chat” 3. User does not enter anything in the chat box 4. User then press enter to send the message to the group | Message: | The message will not be sent to the group | The message is not sent to the group | Pass |

## White Box Testing

***Test Case 1: Login Function***

***Test Case 2: Register Function***

***Test Case 3: Reset Password Function***

***Test Case 4: Find Food - Get Current Location***

***Test Case 5: Find Food - Search Food Function***

***Test Case 6: Find Food - Recommend Food Function***

***Test Case 7: Change Password Function***

***Test Case 8: Edit Profile Function***

***Test Case 9: Create Group Function***

***Test Case 10: Create Next Meeting Function***

***Test Case 11: Edit Next Meeting Function***

***Test Case 12: Join Group Function***

***Test Case 13: Find Groups Function***

***Test Case 14: Edit Group Function***

***Test Case 15: Chat in Group Function***

# Appendix A: Data Dictionary

| Index | Terms | Definition |
| --- | --- | --- |
| 1. | Account | For a user to use the website’s functionality, they must register for an account |
| 2. | Address | The official address of the food directory |
| 3. | Alphanumeric | The character is either an alphabet or a number |
| 4. | Contact | Phone information to call the place |
| 5. | Cookie | A cookie is created for a website to track, personalize and save information about each user’s session |
| 6. | Display Name | The user’s customisable name  Displayed with ‘@’ as a prefix |
| 7. | Drop-down selection | A drop down text box for the user to select options |
| 8. | Email Address | The user’s email address |
| 9. | Filter | Narrows the search display results to the user’s selection |
| 10. | Food Preference | The food that the user prefers to eat, including, but not limited to, Vegetarian, Halal, No Beef, No Alcohol, Spicy, etc. |
| 11. | Global Positioning System (GPS) | Space-based radio-navigation system  Allows the website to know the location of the user |
| 12. | Google Maps API | A set of application programming interfaces that lets us use the power of Google Maps in our website |
| 13. | Group | A cluster of user accounts with similar food and regional preferences that can meet up to have a meal together |
| 14. | Group Chat | A place for the group members to talk and discuss about future meetings |
| 15. | Group Member | A user is a group member after joining a group |
| 16. | Group Owner | A user is a group member after creation of a group |
| 17. | Group Profile | The group’s details such as the group’s name, picture, description, food preferences, and current group members’ name and username |
| 18. | Home Page | The default page shown after signing in  Upcoming meetings will also be shown here |
| 19. | Input text box | An editable text field that allows the user to change a particular item’s text while fulfilling a certain requirement |
| 20. | Landing Page | An area for first time users with new cookies |
| 21. | Login Page | An area for user to sign in to their user account |
| 22. | Marker | A pin icon shown in Google Maps that shows exactly where is the location |
| 23. | Mask | The password is hidden with asterisks |
| 24. | Meeting | The upcoming group activity that the group members have decided to meet up for |
| 25. | Option | A clickable button that the user can interact with |
| 26. | Password | The user’s password |
| 29. | Profile | The user account’s profile with information created from registration |
| 31. | Ratings | An average of scores rated by people who have been to a place on a scale from 1 to 5 |
| 32. | Regional Preference | The region where the user prefers for meetings  Namely, North, South, East, West, Central |
| 33. | Registration Page | An area for user to create their user account |
| 34. | Search | A text field that the user can type in to find groups and locations |
| 35. | Tourism Info Hub API | A set of application programming interfaces that the Singapore Government has publicly released to be used to fetch food location and its details |
| 36. | Unicode | An international encoding standard for use with different languages and script |
| 37. | Username | The user’s unique identifier |

