Functional Requirements	1 User login 1.1 When a user wants to login to the application, the user must enter their email and password 1.1.1 The email must be text of at least 1 character and within 512 characters 1.1.2 The email must have a valid Gmail domain 1.2 The email and password must match the one in the database to successfully login 2 User creating a room 2.1 A user can create a room with a unique room code 2.1.1 The unique room code will be randomly generated 2.1.2 The unique room code will be 8 characters in length with a mix of numbers and characters 2.2 A user can enter a unique room code to join an existing room 3 User Joined a room 3.1 A user can enter their unique name to be displayed in the room 3.1.1 The name of the user must be a text of at least 5 characters and within 16 characters. 3.2 A user can enter their available dates on a displayed calendar 3.3 A user can choose their preferred meals 3.3.1 The user will tick checkboxes for breakfast, lunch, dinner and/or supper 3.4 The user can choose their preferred area of dining 3.4.1 The user will tick checkboxes for their chosen preferred area 3.5 The user can choose their preferred cuisine 3.5.1 The user will tick checkboxes for their chosen preferred cuisine 4 Application filtering and deciding the commonly available time slot 4.1 The application will filter the commonly available time slots of all members 4.2 The application will display the commonly available time slots to all members in DDMMYY | HH:MM format 4.2.1 The time slots can be edited by users 4.2.2 The areas chosen can be edited by users 4.2.3 The cuisines chosen can be edited by users 5 User voting for recommended restaurants 5.1 The application will have a “generate” button to generate a list of recommended restaurants 5.1.1 The list is generated using the commonly available time slots, preferred areas and cuisine as a filter 5.1.2 The list is further filtered based on restaurant popularity amongst other users of the application 5.1.3 The user can choose to recommend visiting a restaurant listed with upvote 5.1.3.1 The application will use a points system where a

	recommendation adds one point 5.1.4 The user can choose to discourage visiting a restaurant listed with downvote 5.1.4.1 The application will use a points system where a recommendation adds one point and discouragement deducts one point 6 A tiebreaker is done using a spinning wheel 6.1 The preferred restaurant is decided by the restaurant with the highest points 6.2 A voting tie will be broken by spinning a wheel with the possible restaurants listed 6.2.1 The wheel will use a random number generator to pick a restaurant using its index 6.3 The wheel can be respun if there is below 70% voting agreement between users 7 The application will send reminders to the user ? 7.1 The application will have a confirmation button that users will click after finalisation 7.1 The system will show an option to add the event to the calendar 7.2 There will be a confirmation email sent to the Gmail of the users 7.2.1 The email sent will contain an attachment to add details of the event to a user's calendar 7.2.2 The event added to the user's calendar will remind the user 1 day before the date and time of the event 8 Each room will have their own chat 8.1 User can send messages through the chat 8.1.1 Each message will show the message, creator, and date & time 8.1.2 Each message must be at least 1 character and within 500 characters 8.2 All users in the same room must be able to see the message 9 Users can exit and close the room 9.1 Once the date of event has passed, users can choose to exit the room 9.2 If the user is the room administrator, the user can choose to close the room 10 Users can manage their friends list 10.1 Users can add friends through unique member ID 10.2 Users should be able to view their list of friends 10.3 User can invite their friends into the rooms they joined 11 Users can view all schedules of rooms that they are in 11.1 Users must be able to view all upcoming events scheduled for rooms that they are in.

	12 Users can manage their profile 12.1 Users can edit their profile 12.1.1 Users can change their name to be displayed in the group 12.1.1.1 The name of the user must be a text of at least 5 characters and within 16 characters. 12.1.2 The user can upload a photo to create or change their profile picture 12.1.2.1 The photo must be a 1:1 aspect ratio of a maximum of 500x500 pixels 12.2 Users can view past rooms that they have previously joined and closed
Non-Functional Requirements	• The system should respond to the user's actions within 2 seconds • The application should be able to load up within 5 seconds when the user first opens it • The database must be updatable with new restaurants • The language used by the application must be displayed in the local language or one chosen by the user ? • A new message must show up within 15 seconds of sending

Data Dictionary

Term	Definition
Email	A text field that contains the user's email address. Must be at least 1 character and within 512 characters, with a valid Gmail domain.
Password	A text field containing the user's password. Used for authentication and must match the stored password in the database.
Room PIN	A unique identifier for a room. It is randomly generated and consists of 8 characters, including numbers and letters.
User Display Name	A unique name chosen by the user for display within a room. Must be between 1 and 16 characters.
Available Dates	The dates a user selects on a calendar to indicate their availability.
Commonly Available Time Slots	Time slots that all room members are available for, calculated based on individual inputs.
Time Slot Format	Displayed in the format DDMMYY | HH:MM
Editable Preferences	Time slots, areas, and cuisines that users can modify after selection.
Restaurant Recommendation List	A list of recommended restaurants generated based on commonly available time slots, preferred areas, and cuisine preferences.

Restaurant Popularity Score	A ranking system that increases when users recommend a restaurant and decreases when users discourage it
Voting System	A mechanism where users vote for their preferred restaurant from the recommendation list
Tie-Breaker Mechanism	A system using a random number generator to spin a wheel and select a restaurant when votes are tied
Chat	A system where user can view message and send message to
Room	A chat room where users can communicate and schedule an upcoming event
Profile	A customised set of text that each user has that includes their profile picture and name

Use Cases

for

OngLaiDecides

Version 1.0 approved

Tiffany Mun, Boey

Ong Lai Tarts :D

05/02/2025

Revision History

Name	Date	Reason For Changes	Version

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

Guidance for Use Case Template

Document each use case using the template shown in the Appendix. This section provides a description of each section in the use case template.

