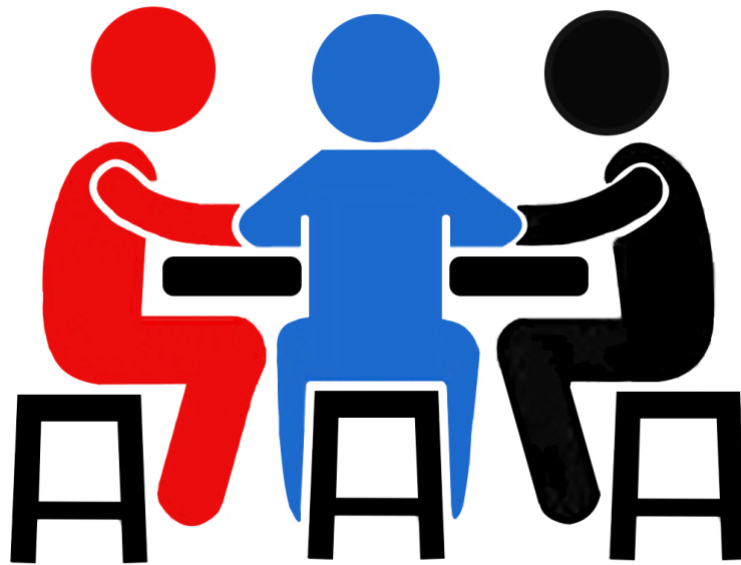


USE CASES

Making Friends Over Food App:



“Tables”

Team Never Eat Alone

Group Members:

Christopher Weaver, Sushanth Mukkamalla,
Vi Phung, Zihao Zhou, Khoa Bui, Xuan Zhang,
Zhening Huang, Yinlong Qian, Jinwei Ren,
Yuhong Sun, Sean Yeh

Table of Contents

Use Cases:	Title
UC 1.1	Sign Up
UC 1.2	Login
UC 1.3	Log Out
UC 1.4	Password Recovery
UC 2.1	Create Personal Profile
UC 2.2	Edit Personal Profile
UC 3.1	Search for Other Users
UC 3.2	Find Random Users
UC 4.1	Select Matched Users
UC 4.2	Send Invitation
UC 4.3	Receive Invitation
UC 5.1	Create Chat Room
UC 5.2	Send Messages in Chat Room
UC 5.6	Set When and Where to Meet
UC 5.7	Edit When and Where to Meet
UC 5.8	Leave Chat Room
UC 7.1	History of Everyone Met

Next iteration planned:

UC 2.3	Set Status
UC 5.3	Mute Notifications for Chat Room
UC 5.4	Create Poll in Chat Room
UC 5.5	Add Additional Users to Chat Room
UC 6.1	Meeting Reminder Notification

Priority	Description
1	Must Implement
2	Should Implement
3	Want to Implement
4	Won't Implement

Status
Planned
In-Progress
Completed
Next Iteration

Meeting new people, especially in unfamiliar and uncomfortable environments, can be difficult. Fortunately, our app, “Tables,” provides an elegant, yet pragmatic, way to do so. At its core, Tables is an app for UCSD students to find and grab a meal with others who they are likely to get along with, and thus, likely to form long and lasting friendships. What are the primary features a user of Tables can expect? Users of the application will create a profile that represents them, search for other users who they are likely to connect with, and, once they have matched with others, be able to communicate with each other to set a location and time to eat.

A user begins by creating their profile, which, by representing who they are to others and to the system, allows them to match with people who share similar personalities and hobbies. Within their profile they can cite interests and hobbies, describe what kind of food they like, and state the classes and subjects they are in/are interested in. With this information, other users can get a feel of the user and decide whether they are someone they would get along with, and vice versa; furthermore, this information is used by the system to better optimize the displaying of others, so that the user is not bogged down by having to sift through those whom they have little or nothing in common with. In short, the user profile is a tool used by both the user and the system to provide an intelligent matching system.

