**Interface Use Case Models**

**Use Case:** Create room

**Related Use Cases:** Generalization of add members

**Steps:**

| **Precondition:** User is logged into system and has selected create new group option |  |
| --- | --- |
| Actor actions | System response |
| 1. Select new group |  |
| 2. Give name to group | 3. Save group |
|  | 4. Return dialog to select members |
| **Postcondition:** User has successfully created a group |  |

**Use Case:** Add members to group

**Related Use Cases:** Specialization of create room

**Steps:**

| **Precondition:** User has successfully created a group, is logged in, and in add member to group menu. |  |
| --- | --- |
| Actor actions | System response |
|  | 1. Give dialog |
| 2. Select text field |  |
| 3. Type user email |  |
| 4. Click submit | 5. Check if user is there |
|  | 6. Add user to group |
| **Postcondition:** User has successfully added member to the group |  |

**Use Case:** Add members to group that does not exist

**Related Use Cases:** Specialization of create room

**Steps:**

| **Precondition:** User has inputted an email that does not belong to an account |  |
| --- | --- |
| Actor actions | System response |
|  | 1. Give dialog |
| 2. Select text field |  |
| 3. Type user email |  |
| 4. Click submit | 5. Check if user is there |
|  | 6. Return error dialog |
| **Postcondition:** User has received error dialog and is able to try another email |  |

**Use Case:** Share lists with groups

**Related Use Cases:** Generalization of edit group lists

**Steps:**

| **Precondition:** User is logged in and is in a sub menu for the desired group |  |
| --- | --- |
| Actor actions | System response |
| 1. Selects option to create a list |  |
| 2. Give list a name |  |
| 4. Adds items to list |  |
| 5. Saves and submits list | 6. Save list to database |
|  | 7. List is updated for every user in group |
| **Postcondition:** List has been created and shared with everyone in group |  |

**Use Case:** Add item to a list

**Related Use Cases:** Specialization of create group lists

**Steps:**

| **Precondition:** User is logged in and is in a sub menu for the desired group |  |
| --- | --- |
| Actor actions | System response |
| 1. Selects list to edit |  |
| 2. Select edit list option |  |
| 4. Adds or removes item from list |  |
| 5. Saves and submits list | 6. Save list to database |
|  | 7. List is updated for every user in group |
| **Postcondition:** List has been edited and shared with everyone in group |  |

**Use Case:** Check off item on list

**Related Use Cases:** Specialization of create group lists

**Steps:**

| **Precondition:** User is logged in and is in a sub menu for the desired group |  |
| --- | --- |
| Actor actions | System response |
| 1. Selects list to edit |  |
| 2. Select items to check off |  |
| 4. Save changes | 5. Changes saved to database |
|  | 6. . List is updated for every user in group |
| **Postcondition:** List has been edited and shared with everyone in group |  |

**Use Case:** User logged into system

**Related Use Cases:** Generalization of user creating an account

**Steps:**

| **Precondition:** User has opened application without being logged in. |  |
| --- | --- |
| Actor actions | System response |
| 1. Select login option |  |
| 2. Select text field for username |  |
| 3. Enters username |  |
| 4. Select text field for password |  |
| 5. Enters password |  |
| 6. Submits information | 7. Checks if information is correct |
|  | 8. Logs user in |
| **Postcondition:** User has been logged into their account |  |

**Use Case:** User enter invalid information

**Related Use Cases:** Specialization of user login

**Steps:**

| **Precondition:** User has entered invalid account login information |  |
| --- | --- |
| Actor actions | System response |
|  | 1. Sends error message to user |
| 2. User reenters credentials again |  |
| 3. User submits information | 4. Checks if information is correct |
|  | 5. Logs user in |
| **Postcondition:** User has correctly reentered their credentials |  |

**Use Case:** Creating an account

**Related Use Cases:** Specialization of user login

**Steps:**

| **Precondition:** User has not created an account and has selected create an account. |  |
| --- | --- |
| Actor actions | System response |
| 1. Select text field for username |  |
| 2. Enters desired username |  |
| 3. Select text field for password |  |
| 4. Enter password |  |
| 5. Submits information | 6. Checks if information is valid |
|  | 7. Saves user to database |
| **Postcondition:** User has created an account |  |

**Use Case:** View outstanding debt

**Related Use Cases:** Specialization of viewing charges

**Steps:**

| **Precondition:** User is logged in and has selected to look at current outstanding debt. |  |
| --- | --- |
| Actor actions | System response |
| 1. Select person who user owes money to | 2. Displays history of transactions between users |
| 3. Selects option to pay person |  |
| 4. Enters amount of money paired | 5. System saves payment |
| **Postcondition:** System has updated debt is user made payment |  |

**Use Case:** View individual charges accessed

**Related Use Cases:** Specialization of viewing charges

**Steps:**

| **Precondition:** User is logged in and has selected to look at monthly charges |  |
| --- | --- |
| Actor actions | System response |
|  | 1. Display the charges |
| 2. User views charges |  |
| 3. User exits dialog | 4. Return to previous menu |
| **Postcondition:** User is in the previous menu |  |