1. Use Case Identification

1.1. Use Case ID

Give each use case a unique numeric identifier, in hierarchical form: X.Y. Related use cases can be grouped in the hierarchy. Functional requirements can be traced back to a labeled use case.

1.2. Use Case Name

State a concise, results-oriented name for the use case. These reflect the tasks the user needs to be able to accomplish using the system. Include an action verb and a noun. Some examples:

- View part number information.
- Manually mark hypertext source and establish link to target.
- Place an order for a CD with the updated software version.

1.3. Use Case History

1.3.1 Created By

Supply the name of the person who initially documented this use case.

1.3.2 Date Created

Enter the date on which the use case was initially documented.

1.3.3 Last Updated By

Supply the name of the person who performed the most recent update to the use case description.

1.3.4 Date Last Updated

Enter the date on which the use case was most recently updated.

2. Use Case Definition

2.1. Actor

An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor(s) that will be performing this use case.

2.2. Description

Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.

2.3. Preconditions

List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each precondition. Examples:

1. User's identity has been authenticated.
2. User's computer has sufficient free memory available to launch task.

2.4. Postconditions

Describe the state of the system at the conclusion of the use case execution. Number each postcondition. Examples:

1. Document contains only valid SGML tags.
2. Price of item in database has been updated with new value.

2.5. Priority

Indicate the relative priority of implementing the functionality required to allow this use case to be executed. The priority scheme used must be the same as that used in the software requirements specification.

2.6. Frequency of Use

Estimate the number of times this use case will be performed by the actors per some appropriate unit of time.

2.7. Flow of Events

Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. This description may be written as an answer to the hypothetical question, "How do I <accomplish the task stated in the use case name>?" This is best done as a numbered list of actions performed by the actor, alternating with responses provided by the system.

2.8. Alternative Flows

Document other, legitimate usage scenarios that can take place within this use case separately in this section. State the alternative course, and describe any differences in the sequence of steps that take place. Number each alternative course using the Use Case ID as a prefix, followed by "AC" to indicate "Alternative Course". Example: X.Y.AC.1.

2.9. Exceptions

Describe any anticipated error conditions that could occur during execution of the use case, and define how the system is to respond to those conditions. Also, describe how the system is to respond if the use

case execution fails for some unanticipated reason. Number each exception using the Use Case ID as a prefix, followed by “EX” to indicate “Exception”. Example: X.Y.EX.1.

2.10. Includes

List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality.

2.11. Special Requirements

Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.

2.12. Assumptions

List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.

2.13. Notes and Issues

List any additional comments about this use case or any remaining open issues or TBDs (To Be Determineds) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.

Use Case ID:	1.1		
Use Case Name:	Login		
Created By:	Boey, Tiffany	Last Updated By:	04.02.25
Date Created:	04.02.25	Date Last Updated:	04.02.25

Actor:	Group Owner/User, Oauth
Description:	Users login into their account
Preconditions:	N.A.
Postconditions:	Users authenticated their account and login
Priority:	
Frequency of Use:	
Flow of Events:	1. The user goes to the website 2. The user enters their email/phone number and password 3. The system requests for Oauth. 4. Oauth returns successfully. 5. The system allows user to login and bring them to the main page.
Alternative Flows:	1.1-AF-S1: Incorrect Email/PhoneNumber/Password 1. If Oauth returns unsuccessfully, the system prompts "Incorrect Email or Password" 2. User can try to login again 1.1-AF-S2: No account 1. If user does not have an account, user will be prompted to create an account 2. After creation of account, user will be allowed to login automatically after Oauth 1.1-EX-S1: Failed to login 5 times 1. User's account will be locked. 2. Reset password to unlock account
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	2.1		
Use Case Name:	Create Group		
Created By:	Boey, Tiffany	Last Updated By:	04.02.25
Date Created:	04.02.25	Date Last Updated:	04.02.25

Actor:	User
Description:	User creates a group
Preconditions:	Users have logged into their account
Postconditions:	Users creates a new group
Priority:	
Frequency of Use:	
Flow of Events:	1. User chooses " Create Group" under "Home" Page 2. The system prompts the user to input a group name. 3. The user inputs group name and select "Create Group" 4. System generates a randomised 8 character alphanumeric code for the new group. 5. System prompts user to add a nickname 6. The page will display a group with the group name and the randomised code.
Alternative Flows:	2.1-AF-S1: User doesn't input a group name 1. The system prompts the user "There needs to be a group name." . 2. Return to step 2 2.2-AF-S2: User does not input a nickname 1. System uses name from gmail as username 2. Return to step 6.
Exceptions:	EX-S1: If user has more than X active groups 1. The system will prompt "You are not allowed to create more groups!" 2. The system returns to the main page.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	2.2		
Use Case Name:	Join Group		
Created By:	Boey, Tiffany	Last Updated By:	04.02.25
Date Created:	04.02.25	Date Last Updated:	04.02.25

Actor:	User
Description:	User joins a group
Preconditions:	Users have logged into their account
Postconditions:	Users join an existing group
Priority:	
Frequency of Use:	
Flow of Events:	1. User chooses “Join a Room” under “Home” Page 2. System prompts user to input a code 3. System searches for the code 4. System prompts the user to add a nickname. 5. User joins the group
Alternative Flows:	2.2-AF-S1: Code does not exist 1. System prompts user “Group does not exist!” 2. Return to step 2 2.2-AF-S2: User does not input a nickname 1. System uses name from gmail as username 2. Return to step 5.
Exceptions:	EX-S1: User selects “<=” 1. Return to previous page
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	3.2		
Use Case Name:	Select Date, Time, Food and Location Preferences ?		
Created By:	Boey, Tiffany	Last Updated By:	04.02.25
Date Created:	04.02.25	Date Last Updated:	04.02.25

