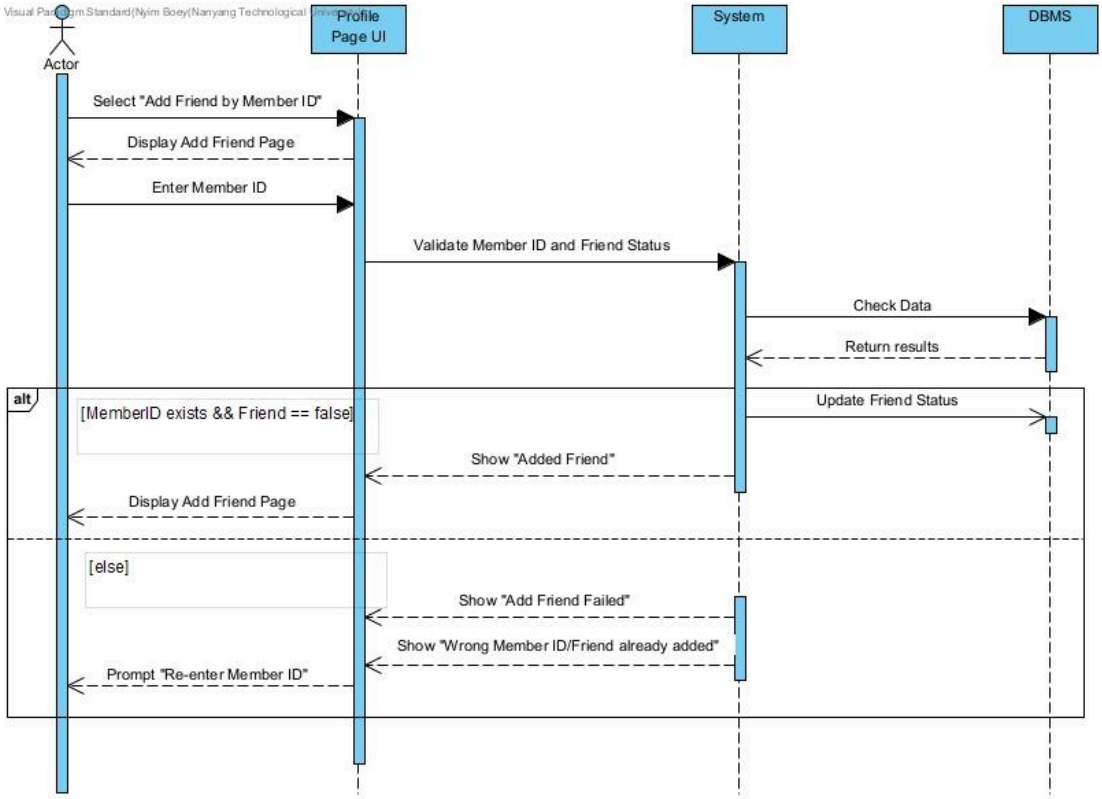
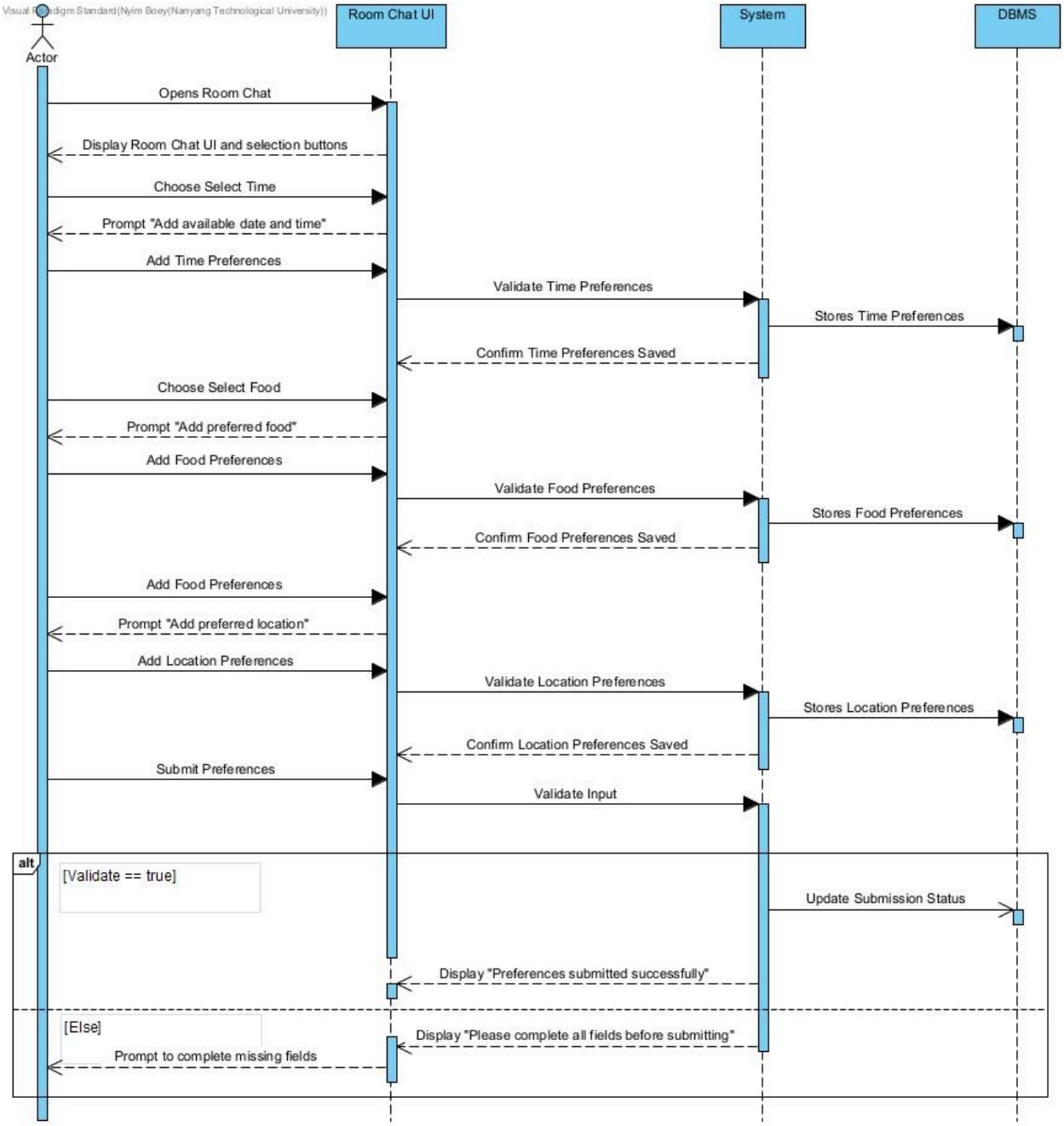