# Appendix B: Analysis Models

1. **Use Case Diagram**
2. **Use Case Description**

1. Register

| Use Case ID: | UC-0001 | | |
| --- | --- | --- | --- |
| Use Case Name: | Register | | |
| Created By: | Shavonne | Last Updated By: | Daryl |
| Date Created: | 16/10/23 | Date Last Updated: | 16/11/23 |

| Actor: | User |
| --- | --- |
| Description: | Allows users to create an account for the system |
| Preconditions: | - |
| Postconditions: | An account is created |
| Priority: | High |
| Frequency of Use: | Once per new User |
| Flow of Events: | 1. Use Case starts when the user clicks on the “Register” button. 2. The system prompts the user to input the sign up form. 3. The user enters account details. 4. The user clicks “Submit” to confirm their information. 5. The system validates the input data. 6. Use case ends. |
| Alternative Flows: | AF-S3.1: System checked that user entered an invalid input for email (wrong format or duplicated)   1. System displays an error message. 2. System goes back to step 2.   AF-S3.2: System checked that user did not satisfy password requirements   1. System displays an error message. 2. System goes back to step 2.   AF-S3.3: System checked that user did not enter the correct password in the confirm password field   1. System displays an error message. 2. System goes back to step 2.   AF-S3.4: System checked that user entered an invalid username (duplicated username)   1. System displays an error message. 2. System goes back to step 2. |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

2. Sign In

| Use Case ID: | UC-0002 | | |
| --- | --- | --- | --- |
| Use Case Name: | Sign In | | |
| Created By: | Zhen Xuan | Last Updated By: | Daryl |
| Date Created: | 16/10/23 | Date Last Updated: | 16/11/23 |

| Actor: | User |
| --- | --- |
| Description: | Allows users to sign in to their account |
| Preconditions: | The user has a verified account |
| Postconditions: | A unique session will be created for the user |
| Priority: | High |
| Frequency of Use: | Once per session |
| Flow of Events: | 1. Use case starts when the user clicks on the “Login” button in the Landing Page. 2. The system displays the Login Page. 3. The user enters an email address or a username. 4. The user enters the password. 5. The system validates data and authenticates the user’s account. 6. The system authenticates the user using the included use case “Authenticate User”. 7. The user is signed in. 8. The system redirects to the Homepage of the website. 9. Use case ends. |
| Alternative Flows: | AF-S5.1: Email or Username is invalid   1. System displays a message “Email or Username is invalid!” 2. System returns to step 3.   AF-S5.2: Password is invalid   1. System displays a message “Password is invalid!” 2. System returns to step 3. |
| Exceptions: | EX.1: User does not have an account   1. User selects the “Create Account” option.   EX.2: User forgot their password   1. User selects the “Forgot Password?” option. |
| Includes: | Authenticate User |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

3. Sign out

| Use Case ID: | UC-0003 | | |
| --- | --- | --- | --- |
| Use Case Name: | Sign out | | |
| Created By: | Raphael | Last Updated By: | Daryl |
| Date Created: | 27/09/23 | Date Last Updated: | 16/11/23 |

| Actor: | User |
| --- | --- |
| Description: | Allows a user to sign out of his account |
| Preconditions: | User is signed in |
| Postconditions: | User’s session will be terminated |
| Priority: | Low |
| Frequency of Use: | 0-1 per log in |
| Flow of Events: | 1. Use case starts when the user clicks on the “Sign out” button. 2. The system prompts the user to “Confirm” or “Cancel”. 3. The user clicks on “Confirm”. 4. The system terminates the login session. 5. The system displays the landing page. 6. Use case ends. |
| Alternative Flows: | - |
| Exceptions: | EX.1: User clicks on “Cancel”.   1. System closes the prompt and displays the page the user was on. |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

4. View Profile

| Use Case ID: | UC-0004 | | |
| --- | --- | --- | --- |
| Use Case Name: | View Profile | | |
| Created By: | Kevin | Last Updated By: | Daryl |
| Date Created: | 16/10/23 | Date Last Updated: | 16/11/23 |

| Actor: | User |
| --- | --- |
| Description: | Allows a user to view a user profile |
| Preconditions: | User is logged in to view his profile |
| Postconditions: | - |
| Priority: | Medium |
| Frequency of Use: | 1-2 times per login |
| Flow of Events: | 1. Use case starts when the user clicks on “Profile”. 2. The system fetches user profile details from the included use case “Fetch Data”. 3. The system displays the Profile page of the logged in user. 4. Use case ends. |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | Fetch Data |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

