**Use Case Name**: JoinRoon

**Scope**: System Use Case

**Level**: User Goal

**Main participants**: User

**Stakeholders and users whose concerns:** User who had logged in: join an existed room to play a game.

**Prerequisites**: Users login their account.

**Postcondition**: User join room successfully.

**Mainstream**:

1. Users click the "JoinRoom" button on the page.

2. The system receives the user's request, jumps to the room list page, searched the rooms that have been created and displays the list of rooms for users to choose.

3. Users choose a room, and click "join" button to join the room.

4. The system adds users to the room and jumps to the room page.

**Expansion process**

No room existed:

1. In step 2, no room has been created. The system prompts the user to create a room.

Failed to join room:

1. In step 4, an error occurs such as the internet instability when the user join the room. The System prompts the user the failure of join room, prompts the user to try again or choose another room and reload the reloads the list of rooms.

**Special Needs**: None.

**Frequency:** It may happen at any time, but in general, the expectation of times that a user join a room per game is less than 1, so the rate will not be too high.

**Use Case Name**: SelectRole

**Scope**: System Use Case

**Level**: Subfunction

**Main participants**: User

**Stakeholders and users whose concerns:** User who had joined a room: select a role that the user will use in the next game.

**Prerequisites**: Users join an existed room successfully.

**Postcondition**: The user selects a role successfully.

**Mainstream**:

1. The user clicks the box of himself or herself on the page.
2. The system pops up a role select page that is less than the room page, which contains all the roles that the user owns.
3. The user clicks a role to select a role, click “Select” button to confirm the choice.
4. The system hides the role select page and change the role information of the user on the room page.

**Expansion process**

The user doesn’t select a role:

1. In step 1, the user doesn’t select a role and then click the “Prepare” button. After the game begins, the system will assign a random role to the user that the user owns.

The role has been chosen by the other user:

1. In step 3, the role has been selected by the other user. The system prompts the user to select again.

**Special Needs**: None.

**Frequency:** It may happen at any time after it that the user joins a room and before the user prepares..

**Use Case Name**: Prepare

**Scope**: System Use Case

**Level**: Subfunction

**Main participants**: User

**Stakeholders and users whose concerns:** User who has joined a room: claim that the user has prepared to start a game.

**Prerequisites**: Users join a room.

**Postcondition**: The user prepares successfully.

**Mainstream**:

1. The user clicks “Prepare” button to claim that he or she is prepared.
2. The “Prepare” button turns to “Cancel” button. Users can click it to cancel prepare state.
3. When all the users are prepared, the game start.

**Expansion process**: None.

**Special Needs**: None.

**Frequency:** It may happen at any time after it that the user joins a room and before the game start.