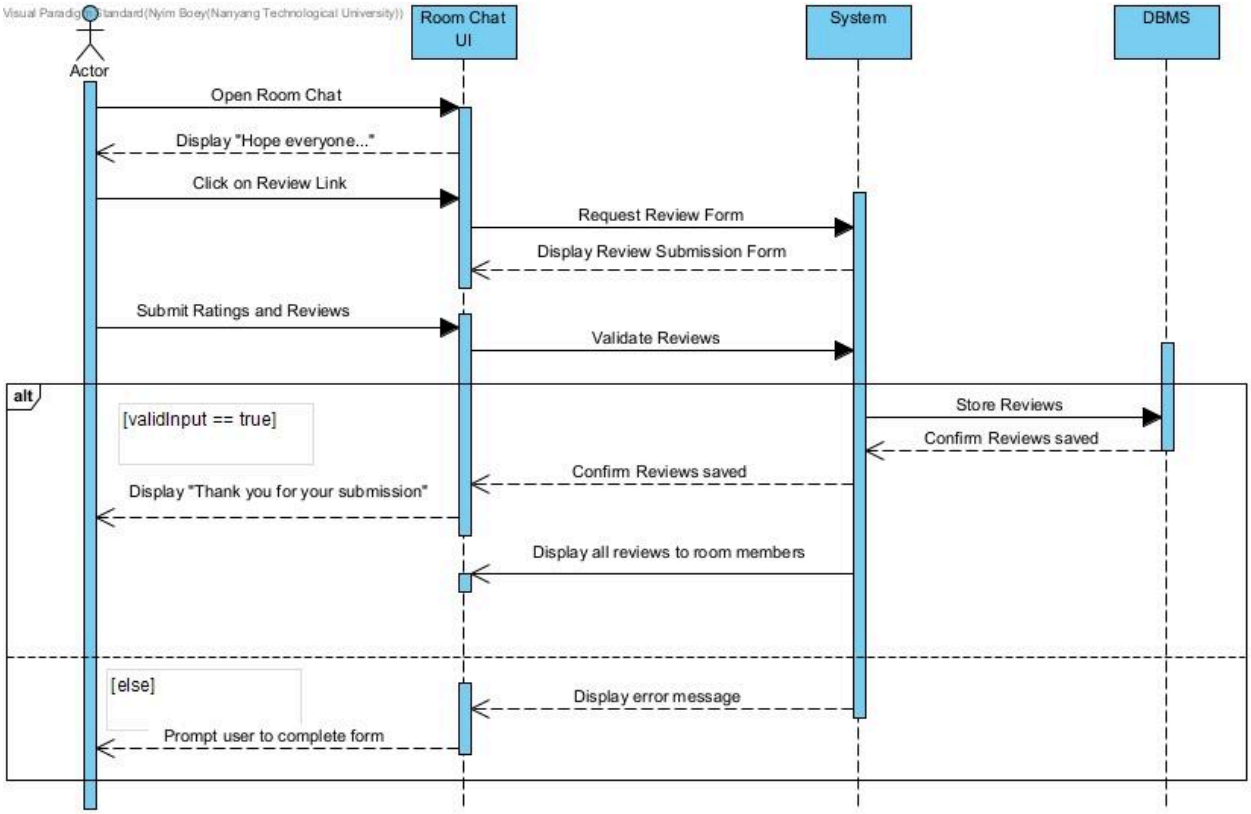
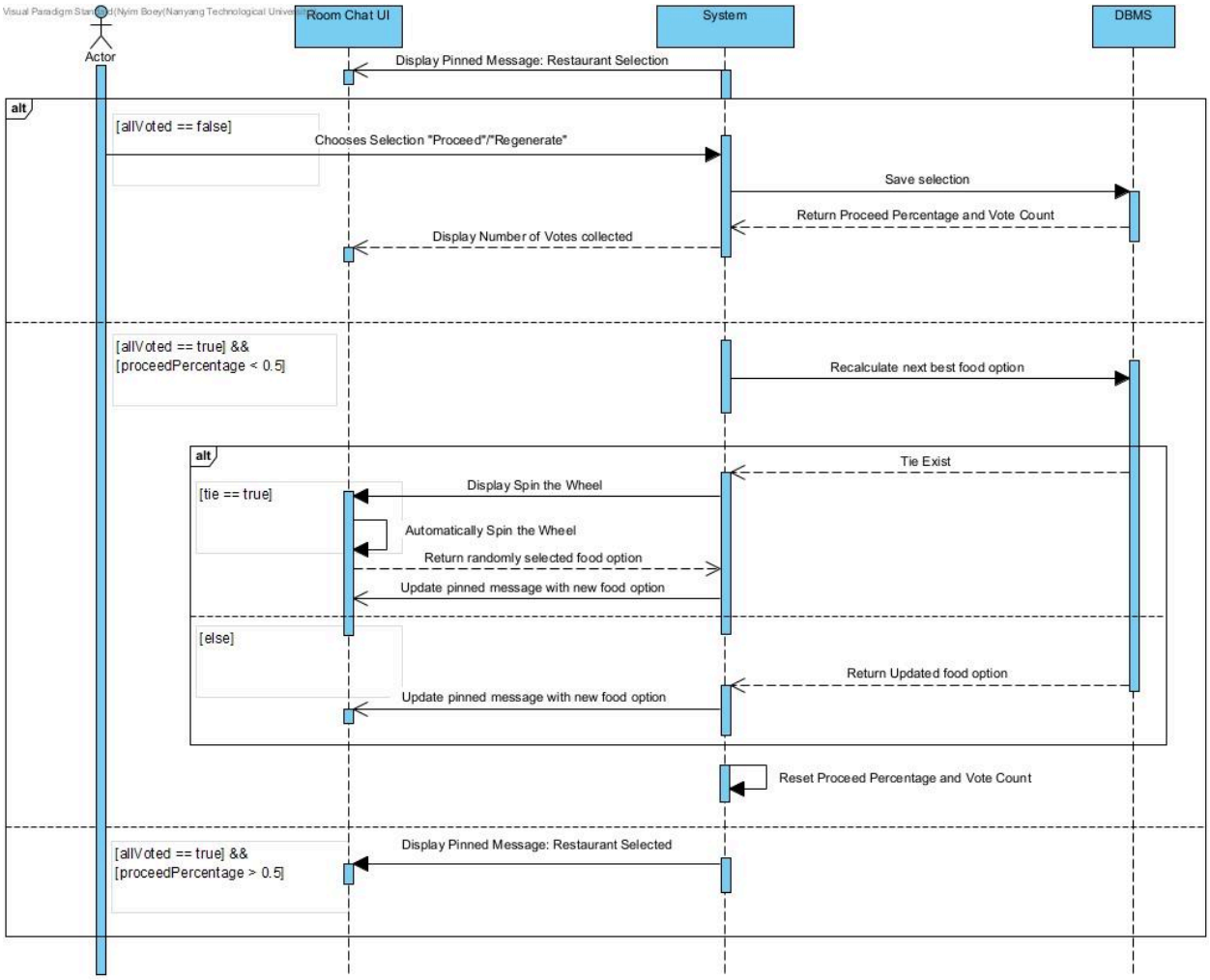
