## 

## 

## 

## 

## **ChatMaps Deliverable 2**

Use Case Models

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

COS420

3 March 2024

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

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

**Use Case Diagram #1**

<<include>>

<<extend>>

User

Admin

Find friend

Find friend by typing name

Find friend by selecting from list

Attempt to find friend that doesn’t exist

Select from list

Finding Friends

**Related User Stories**

1. The system shall allow the user to add friends via their username.
2. The system shall let users invite friends to chat rooms that they’re currently connected to.
3. The system shall have a friends list to show who is currently online.

**Use Case Descriptions**

*Use Case 1: Find Friends*

|  |  |
| --- | --- |
| Description: | The Finding Friends use case is intended for finding other users of the application who have been made friends. |
| System Under Design: | Friend System |
| Primary Actor: | End User |
| Participants: | User and Admin |
| Goals: | Successfully find desired friend |
| 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 in friends list | 2. Confirm dialogue appears |
| 3. Confirm selection | 4. Dialogue disappears |

*Use Case 2: Find friends by typing name*

|  |  |
| --- | --- |
| Description: | The Finding Friends use case is intended for finding other users of the application who have been made friends by typing their name. |
| System Under Design: | Friend System |
| Primary Actor: | End User |
| Participants: | User and Admin |
| Goals: | Successfully find desired friend |
| 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. 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: Find friend that doesn’t exist*

|  |  |
| --- | --- |
| Description: | The Finding Friend that doesn’t exist use case is intended for dealing with when a name is searched that doesn’t match a friend. |
| System Under Design: | Searching System |
| Primary Actor: | End User |
| Participants: | User and Admin |
| Goals: | Successfully find desired friend |
| Following Use Cases: |  |
| Invariant: |  |
| Precondition: | The user must have not added the searched user as a friend. |
| Success Post Condition: | The system indicates the desired friend does not exist. |

|  |  |
| --- | --- |
| 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 selecting name*

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

|  |  |
| --- | --- |
| Description: | The Finding Friends by selecting name use case is intended for finding other users of the application who have been made friends by selecting their name in a list. |
| System Under Design: | Friend System |
| Primary Actor: | End User |
| Participants: | User and Admin |
| Goals: | Successfully find desired friend |
| 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. |

*Use case 5: Select from list*

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

|  |  |
| --- | --- |
| Description: | The Select from list use case is for selecting users in a friends list. |
| System Under Design: | Friend System |
| Primary Actor: | End User |
| Participants: | User and 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. |

**Use Case Diagram #2**

Login System

Admin

End User

<<include>>

<<extend>>

<<include>>

*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: | 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: |  |

|  |  |
| --- | --- |
| ACTOR:  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 in their respective boxes.  4) The End User or Admin clicks on the login button. | SYSTEM RESPONSES:  2) The system displays the login page.  5) The system redirects the user to the app page. |

*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: |  |

|  |  |
| --- | --- |
| ACTOR:  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. | SYSTEM RESPONSES:  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: |  |
| Success Post Condition: |  |

|  |  |
| --- | --- |
| ACTOR:  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. | SYSTEM RESPONSES:  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*

|  |  |
| --- | --- |
| 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 terminating their current process. |
| System Under Design: | Login System |
| Primary Actor: | End User |
| Participants: | Admin |
| Goals: | Successfully terminate the process |
| Following Use Cases: | Logout |
| Invariant: |  |
| Precondition: |  |
| Success Post Condition: |  |

|  |  |
| --- | --- |
| 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. |