**Use Case:** Add members to group

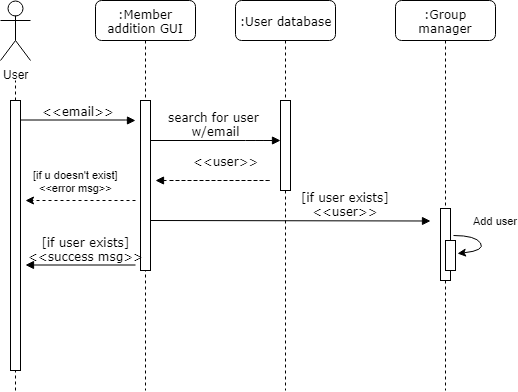
**User Story IDs Addressed:** 011

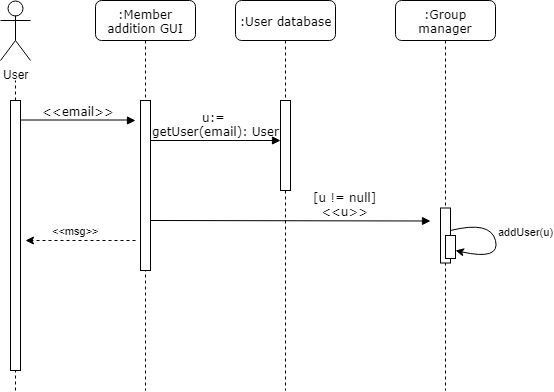
**Related Use Cases:** Included in “Create room”

**Steps:**

| **Pre-Conditions: User has successfully created a group, is logged in, and in add member to group menu.** | |
| --- | --- |
| Actor actions (User) | 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\* |

1. Member addition GUI queries the user database with the inputted email
2. If the user database returns a user
   1. Member addition GUI sends user to group manager
   2. Group manager adds user in group manager
   3. Member addition GUI displays success message
3. Else:
   1. Member addition GUI displays error message to user





**Use Case:** Create room

**User Story IDs Addressed:** 011

**Related Use Cases:** Includes “Add members to group”

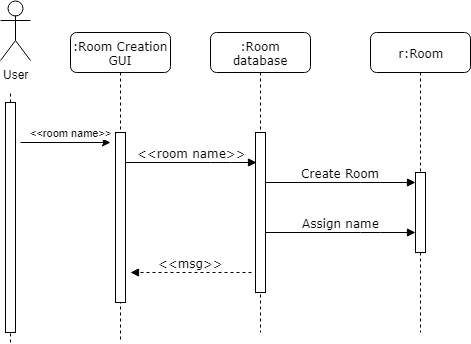
**Steps:**

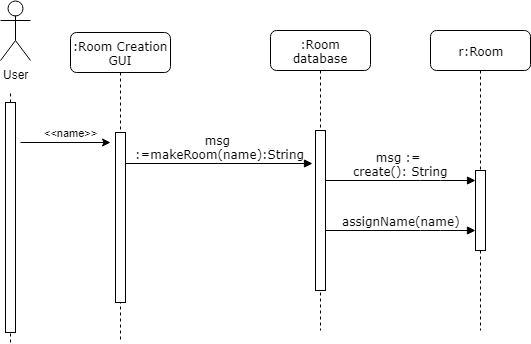
| **Pre-Conditions: User is logged into system and has selected create new group option.** | |
| --- | --- |
| Actor actions (User) | System response |
| 1. Select new group |  |
| 2. Give name to group | 3. Save group\* |
|  | 4. Return dialog to select members |

**Post-Conditions: User has successfully created a group.**

2. User inputs group name to the room creation GUI

1. Room creation GUI sends room name to room database
2. Room database creates new room in room database
3. Room database assigns name to room
4. Room database sends message to room creation GUI





**Use Case:** Create new shared expense

**User Story IDs Addressed:** 001, 003

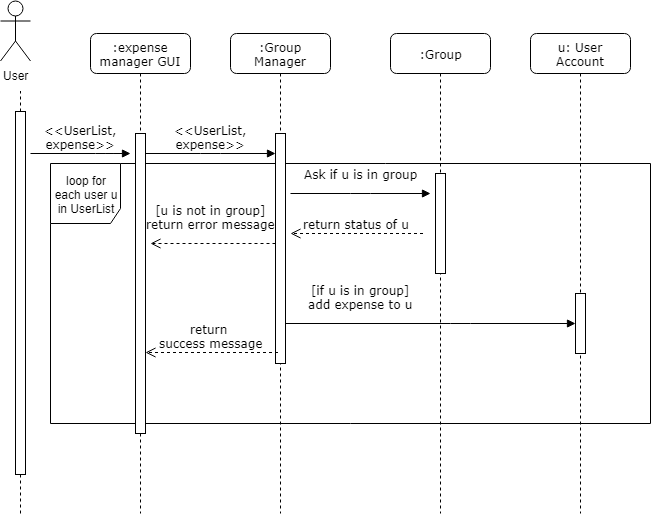
**Related Use Cases:** Generalizes “Create recurring, evenly split charge”; “Create standalone charge”