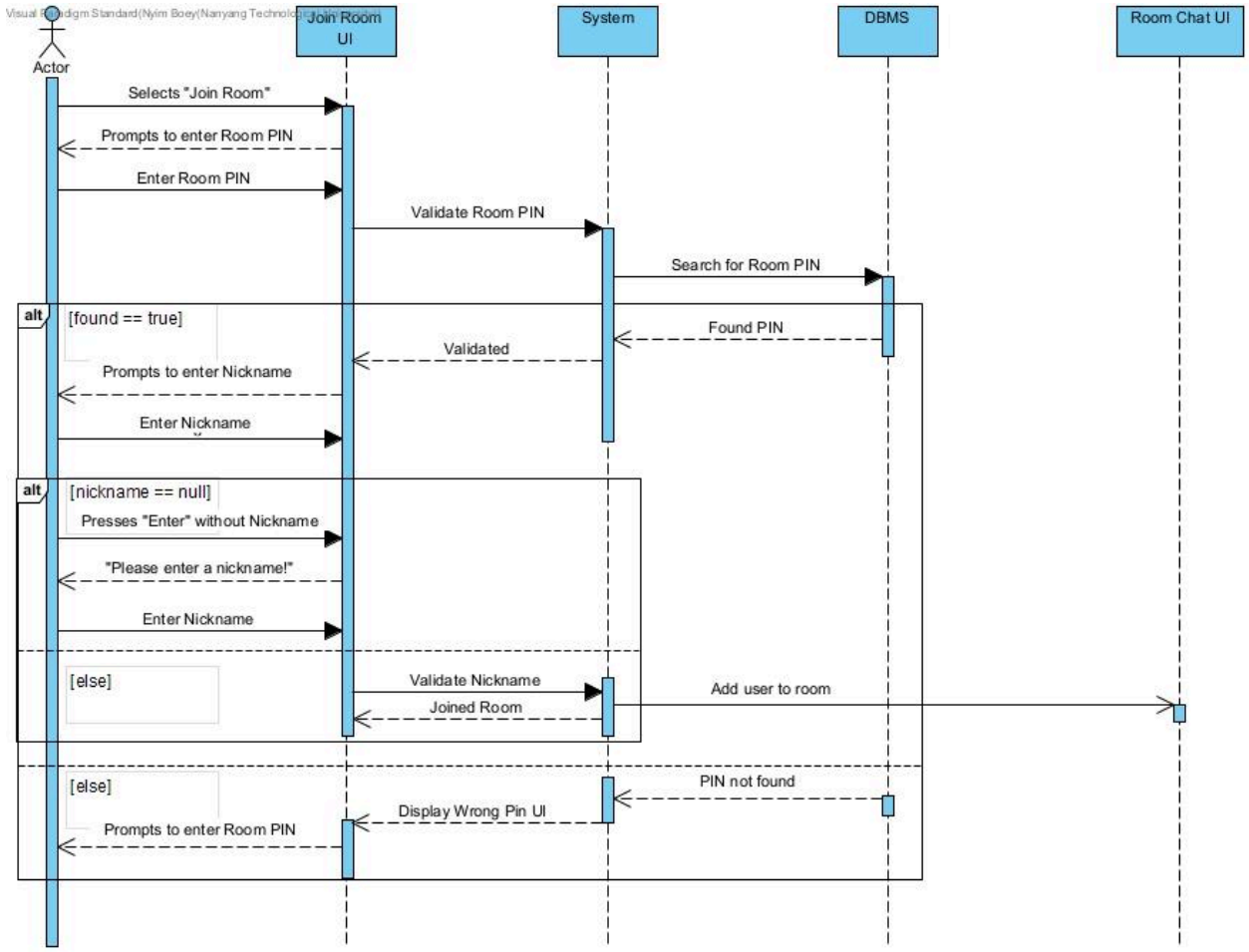
Sequence diagram:



