

# ChatMaps Deliverable 5

Test Plan

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 Testing

---

Title: Login by Typing Valid Credentials

Actors: End User, Admin

Requirements: End User or Admin is on the homepage.

Main Scenario:

1. The End User or Admin types in their username and password.
2. The End User or Admin then clicks on the login button.
3. The system validates their credentials.
4. The system redirects them to the landing page after successful login.

Alternatives:

- 3a. The system is unable to validate the users credentials with the database.
- 4a. The system fails to redirect the End User or Admin to the landing page.

Test Situations:

1. The End User or Admin appears online to the system or other users.
2. Enable visual strings to double check usernames and passwords.
3. Click on reset password to confirm an account already exists.

Test Coverage:

Base: main scenarios (4) / alternative scenarios (2) covered.

Test: All cases covered.

Coverage: 100%

---

Title: Add Friend by Selection

Actors: End User, Admin

Requirements: an internet connection and web browser.

Main Scenario:

1. The End User or Admin selects the username or name of a user to make their friend.
2. The End User or Admin finds the friend's profile.
3. The End User or Admin clicks the request button.
4. The system sends the request to the desired user.

Alternatives:

- 1a. The End User or Admin selects an incorrect name.
- 4a. The system fails to send the request to the desired user.

Test Situations:

1. A user attempts to add an already added friend.
2. A user attempts to add a friend using a blank username.
3. A user attempts to add a friend that has deleted their account.

Test Coverage:

Base: main scenarios (4) / alternative scenarios (2) covered.

Test: All test cases covered.

Coverage: 100%

---

Title: Create a Room

Actors: End User, Admin

Requirements: an internet connection and web browser.

Main Scenario:

1. The End User or Admin selects the 'Create' tab.
2. The End User or Admin enters the name of the room.
3. The End User or Admin enters the description of the room.
4. The End User or Admin clicks the create button.
5. The system creates a room at the location of the user.

Alternatives:

- 1a. The system fails to create a room.
- 2a. The room name is either too long or contains invalid characters.
- 3a. The End User or Admin leaves the description blank.
- 5a. The system fails to pinpoint the location of the user.

Test Situations:

1. An End User or Admin attempts to create a room.
2. An End User or Admin denies location permissions while attempting to create a room.
3. An End User or Admin attempts to create a room using a different language.

Test Coverage:

Base: main scenarios (5) / alternative scenarios (4) covered.

Test: All test cases covered.

Coverage: 100%

---

Title: Accept a Friend Request

Actors: End User, Admin

Requirements: an internet connection and web browser.

Main Scenario:

1. The End User or Admin navigates to the friend request tab.
2. The user finds the request.
3. The user clicks on the accept button.
4. The system adds the friend to the list of friends.

Alternatives:

- 2a. The End User or Admin cannot find the request.
- 4a. The system fails to append a new friend to their friends list.

Test Situations:

1. User has the username of the added friend in the friends tab.
2. Both users accept each other's friend requests at the same time.

Test Coverage:

Base: main scenarios (4) / alternative scenarios (2) covered.

Test: All test cases covered.

Coverage: 100%

---

Title: Join a Room

Actors: End User, Admin

Requirements: an internet connection and web browser.

Main Scenario:

1. The End User or Admin navigates to the rooms tab.
2. The user clicks on a room they wish to join..
3. The user clicks on the addition symbol button to add the room.
4. The system redirects the user to the chat room.

Alternatives:

- 3a. The End User or Admin is already a member of the room they're trying to join.
- 1a. No nearby rooms exist.

Test Situations:

1. An End User or Admin attempts to join a room.
2. An End User or Admin attempts to join a room without location permissions enabled.

Test Coverage:

Base: main scenarios (4) / alternative scenarios (2) covered.

Test: All test cases covered.