Actor:	User
Description:	Users input their availability (time), food preferences, and location for a group outing and submits the information.
Preconditions:	Users joined a group
Postconditions:	The user's time availability, food preferences, and location are successfully submitted. The group submission status is updated.
Priority:	
Frequency of Use:	
Flow of Events:	1. After creating/joining a group, the system redirects the user to a group chat with other members. 2. The user interacts with the buttons on the left panel: a. "Add Time" to input their available date and time. b. "Add Food" to specify top 5 food cuisine preferences. c. "Add Location" to suggest or select their top 3 preferred locations. 3. Users select their available date and time. 4. Users specify their food preferences. 5. Users select a preferred location. 6. The user selects the “Submit” button. 7. The system validates the input: Ensures all required fields (time, food, location) are completed. 8. If all validations pass, the system updates the submission status (e.g., "3/5 has submitted!") and confirms the user's input. 9. Once all group members have submitted, the system checks for overlapping availability across all inputs. 10. If a matching date and time are found: - The system generates a message in the group chat: "Date and Time: DD/MM/YY HH:MM." 11. If no coinciding date and time are found: - The system notifies the group: "No common date and time available. Please review your submissions and update your availability."
Alternative Flows:	3.2-AF-S1: User presses "Submit" without completing all required fields (time, food, location). - The system prompts, “Please complete all required fields (time, food, location) before submitting!” and returns to step 2.
Exceptions:	3.2-EX-S1: User selects “<=” 1. System returns to the previous page
Includes:	
Special Requirements:	

Assumptions:	
Notes and Issues:	

Use Case ID:	4.1		
Use Case Name:	Regenerate Schedule		
Created By:	Boey, Tiffany	Last Updated By:	04.02.25
Date Created:	04.02.25	Date Last Updated:	04.02.25

Actor:	User
Description:	User reschedules a previously generated schedule if the proposed date and time do not work for the group.
Preconditions:	A schedule has already been generated. All group members have submitted their time, food, and area preferences.
Postconditions:	A new schedule is successfully regenerated and displayed to all group members.
Priority:	
Frequency of Use:	
Flow of Events:	1. A previously generated schedule is displayed as a pinned message: "System Generated Message: Date and Time: [DD/MM/YY HH:MM]." 2. User selects the "Regenerate Time/Date" button. 3. The system prompts: "Generating a new schedule based on preferences." 4. The system recalculates: ? 5. Finds the next most common available time and location based on group preferences. 6. Updates the food selection if necessary. 7. The system displays a new pinned message: "Updated System Generated Message: Date and Time: [DD/MM/YY HH:MM]." 8. The group reviews the updated schedule. 9. If users are satisfied, they select the "Proceed" button to finalize the schedule else they can select "Regenerate Time/Date" til they are satisfied.
Alternative Flows:	4.1-AF-S1: There are no other common available time/date 1. The system prompts: "There are no common available dates/times for all." 4.2-AF-S2: User selects "Regenerate Time/Date" repeatedly. 1. The system continues to calculate and display the next best options until all possibilities are exhausted.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	5.1		
Use Case Name:	Voting food options		
Created By:	Boey, Tiffany	Last Updated By:	04.02.25
Date Created:	04.02.25	Date Last Updated:	04.02.25

Actor:	User
Description:	User upvote/downvote food options. They can view reviews for each food option before voting.
Preconditions:	A schedule has been generated at least once. Food options are available for voting.
Postconditions:	Users successfully upvote or downvote food options.
Priority:	
Frequency of Use:	
Flow of Events:	1. User clicks the "Vote Food" button. 2. The system redirects the user to a page displaying the list of food options. 3. Users can: - Upvote a food option by clicking the up arrow (+1). - Downvote a food option by clicking the down arrow (-1). 4. Users can click on any food option to view its reviews, including ratings and comments. 5. After finalizing their votes, users click the "Submit" button. 6. The system: - Records the user's votes real-time. - Updates the group submission count (e.g., "3/5 members have submitted"). - Redirects the user back to the main group page. 7. Once all group members complete their votes for food options by clicking "Submit." The system calculates the highest-voted food option. 8. The system generates and pins a message: "System Generated Message: Date and Time: [DD/MM/YY HH:MM], Location: [Selected Restaurant]."
Alternative Flows:	5.1-AF-S1: A user does not cast any votes and tries to click "Submit." 1. The system prompts, “Please vote for at least one food option before submitting.” 2. Return to step 3.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	5.2		
Use Case Name:	Regenerate Food Option		
Created By:	Boey, Tiffany	Last Updated By:	04.02.25
Date Created:	04.02.25	Date Last Updated:	04.02.25

Actor:	User
Description:	The system allows users to vote for food options, chooses a top-voted food option for the users. System also resolves conflicts in case of tied votes, and confirms the selected food option.
Preconditions:	Schedule has been generated at least once
Postconditions:	System display a food option
Priority:	
Frequency of Use:	
Flow of Events:	1. A previously generated system message is displayed as a pinned message: "System Generated Message: Date and Time: [DD/MM/YY HH:MM], Location: [Selected Restaurant]." 2. If users are satisfied with the pinned message: - They click the "Proceed" button to confirm the schedule. 3. If users are not satisfied with the pinned message: - They tap the "Regenerate Food" button. 4. The system recalculates and generates the next best food option based on the votes or ratings. 5. The pinned message updates with the new food option. 6. Users review the updated selection and either re-enter "proceed" or "regenerate" until satisfied.
Alternative Flows:	5.2-AF-S1: No food option receives votes. 1. The system selects a food option based on the highest overall rating and generates a pinned message. 5.2-AF-S2: Food options are tied in upvotes 2. A spin the wheel will appear 3. The group owner will spin the wheel 4. A random food option will be displayed 5. Users can agree or disagree with the randomly chosen option 6. If all users disagree, the option is eliminated and re-spin the wheel; else, the option will be displayed as the final choice 7. Return to step 2
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	7.1		
Use Case Name:	Add to Calendar		
Created By:	Boey, Tiffany	Last Updated By:	04.02.25
Date Created:	04.02.25	Date Last Updated:	04.02.25