After the profile has been created, a user can create a search request to find other users. This request can be filtered based on an algorithm that takes into account, depending on what the user desires, similar interests, major, desired meetup time, classes. One of the hardest parts about meeting others and creating friendships is initially finding something to talk about; however, filtering the search results will give users a starting point of conversation, and from there, hopefully the conversation will be able to carry on to other topics. The search request improves and simplifies the process of finding others you are interested in meeting and creating friendship with.

After two users have confirmed that they would like to meet with each other, they will be able to talk in a private group chat to finalize a meeting time and location. The chat will have multiple features such as creating a poll, automatically picking a time when all users are free, and picking locations from a map of campus. Users will waste less time messaging each other back and forth about when and where they want to meet, so they can have more time to talk about their similar interests during a meeting. These features, which are not found in text messaging applications, are designed to make finalizing the location and time as simple as possible.

At the heart of the Tables experience is the ability to meet new people who you can relate to easily and efficiently. This is brought about by three key features: the creation of a user profile, the search for other users, and communication between users to set up a time and place to grab a bite. The profile allows each user to express who they are, which will be the main tool that both users and the system use to perform preliminary screening for potential new friends. The search system is the primary driver of the application. It takes advantage of the profiles to give users an ordered list of those that they are likely to get along with. Finally, the chat and location function allows users to easily choose and agree on location to grab a table. Tables solves the age old problem of meeting new people comfortably.

<u>Use Case #1.1</u>	Sign Up
Description:	The actor creates an account UCSD email in the application with an UCSD email and password.
Actor:	The user of the application.
User Goals:	Create an account to begin taking advantage of the applications functions.
Desired Outcome:	The user will successfully create an account that will be saved in the database and will be able to use for the duration of their time with the application.
Dependent Use Cases:	None.
Requirements:	SR 1.1
Pre-Conditions:	The user installs the application, opens it without an account and have a UCSD email.
Post-Conditions:	An account is registered in the online database and can be accessed so that the account can be logged into/have information stored with it.
Trigger:	The user would like to begin using the functionality of the application.
Workflow:	<ol style="list-style-type: none"> 1. The user shall press the "Create New Account" button on the login screen. 2. The user shall enter in their UCSD email, name, and password. 3. The system shall accept the above information and conditionally store it in the database. 4. The system shall tell the user in the application to verify their account. 5. The system shall send a verification email to the user. 6. The user shall verify their account via email link.
Alternate Workflow:	<ol style="list-style-type: none"> 1. The user shall enter email that has already been signed up. The system shall display "Registration Unsuccessful". 2. The user shall enter email that is not @ucsd.edu. The system shall toast a message says "Enter a valid UCSD email". 3. The user shall enter email that does not exist. The system shall send an email to the entered address, but the user will not be able to verify that email.
Status:	Completed
Priority:	1

<u>Use Case #1.2</u> Login	
Description:	The actor logs into the application using the registered account information through a login screen.
Actor:	The user of the application.
User Goals:	The user wants to be able to login and continue using the application.
Desired Outcome:	The user will successfully login and use the app.
Dependent Use Cases:	UC 1.1
Requirements:	SR 1.2
Pre-Conditions:	The user signs up for an account and has information ready to login.
Post-Conditions:	The application will direct the user to the main screen.
Trigger:	The user would like to login to the application.
Workflow:	<ol style="list-style-type: none"> 1. The user shall enter the username in the "Username" box. 2. The user shall enter in the password in the "Password" box. 3. The system shall accept the input data. 4. The system shall look up input in database. 5. The system shall verify if there is existing account. 6. The system shall go to the search tab if the user has already created a personal profile or to the create account page if the user has not.
Alternate Workflow:	<ol style="list-style-type: none"> 1. The user shall enter incorrect username. The system shall display "Sign In Failed". 2. The user shall enter incorrect password. The system shall display "Sign In Failed".
Status:	Completed
Priority:	1

