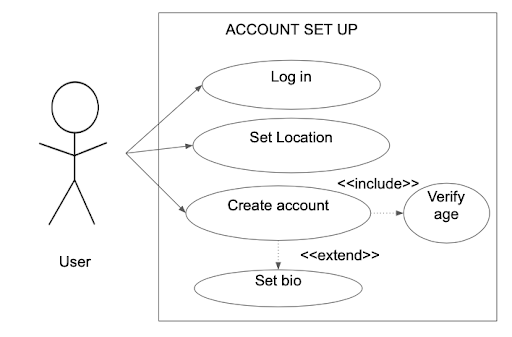
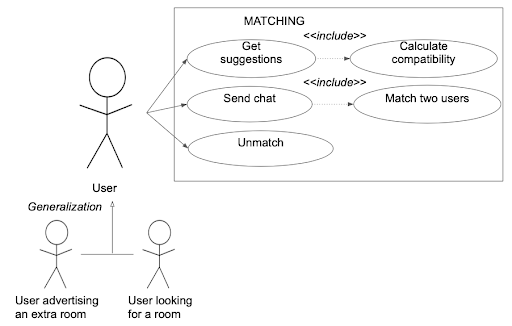
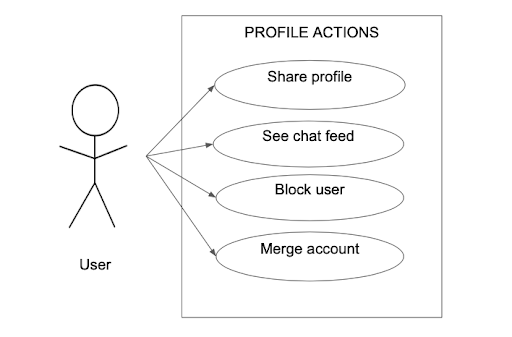
***Use Cases models:***

1. Matching
2. Account Setup
3. Profile Actions





***Use case list:***

1. Login
2. Set Location
3. Create Account
4. Verify Age
5. Set Biography
6. Get roommate suggestion
7. Calculate Compatibility
8. Send Chat to another profile
9. Unmatch with matched user
10. Match Two users
11. Share Profile
12. See Chat Feed
13. Block User
14. Merge Accounts
15. Upload Profile Pictures
16. Set Username
17. Set Password
18. Location Permissions
19. Set preferences (filters)
20. Set college

**1)**

| **Title:** | **Log In** |
| --- | --- |
| **Description:** | This use case describes the steps for login. |
| **System Under Design:** | RoomUp |
| **Primary Actor:** | RoomUp User |
| **Participants:** | RoomUp User |
| **Goal:** | Login |
| **Following Use Cases:** | NA |
| **Precondition:** | RoomUp displays login screen |
| **Success Post-condition:** | Users are sent to the profile update page to input their personal information. |

| **Actor:**  1) User types username and password  4) User has access to the account | **System Responses:**  2) System verifies username and password  3) System grants access to the user’s account |
| --- | --- |

**2)**

| **Title:** | **Set Location** |
| --- | --- |
| **Description:** | Allow the app to save a location to the user profile |
| **System Under Design:** | RoomUp |
| **Primary Actor:** | RoomUp User (seeking room/offering room) |
| **Participants:** | RoomUp User (seeking room/offering room)(opposite of which one was chosen as primary actor) |
| **Goal:** | Add a location to a user account |
| **Following Use Cases:** | <<include>>Calculate compatibility  <<extends>>Create Account |
| **Precondition:** | Location Permissions - If a user chooses to use the API to set their current location, then we will need one time permissions from the user. |
| **Success Post-condition:** | User location is available to view and saved to a user account |

| **Actor:**   1. User clicks “add location”’   **3.(a) user gives one time permissions**  **(b) user enters location into search** | **System Responses:**   1. **(a) set location using current location**   **(b) set location by search**  **4. (a) System uses fuse location provider**  **(b) use google API**  **5. System saves location to user account** |
| --- | --- |