Actor:	User
Description:	User adds schedule to calendar
Preconditions:	Schedule has been generated at least once
Postconditions:	Users' device downloads the .ics file.
Priority:	
Frequency of Use:	
Flow of Events:	1. User selects "Add to Calendar" 2. System sends a confirmation email and .ics attachment with the email 3. If .ics file is utilised by user, it will remind user a day before the event
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	9.1		
Use Case Name:	Exit Room		
Created By:	Boey, Tiffany	Last Updated By:	04.02.25
Date Created:	04.02.25	Date Last Updated:	04.02.25

Actor:	User
Description:	User exits a group
Preconditions:	Users is member of the group
Postconditions:	Users exits the group
Priority:	
Frequency of Use:	
Flow of Events:	1. User selects a room by right clicking on a group row 2. User selects “Exit Room” button 3. The system prompts “Do you want to exit this Room?” 4. The user chooses “Exit” 5. The system deletes the room for the user and user’s related data in the group
Alternative Flows:	
Exceptions:	2.3-EX-S1: User selects “Cancel” or “x” 1. The system returns to the data page
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	9.2		
Use Case Name:	Close Room		
Created By:	Boey, Tiffany	Last Updated By:	04.02.25
Date Created:	04.02.25	Date Last Updated:	04.02.25

Actor:	User
Description:	User closes Room.
Preconditions:	The outing has been completed.
Postconditions:	The room is successfully closed, and all related interactions are disabled. A final confirmation message is displayed to users.
Priority:	
Frequency of Use:	
Flow of Events:	<u>Manual Room Closure</u> 1. After the outing, the system displays the "Close Room" button on the group chat interface. 2. The user selects the "Close Room" button. 3. The system prompts for confirmation:"Are you sure you want to close the room? Once closed, no further actions can be performed." 4. If the user confirms: 5. The system closes the room. 6. The system displays a final message: "The room has been closed. Thank you for using [App Name]!" 7. All interactions in the group chat, such as voting and reviews, are disabled. 8. If the user cancels: 9. The system keeps the room open. <u>Automatic Room Closure</u> 1. One week after the outing, the system checks the room's status. 2. If the room is still open, the system automatically closes it. 3. The system generates a final message:"The room has been automatically closed after one week. Thank you for using [App Name]!" 4. All interactions in the group chat are disabled.
Alternative Flows:	7.1-AF-S1: User clicks "Leave a Review" but exits without submitting. 1. The system does not record any feedback and maintains the pinned review reminder. 7.2-AF-S1: 1. The user attempts to close the room before all members have submitted reviews.

	2. System Response: "Not all members have submitted reviews. Are you sure you want to close the room?" 3. If the user confirms, proceed to step 4. 4. If the user cancels, return to step 2.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	10.1		
Use Case Name:	Add Friends		
Created By:	Boey, Tiffany	Last Updated By:	04.02.25
Date Created:	04.02.25	Date Last Updated:	04.02.25

Actor:	User
Description:	User adds a friend to their friends list by entering the friend's unique Member ID.
Preconditions:	1. The user must be logged into their account. 2. The system must provide the user with their unique Member ID. 3. The user must know the Member ID of the friend they wish to add.
Postconditions:	The specified Member ID is successfully added to the user's friends list.
Priority:	
Frequency of Use:	
Flow of Events:	1. The user chooses "Add Friend by Member ID" under the "Friends" page 2. The system displays the "Add Friend by Member ID" screen with an input field for the Member ID. 3. The user enters the Member ID of the friend and clicks the "Enter" button. 4. The system verifies if the entered Member ID exists: 5. If valid: The system adds the friend to the user's friends list and displays a confirmation message: "Friend added successfully!" 6. If invalid: The system displays an error message: "Member ID not found. Please try again."
Alternative Flows:	2.1-AF-S1: The user does not input a Member ID and clicks "Enter." 1. The system displays a message: "You have no past rooms to display." 2.2-AF-S2: The user enters a Member ID that already exists in their friends list. 1. The system displays an error message: "This Member ID is already your friend."
Exceptions:	EX-S1: A system error occurs during the friend addition process (e.g., network issue). 1. The system displays an error message: "Unable to add friend at this time. Please try again later." 2. Return to the "Friends" page.
Includes:	
Special Requirements:	
Assumptions:	

Notes and Issues:	

Use Case ID:	10.2		
Use Case Name:	View Friends		
Created By:	Boey, Tiffany	Last Updated By:	04.02.25
Date Created:	04.02.25	Date Last Updated:	04.02.25

Actor:	User
Description:	User views a list of their added friends
Preconditions:	1. The user must be logged into their account. 2. The user must have at least one friend added to their friends list.
Postconditions:	Users view their current list of friends.
Priority:	
Frequency of Use:	
Flow of Events:	1. The user chooses “View Friends” under the "Friends" page 2. The system retrieves and displays a list of friends, including: a. Friend's name b. Friend's Member ID c. Avatar/profile picture 3. The user can scroll through the list to view all their friends.
Alternative Flows:	2.1-AF-S1: The user has no friends added to their list. 2. The system displays a message: "You have no friends in your list yet. Use 'Add Friend by Member ID' to get started!"
Exceptions:	EX-S1: A system error occurs while retrieving the friends list (e.g., network issue). 3. The system displays an error message: "Unable to load friends list. Please try again later." 4. Return to the "Friends" page.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	10.3		
Use Case Name:	Invite Friend		
Created By:	Boey, Tiffany	Last Updated By:	04.02.25
Date Created:	04.02.25	Date Last Updated:	04.02.25