Use Case #1.3 Log Out	
Description:	The actor logs out of the application using the “Log out” button.
Actor:	The user of the application.
User Goals:	The user wants to be able to log out and stop using the app.
Desired Outcome:	The user will successfully log out and be directed back to the login screen.
Dependent Use Cases:	UC 1.1, UC 1.2
Requirements:	SR 1.3
Pre-Conditions:	The user already has a registered account and logged onto the application.
Post-Conditions:	The user cannot see any information specific to the account that was previously logged on and gets directed to the login screen.
Trigger:	The user would like to log out of the application.
Workflow:	<ol style="list-style-type: none"> 1. The user shall open the navigation bar by clicking the three line icon on the top left corner. 2. The user shall press the “Log Out” option in the navigation bar. 3. The system shall recognize that “Log Out” has been clicked. 4. The system shall stop current function/features. 5. The system shall log out of the user’s account. 6. The system shall direct user to login screen.
Alternate Workflow:	None.
Status:	Completed
Priority:	1

Use Case #1.4 Password Recovery	
Description:	The actor resets their password with the “Forgot password” button on the login screen.
Actor:	The user of the application.
User Goals:	The user wants to be able to change their password.
Desired Outcome:	The user will successfully create a new password for their account.
Dependent Use Cases:	UC 1.1, 1.2
Requirements:	N/A.
Pre-Conditions:	The user already has a registered account.
Post-Conditions:	The user is directed to the login screen.
Trigger:	The user has forgotten their password and would like to be able to login the app.
Workflow:	<ol style="list-style-type: none"> 1. The user shall press the “Forgot Password?” button. 2. The system shall recognize that “Forgot Password?” has been clicked. 3. The system shall direct user to the password reset screen. 4. The system shall ask user to enter their email. 5. The user shall enter their UCSB email. 6. The user shall click “Reset Password”. 7. The system shall show a message saying that a recovery password link has been sent to user’s email. “Password reset email has been sent.”. 8. The user shall open their email. 9. The user shall click on the password reset link. 10. The user shall enter a new password. 11. The user shall click “SAVE”. 12. The system shall save the new password.
Alternate Workflow:	<ol style="list-style-type: none"> 1. The user shall enter an invalid email on Recovery Password screen. The system shall display “Email failed to send, please check the email address.”. 2. The user shall click the back arrow on the bottom navigation bar to cancel their action. The system shall re-direct user back to login screen.
Status:	Completed
Priority:	1

Use Case #2.1 Create Personal Profile	
Description:	The actor can create a personal profile by inputting data (i.e. hobbies, classes, college, year, etc)
Actor:	The user of the application.
User Goals:	The user wants to show what they are interested in and anything else that they would like to share publicly.
Desired Outcome:	The user will have a profile with all the newly input details available to be seen by the system, themselves (on the “My Profile” screen), and other users.
Dependent Use Cases:	UC 1.1, UC 1.2
Requirements:	SR 2.1
Pre-Conditions:	The user has a registered account and is logged in.
Post-Conditions:	The user’s new profile is stored in the database.
Trigger:	The user would like to begin using the application and share their interests publicly.
Workflow:	<ol style="list-style-type: none"> 1. The user shall log in for the first time. 2. The system shall direct the user to the create personal profile screen. 3. The user shall fill in all the inputs on the screen. 4. The user shall click “Continue” at the bottom of the screen. 5. The system shall save all the data that the user has inputted on the database. 6. The system shall direct the user to the search tab.
Alternate Workflow:	<ol style="list-style-type: none"> 1. The user shall click the back arrow on the bottom navigation bar to cancel the creation of their profile. The system shall close the application.
Status:	Completed
Priority:	1

