

ChatMaps Deliverable 5

Use Case Models

Stephen Goodridge, Clark LaChance, Nicholas Pease, Joseph Gallant, Aidan Bradley

COS420

21 April 2024

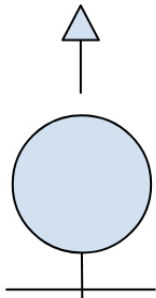
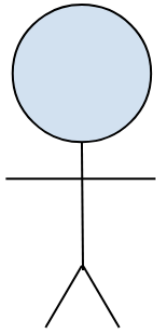
Github: <https://github.com/ChatMaps/ChatMaps>

Kanban: <https://trello.com/b/TaygvBv7/chatmaps>

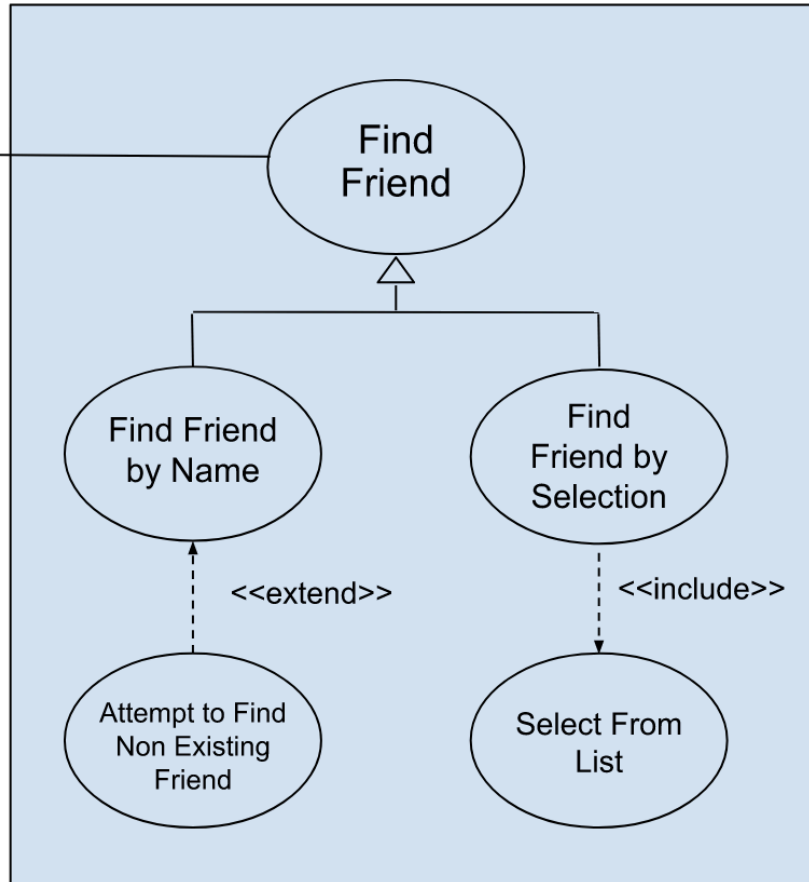
Use Case Diagram #1

Friend System

End User



Admin



Use Case Descriptions

Use Case 1: Find Friend

| | |
|-------------------------|---|
| Title: | Find Friend |
| Description: | The Find Friend use case is intended for finding a user you're already friends with, so that you can connect with them in some way. |
| System Under Design: | Friend System |
| Primary Actor: | End User |
| Participants: | End User, Admin |
| Goals: | Successfully find a desired friend and take some action. |
| Following Use Cases: | |
| Invariant: | |
| Precondition: | Expects the user to have the other user as an added friend. |
| Success Post Condition: | The user can perform certain actions with a desired friend like direct message them or invite them to a chat room. |

| Actor Actions | System Responses |
|--------------------------------|-----------------------------|
| 1. Find friend in friends list | 2. Confirm dialogue appears |
| 3. Confirm selection | 4. Dialogue disappears |

Use Case 2: Find Friend by Name

| | |
|-------------------------|---|
| Title: | Find Friend by Name |
| Description: | The Find Friend by Name use case is intended for finding other users of the application who have been made friends, by typing their name in a search bar. |
| System Under Design: | Friend System |
| Primary Actor: | End User |
| Participants: | End User, Admin |
| Goals: | Successfully find the desired friend and take some action. |
| Following Use Cases: | |
| Invariant: | |
| Precondition: | Expects the user to have added the other user as a friend. |
| Success Post Condition: | The user can perform certain actions with the desired friend. |

| Actor Actions | System Responses |
|-------------------------------------|---------------------------------|
| 1. Click search bar in friends list | 2. Cursor appears in text box |
| 3. Type the name of friend | 4. Friend appears in result bar |
| 5. Click friend name | |
| 6. Click "Invite" button | |