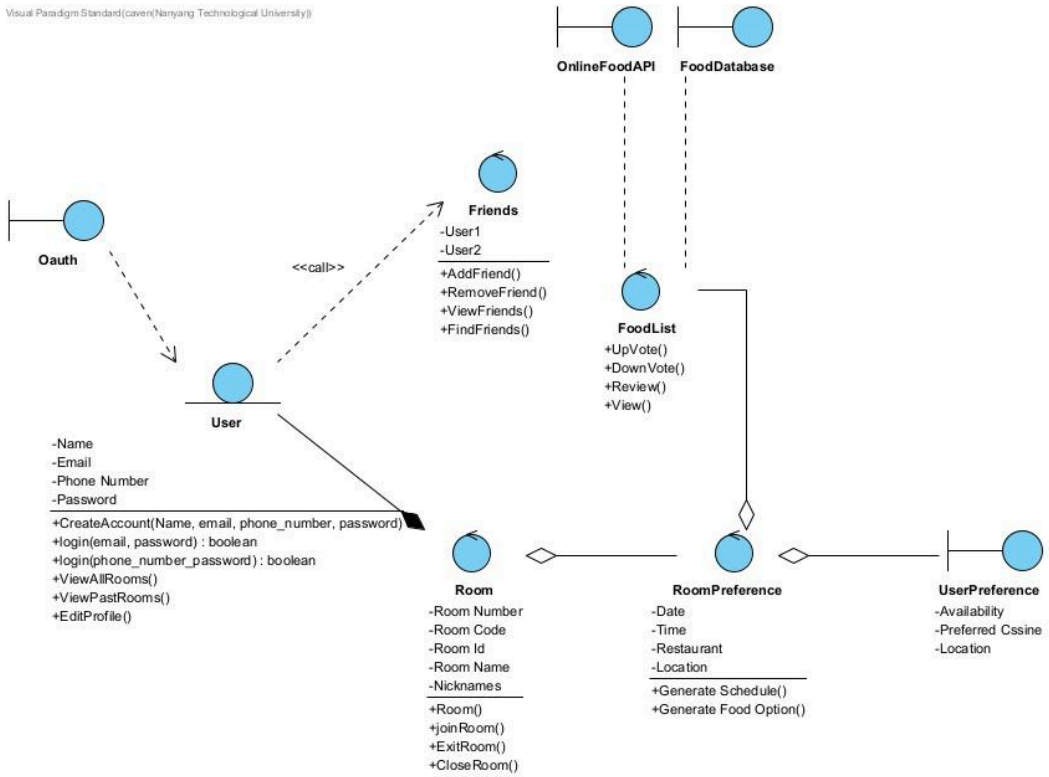

Visual Paradigm Standard(Nyim Boey(Nanyang Technological University))
Actor
Room Chat UI
System
DBMS
Opens Room Chat
Display Room Chat UI and selection buttons
Choose Select Time
Prompt "Add available date and time"
Add Time Preferences
Validate Time Preferences
Stores Time Preferences
Confirm Time Preferences Saved
Choose Select Food
Prompt "Add preferred food"
Add Food Preferences
Validate Food Preferences
Stores Food Preferences
Confirm Food Preferences Saved
Add Food Preferences
Prompt "Add preferred location"
Add Location Preferences
Validate Location Preferences
Stores Location Preferences
Confirm Location Preferences Saved
Submit Preferences
Validate Input
alt
[Validate == true]
Update Submission Status
Display "Preferences submitted successfully"
[Else]
Display "Please complete all fields before submitting"
Prompt to complete missing fields

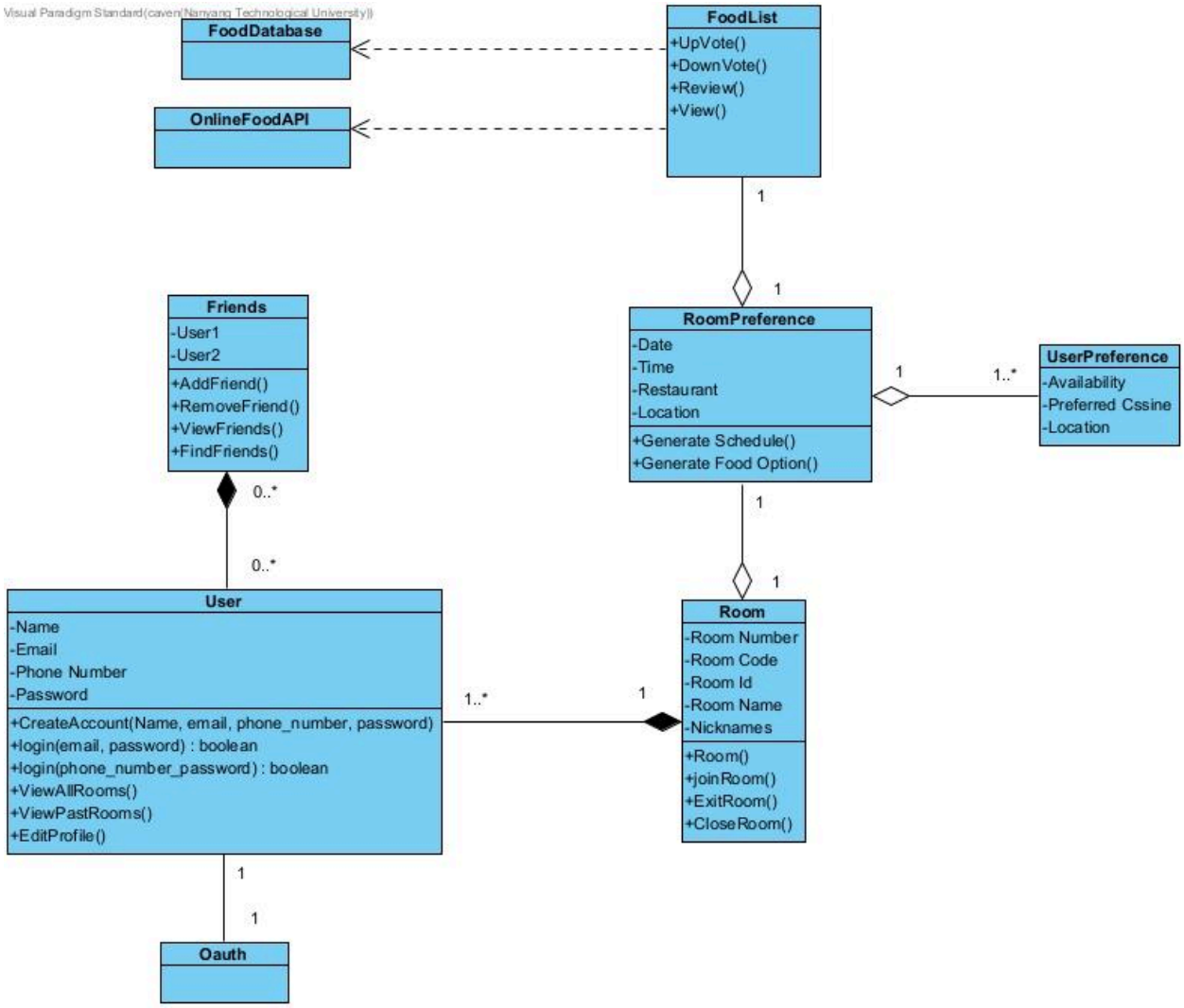

Visual Paradigm Standard(Nyim Boey(Nanyang Technological University))
Actor
Room Chat UI
System
DBMS
Open Room Chat
Display "Hope everyone..."
Click on Review Link
Request Review Form
Display Review Submission Form
Submit Ratings and Reviews
Validate Reviews
alt
[validInput == true]
Store Reviews
Confirm Reviews saved
Confirm Reviews saved
Display "Thank you for your submission"
Display all reviews to room members
[else]
Display error message
Prompt user to complete form


Actor
Room Chat UI
System
DBMS
Display Pinned Message: Restaurant Selection
alt
[allVoted == false]
Chooses Selection "Proceed"/"Regenerate"
Save selection
Return Proceed Percentage and Vote Count
Display Number of Votes collected
[allVoted == true] &&
[proceedPercentage < 0.5]
Recalculate next best food option
alt
Tie Exist
[tie == true]
Display Spin the Wheel
Automatically Spin the Wheel
Return randomly selected food option
Update pinned message with new food option
[else]
Return Updated food option
Update pinned message with new food option
Reset Proceed Percentage and Vote Count
[allVoted == true] &&
[proceedPercentage > 0.5]
Display Pinned Message: Restaurant Selected