Use Case #2.2 Edit Personal Profile	
Description:	The actor can edit their personal profile by inputting new data or editing existing profile.
Actor:	The user of the application.
User Goals:	The user wants to update interests, courses, etc. or delete any data that they would like to keep in private.
Desired Outcome:	The user will have a profile with all the newly input details available to be seen by the system, themselves (on the “My Profile” screen), and other users.
Dependent Use Cases:	UC 1.1, UC 1.2, UC 2.1.
Requirements:	SR 2.2.
Pre-Conditions:	The user already has a registered account, is logged on and has an existing profile.
Post-Conditions:	The database is updated to reflect the edited profile.
Trigger:	The user would like to change their profile information.
Workflow:	<ol style="list-style-type: none"> 1. The user shall open the navigation bar by clicking the three line icon on the top left corner. 2. The user shall click on the “Edit Profile” option. 3. The system shall direct to the edit profile screen. 4. The system shall fill in all the information the user has previously entered. 5. The user shall change the information as desired. 6. The user shall click “Continue” at the end of the screen after they finish. 7. The system shall save all the data that user has inputted to the database. 8. The system shall display “Profile has been edited.”. 9. The system shall direct user back to the navigation drawer.
Alternate Workflow:	<ol style="list-style-type: none"> 1. The user shall click the back arrow on the top left to cancel the creation of their profile. The system shall take the user back to the navigation drawer.
Status:	Completed
Priority:	1

<u>Use Case #3.1</u>	Search for Other Users
Description:	The user gets a list of other users. The search list should display each user's name and tags of interests. The users at the top of the search list should have similar interest levels as those of the user respectively.
Actor:	The user of the application.
User Goals:	The user will be able to find people who they may be interested to meet with.
Desired Outcome:	The user successfully find potential new friends that they can hang out with over a meal.
Dependent Use Cases:	UC 1.1, UC 1.2, UC 2.1.
Requirements:	SR 3.1.
Pre-Conditions:	The user has logged in and set up their profile.
Post-Conditions:	The app displays a list of users, those who are the best match to the user at the top.
Trigger:	The user wants to find people that they are interested to meet with.
Workflow:	<ol style="list-style-type: none"> 1. The user shall go to the "SEARCH" tab. 2. The user shall click on the "Search" button. 3. The system shall run an algorithm to rank other users based on interest similarities. 4. The system shall list users in order of best match to worst match, according to their interests.
Alternate Workflow:	None.
Status:	Completed
Priority:	1

Use Case #3.2 Find Random Users	
Description:	The user finds random available users to chat with.
Actor:	The user of the application.
User Goals:	The user will be able to find random people who they may be interested to meet with.
Desired Outcome:	The user successfully finds friends that they can hang out with over a meal.
Dependent Use Cases:	UC 1.1, UC 1.2, UC 2.1
Requirements:	SR 3.2.
Pre-Conditions:	The user has logged in and has set up their profile.
Post-Conditions:	The app displays random users, one at a time.
Trigger:	The user wants to find people they are interested to meet with.
Workflow:	<ol style="list-style-type: none"> 1. The user shall go to the “SEARCH” tab. 2. The user shall click on the “Random Search” button. 3. The system shall list users in a random order.
Alternate Workflow:	None.
Status:	Completed
Priority:	1

Use Case #4.1 Select Matched Users	
Description:	The user can select people who they may be interested to meet and eat with.
Actor:	The user of the application.
User Goals:	The user wants to choose who they may want to meet with to eat.
Desired Outcome:	The user successfully informs the application of the people they are considering meeting and eating with.
Dependent Use Cases:	UC 1.1, UC 1.2, UC 3.1
Requirements:	SR 4.1.
Pre-Conditions:	The user has started a search and received a list of users.
Post-Conditions:	The system receives the user's choices of their potential new friends and sends them an invite notification.
Trigger:	The user wants to tell the application people they are interested in meeting and eating with.
Workflow:	<ol style="list-style-type: none"> 1. The user shall look at the other users profiles names and tags on the list. 2. The user shall swipe up and down to scroll the list of users. 3. The user shall tap on a user's profile to view it in detail if they want to know more details. 4. The user shall press the arrow in upper left corner to return to the list. 5. The user shall press the toggle on the right side, next to other user's name to signify that they are interested in meeting with them. 6. The system shall record all users that have been selected.
Alternate Workflow:	<ol style="list-style-type: none"> 1. The user shall click the arrow on top left to stop searching for other users. The system shall hide the list and redirect the user back to the search tab.
Status:	Completed
Priority:	1