###### 

5. Edit Profile

| Use Case ID: | UC-0005 | | |
| --- | --- | --- | --- |
| Use Case Name: | Edit Profile | | |
| Created By: | Jia Qi | Last Updated By: | Daryl |
| Date Created: | 27/09/23 | Date Last Updated: | 16/11/23 |

| Actor: | User |
| --- | --- |
| Description: | Allows User to edit profile |
| Preconditions: | User is in the View Profile Page |
| Postconditions: | The new details reflect the user’s profile |
| Priority: | Medium |
| Frequency of Use: | 1-2 times per login |
| Flow of Events: | 1. Use Case starts when User selects the “Edit Profile” button. 2. System displays form to edit profile. 3. The user edits the form. 4. The user submits the form. 5. The system redirects User to “View Profile” page. 6. Use Case ends. |
| Alternative Flows: | AF-S4.1: Email entered is invalid   1. System prompts the user to enter a new email. 2. System returns to step 3.   AF-S4.2: Email entered is already in the database   1. System prompts the user to enter a new email. 2. System returns to step 3.   AF-S4.3: Username entered is already in the database   1. System prompts the user to enter a new username. 2. System returns to step 3. |
| Exceptions: | EX.1: User clicks on “Cancel”.   1. User returns to View Profile Page. |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

###### 

6. Create Group

| Use Case ID: | UC-0006 | | |
| --- | --- | --- | --- |
| Use Case Name: | Create Group | | |
| Created By: | Daryl | Last Updated By: | Daryl |
| Date Created: | 27/09/23 | Date Last Updated: | 16/11/23 |

| Actor: | User |
| --- | --- |
| Description: | Allows User to create a group |
| Preconditions: | The user is logged in and the user can only create groups under “View My Groups” |
| Postconditions: | A group is created with the user as the owner |
| Priority: | Medium |
| Frequency of Use: | 0-2 times per login |
| Flow of Events: | 1. Use Case starts when the user selects “+” to create a group. 2. The system displays a form. 3. The user fills in the necessary details of the group. 4. The user selects “Confirm”. 5. The system creates the group. 6. The system displays the newly created group in “View My Groups” 7. Use Case ends. |
| Alternative Flows: | AF-S4.1: User did not fill in the required field   1. System prompts the user to fill in the form. 2. System return to step 3. |
| Exceptions: | EX.1: User selects “Cancel”   1. System closes the form |
| Includes: | View My Groups |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

###### 

7. View Map

| Use Case ID: | UC-0007 | | |
| --- | --- | --- | --- |
| Use Case Name: | View Map | | |
| Created By: | Shavonne | Last Updated By: | Daryl |
| Date Created: | 27/09/23 | Date Last Updated: | 16/11/23 |

| Actor: | User |
| --- | --- |
| Description: | Allows User to view google map when user clicks “Find Food” |
| Preconditions: | The user is logged in |
| Postconditions: | - |
| Priority: | Medium |
| Frequency of Use: | 1-4 times per login |
| Flow of Events: | 1. Use Case starts when the user selects the “Find Food” button in the navigation menu. 2. The system sends a fetch request for map data to Google Maps API. 3. The system receives requested data from Google Maps API 4. The system displays a google map with a search box. 5. Use case ends. |
| Alternative Flows: | - |
| Exceptions: | EX.1: Google API down   1. System does not show google map |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

###### 

8. Edit Group Details

| Use Case ID: | UC-0008 | | |
| --- | --- | --- | --- |
| Use Case Name: | Edit Group Details | | |
| Created By: | Zhen Xuan | Last Updated By: | Daryl |
| Date Created: | 27/09/23 | Date Last Updated: | 16/11/23 |

| Actor: | User |
| --- | --- |
| Description: | Allows the user to edit the group details in the “View Group Detail” page |
| Preconditions: | The group exists and the user is the group owner |
| Postconditions: | The group details will change |
| Priority: | Medium |
| Frequency of Use: | 0-3 times |
| Flow of Events: | 1. Use case starts when the user clicks on the “Edit” button. 2. The system redirects and displays editable text boxes for the details of the group. 3. The user edits the group details. 4. The user selects “Save” button. 5. The system displays the edited group details. 6. Use case ends. |
| Alternative Flows: | AF-S4.1: User did not fill in the required field   1. System prompts the user to fill in the form. 2. System return to step 3. |
| Exceptions: | EX.1: User clicks cancel button   1. System redirects the user back to the “View Group Detail” page. |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