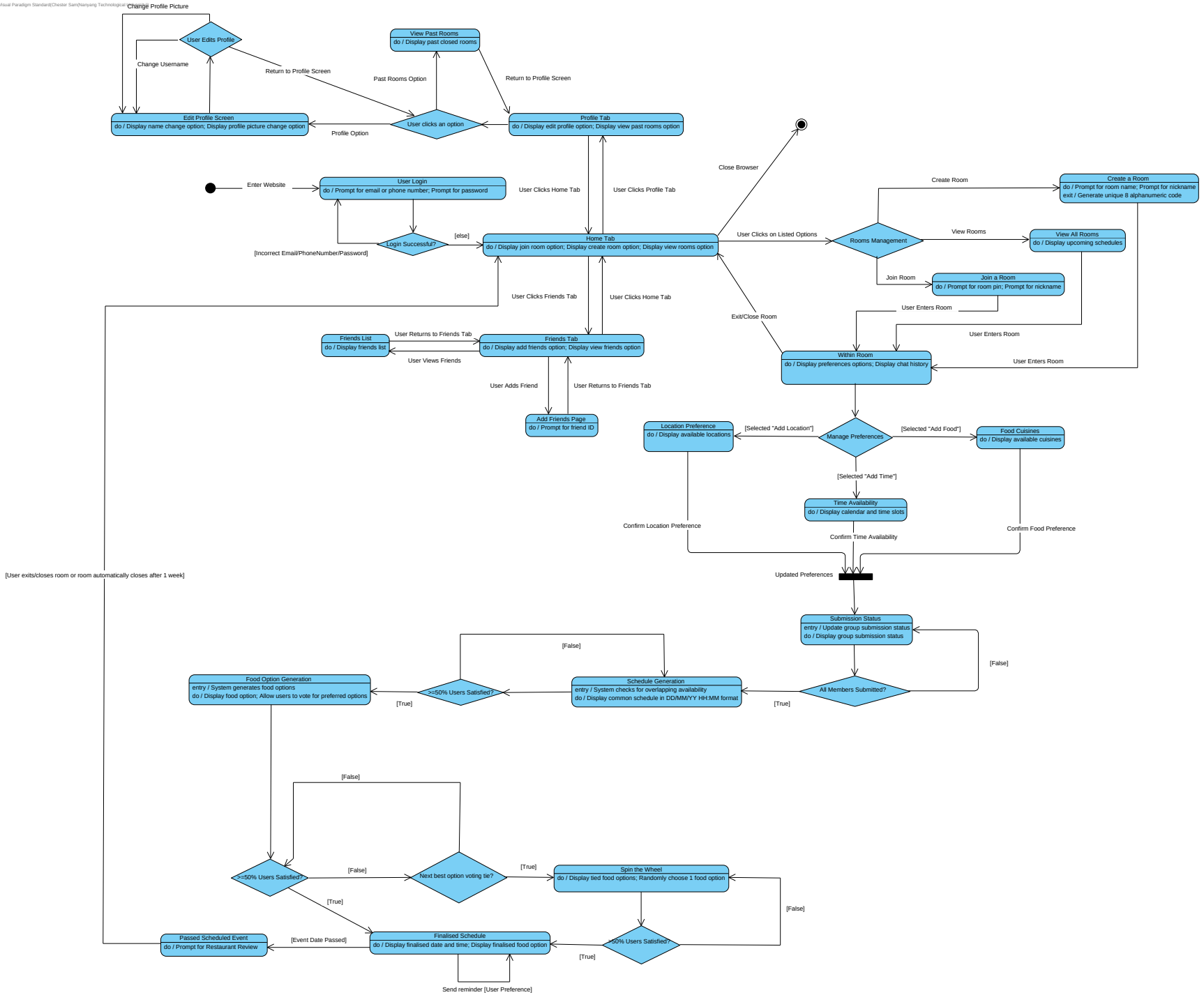

Actor
Join Room UI
System
DBMS
Room Chat UI
Selects "Join Room"
Prompts to enter Room PIN
Enter Room PIN
Validate Room PIN
Search for Room PIN
alt
[found == true]
Found PIN
Validated
Prompts to enter Nickname
Enter Nickname
alt
[nickname == null]
Presses "Enter" without Nickname
"Please enter a nickname!"
Enter Nickname
[else]
Validate Nickname
Joined Room
Add user to room
[else]
PIN not found
Display Wrong Pin UI
Prompts to enter Room PIN

Class diagram:




Visual Paradigm Standard(caven(Nanyang Technological University))
FoodDatabase
OnlineFoodAPI
FoodList
+UpVote()
+DownVote()
+Review()
+View()
1
1
Friends
-User1
-User2
+AddFriend()
+RemoveFriend()
+ViewFriends()
+FindFriends()
RoomPreference
-Date
-Time
-Restaurant
-Location
+Generate Schedule()
+Generate Food Option()
1
1..*
UserPreference
-Availability
-Preferred Cssine
-Location
0..*
0..*
1
1
User
-Name
-Email
-Phone Number
-Password
+CreateAccount(Name, email, phone_number, password)
+login(email, password) : boolean
+login(phone_number_password) : boolean
+ViewAllRooms()
+ViewPastRooms()
+EditProfile()
1..*
1
Room
-Room Number
-Room Code
-Room Id
-Room Name
-Nicknames
+Room()
+joinRoom()
+ExitRoom()
+CloseRoom()
1
1
Oauth

Visual Paradigm Standard(Chester Sam(Nanyang Technological University))



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
Tiffany	20.02.25	Revision for errors and additional use cases	2.0

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

1. 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 roomed 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:	Room 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 users to login and bring them to the main page.
Alternative Flows:	AF.3: Incorrect Email/PhoneNumber/Password 1. If Oauth returns unsuccessfully, the system prompts "Incorrect Email or Password" 2. Returns to step 2
Exceptions:	1.1.EX.2: No account 1. If the user does not have an account, the system will prompt "Account not found, please create an account!" 1.1.EX.3: Failed to login 5 times 1. User's account will be locked. 2. Reset password to unlock account
Includes:	
Special Requirements:	
Assumptions:	User has a gmail account
Notes and Issues:	

Use Case ID:	2.1		
Use Case Name:	Create 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 creates a room
Preconditions:	Users have logged into their account
Postconditions:	Users creates a new room The user is automatically entered into the room (via Use Case 3.3: Enter Room).
Priority:	