Actor:	User
Description:	User invites friends to join the group by sharing the group PIN.
Preconditions:	The user has created or joined a group. The user is authenticated and logged in.
Postconditions:	Invited friends receive the group PIN and can join the group.
Priority:	
Frequency of Use:	
Flow of Events:	1. User selects the "Invite" button in the group chat interface. 2. The system displays the group PIN (e.g., "PIN: 1234ABCD") and a prompt to share it with others. 3. User copies the PIN or shares it via external applications (e.g., messaging or email). 4. Invited friends use the PIN to join the group through the "Join Group" option on the home page.
Alternative Flows:	2.5-AF-S1: User clicks "Invite" but does not share the PIN. - The system keeps the PIN visible in the group chat interface for future sharing.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	11.1		
Use Case Name:	View All Schedules/ Current Rooms Joined		
Created By:	Boey, Tiffany	Last Updated By:	04.02.25
Date Created:	04.02.25	Date Last Updated:	04.02.25

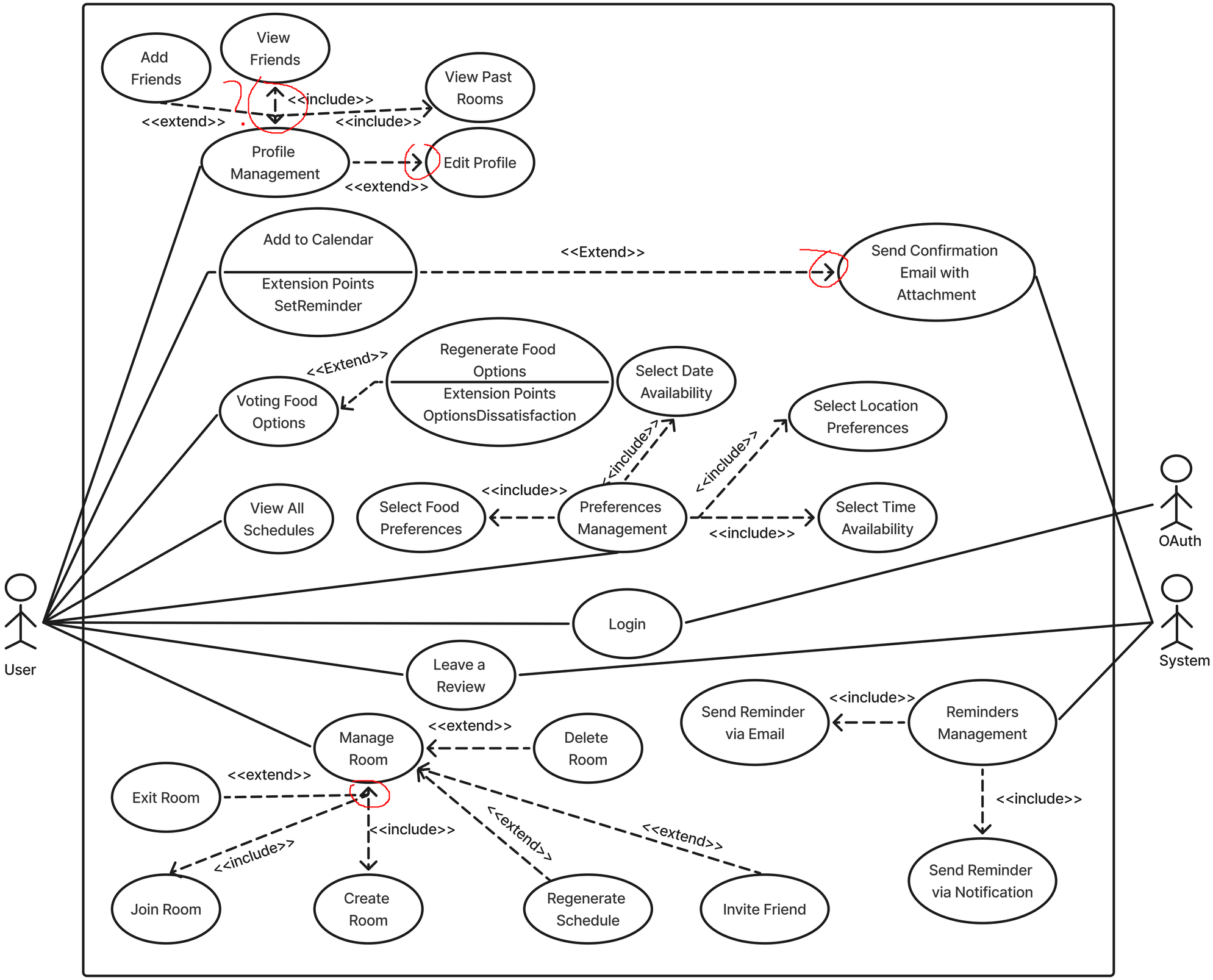
Actor:	User
Description:	User views all generated schedules
Preconditions:	Users is authenticated and logged in
Postconditions:	Users views all schedules
Priority:	
Frequency of Use:	
Flow of Events:	1. User selects “Joined Rooms” under “Home” Page 2. The system pops up a box that shows all schedules generated from all groups the user is in currently, in sequence of date and time
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