Use Case #4.2 Send Invitation	
Description:	The user can send invite notifications to all the people they have selected from the list.
Actor:	The user of the application.
User Goals:	The people selected on the list get to know they are invited by the user.
Desired Outcome:	The people selected by the user receive an invitation notification.
Dependent Use Cases:	UC 1.1, UC 1.2, UC 3.1, UC 3.2, UC 4.1
Requirements:	SR 4.2.
Pre-Conditions:	The user has run a search for other users. The user has selected at least one person on the list.
Post-Conditions:	The system sends invite notifications to the selected users.
Trigger:	The user wants people they have selected on the list to get the invitation notification.
Workflow:	<ol style="list-style-type: none"> 1. The user shall click “Send Invitation(s)”. 2. The system shall send invitation notifications to the other user who was selected. 3. The system shall toast a message to the user “Sending invite(s) to: *names*”.
Alternate Workflow:	<ol style="list-style-type: none"> 1. The user shall click on the back arrow on the top left corner. System shall hide the list redirect the user back to the search tab.
Status:	Implemented
Priority:	1

Use Case #4.3 Receive Invitation	
Description:	The user should see the notice that someone want to dine with him. The user can then choose to accept or reject such invitation.
Actor:	The user of the application.
User Goals:	The user will be able to accept/decline this invitation.
Desired Outcome:	If the user accepts the invitation, the inviter will be able to chat with them. If the user declines the invitation, the inviter will see a notification.
Dependent Use Cases:	UC 1.1, UC 1.2, UC 4.1, UC 4.2
Requirements:	SR 4.3.
Pre-Conditions:	The user has received an invite notification.
Post-Conditions:	If the invite is accepted, a chat room will be created between the two users.
Trigger:	The user received an invitation.
Workflow:	<ol style="list-style-type: none"> 1. The user shall click on the “View Profile” button to see the inviter’s basic information. 2. The user shall click on the “Accept” button if they want to grab a meal with the inviter. 3. The system shall record that the user accepted the invite and create a chat room between the two users. 4. The user shall be able to chat with the inviter.
Alternate Workflow:	<ol style="list-style-type: none"> 1. The user shall click on the “Decline” button if they do not want to grab a meal with the inviter. System shall record that the user has declined the invite.
Status:	Completed
Priority:	1

<u>Use Case #5.1</u>	Create Chat Room
Description:	The user should be able to create a new chat room with people they have already matched with.
Actor:	The user of the application.
User Goals:	The user wants to create a chat room to communicate with a certain person.
Desired Outcome:	The user has a new chat room which they can invite other users to.
Dependent Use Cases:	UC 1.1, UC 1.2, UC 4.1, UC 4.2, UC 4.3.
Requirements:	SR 5.1.
Pre-Conditions:	The user has accepted the invitation that has been sent to them.
Post-Conditions:	A chat room room with two users is created.
Trigger:	The user accepted the invitation.
Workflow:	<ol style="list-style-type: none"> 1. The system shall update the database saying that user has accepted the invitation. 2. The system shall create an empty chat room between the user and the inviter to store all their messages.
Alternate Workflow:	None.
Status:	Completed.
Priority:	1

Use Case #5.2 Send Messages in Chat Room	
Description:	The user should be able to chat with another user or multiple users.
Actor:	The user of the application.
User Goals:	User created chat room to communicate with another user, potentially to set up a meet or exchange information.
Desired Outcome:	The message is sent to the receiving user with minimal delay.
Dependent Use Cases:	UC 1.1, UC 1.2, UC 4.1, UC 4.2, UC 4.3, UC 5.1
Requirements:	SR 5.2.
Pre-Conditions:	A chat room has been created.
Post-Conditions:	A message is sent from one user to another.
Trigger:	A user wants to communicate with other user who they want to meet.
Workflow:	<ol style="list-style-type: none"> 1. The user shall click on the editable box to insert text. 2. The user shall enter in the message they wish to send. 3. The user shall click send by clicking the airplane send icon on the right side of the text box. 4. The system shall store the message in the database. 5. The system shall transfer this message to the other users account. 6. The users shall see the messages on their chat room.
Alternate Workflow:	<ol style="list-style-type: none"> 1. The user shall click on the editable box to insert text. 2. The user shall enter in the message they wish to send. 3. The user shall click on the back arrow on the top left corner. System shall clear the message the user wrote and redirect the user back to the chat tab.
Status:	Completed.
Priority:	1