Frequency of Use:	
Flow of Events:	1. User chooses " Create Room" under "Home" Page 2. The system prompts the user to input a room name. 3. The user inputs room name and select "Create Room" 4. System generates a randomised 8 character alphanumeric code for the new room. 5. System prompts user to add a nickname 6. The page will display a room with the room name and the randomised code. 7. The system automatically triggers Use Case 4: Enter Room, allowing the user to enter the newly created room.
Alternative Flows:	AF.2: User doesn't input a room name 1. The system prompts the user "There needs to be a room name." 2. Return to step 2 AF.5: User does not input a nickname and presses "Enter" 1. System uses name from gmail as username 2. Return to step 6.
Exceptions:	2.1.EX.1: If user has more than 5 active rooms 1. The system will prompt "You are not allowed to create more rooms!" 2. The system returns to the main page.
Includes:	Use Case ID 4
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	3.1		
Use Case Name:	Join 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 joins a room
Preconditions:	Users have logged into their account
Postconditions:	Users join an existing room The user is entered into the room (via Use Case ID 4: Enter Room).
Priority:	
Frequency of Use:	
Flow of Events:	1. User chooses “Join a Room” under “Home” Page 2. System prompts user to input a room PIN 3. System searches for the room PIN 4. System prompts the user to add a nickname. 5. User enters the room (via Use Case ID 4: Enter Room)
Alternative Flows:	AF.2: Code does not exist 1. System prompts user “Room does not exist!” 2. Return to step 2 AF.4: User does not input a nickname and presses “Enter” 1. System prompts users “Enter a nickname!” 2. Return to step 4.
Exceptions:	3.1.EX.1-Any Step: User selects the "backward button" 1. Return to previous page
Includes:	Use Case ID 4
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	3.2		
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 room by sharing the room PIN.
Preconditions:	The user has created or joined a room. The user is authenticated and logged in.
Postconditions:	Invited friends receive the room PIN and can join the room.
Priority:	
Frequency of Use:	
Flow of Events:	1. User selects the "Invite" button in the room chat interface.

	2. The system displays the room 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 users use the PIN to join the room through the "Join Room" option on the home page.
Alternative Flows:	AF.3: User clicks "Invite" but does not share the PIN. - The system keeps the PIN visible in the room chat interface for future sharing.
Exceptions:	3.2.EX-Any Step: User selects the "backward button" 1. System returns to the previous page
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	3.3		
Use Case Name:	Select 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 room outing and submits the information.
Preconditions:	Users joined a room
Postconditions:	The user's time availability, food preferences, and location are successfully submitted. The room submission status is updated.
Priority:	
Frequency of Use:	
Flow of Events:	1. After creating/joining a room, the system redirects the user to a room 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 members have submitted!") and confirms the user's input. 9. Once all room 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 room chat: "Date and Time: DD/MM/YY HH:MM."
Alternative Flows:	AF.6: 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 - System returns to step 2. AF.9: No Common Date and Time Available - If no coinciding date and time are found, the system notifies the room: - "No common date and time available. Please review your submissions and update your availability." - System returns to step 2.
Exceptions:	3.3.EX-Any Step: User selects the "backward button" 2. System returns to the previous page
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	4.1		
Use Case Name:	Enter 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:	Users can access the room interface after joining, creating, or selecting a room from the "Joined Rooms" section. Upon entry, the system displays room details and enables chat functionality.
Preconditions:	The user must be logged into the application. The user must have joined, created, or selected a room from "Joined Rooms." The room must be active for the user to enter.
Postconditions:	If the room is active, the user is granted access and can view room details and participate in chat.
Priority:	
Frequency of Use:	
Flow of Events:	1. The user joins, creates, or selects a room from the "Joined Rooms" section. 2. The system displays Room Name (at the top left of the screen) and Unique Room PIN (for sharing with others). 3. The system includes "Room Chat Message" functionality (Use Case 4.2), allowing users to send and receive messages.

Alternative Flows:	
Exceptions:	4.1.EX-Any Step: User selects the "backward button" 3. System returns to the previous page 4.1.EX-1: The user has not joined any rooms. The system displays a message: "No rooms found. Join or create a room to get started!"
Includes:	Use Case ID 4.2
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	4.2		
Use Case Name:	Room Chat Message		
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 communicate with other members in the room by sending and receiving messages through the room chat system.
Preconditions:	The user must have joined an existing active room. The room must have at least one other participant to enable messaging.
Postconditions:	The message is successfully sent and displayed in the chat. The message includes the sender's name, profile picture, timestamp, and content. All members in the room can view the message in real time.
Priority:	
Frequency of Use:	
Flow of Events:	1. After entering a room, the system redirects the user to the room chat interface. 2. The user enters a message in the text input field. 3. The user presses the "Send" button. 4. The system checks if the message sent contains at least 1 character and does not exceed 500 characters. 5. If validation passes, the system displays the message in the chat window with the sender's name, profile picture, timestamp, and content. 6. The system sends the message to all room members in real time.
Alternative Flows:	AF.6: User has no internet connection while sending a message. - The system displays a "No connection. Message failed to send." prompt.

	- The system offers a "Retry" button. - Return to step 2 AF.7: User sends an empty message or invalid message - System displays an error message: "Message cannot be empty." - Return to step 2
Exceptions:	4.2.EX-Any Step: User selects the "backward button" 4. System returns to the previous page
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	5.1		
Use Case Name:	Generate Schedule		
Created By:	Boey, Tiffany	Last Updated By:	04.02.25
Date Created:	04.02.25	Date Last Updated:	04.02.25

Actor:	System
Description:	The system automatically generates the most suitable time slot for the group based on members' submitted availability, food, and location preferences.
Preconditions:	All room members have submitted their availability, food, and location preferences.
Postconditions:	A time slot is successfully generated and displayed to all room members. Users have the option to either confirm or request a different time slot.
Priority:	
Frequency of Use:	
Flow of Events:	1. Once all members have submitted their availability, food, and location preferences, the system filters the submitted data to determine the most suitable time slot. 2. The system selects the earliest available time slot that matches the majority of users. 3. The system displays a pinned system message in the chat: “System Generated Time Slot: [DD/MM/YY HH:MM].” 4. Users review the proposed time slot. 5. If users accept the suggested time slot, they select the "Proceed" button to confirm it. 6. If users are not satisfied, they can request to regenerate a new time slot (triggers Use Case 5.2).
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	

Notes and Issues:	