**3)**

| **Title:** | **Create** **Account** |
| --- | --- |
| **Description:** | Create an account with a UID for each user, which will hold account information such as login information, name, age, D.O.B, etc. |
| **System Under Design:** | RoomUp |
| **Primary Actor:** | RoomUp User |
| **Participants:** | NA |
| **Goal:** | Have a unique UID for each user so they can save their account information and login successfully. |
| **Following Use Cases:** | <<include>> Age verification  <<extend>> Add biography |
| **Precondition:** | -This case assumes that the user has a UID from creating an account. |
| **Success Post-condition:** | User has access to our app and its features. |

| **Actor:**   1. User clicks the create account button   3. User inputs a new username, password. Then they click the continue button.  5. User input personal information. | **System Responses:**   1. System takes them to the create account login page.   4. The system brings them to the account creation page which includes boxes for name, age, and other profile information. |
| --- | --- |

**4)**

| **Title:** | **Verify Age** |
| --- | --- |
| **Description:** | The user must verify their age (+18 years old) otherwise the system must deny them service. |
| **System Under Design:** | RoomUp |
| **Primary Actor:** | RoomUp User |
| **Participants:** | New RoomUp user |
| **Goal:** | To inhibit underage users from signing up as they are not old enough. |
| **Following Use Cases:** | Users must verify age in order to create an account for RoomUp. |
| **Precondition:** | -Users must check a box stating that they are over 18 years old. |
| **Success Post-condition:** | User gets access to our app and its features. |

| **Actor:**   1. User fills out account info.   3. User checks box.  5.   1. User fills out rest of information 2. User clicks the continue button. | **System Responses:**   1. System displays a box asking the user to verify that they are over 18 years old.   4. System verifies that the user checked the box (Which means they verify they are over 18 years old). (True)   1. If the user didn’t check the box display warning that they must check the box. 2. If the user did check the box:    1. Check their D.O.B to see if they are 18 years old       1. If No: Display warning and block them from creating an account until they check the box       2. If Yes, move to next step    2. Then bring them to the homepage of the app. |
| --- | --- |

**5)**

| **Title:** | **Set Biography** |
| --- | --- |
| **Description:** | Allow user to add a description of themselves to their account |
| **System Under Design:** | RoomUp |
| **Primary Actor:** | RoomUp user |
| **Participants:** | N/A |
| **Goal:** | Save a user given description to a user account so it can be displayed in the home feed when they are suggested to another user. |
| **Following Use Cases:** | <<extends>> Create Account |
| **Precondition:** | -User must create a username and password. |
| **Success Post-condition:** | Save and immediately be able to view the user biography. |

| **Actor:**   1. Have the user click “add bio” button   3. Type biography into text box | **System Responses:**   1. Display a text box for the user to type their biography into 2. Save text to account and display to users |
| --- | --- |

6)

| **Title:** | **Get Roommate Suggestion** |
| --- | --- |
| **Description:** | The roommate suggestion is highly dependent on if the two users swipe up on each other in the swipe page. If not, then the user will not get a user suggestion. |
| **System Under Design:** | RoomUp |
| **Primary Actor:** | RoomUp user |
| **Participants:** | RoomUp user ‘a’ and RoomUp user ‘b’. Two RoomUp users are involved. |
| **Goal:** | The goal is to match two people only if they swipe up on each other so they can eventually start chatting with each other. |
| **Following Use Cases:** | Computer compatibility is an <<include>> as it is required to compute the compatibility of two users before giving them the suggestion. |
| **Precondition:** | -Two users must swipe up on each other, and the compute compatibility use case must compute if they are compatible or not. |
| **Success Post-condition:** | Two users are matched and can chat with each other. |

| **Actor:**   1. User ‘a’ swipe up on people they like.    1. User ‘b’ swipes up on users they like.    2. Both users swipe up on each other. | **System Responses:**   1. Use case “Compute compatibility” will measure if the users are a good match.    1. If yes → Send notification to both users saying they matched.    2. If no → do nothing. |
| --- | --- |