Use Case #5.6 Set When and Where to Meet	
Description:	The users of the chat room will be able to set when and where to meet.
Actor:	The user of the application.
User Goals:	The users of the chat room will be able to set the meeting location and time so they can access to the information at all time without looking back at previous messages.
Desired Outcome:	The users will successfully be able to set when and where to meet in the chat room.
Dependent Use Cases:	UC 1.1, UC 1.2, UC 4.1, UC 4.2, UC 4.3, UC 5.1.
Requirements:	SR 5.6.
Pre-Conditions:	A chat room has been created. The user is in the chat room.
Post-Conditions:	The meeting time and location will be set, saved, and displayed to the user.
Trigger:	The users of a chat room want to initialize the location and time of the meeting.
Workflow:	<ol style="list-style-type: none"> 1. The user shall click on the toolbar icon on the top right corner. 2. The user shall click on the “Set Time/Location” option. 3. The system shall open the “Enter Time and Location” screen. 4. The user shall select a time to meet by clicking on the clock. 5. The user shall select a date to meet by clicking on the calendar. 6. The user shall input a location they want to meet the other user at. 7. The user shall click on the “Submit” button. 8. The system shall save the data to the database. 9. The system shall redirect the user back to the chat room. 10. The system shall display the location, date, and time at the top of the chat room.
Alternate Workflow:	<ol style="list-style-type: none"> 1. The user shall click the back arrow on the top left corner. The system shall redirect user back to the chat room and disregard all the information that has been entered.
Status:	Completed.

Priority:	1
-----------	---

Use Case #5.7 Edit When and Where to Meet	
Description:	The users of the chat room will be able to edit when and where to meet.
Actor:	The user of the application.
User Goals:	The users of the chat room will be able to update the meeting location and time.
Desired Outcome:	The users will successfully be able to edit when and where to meet in the chat room.
Dependent Use Cases:	UC 1.1, UC 1.2, UC 4.1, UC 4.2, UC 4.3, UC 5.1, UC 5.6
Requirements:	SR 5.7.
Pre-Conditions:	A chat room has been created, meeting time and location already set. The user is in the chat room.
Post-Conditions:	The meeting time and location will be updated and saved.
Trigger:	The users of a chat room want to update the location and time of the meeting.
Workflow:	<ol style="list-style-type: none"> 1. The user shall click on the toolbar icon on the top right corner. 2. The user shall click on the “Set Time/Location” option. 3. The system shall open the “Enter Time and Location” screen. 4. The user shall select a time to meet by clicking on the clock. 5. The user shall select a date to meet by clicking on the calendar. 6. The user shall input a location they want to meet the other user at. 7. The user shall click on the “Submit” button. 8. The system shall save the data to the database. 9. The system shall redirect the user back to the chat room. 10. The system shall display the location, date, and time at the top of the chat room.
Alternate Workflow:	<ol style="list-style-type: none"> 1. The user shall click the back arrow on the top left corner. The system shall redirect user back to the chat room and disregard all the information that has been entered.
Status:	Completed.