Use Case ID:	5.2		
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:	System
Description:	User reschedules a previously generated schedule if the proposed date and time do not work for the room.
Preconditions:	A schedule has already been generated. All room members have submitted their time, food, and area preferences.
Postconditions:	A new schedule is successfully regenerated and displayed to all room 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 and finds the next most common available time and location based on room preferences. 5. The system displays a new pinned message: “Updated System Generated Message: Date and Time: [DD/MM/YY HH:MM].” 6. The room reviews the updated schedule. 7. If users are satisfied, they select the "Proceed" button to finalize the schedule else they can select “Regenerate Time/Date” till they are satisfied.
Alternative Flows:	AF.2: User selects "Regenerate Time/Date" multiple times. 1. The system filters and calculates the next best option based on available preferences. 2. If all possible time slots have been exhausted, the system will return to the first available time slot and continue cycling through the options. 3. Return to Step 4.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	6.1		
Use Case Name:	Generate Restaurant List		
Created By:	Boey, Tiffany	Last Updated By:	04.02.25
Date Created:	04.02.25	Date Last Updated:	04.02.25

Actor:	System
Description:	Users generate a recommended list of restaurants based on the group's availability, preferred dining areas, and cuisine preferences. The generated list is the same for all room members, ensuring a consistent voting process.
Preconditions:	A time slot has been selected and confirmed. Users have submitted their preferred dining areas and cuisine types.
Postconditions:	The system successfully generates and displays the same list of recommended restaurants to all room members. Users can proceed with voting on the restaurant options.
Priority:	
Frequency of Use:	
Flow of Events:	1. After confirming the time slot, the system provides a "Generate" button for users to create a list of recommended restaurants. 2. A user clicks the "Generate" button to trigger the system to compile a restaurant list. 3. The system filters and selects restaurants based on the group's most commonly available time slot, preferred dining areas and preferred cuisine types. 4. The system displays the generated restaurant list in the room chat. 5. Users review the restaurant list and proceed to voting (handled in Use Case 6.2).
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	1. At least one restaurant meets the group's criteria. 2. Users understand that voting will be based on the generated list.
Notes and Issues:	

Use Case ID:	6.2		
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:	Users vote on the recommended restaurant list by upvoting or downvoting options. Users can view restaurant details and reviews

	before voting. The restaurant with the highest votes will be selected for the group.
Preconditions:	A time slot has been confirmed for the event. The system has generated a list of recommended restaurants based on group preferences. At least one restaurant is available for voting.
Postconditions:	Users successfully vote on restaurant options. The system records all votes and updates the rankings in real time. The restaurant with the highest votes is selected for the group.
Priority:	
Frequency of Use:	
Flow of Events:	1. The user clicks the "Vote Food" button in the room chat. 2. The system redirects the user to a page displaying the recommended restaurant list. 3. Users can either “upvote” a restaurant by clicking the up arrow (+1 point), or “downvote” a restaurant by clicking the down arrow (-1 point). 4. Users can click on any restaurant to view details, including ratings, reviews, and location information. 5. After finalizing their votes, users click the "Submit" button". 6. The system records the user's votes in real time and updates the room submission count (e.g., "3/5 members have submitted"). 7. System then redirects the user back to the Room Chat Message page. 8. Once all room members complete their votes, the system calculates the highest-voted restaurant. 9. The system generates and pins a message: "System Generated Message: Date and Time: [DD/MM/YY HH:MM], Location: [Selected Restaurant]." 10. In the case, the votes are tied, the system will choose randomly (handled in Use Case 6.3).
Alternative Flows:	AF.5: 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:	6.2.EX-1: A user does not vote. - The system proceeds once time runs out (10 minutes).
Includes:	
Special Requirements:	
Assumptions:	All members will vote for an option
Notes and Issues:	

Use Case ID:	6.3		
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, Systen
Description:	The system resolves restaurant selection conflicts when two or more restaurants receive the highest number of points. Users can vote to keep the old restaurant, accept the newly selected restaurant, or request a re-spin. The final confirmed restaurant is displayed in the group chat.
Preconditions:	A time slot has been confirmed for the event. The system has generated a list of recommended restaurants and users have completed voting. Two or more restaurants have received the same highest number of votes.
Postconditions:	1. The system randomly selects one restaurant from the tied options. 2. Users vote to either “Accept New Restaurant”, or “Regenerate”. 3. If “Regenerate” receives more than 50% votes, the system spins again, updating the “Accept New Restaurant” option. 4. The process repeats until a final restaurant is selected and displayed in the group chat.
Priority:	
Frequency of Use:	
Flow of Events:	1. A system-generated pinned message displays the restaurant with the highest votes: “Selected Restaurant - [Restaurant Name]." 2. System will choose the next highest voted restaurant to be displayed “Selected Restaurant - [Restaurant Name]." 3. If there is a tie between two or more restaurants, the system selects a restaurant randomly by a spinning wheel. New restaurant selected randomly is displayed by the system: “Selected Restaurant - [Restaurant Name]." 4. Users vote on one of the following options: “Accept New Restaurant” or “Regenerate”. 5. If more than 50% vote to “Regenerate”, the system spins again, else the restaurant is accepted. 6. If “Regenerate” is chosen, return to step 2. 7. The process repeats until a restaurant is accepted. 8. The final restaurant selection is displayed in the group chat: 9. “System Generated Message: Final Restaurant - [Restaurant Name].”
Alternative Flows:	
Exceptions:	6.3.EX-1: A user does not vote during the tiebreaker. - The system proceeds once time runs out (10 minutes). 6.3.EX-6: All tiered options are exhausted after multiple re-spins.

	1. If more than 50% of users reject the final selection, the system allows manual selection from the tied options.
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 event to calendar
Preconditions:	The final restaurant and time slot must be confirmed. The user must be logged in.
Postconditions:	The system sends a confirmation email with event details. The user receives an .ics file attachment for importing the event into their calendar.
Priority:	
Frequency of Use:	
Flow of Events:	1. The system provides an "Add to Calendar" button. 2. The user clicks "Add to Calendar." 3. The system triggers the "Send Confirmation Email" use case.
Alternative Flows:	
Exceptions:	
Includes:	Use Case ID 7.2
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	7.2		
Use Case Name:	Send Confirmation Email with Attachment		
Created By:	Boey, Tiffany	Last Updated By:	04.02.25
Date Created:	04.02.25	Date Last Updated:	04.02.25

Actor:	System
Description:	When a user clicks "Add to Calendar," the system automatically sends a confirmation email containing the event details and a downloadable calendar attachment.
Preconditions:	The user must click "Add to Calendar."