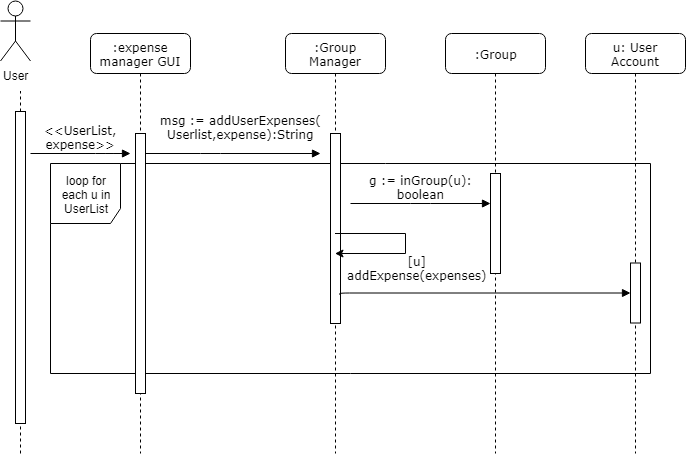
**Steps:**

| Pre-Conditions: Assumes the user has logged into the app, is viewing the group of interest, and has incurred an expense that needs to be shared with the group. | |
| --- | --- |
| Actor actions (User) | System response |
| 1. Select “Create new charge” | 2. Dialogue box appears |
| 3. Input charge details |  |
| 4. Select “Confirm” or enter\* | 5. Dialogue box disappears |
| Post-Conditions: All included group members may see the new charge on their account. | |

3. User sends amount to charge and list of users to split the charge with to expense manager GUI

1. Expense manager GUI sends charge details to group manager
2. For each user in the list of users to split the charge with:
   1. Group manager checks if user is in the group
   2. If user is in group:
      1. Group manager adds charge to user account
   3. Else:
      1. Group manager sends error message to expense manager GUI
3. Group manager sends completion message to expense manager GUI





**Use Case:** Log In

**User Story IDs Addressed:** 012

**Related Use Cases:** Includes “Create Account”

**Steps:**

|  |  |
| --- | --- |
| **Pre-Conditions: User has opened application without being logged in.** | |
| Actor actions (User) | System response |
| 1. Select login option |  |
| 2.1 Enter username  2.2 Enter password  Alternative: User does not have account  2.1 “Create Account” |  |
| 6. Submits information | 7. Checks if information is correct\* |
|  | 8. Logs user in\* |
| **Post-Conditions: User has been logged into their account.** | |

6. User submits username and password

7. Login GUI sends username and password to login controller

7.1. Login controller sends login information to user database for validation

7.2. If the user exists:

7.2.1. User database sends validation to login controller

7.2.2. Login controller logs the user in

7.2.3. Login controller sends confirmation message to login GUI

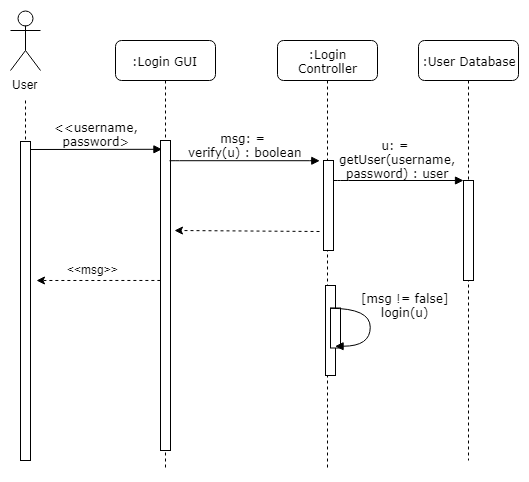
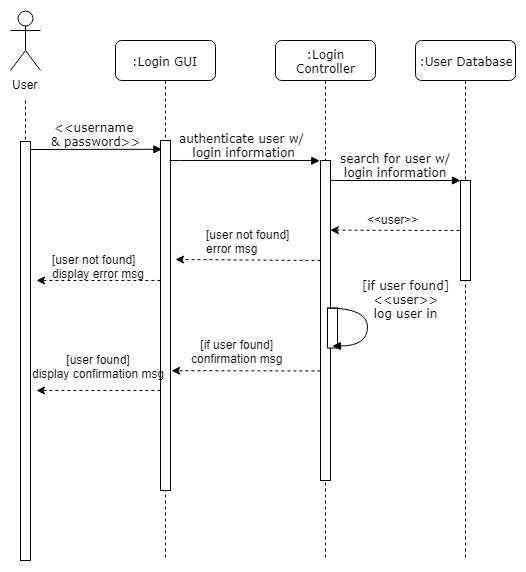
7.2.3. Login GUI displays confirmation message to user