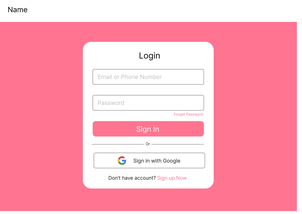
Use Case ID:	12.1		
Use Case Name:	Edit Profile		
Created By:	Boey, Tiffany	Last Updated By:	04.02.25
Date Created:	04.02.25	Date Last Updated:	04.02.25

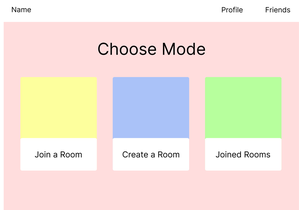
Actor:	User
Description:	User edits their profile
Preconditions:	Users have logged into their account
Postconditions:	User's Profile is updated
Priority:	
Frequency of Use:	
Flow of Events:	1. The user chooses "Edit Profile" under the "Profile" page. 2. The system displays the current profile with editable fields (e.g., name and profile picture). 3. The user updates the name by entering it in the input field and clicking "Submit." 4. If the user clicks the profile picture edit icon, they can upload or change their profile picture. 5. The system saves the changes and updates the profile. 6. The updated profile is displayed, including the new name and/or profile picture.
Alternative Flows:	2.1-AF-S1: User does not input a new name. 1. The system retains the current name and displays a message: "Name cannot be blank. Please enter a valid name." 2.2-AF-S2: The user does not upload a new profile picture. 1. The system retains the current profile picture.
Exceptions:	EX-S1: System error during the update process (e.g., failed image upload or server issue). 1. The system displays an error message: "Failed to update profile. Please try again later." 2. Return to the "Profile" page.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

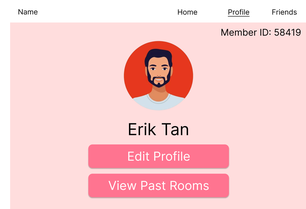
Use Case ID:	12.2		
Use Case Name:	View Past Rooms		
Created By:	Boey, Tiffany	Last Updated By:	04.02.25
Date Created:	04.02.25	Date Last Updated:	04.02.25

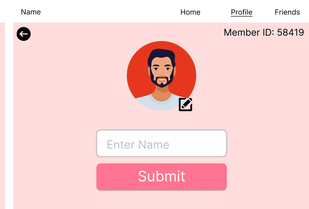
Actor:	Group Owner
Description:	User view past rooms
Preconditions:	Users have logged into their account
Postconditions:	User view past rooms
Priority:	
Frequency of Use:	
Flow of Events:	1. The user chooses “View Past Rooms” under the "Profile" page. 2. The system retrieves and displays a list of past rooms. 3. Each entry includes: - Room Name - Room PIN - Number of Participants - Date & Time - Date room was closed 4. The user can scroll through the list to view multiple past rooms.
Alternative Flows:	2.1-AF-S1: The user has no past rooms. 1. The system displays a message: "You have no past rooms to display."
Exceptions:	EX-S1: A system error occurs while retrieving past rooms (e.g., a network issue). 1. The system displays an error message: "Failed to load past rooms. Please try again later." 2. Return to the "Profile" page.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

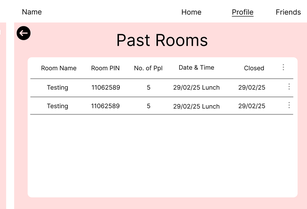

Add Friends
View Friends
View Past Rooms
<<include>>
<<extend>>
<<include>>
Profile Management
Edit Profile
<<extend>>
Add to Calendar
Extension Points
SetReminder
<<Extend>>
Send Confirmation Email with Attachment
<<Extend>>
Regenerate Food Options
Extension Points
OptionsDissatisfaction
Voting Food Options
Select Date Availability
Select Location Preferences
<<include>>
<<include>>
View All Schedules
Select Food Preferences
<<include>>
Preferences Management
<<include>>
Select Time Availability
OAuth
User
Login
System
Leave a Review
Send Reminder via Email
<<include>>
Reminders Management
Manage Room
<<extend>>
Delete Room
Exit Room
<<extend>>
<<include>>
<<include>>
<<extend>>
<<extend>>
<<include>>
Send Reminder via Notification
Join Room
Create Room
Regenerate Schedule
Invite Friend

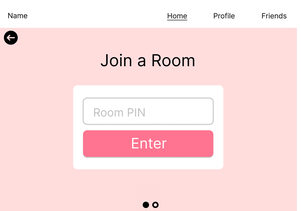

Name
Login
Email or Phone Number
Password
Forgot Password
Sign In
Or
Sign in with Google
Don't have account? Sign up Now

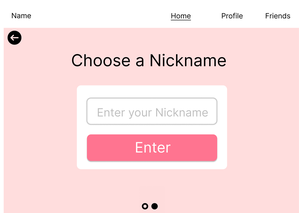

Name
Profile
Friends
Choose Mode
Join a Room
Create a Room
Joined Rooms

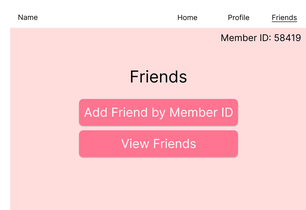

Name
Home
Profile
Friends
Member ID: 58419
Erik Tan
Edit Profile
View Past Rooms

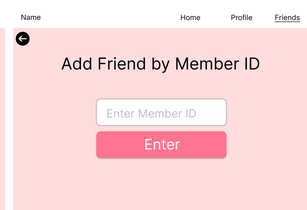

Name
Home
Profile
Friends
Member ID: 58419
Enter Name
Submit

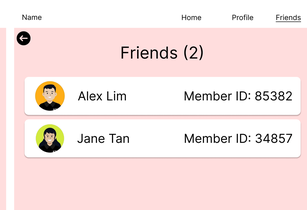

Name
Home
Profile
Friends
Past Rooms
Room Name
Room PIN
No. of Ppl
Date & Time
Closed
Testing
11062589
5
29/02/25 Lunch
29/02/25
Testing
11062589
5
29/02/25 Lunch
29/02/25