Coverage: 100%

---

## Unit Testing

### Test Case 1 (Code)

**Object: Chat**

**Input:**

exampleChatObj {JSON} - contains mock chat message [body, issystem, timestamp, user]

exampleUser {JSON} - contains mock user object [uid, username, lastOnline]

**Output:**

React Component <Chat>

---

### Test Case 2 (Code)

**Object: ChatRoomSidebar**

**Input:**

exampleRoom {JSON} - contains mock chatroom [name, description]

**Output:**

React Component <ChatRoomSidebar>

---

### Test Case 3 (Code)

**Object: DM**

**Input:**

exampleUser {JSON} - contains mock user object [uid, username, lastOnline, firstName, lastName, pfp]

exampleFriend {JSON} - contains mock user object [uid, username, lastOnline, firstName, lastName, pfp]

**Output:**

React Component <DM>

---

### Test Case 4 (Code)

**Object: Member**

**Input:**

exampleUser {JSON} - contains mock user object [uid, username, lastOnline, firstName, lastName, pfp]

**Output:**

React Component <Member>

---

### Test Case 5 (Code)

**Object: RMF (Rich Message Formatting)**

**Input:**

Message {String} - String containing text, urls, and images

**Output:**

\*[] React Components and Strings

---

### **Test Case 6**

#### **Object: Friend**

#### **Input:**

exampleUser {JSON} - contains mock user object [uid, username, lastOnline, firstName, lastName, pfp]

exampleFriend {JSON} - contains mock user object [uid, username, lastOnline, firstName, lastName, pfp]

#### **Output:**

React Component <Friend>

---

### **Test Case 7**

#### **Object: FriendRequest**

#### **Input:**

exampleReceivingUser {JSON} - contains mock user object [uid, username, lastOnline, firstName, lastName, pfp]

exampleSendingUser {JSON} - contains mock user object [uid, username, lastOnline, firstName, lastName, pfp]

#### **Output:**

React Component <FriendRequest>

---

### **Test Case 8**

#### **Object: Geo**

#### **Input:**

loc {JSON} - contains moc location data [latitude and longitude]

zoom {Boolean} - contains logic to enable or disable zoom

moveable {Boolean} - contains logic to enable or disable moving the map

exampleUser {JSON} - contains mock user object [uid, username, lastOnline, firstName, lastName, pfp]

#### **Output:**

React Component <Geo>

---

### **Test Case 9**

#### **Object: Notification**

#### **Input:**

data {JSON} - mock notification data object [suser, ruser, title, byline, action]

#### **Output:**

React Component <Notification>

---

### **Test Case 10**

#### **Object: HomePage**

#### **Input:**

loc {JSON} - contains moc location data [latitude and longitude]