9. Remove Members

| Use Case ID: | UC-0009 | | |
| --- | --- | --- | --- |
| Use Case Name: | Remove Members | | |
| Created By: | Raphael | Last Updated By: | Daryl |
| Date Created: | 27/09/23 | Date Last Updated: | 16/11/23 |

| Actor: | - |
| --- | --- |
| Description: | When the owner deletes group, all members will be removed from the group |
| Preconditions: | The group has to exist |
| Postconditions: | All members of the group will be removed |
| Priority: | Medium |
| Frequency of Use: | 1 time per group |
| Flow of Events: | 1. Use case starts during the execution of “Delete Group” use case. 2. The system sends a query to the database to remove all users from the group. 3. Use case ends. |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

10. View My Groups

| Use Case ID: | UC-0010 | | |
| --- | --- | --- | --- |
| Use Case Name: | View My Groups | | |
| Created By: | Kevin | Last Updated By: | Daryl |
| Date Created: | 27/09/23 | Date Last Updated: | 16/11/23 |

| Actor: | User |
| --- | --- |
| Description: | Allow User to view all groups they created or are subscribed to |
| Preconditions: | User is logged in |
| Postconditions: | System displays all the groups correctly |
| Priority: | High |
| Frequency of Use: | Once per 5 minutes |
| Flow of Events: | 1. Use case starts when the user clicks on the “My Groups” button. 2. The system fetches user’s group data from the included use case “Fetch Data”. 3. The system displays all users’ joined groups in the system. 4. Use case ends. |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | Fetch Data |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

11. Chat In Groups

| Use Case ID: | UC-0011 | | |
| --- | --- | --- | --- |
| Use Case Name: | Chat In Groups | | |
| Created By: | Jia Qi | Last Updated By: | Daryl |
| Date Created: | 27/09/23 | Date Last Updated: | 16/11/23 |

| Actor: | User |
| --- | --- |
| Description: | Allow User to message in group chats |
| Preconditions: | User is logged in  User is a member of the selected group chat |
| Postconditions: | A user message is sent to the selected group chat |
| Priority: | High |
| Frequency of Use: | Once per 5 minutes |
| Flow of Events: | 1. Use case starts when the user clicks on the message box on the group chat. 2. The user enters a message in the text box. 3. The user submits the message. 4. The system displays the message from User to the chat group. 5. Use case ends. |
| Alternative Flows: | - |
| Exceptions: | EX.1: User did not enter any messages in the text box   1. No new messages are sent nor displayed |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

12. Search Location

| Use Case ID: | UC-0012 | | |
| --- | --- | --- | --- |
| Use Case Name: | Search Location | | |
| Created By: | Daryl | Last Updated By: | Daryl |
| Date Created: | 27/09/23 | Date Last Updated: | 16/11/23 |

| Actor: | User |
| --- | --- |
| Description: | Allow User to search places to eat based on location |
| Preconditions: | The user is logged in  The user is on the “View Map” page |
| Postconditions: | System displays details of the food directory |
| Priority: | Medium |
| Frequency of Use: | 1-4 per login |
| Flow of Events: | 1. Use Case starts when User has done “View Map”. 2. The user searches for location in the search box. 3. The system displays the food places on Google Maps as markers. 4. The user clicks on a marker to view details of the food directory. 5. Use case ends. |
| Alternative Flows: | AF-S3: User filters food using recommended cuisine tags   1. System displays food based on the selected tag. 2. System returns to step 4. |
| Exceptions: | EX.1: User enters words that cannot be found on google map   1. System displays a message “No result found”.   EX.2: Google Maps API is down   1. System displays an error message.   EX.3: Tourism API is down   1. System displays an error message. |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