Use Case 3: Attempt to Find Non Existing Friend

| | |
|-------------------------|---|
| Title: | Attempt to Find Non Existing Friend |
| Description: | The Attempt to Find a Non Existing Friend use case is used when you can't find the desired friend by scrolling through the friends list and results in the End User having to search their friends list by typing their name. |
| System Under Design: | Friend System |
| Primary Actor: | End User |
| Participants: | End User, Admin |
| Goals: | Successfully find the desired friend and take some action. |
| Following Use Cases: | Find Friend by Name |
| Invariant: | |
| Precondition: | The End User must have not added the searched user as a friend in the past or the End User missed the desired friend's profile name while scrolling through the friends list. |
| Success Post Condition: | |

| Actor Actions | System Responses |
|-------------------------------------|---|
| 1. Click search bar in friends list | 2. Cursor appears in text box |
| 3. Type the name of friend | 4. System indicates friend does not exist |
| 5. Change friend name | 6. Correct friend appears |
| 7. Click "Invite" button | |

Use case 4: Find Friend by Selection

| | |
|-------------------------|---|
| Title: | Find Friend by Selection |
| Description: | The Find Friends by Selection use case, an extension of the Select From List use case, is used by End Users to find a desired friend by scrolling through their friends list. Allowing the End User to select the desired friend and perform some action. |
| System Under Design: | Friend System |
| Primary Actor: | End User |
| Participants: | End User, Admin |
| Goals: | Successfully find a desired friend and take some action. |
| Following Use Cases: | |
| Invariant: | |
| Precondition: | Expects the user to have added another user as a friend. |
| Success Post Condition: | The user can perform certain actions with a desired friend. |

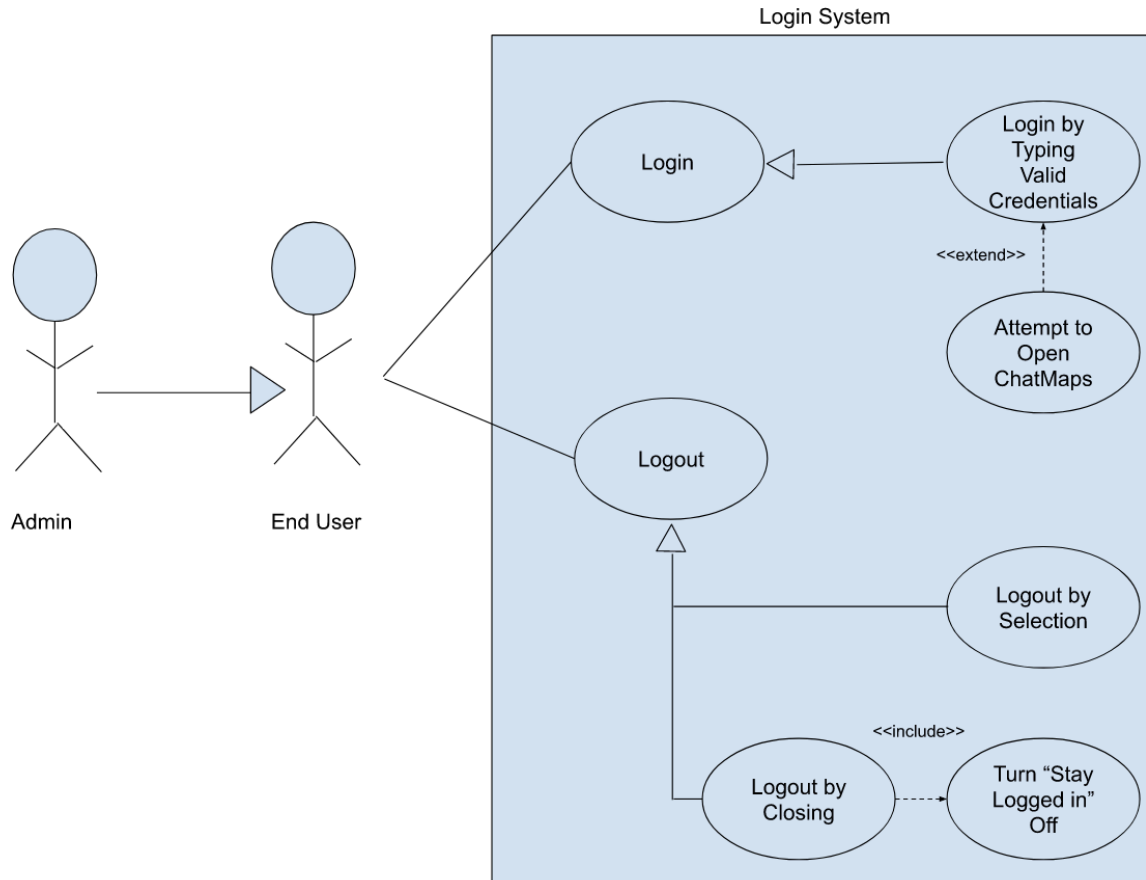
| Actor Actions | System Responses |
|--|-----------------------------|
| 1. Find friend by selecting name in friends list | 2. Confirm dialogue appears |
| 3. Confirm selection | 4. Dialogue disappears |