	The user must have a registered Gmail address.
Postconditions:	The system successfully sends an email with event details and an .ics file attachment. The user receives the email and can add the event to their calendar.
Priority:	
Frequency of Use:	
Flow of Events:	1. The user clicks "Add to Calendar." 2. The system retrieves Event Date and Time, Restaurant name and location, Room PIN 3. The system generates an .ics file containing the event details. 4. The system sends the confirmation email with the event details and attachment to the user's registered Gmail address.
Alternative Flows:	AF.4: The user's email client does not support .ics files. - The user must manually add the event using the provided details.
Exceptions:	7.2.EX-1: Email delivery failure. The system retries sending the email. If it fails, an error notification is displayed: "Failed to send confirmation email. Please check your email settings."
Includes:	
Special Requirements:	
Assumptions:	The user will download the .ics file
Notes and Issues:	

Use Case ID:	8.1		
Use Case Name:	Leave a Review		
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 leaves a review for the restaurant after the outing. The review allows users to provide feedback based on their experience during the outing.
Preconditions:	The outing has been completed.
Postconditions:	The review is successfully submitted and recorded in the system.
Priority:	
Frequency of Use:	
Flow of Events:	1. After the outing, the system displays the "Leave a Review" button on the room chat interface. 2. A system-generated message is pinned in the room chat: "Hope everyone has enjoyed the outing! Please leave a review for the restaurant to help us improve user experience and food selection, thank you." 3. The user clicks the "Leave a Review" button again to proceed. 4. The system displays a review submission form with a text box for feedback and rating options. 5. The user provides a rating and writes their review. 6. The user submits the review. 7. The system saves the review and displays a confirmation message: "Thank you for your feedback!" 8. The review is made visible to all room members and can be viewed at any time.
Alternative Flows:	AF.3: User clicks "Leave a Review" but exits without submitting. 1. The system does not record any feedback and maintains the pinned review reminder. 2. Return to step 3
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 room
Preconditions:	The user is a member of the room. The room is still active.
Postconditions:	The user exits the room and is no longer listed as a participant. The room is removed from the user's "Joined Rooms" list. The user's data related to the room is deleted, except for any sent messages.
Priority:	
Frequency of Use:	
Flow of Events:	1. The user navigates to the "Joined Rooms" section. 2. The user clicks on the three-dot menu next to the room they want to leave. 3. The user selects the "Exit Room" option. 4. The system displays a confirmation prompt: "Are you sure you want to exit this room?" 5. The user selects "Exit" to confirm. 6. The system removes the user from the room and deletes their associated data in that room (excluding messages).
Alternative Flows:	AF.4: User Cancels Exit Request - If the user selects "Cancel" or closes the prompt, the system returns to the "Joined Rooms" section without making any changes.
Exceptions:	
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:	Manual Room Closure 1. After the outing, the system displays the "Close Room" button on the room 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, the system closes the room and displays a final message: "The room has been closed. Thank you for using OngLaiDecides!" 5. All interactions in the room chat, such as voting and reviews, are disabled. 6. If the user cancels, the system keeps the room open. Automatic Room Closure 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 OngLaiDecides!" 4. All interactions in the room chat are disabled.
Alternative Flows:	AF.2: User clicks "Close Room" but exits without confirming. 1. The system does not close the room and keeps the room chat open. AF.3: The user attempts to close the room before all members have submitted reviews. 1. System Response: "Not all members have submitted reviews. Are you sure you want to close the room?" 2. If the user confirms, proceed to step 4. 3. If the user cancels, return to step 1.
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:	Users add 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:	AF.3: The user does not input a Member ID and clicks "Enter." 1. The system displays a message: "This user does not exist." AF.4: 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:	10.1.EX.3: A system error occurs during the friend addition process (e.g., network issue). 1. The system displays an error message: "Unable to add friends 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 Friend's name, Friend's Member ID, Avatar/profile picture 3. The user can scroll through the list to view all their friends.
Alternative Flows:	AF.2: The user has no friends added to their list. 1. The system displays a message: "You have no friends in your list yet. Use 'Add Friend by Member ID' to get started!"
Exceptions:	10.2.EX.1: A system error occurs while retrieving the friends list (e.g., network issue). 1. The system displays an error message: "Unable to load friends list. Please try again later." 2. Return to the "Friends" page.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	11.1		
Use Case Name:	View All Rooms		
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 “View All Rooms” under “Home” Page

	2. The system shows all schedules generated from all rooms 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:	AF.3: 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." AF.4: The user does not upload a new profile picture. 1. The system retains the current profile picture.
Exceptions:	12.1.EX.5: 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:	User
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 and Date room was closed 4. The user can scroll through the list to view multiple past rooms.
Alternative Flows:	AF.1: The user has no past rooms. 1. The system displays a message: "You have no past rooms to display."
Exceptions:	12.2.EX.1: 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:	

Use Case ID:	13.1		
Use Case Name:	Send Reminder via Notification		
Created By:	Boey, Tiffany	Last Updated By:	04.02.25
Date Created:	04.02.25	Date Last Updated:	04.02.25

Actor:	System
Description:	System sends a push notification reminder to users one day before the scheduled event.
Preconditions:	The user must have push notifications enabled on their device.
Postconditions:	The user receives a push notification reminder one day before the event. The notification includes event details such as date, time, restaurant name, and location.
Priority:	

Frequency of Use:	
Flow of Events:	1. The system checks for confirmed events scheduled for the next day. 2. If an event exists, the system generates a push notification that mentions date, time, restaurant name, and location.. 3. The system sends the notification to all users who are part of the event.
Alternative Flows:	
Exceptions:	13.1.EX.1: User has disabled push notifications. 1. The system does not send a push notification to that user.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	14.1		
Use Case Name:	Send Reminder via Email		
Created By:	Boey, Tiffany	Last Updated By:	04.02.25
Date Created:	04.02.25	Date Last Updated:	04.02.25

Actor:	System
Description:	System sends an email reminder to all users one day before the scheduled event that includes event details.
Preconditions:	The user must have a registered email address in the system.
Postconditions:	The user receives an email reminder one day before the event. The email contains event details (date, time, restaurant, and location).
Priority:	
Frequency of Use:	
Flow of Events:	1. The system checks for confirmed events scheduled for the next day. 2. If an event exists, the system generates an email reminder. 3. The system sends the email to all users who are part of the event. 4. The email includes Event Date and Time, Restaurant Name and Location
Alternative Flows:	AF.1: User's email address is invalid or unavailable. 1. The system logs the failure and does not send the reminder to that user.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	