13. View Group Chat

| Use Case ID: | UC-0013 | | |
| --- | --- | --- | --- |
| Use Case Name: | View Group Chat | | |
| Created By: | Shavonne | Last Updated By: | Daryl |
| Date Created: | 27/09/23 | Date Last Updated: | 16/11/23 |

| Actor: | User |
| --- | --- |
| Description: | Displays all the messages in the group chat |
| Preconditions: | The user is logged in  The user is in “View Group Details” |
| Postconditions: | The chat messages for the selected group are displayed |
| Priority: | High |
| Frequency of Use: | 0-5 times per session |
| Flow of Events: | 1. Use case starts when the user selects the header “Group Chat” in “View Group Details”. 2. The system fetches group chat details from the included use case “Fetch Data”. 3. The system displays all messages in the selected group from newest to oldest, and their corresponding senders’ usernames and display names. 4. Use case ends. |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | Fetch Data |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

###### 

14. Delete Group

| Use Case ID: | UC-0014 | | |
| --- | --- | --- | --- |
| Use Case Name: | Delete Group | | |
| Created By: | Zhen Xuan | Last Updated By: | Daryl |
| Date Created: | 27/09/23 | Date Last Updated: | 16/11/23 |

| Actor: | User |
| --- | --- |
| Description: | Allows the group to be deleted |
| Preconditions: | The user is a group owner  The user is in “View Group Details” |
| Postconditions: | The selected group is deleted from the database |
| Priority: | Low |
| Frequency of Use: | 0-1 times |
| Flow of Events: | 1. Use case starts when the user clicks on “Delete Group”. 2. The system will prompt the user for confirmation. 3. Tser clicks the confirmation button. 4. The system will send a query to the database to delete the group. 5. The group will be deleted. 6. The user will leave the group using the included use case “Remove Member”. 7. Use case ends. |
| Alternative Flows: | - |
| Exceptions: | EX.1: User clicks “Cancel”   1. Group would not be deleted. |
| Includes: | Remove Member |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

###### 

15. Leave Group

| Use Case ID: | UC-0015 | | |
| --- | --- | --- | --- |
| Use Case Name: | Leave Group | | |
| Created By: | Raphael | Last Updated By: | Daryl |
| Date Created: | 27/09/23 | Date Last Updated: | 16/11/23 |

| Actor: | User |
| --- | --- |
| Description: | The user leaves the group |
| Preconditions: | The user is logged in  The user is in “View Group Details” |
| Postconditions: | The group will be deleted for the user |
| Priority: | Low |
| Frequency of Use: | 1 time per group |
| Flow of Events: | 1. Use case starts when the user clicks on “Leave Group”. 2. The system prompts for confirmation. 3. The user clicks the confirmation button. 4. The system sends a query to the database to remove the user from the group. 5. Use case ends. |
| Alternative Flows: | - |
| Exceptions: | EX.1: The user selects “Cancel”   1. User does not leave group |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

###### 

16. View Group Details

| Use Case ID: | UC-0016 | | |
| --- | --- | --- | --- |
| Use Case Name: | View Group Details | | |
| Created By: | Kevin | Last Updated By: | Daryl |
| Date Created: | 27/09/23 | Date Last Updated: | 16/11/23 |

| Actor: | User |
| --- | --- |
| Description: | Displays the groups details |
| Preconditions: | The user is logged in  The user is a member of the group |
| Postconditions: | The group details will be shown |
| Priority: | High |
| Frequency of Use: | 0-5 times |
| Flow of Events: | 1. Use case starts when the clicks on a group listing in “View My Groups”. 2. The system fetches data using the included use case “Fetch Data”. 3. The system displays all the details of the group. 4. Use case ends. |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | Fetch Data |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