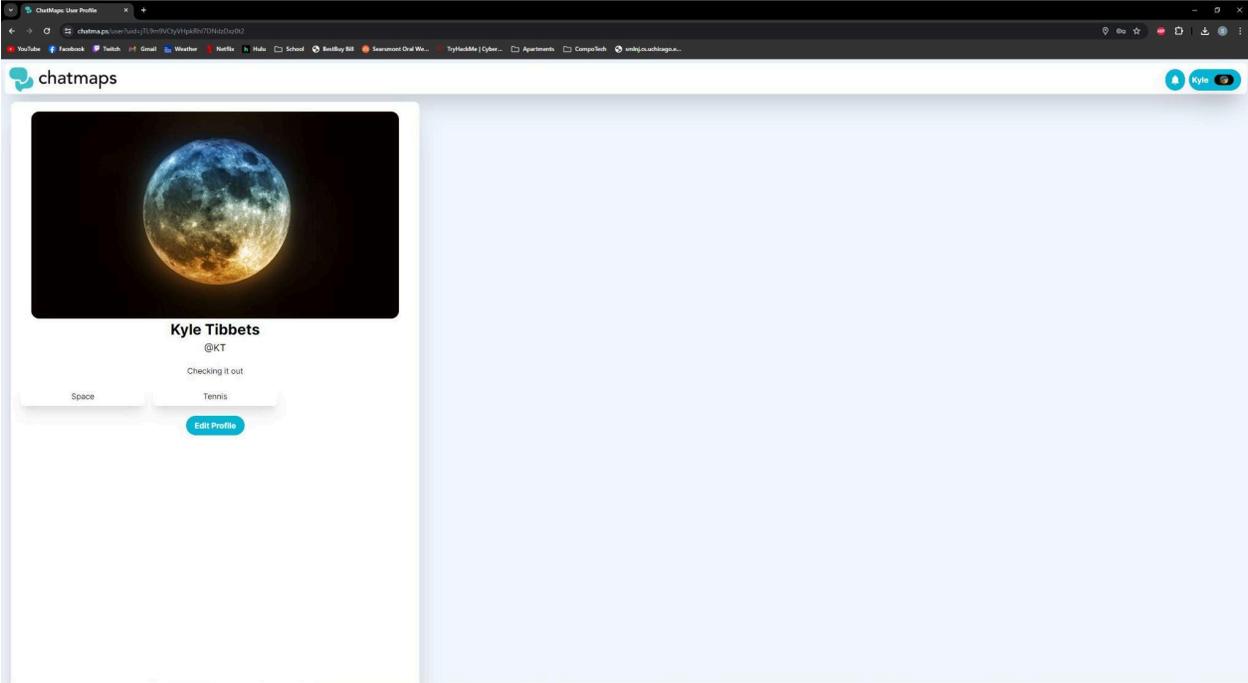
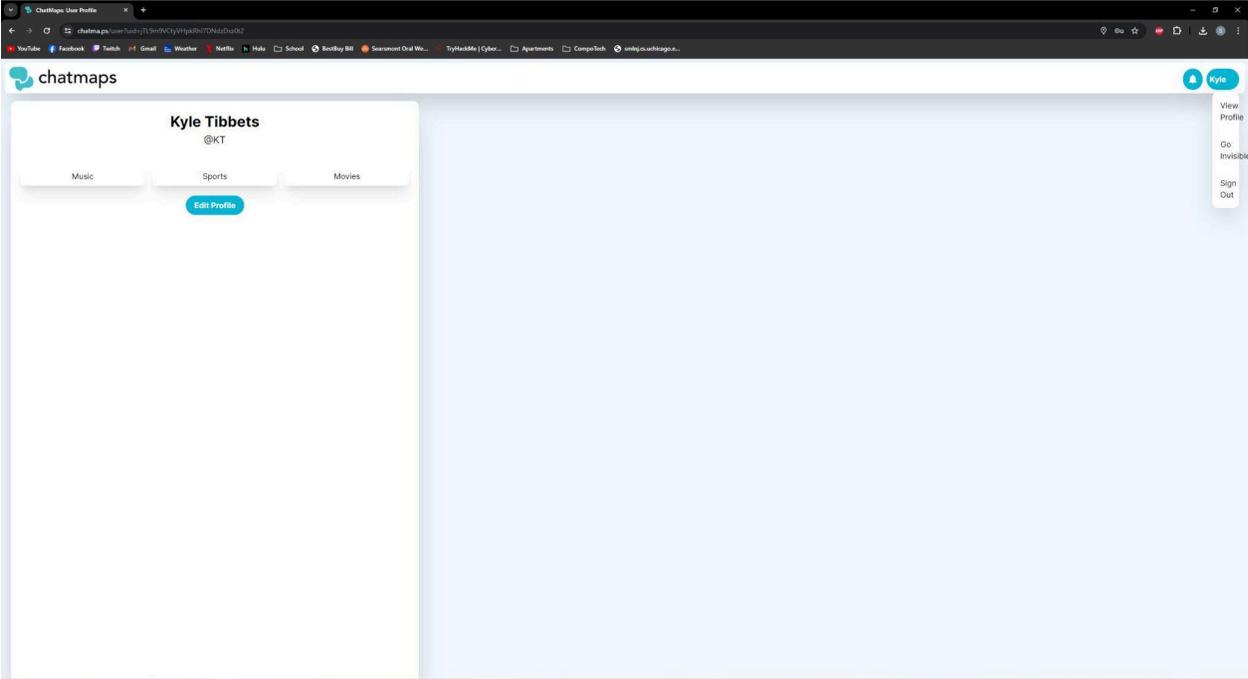
exampleUser {JSON} - contains mock user object [uid, username, lastOnline, firstName, lastName, pfp]

**Output:**

React Component <HomePage>

---

# Acceptance Testing



User: Kyle Tibbets

Feedback:

- Easy to navigate UI
- Very customizable profile page
- Liked the GIF feature in chat rooms
- Small bugs around dropdown menu size when no profile picture is added.