Priority:	1
-----------	---

<u>Use Case #5.8</u> Leave Chat Room	
Description:	The user will be able to leave a chat room and remove it from the list of active chat rooms.
Actor:	The user of the application.
User Goals:	The user will be able to leave a chat room.
Desired Outcome:	The user will be able to leave a chat room and stop receiving notifications from it.
Dependent Use Cases:	UC 1.1, UC 1.2, UC 4.1, UC 4.2, UC 4.3, UC 5.1.
Requirements:	SR 5.8.
Pre-Conditions:	A chat room has been created. The user is in a chat room.
Post-Conditions:	The chat room will be removed from the list of active chat rooms.
Trigger:	The user wants to leave a chat room.
Workflow:	<ol style="list-style-type: none"> 1. The user shall click on the toolbar icon on the top right corner. 2. The user shall click on the “Leave Chat” option. 3. The system shall save to the database that this user has left the chat room. 4. The system shall redirect user back to the chat tab. 5. The system shall display “You have left this chat.” under the name of the user you have left the chat room from.
Alternate Workflow:	None.
Status:	Completed.
Priority:	1

<u>Use Case #7.1</u>	Get History of Everyone Met
Description:	The user will have the ability to see other user's names they have interacted with in this app. This includes invites they have declined, current chat rooms, left chat rooms, and pending invitations.
Actor:	The user of the application.
User Goals:	The User can look through information of everyone they have met with so they will have the ability to contact them.
Desired Outcome:	The user will have the ability to see other user's names they have interacted with in this app.
Dependent Use Cases:	UC 1.1, UC 1.2, UC 2.1, UC 4.2, UC 4.3, UC 5.7, UC 5.8.
Requirements:	SR 7.1.
Pre-Conditions:	The user has interacted with other users on the app.
Post-Conditions:	The users will be able to view interaction history.
Trigger:	The user wants to see all the users they have interacted with in the past, or currently interacting with.
Workflow:	<ol style="list-style-type: none"> 1. The user shall go to the chat tab. 2. The system shall display an invitation message if another user has sent them an invite. 3. The system shall display "*name* has not responded to your invitation" under the other users name if they have not responded to the users invitation. 4. The system shall display "You have left this chat" if the current user is the one that has left the chat room. 5. The system shall display "*name* has left the chat room" under the other users name if they are the one that has left the chat room. 6. The system shall display the location and time for currently open chat rooms.
Alternate Workflow:	None.
Status:	Completed
Priority:	1

Next iteration:

Use Case #2.3 Set Status	
Description:	The actor will be able to set their status (i.e. setting it to available, hidden, away, or busy).
Actor:	The user of the application.
User Goals:	The user wishes to express their current status.
Desired Outcome:	The user will successfully update their status.
Dependent Use Cases:	UC 1.1, UC 1.2.
Requirements:	SR 2.3.
Pre-Conditions:	The user is logged on and has navigated to the status box.
Post-Conditions:	The system saves the user's new status in database and displays it to the appropriate users.
Trigger:	The user would like to express their status.
Workflow:	<ol style="list-style-type: none">1. The user shall open the navigation bar by clicking the three line icon on the top left corner.2. The user shall click on the “Set Status” option.3. The system shall direct the user to the set status screen.4. The user shall select one of the options: “available”, “busy”, “hidden”, “away”, or “busy”.5. The user shall click the “Set” button.6. The system shall record the user’s status in the database.7. The system shall redirect user back to the navigation bar.
Alternate Workflow:	<ol style="list-style-type: none">1. The user shall click the back arrow on the top left corner. The system shall redirect user back to the chat room and disregard all the information that has been entered.
Status:	Planned for Next Iteration
Priority:	4

Use Case #5.3 Mute Notification in Chat Room	
Description:	The user will be able to toggle receiving notification messages for chat rooms.
Actor:	The user of the application.
User Goals:	The user wants to create a chat room to communicate with a certain person/group of people.
Desired Outcome:	The user can mute a chat room if they are getting annoyed with all the notifications they are receiving.
Dependent Use Cases:	UC 1.1, UC 1.2, UC 4.1, UC 4.2, UC 4.3, UC 5.1, UC 5.2.
Requirements:	SR 5.3.
Pre-Conditions:	A chat room has been created and messages are sent.
Post-Conditions:	The user has muted the chat room to stop receiving notifications.
Trigger:	The user may not want to be distracted at the moment and would like to turn off their notifications.
Workflow:	<ol style="list-style-type: none"> 1. The user shall click on the chat room that they want to mute. 2. The user shall select the mute notification icon at the top of the chat. 3. The system shall record that this chat room is muted. 4. The user shall no longer receive notifications regarding messages from chat room.
Alternate Workflow:	None.
Status:	Planned for Next Iteration
Priority:	4