###### 

17. View Next Meeting

| Use Case ID: | UC-0017 | | |
| --- | --- | --- | --- |
| Use Case Name: | View Next Meeting | | |
| Created By: | Jia Qi | Last Updated By: | Daryl |
| Date Created: | 27/09/23 | Date Last Updated: | 16/11/23 |

| Actor: | User |
| --- | --- |
| Description: | The details of the meeting will be displayed |
| Preconditions: | The user is logged in  The user is in “View Group Details” |
| Postconditions: | System will display the next meeting details |
| Priority: | High |
| Frequency of Use: | 0-5 times per login |
| Flow of Events: | 1. Use case starts when the user clicks on “Next Meeting” in “View Group Details”. 2. The system fetches data from the included use case “Fetch Data”. 3. The system will display next meeting details. 4. The system sends a fetch request for map data to Google Maps API. 5. The system receives requested data from Google Maps API. 6. The system displays Google Maps. 7. Use case ends. |
| Alternative Flows: | - |
| Exceptions: | EX.1: No upcoming meeting is found   1. System will display “There are no upcoming meetings.”   EX.2: Google API down   1. System does not show google map |
| Includes: | Fetch Data |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

###### 

18. Edit Meeting Details

| Use Case ID: | UC-0018 | | |
| --- | --- | --- | --- |
| Use Case Name: | Edit Meeting Details | | |
| Created By: | Daryl | Last Updated By: | Daryl |
| Date Created: | 27/09/23 | Date Last Updated: | 16/11/23 |

| Actor: | User |
| --- | --- |
| Description: | Allows the user to edit the group meetings details |
| Preconditions: | The user is logged in and the user is the group owner  The user is in “View Next Meeting”. |
| Postconditions: | The group details will be shown |
| Priority: | High |
| Frequency of Use: | 0-5 times per session |
| Flow of Events: | 1. Use case starts when the user clicks on the edit or create button in the “View Next Meeting” page. 2. The system will unlock the fields of Date, Time and Description. 3. The user edits any of the fields. 4. The user searches for location in the search box. 5. The system displays the food places on Google Maps as markers. 6. The user clicks on a marker to view details of the food directory. 7. The user clicks the confirmation button to save location name and address. 8. The user clicks the save button. 9. The system saves the updated meeting details. 10. Use case ends. |
| Alternative Flows: | AF-S8: System checks that the user has not entered any fields   1. System displays a message to prompt the user to enter the required fields. 2. System returns to step 3. |
| Exceptions: | EX.1: User selects “Cancel”   1. System does not save any changes. |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

###### 

19. Find Groups

| Use Case ID: | UC-0019 | | |
| --- | --- | --- | --- |
| Use Case Name: | Find Groups | | |
| Created By: | Shavonne | Last Updated By: | Daryl |
| Date Created: | 27/09/23 | Date Last Updated: | 16/11/23 |

| Actor: | User |
| --- | --- |
| Description: | User is logged in  User clicks the “Find Groups” button in the website navigation bar |
| Preconditions: | User has successfully logged into the website |
| Postconditions: | System displays all available groups for joining |
| Priority: | High |
| Frequency of Use: | Once per click |
| Flow of Events: | 1. Use case begins when the user clicks the “Find Groups” button in the navigation bar. 2. The system fetches all available groups from the included use case “Fetch Data”. 3. The system displays all available groups for joining and their relevant details. 4. Use case ends. |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | Fetch Data |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

###### 

20. Join Group

| Use Case ID: | UC-0020 | | |
| --- | --- | --- | --- |
| Use Case Name: | Join Group | | |
| Created By: | Zhen Xuan | Last Updated By: | Daryl |
| Date Created: | 27/09/23 | Date Last Updated: | 16/11/23 |

| Actor: | User |
| --- | --- |
| Description: | User clicks the join button and joins an available group under “Find Groups” |
| Preconditions: | System displays all available groups for joining |
| Postconditions: | User successfully joins the selected available group |
| Priority: | High |
| Frequency of Use: | Once per click |
| Flow of Events: | 1. Use case starts when the user clicks the join button of the selected group. 2. The system will inform the user of the successful addition to the group. 3. Use case ends. |
| Alternative Flows: | - |
| Exceptions: | EX.1: User clicks the join button of a group he is already a member of   1. System displays a message “You are already in this group!”. |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