7.3. else:

7.3.1. User database sends confirmation failure to login controller

7.3.2. Login controller sends error message to login GUI

7.3.3. Login GUI displays error message to user



**Use Case:** Create Account

**User Story IDs Addressed:** 012

**Related Use Cases:** Included in “Log In”

**Steps:**

|  |  |
| --- | --- |
| **Pre-Conditions: User has not created an account and has selected create an account.** | |
| Actor actions (User) | System response |
| 1. Select text field for username |  |
| 2. Enters desired username |  |
| 3. Select text field for password |  |
| 4. Enter desired password |  |
| 5. Submits information | 6. Checks if information meets minimum system requirements\* |
|  | 7. Saves user to database\* |
| **Post-Conditions: User has created an account.** | |

5. User submits username, email and password

6. Register checks is username is available

6.1. Register sends information to user data base to check if there is already a user

6.2. No account already exists

6.2.1. Register controller checks if the user information meets the minimum requirements

6.2.2. Requirements met

6.2.2.1. Register controller creates an account with input information

6.2.2.2. Account saved to user database

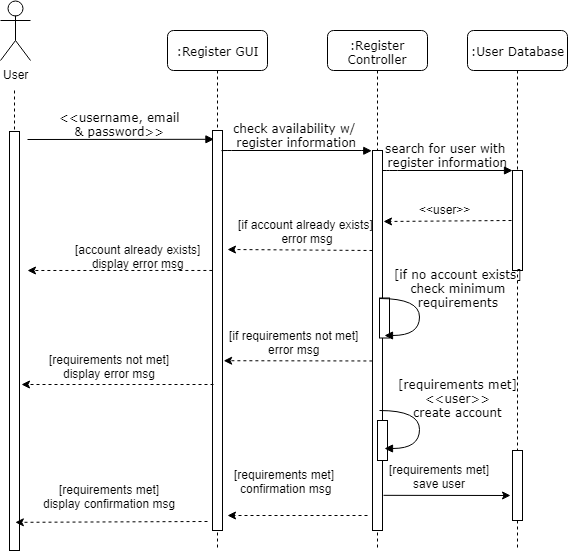
6.3.2.1. Register controller create confirmation message

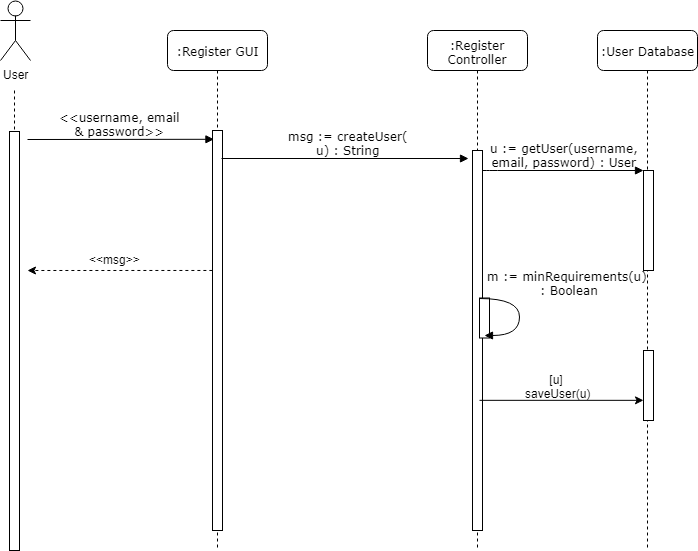
6.2.3. else

6.2.3.1. Register controller creates error message

6.3. else

6.3.1. Register controller creates error message





**Use Case:** Share lists with groups

**User Story IDs Addressed:** 009, 010

**Related Use Cases:** Includes “Edit Lists”

**Steps:**

|  |  |
| --- | --- |
| **Pre-Conditions: User is logged in and is in a sub-menu for the desired group.** | |
| Actor actions (User) | System response |
| 1. Selects option to create a list | 2. Open dialogue box |
| 3. Give list a name |  |
| 4. Add items  (via Include: “Edit list”) |  |
| 5. Save and submits list | 6. Save list to database; exit \* |
| **Post-Conditions: List has been created and shared with everyone in group.** | |

5. User submits list name

6. List GUI sends list name to group

6.1 Group creates a list

6.2. List assigned name

6.3. Group saves list to list data base

6.4. Group sends completion message to list GUI

6.5. List GUI displays completion message