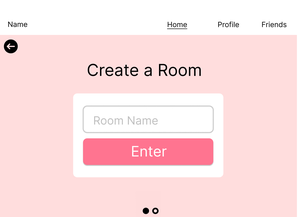

Name
Home
Profile
Friends
Join a Room
Room PIN
Enter

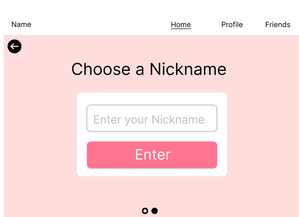

Name
Home
Profile
Friends
Choose a Nickname
Enter your Nickname
Enter

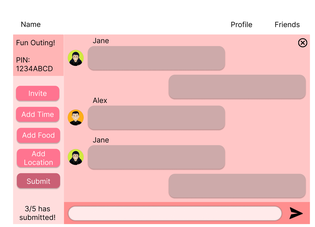

Name
Home
Profile
Friends
Member ID: 58419
Friends
Add Friend by Member ID
View Friends

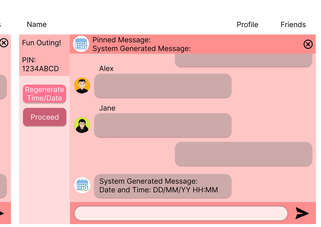

Name
Home
Profile
Friends
Add Friend by Member ID
Enter Member ID
Enter

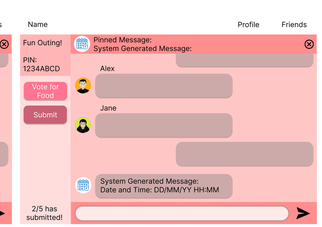

Name
Home
Profile
Friends
Friends (2)
Alex Lim
Member ID: 85382
Jane Tan
Member ID: 34857

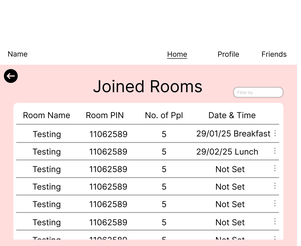

Name
Home
Profile
Friends
Create a Room
Room Name
Enter

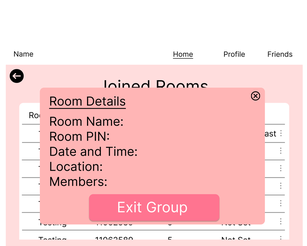

Name
Home
Profile
Friends
Choose a Nickname
Enter your Nickname
Enter

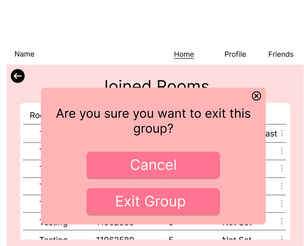

Name
Profile
Friends
Fun Outing!
PIN: 1234ABCD
Jane
Alex
Jane
Invite
Add Time
Add Food
Add Location
Submit
3/5 has submitted!

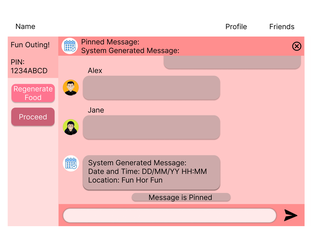

Name
Profile
Friends
Fun Outing!
PIN: 1234ABCD
Pinned Message:
System Generated Message:
Alex
Jane
Regenerate Time/Date
Proceed
System Generated Message:
Date and Time: DD/MM/YY HH:MM

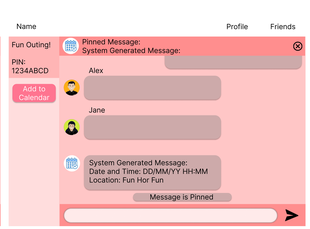

Name
Profile
Friends
Fun Outing!
PIN: 1234ABCD
Pinned Message:
System Generated Message:
Alex
Jane
Vote for Food
Submit
System Generated Message:
Date and Time: DD/MM/YY HH:MM
2/5 has submitted!

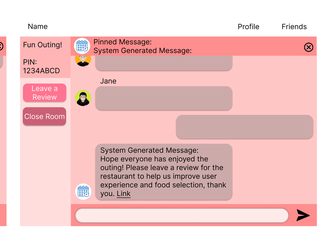

Name
Home
Profile
Friends
Joined Rooms
Filter by
Room Name
Room PIN
No. of Ppl
Date & Time
Testing
11062589
5
29/01/25 Breakfast
Testing
11062589
5
29/02/25 Lunch
Testing
11062589
5
Not Set
Testing
11062589
5
Not Set
Testing
11062589
5
Not Set
Testing
11062589
5
Not Set

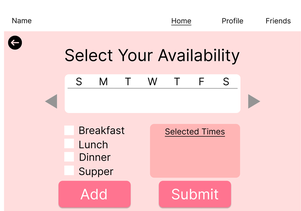

Name
Home
Profile
Friends
Room Details
Room Name:
Room PIN:
Date and Time:
Location:
Members:
Exit Group

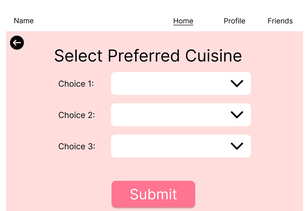

Name
Home
Profile
Friends
Are you sure you want to exit this group?
Cancel
Exit Group