###### 

21. Fetch Data

| Use Case ID: | UC-0021 | | |
| --- | --- | --- | --- |
| Use Case Name: | Fetch Data | | |
| Created By: | Raphael | Last Updated By: | Daryl |
| Date Created: | 27/09/23 | Date Last Updated: | 16/11/23 |

| Actor: | Database |
| --- | --- |
| Description: | Database will fetch the relevant data in response to all queries made by the system, to the system. |
| Preconditions: | System queries a request to the database. |
| Postconditions: | System receives relevant data from the database. |
| Priority: | High |
| Frequency of Use: | Once per request query |
| Flow of Events: | 1. Use case starts when the system makes a request to the database to fetch certain data. 2. Database will execute the query. 3. Database will send the accurate fetched data to the system. 4. System will receive the correct fetched data. 5. Use case ends. |
| Alternative Flows: | - |
| Exceptions: | EX1: Database unable to fetch requested data.   1. System will receive an error message from the database. 2. System will not receive requested data from the database. |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | All search queries are correct and accurate. |
| Notes and Issues: | - |

22. Change Password

| Use Case ID: | UC-0022 | | |
| --- | --- | --- | --- |
| Use Case Name: | Change Password | | |
| Created By: | Kevin | Last Updated By: | Daryl |
| Date Created: | 10/11/23 | Date Last Updated: | 16/11/23 |

| Actor: | User |
| --- | --- |
| Description: | Allows User to change his password |
| Preconditions: | User is logged in  User is on the “View Profile” page |
| Postconditions: | User has changed his password |
| Priority: | High |
| Frequency of Use: | Once per request query |
| Flow of Events: | 1. Use case starts when the user clicks on the “Change Password” button. 2. The system displays a form for the user to change password. 3. The user enters the current password and new password. 4. The user clicks confirm to change password. 5. The system saves updated data. 6. The system redirects the user to “View Profile Page”. 7. Use case ends. |
| Alternative Flows: | AF-S4.1: System checked that user did not enter the correct current password   1. System displays a message “Invalid current password!” 2. System returns to step 3.   AF-S4.2: System checked that user did not satisfy password requirements   1. System displays an error message to prompt the user to re-enter a new password. 2. System returns to step 3. |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

23. Reset Password

| Use Case ID: | UC-0023 | | |
| --- | --- | --- | --- |
| Use Case Name: | Reset Password | | |
| Created By: | Jia Qi | Last Updated By: | Daryl |
| Date Created: | 10/11/23 | Date Last Updated: | 16/11/23 |

| Actor: | User |
| --- | --- |
| Description: | Allows User to reset his password |
| Preconditions: | User has an account registered |
| Postconditions: | User has resetted his password |
| Priority: | Medium |
| Frequency of Use: | Once per request query |
| Flow of Events: | 1. Use case starts when the user clicks on the “Forgot Password” button. 2. The system displays a form for the user to enter his email. 3. The user clicks confirm. 4. The system will check if the email exists in the database from the included use case “Fetch Data”. 5. The system will send a link to his email for password reset. 6. The user clicks on the link. 7. The user enters the new password for his account. 8. The user clicks confirm to reset password. 9. The system saves updated data. 10. The system redirects the user to “Sign In”. 11. Use case ends. |
| Alternative Flows: | AF-S4: System checked that email does not exist in database   1. System displays a message “Email not found!” 2. System returns to step 2.   AF-S7: System checked that user did not satisfy password requirements   1. System displays an error message to prompt the user to re-enter a new password. 2. System returns to step 6. |
| Exceptions: | EX.1: Email service is down   1. No email will be received by user |
| Includes: | Fetch Data |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

1. **Sequence Diagram**
2. **Class Diagram**
3. **State Machine Diagram / Initial Dialog Map**
4. **Key Boundary and Control Classes**

# Appendix C: Design Pattern Writeup

The entire project code base is done using the MVC design pattern where we segregate the Model, View and Controller. The LetsMakanTogether team has written a documentation to specify and note down the different components used and how they interact with each other.

Jia, Q. (n.d.). *Design pattern writeup*. online.fliphtml5.com. <https://online.fliphtml5.com/slbkl/ayvv/>

Source: http://www.frontiernet.net/~kwiegers/process\_assets/srs\_template.doc