7)

| **Title:** | **Calculate Compatibility** |
| --- | --- |
| **Description:** | Calculate compatibility between two users to determine if they should be suggested to one another |
| **System Under Design:** | RoomUp |
| **Primary Actor:** | RoomUp user 1 |
| **Participants:** | RoomUp user 2 |
| **Goal:** | Generate a compatibility score between two users based on the user account information. |
| **Following Use Cases:** | <<include get roommate suggestion>> |
| **Precondition:** | Have a user account created and have all essential information filled out (age, etc.) |
| **Success Post-condition:** | A number score of compatibility is assigned between two users. |

| **Actor:**   1. User 1 creates account 2. User 1 inputs account information | **System Responses:**   1. System prompts user 1 to fill out account information 2. System saves account information to the user 1 account 3. System compares this account to user 2 account. For every information piece that matches +1 is added to the score of compatibility. This score is divided by the number of fields filled out 4. The user pairing is now assigned the compatibility score |
| --- | --- |

8)

| **Title:** | **Send Chat to Another Profile** |
| --- | --- |
| **Description:** | Once two users get matched, they will have the ability to message each other to get to know each other better. They will have unlimited messaging. |
| **System Under Design:** | RoomUp |
| **Primary Actor:** | RoomUp general user. |
| **Participants:** | RoomUp user ‘a’ and RoomUp user ‘b’ |
| **Goal:** | To protect the privacy of users, only those who are matched can message each other. |
| **Following Use Cases:** | <<Include>>Match two users |
| **Precondition:** | -Two users must be matched to chat with each other to protect users from random chats. |
| **Success Post-condition:** | Chat rooms are created and messages between the two users can be sent. |

| **Actor:**   1. User accepts an invitation to chat with a matched user.   3. User ‘a’ sends message (and so forth) | **System Responses:**   1. New chat room between user ‘a’ and user ‘b’ is created.   4. Message is sent to user ‘b’   1. User ‘b’ receives message |
| --- | --- |

9)

| **Title:** | **Unmatch with Matched User** |
| --- | --- |
| **Description:** | If two users are matched and don’t get along, they can have the option to unmatch with the other user to protect themselves, and to avoid conflict. |
| **System Under Design:** | RoomUp |
| **Primary Actor:** | RoomUp general user |
| **Participants:** | Two matched users |
| **Goal:** | To unmatch to users to block chats from unwanted matched users. |
| **Following Use Cases:** | NA |
| **Precondition:** | -The two users must already be matched. |
| **Success Post-condition:** | Two users are no longer matched. |

| **Actor:**   1. User goes to the chat page and clicks on the matched user profile picture.   3. User clicks on the unmatch button | **System Responses:**   1. System shows a button that says “Unmatch” in red text.   4. System unmatched the two users and denies messaging between the two users. |
| --- | --- |

10)

| **Title:** | **Match Two Users** |
| --- | --- |
| **Description:** | Match two users together as potential roommates |
| **System Under Design:** | RoomUp |
| **Primary Actor:** | RoomUp user 1 |
| **Participants:** | RoomUp user 2 |
| **Goal:** | Confirm two users are compatible and allow two users to advance to the next steps of being allowed to chat with each other |
| **Following Use Cases:** | <<include send chat>>  <<include calculate compatibility>> |
| **Precondition:** | Have a compatibility score calculated between user 1 and user 2 |
| **Success Post-condition:** | Identified two users have matched and have this tagged this in their user accounts. Give permissions to chat with each other |

| **Actor:**  2) user 1 and user 2 swipe UP on each others accounts | **System Responses:**   1. If the compatibility score between user 1 and user 2 > 60% then suggest users to each other by displaying the respective accounts in their feed 2. In both accounts set match with the other user to be true 3. Allow users to be have access to the other user in the chat feature with the ability to send a message 4. Display the match at the top of the chat feed |
| --- | --- |

TEMPLATE:

| **Title:** |  |
| --- | --- |
| **Description:** |  |
| **System Under Design:** |  |
| **Primary Actor:** |  |
| **Participants:** |  |
| **Goal:** |  |
| **Following Use Cases:** |  |
| **Invariant:** |  |
| **Precondition:** | - |
| **Success Post-condition:** |  |

| **Actor:** | **System Responses:** |
| --- | --- |