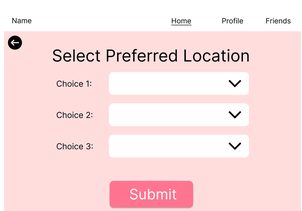

Name
Profile
Friends
Fun Outing!
PIN: 1234ABCD
Pinned Message:
System Generated Message:
Alex
Jane
Regenerate Food
Proceed
System Generated Message:
Date and Time: DD/MM/YY HH:MM
Location: Fun Hor Fun
Message is Pinned

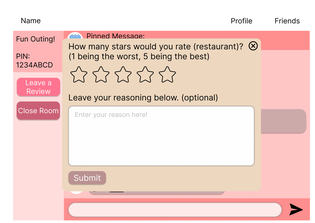

Name
Profile
Friends
Fun Outing!
PIN: 1234ABCD
Pinned Message:
System Generated Message:
Alex
Jane
Add to Calendar
System Generated Message:
Date and Time: DD/MM/YY HH:MM
Location: Fun Hor Fun
Message is Pinned

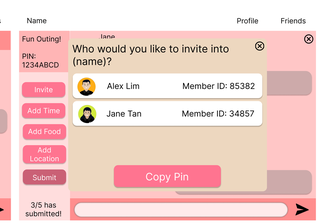

Name
Profile
Friends
Fun Outing!
PIN: 1234ABCD
Pinned Message:
System Generated Message:
Jane
Leave a Review
Close Room
System Generated Message:
Hope everyone has enjoyed the outing! Please leave a review for the restaurant to help us improve user experience and food selection, thank you. Link

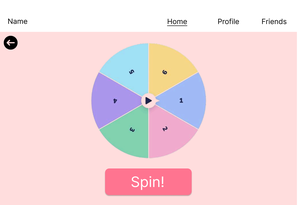

Name
Home
Profile
Friends
Select Your Availability
S M T W T F S
Breakfast
Lunch
Dinner
Supper
Selected Times
Add
Submit

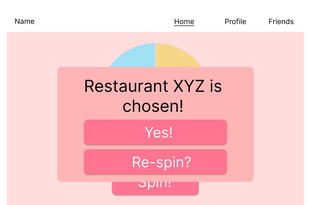

Name
Home
Profile
Friends
Select Preferred Cuisine
Choice 1:
Choice 2:
Choice 3:
Submit

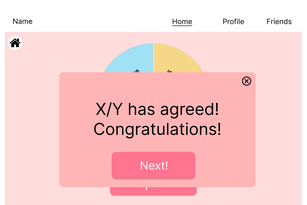

Name
Home
Profile
Friends
Select Preferred Location
Choice 1:
Choice 2:
Choice 3:
Submit

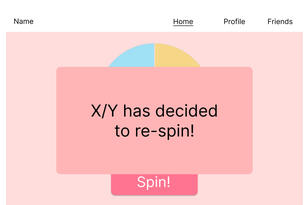

Name
Profile
Friends
Fun Outing!
PIN: 1234ABCD
Pinned Message:
How many stars would you rate (restaurant)?
(1 being the worst, 5 being the best)
Leave a Review
Close Room
Leave your reasoning below. (optional)
Enter your reason here!
Submit

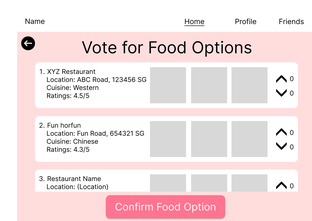

Name
Profile
Friends
Fun Outing!
PIN: 1234ABCD
Who would you like to invite into (name)?
Alex Lim
Member ID: 85382
Jane Tan
Member ID: 34857
Invite
Add Time
Add Food
Add Location
Submit
Copy Pin
3/5 has submitted!

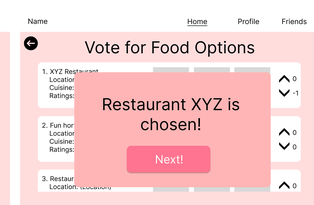

Name
Home
Profile
Friends
Spin!

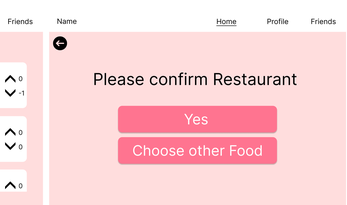

Name
Home
Profile
Friends
Restaurant XYZ is chosen!
Yes!
Re-spin?

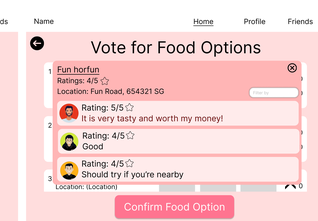

Name
Home
Profile
Friends
X/Y has agreed! Congratulations!
Next!


Name
Home
Profile
Friends
X/Y has decided to re-spin!
Spin!


Name
Home
Profile
Friends
Vote for Food Options
1. XYZ Restaurant
Location: ABC Road, 123456 SG
Cuisine: Western
Ratings: 4.5/5
2. Fun horfun
Location: Fun Road, 654321 SG
Cuisine: Chinese
Ratings: 4.3/5
3. Restaurant Name
Location: (Location)
Confirm Food Option


Name
Home
Profile
Friends
Vote for Food Options
Restaurant XYZ is chosen!
Next!


Name
Home
Profile
Friends
Please confirm Restaurant
Yes
Choose other Food


Name
Home
Profile
Friends
Vote for Food Options
Fun horfun
Ratings: 4/5
Location: Fun Road, 654321 SG
Rating: 5/5
It is very tasty and worth my money!
Rating: 4/5
Good
Rating: 4/5
Should try if you're nearby
Confirm Food Option