Use Case #5.4 Create Poll in Chat Room	
Description:	The user will be able to vote along with others in the chat if they like the suggested time and location.
Actor:	The user of the application.
User Goals:	The participants in the chat room will be able to see if a majority of the participants approve of the suggested time and location.
Desired Outcome:	A graph detailing how many users like the suggested location or time will be displayed.
Dependent Use Cases:	UC 1.1, UC 1.2, UC 4.1, UC 4.2, UC 4.3, UC 5.1.
Requirements:	SR 5.4.
Pre-Conditions:	A chat room has been created.
Post-Conditions:	The results of the poll will be shown in the chat.
Trigger:	A user wants to suggest time or location options to other members of the chat and figure out the best option.
Workflow:	<ol style="list-style-type: none"> 1. The user shall click on the poll button in the chat. 2. The user shall enter multiple times and locations. 3. The user shall select "Create Poll" when both inputs are filled. 4. The system shall receive the time and location data and display the poll in the chat room. 5. Other users shall vote from the poll. 6. The system shall take in inputs and create a graph to show how many people have vote for each option. 7. The system shall save the best time and location options.
Alternate Workflow:	<ol style="list-style-type: none"> 1. The user shall click the back arrow on the top left corner. The system shall redirect user back to the chat room and disregard all the information that has been entered.
Status:	Planned for Next iteration
Priority:	4

Use Case #5.5 Add Additional Users to Chat Room	
Description:	The users in a group chat can add more users to it by sending invitations.
Actor:	The user of the application.
User Goals:	The participants of a group chat will be able to add more users to it.
Desired Outcome:	The users will successfully add an additional user to the chat room.
Dependent Use Cases:	UC 1.1, UC 1.2, UC 4.1, UC 4.2, UC 4.3, UC 5.1.
Requirements:	SR 5.5.
Pre-Conditions:	A chat room has been created.
Post-Conditions:	A new user will be successfully added to the existing group chat.
Trigger:	The user wants to add an additional people to the chat room.
Workflow:	<ol style="list-style-type: none"> 1. The user shall click the add more member button on the group chat room interface. 2. The system shall display a list of users. 3. The user shall select the additional user(s) name(s). 4. The system shall send the invitation to the additional user(s). 5. The additional user shall accept the invitation. 6. The additional user shall successfully join the group chat. 7. The system shall add the new users into the existing chat room.
Alternate Workflow:	<ol style="list-style-type: none"> 1. The user shall decline the invitation and not be added to the group chat. A message is shown to the inviter that the other user has declined the invitation.
Status:	Planned for Next iteration
Priority:	4

<u>Use Case #6.1</u>	Meeting Reminder Notification
Description:	The user will get a notification when it is almost time to meet the other user.
Actor:	The user of the application.
User Goals:	The user will get a reminder notification.
Desired Outcome:	The user will get notified from the app about the upcoming meeting and its time and location.
Dependent Use Cases:	UC 1.1, UC 1.2, UC 4.1, UC 4.2, UC 4.3, UC 5.1, UC 5.6, UC 5.7
Requirements:	SR 6.1.
Pre-Conditions:	The meeting time and location has been set in the chat room.
Post-Conditions:	The user gets notified about the time and location of the meeting.
Trigger:	The user does not want to forget about the scheduled meetings.
Workflow:	<ol style="list-style-type: none"> 1. The system shall remind the users of the chat room an hour before the time they set to meet. 2. The user shall see the reminder notification with time and location of the meeting.
Alternate Workflow:	None.
Status:	Planned for Next iteration
Priority:	4