Use case 5: Select From List

| | |
|-------------------------|--|
| Title: | Select From List |
| Description: | The Select From List use case extends the Find Friend by Selection use case and is used for selecting on friends in a friends list so that some action can be performed. |
| System Under Design: | Friend System |
| Primary Actor: | End User |
| Participants: | User, Admin |
| Goals: | Successfully select a friend from the list. |
| Following Use Cases: | |
| Invariant: | |
| Precondition: | Expects the user to have added another user as a friend. |
| Success Post Condition: | The user can select a desired friend. |

| Actor Actions | System Responses |
|---|--|
| 1. The End User clicks on their friends list. | 2. The friends list expands, displaying all of the End User's current friends. |
| 3. The End User clicks on the desired friend's name in their list of friends. | 4. The system displays actions the End User can take with that friend. |

Use Case Diagram #2



Use case 6: Login

| | |
|-------------------------|--|
| Title: | Login |
| Description: | The Login use case is used to enter the application in order to use its main features. |
| System Under Design: | Login System |
| Primary Actor: | End User |
| Participants: | End User, Admin |
| Goals: | Login successfully |
| Following Use Cases: | Logout |
| Invariant: | |
| Precondition: | This use case assumes that the End User or Admin is not logged in. |
| Success Post Condition: | |

| | |
|--|--|
| <p>ACTOR:</p> <p>1) The End User or Admin enters the website url in their browser.</p> <p>3) The End User or Admin types in their email and password in their respective boxes.</p> <p>4) The End User or Admin clicks on the login button.</p> | <p>SYSTEM RESPONSES:</p> <p>2) The system displays the login page.</p> <p>5) The system redirects the user to the app page.</p> |
|--|--|

Use case 7: Logout

| | |
|-------------------------|--|
| Title: | Logout |
| Description: | The Logout use case is used for safely exiting the application or session. |
| System Under Design: | Login System |
| Primary Actor: | End User |
| Participants: | Admin |
| Goals: | Exit the current session |
| Following Use Cases: | |
| Invariant: | |
| Precondition: | This use case assumes the End User or Admin is already logged in. |
| Success Post Condition: | |

| | |
|--|---|
| ACTOR: 1) The End User or Admin either clicks the logout button or closes the browser. | SYSTEM RESPONSES: 2) The system redirects the user to the login page if the logout button was clicked. 3) The system displays dialogue saying the user has been signed out if the logout button was clicked. 4) Otherwise, the system does not respond. |
|--|---|

Use case 8: Login by typing valid credentials

| | |
|-------------------------|--|
| Title: | Login by Typing Valid Credentials |
| Description: | This use case is a generalization for the Login use case with the added notion that the End user or Admin is entering information associated with an active account. |
| System Under Design: | Login System |
| Primary Actor: | End User |
| Participants: | Admin |
| Goals: | Successfully login without having to make an account |
| Following Use Cases: | Login |
| Invariant: | |
| Precondition: | This use case assumes that the End User or Admin already has an existing account. |
| Success Post Condition: | |

| | |
|---|--|
| <p>ACTOR:</p> <ol style="list-style-type: none">1) The End User or Admin enters the website url in their browser.3) The End User or Admin types in their email and password from a pre-existing account in the respective boxes.4) The End User or Admin clicks on the login button. | <p>SYSTEM RESPONSES:</p> <ol style="list-style-type: none">2) The system displays the login page.5) The system redirects the user to the app page. |
|---|--|

Use case 9: Logout by selection

| | |
|-------------------------|---|
| Title: | Logout by Selection |
| Description: | The Logout by Selection use case is a generalization for the Logout use case, and it's used to exit the session by clicking or "selecting" the logout button. |
| System Under Design: | Login System |
| Primary Actor: | End User |
| Participants: | Admin |
| Goals: | Successfully hit the logout button before closing the browser. |
| Following Use Cases: | Logout |
| Invariant: | |
| Precondition: | The user must be logged in. |
| Success Post Condition: | The user is logged out. |

| | |
|--|--|
| <p>ACTOR:</p> <ol style="list-style-type: none">1) The End User or Admin is already logged in.2) The End User or Admin clicks on the logout button.5) The End User or Admin closes their current browser. | <p>SYSTEM RESPONSES:</p> <ol style="list-style-type: none">3) The system redirects the user to the login page.4) The system displays dialogue informing the user that they have been signed out. |
|--|--|

Use case 10: Logout by closing (UPDATED)

| | |
|-------------------------|--|
| Title: | Logout by Closing |
| Description: | The Logout by Closing use case is a generalization for the Logout use case where the End User or Admin exits their session by closing their browser with the “stay logging in” feature turned off. |
| System Under Design: | Login System |
| Primary Actor: | End User |
| Participants: | Admin |
| Goals: | Successfully logout on browser close |
| Following Use Cases: | Logout |
| Invariant: | |
| Precondition: | The user must be logged in. |
| Success Post Condition: | The user is logged out. |

| | |
|--|---|
| ACTOR: 1) The End User or Admin has the “stay logged in feature” turned off. 2) The End User or Admin closes their current browser/process. | SYSTEM RESPONSES: 3) The system automatically logs the user out of their current session. |
|--|---|