

Final Project Specification: Miller's Hollow Online

Team **Miller's Hollow**

Yiyun Yao(yiyuny) & Jin Wang(jinw2) & Shangjie Chen(shangjie)

I Functionality

I.1 Not in a Game

I.1.1 Authentication

1. Login: Registered Users may login to the website by correct username and password, and get authenticated and authorized
2. Register: Unregistered Users may register to the website by providing username, password and email
3. Activate Account: Registered User may activate the account by click the activation link in the email automatically sent when successfully registered
4. Logout: Authenticated User may logout the website
5. Change Password: Authenticated User may change the password by correct current password
6. Reset Password: User who forget their password may reset the password by providing the email correspond to the account. User may set their new password by click the link in the email automatically sent when password is successfully reset

I.1.2 Profiling & Setting

The main workflow is shown in Figure 1.

1. Editing User Information: Authenticated User may change the first name, last name and email
2. Upload avatar: Authenticated User may upload a picture to change the avatar
3. Set preference game: Authenticated User may set the preference game in the settings to skip the game choose phase in build game and matching game
4. Set waiting time: Authenticated User may set the waiting time in the settings to change the default waiting time before every speech starts
5. Set Music, Display and Log settings: Authenticated User may set the music volume, display settings(turn on/off camera of other user in case the network is not good enough) and log verbosity in the settings

I.1.3 Game Mode

The main workflow is shown in Figure 2.

1. Matching Game: Authenticated User may start a matching game where different unrelated players attend
2. Build Room Game: Authenticated User may build a room game where friends are invited to play game together; in the mean time, a share token is generated for other users to join
3. Join Room Game: Authenticated User may join a room game by a share token

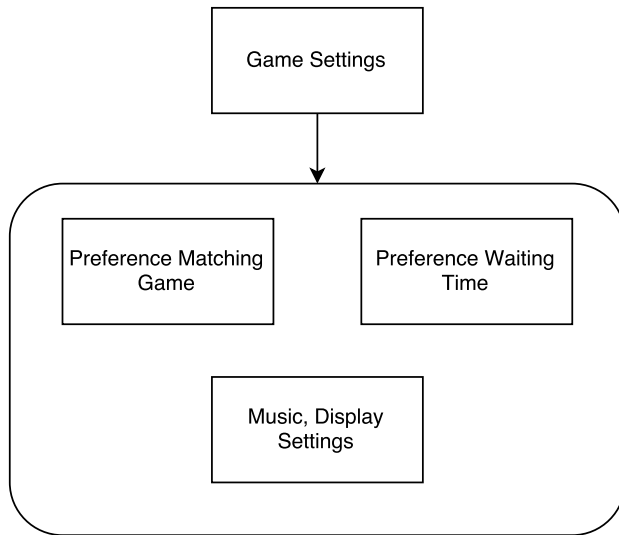


Figure 1: Functionality: Profiling & Setting

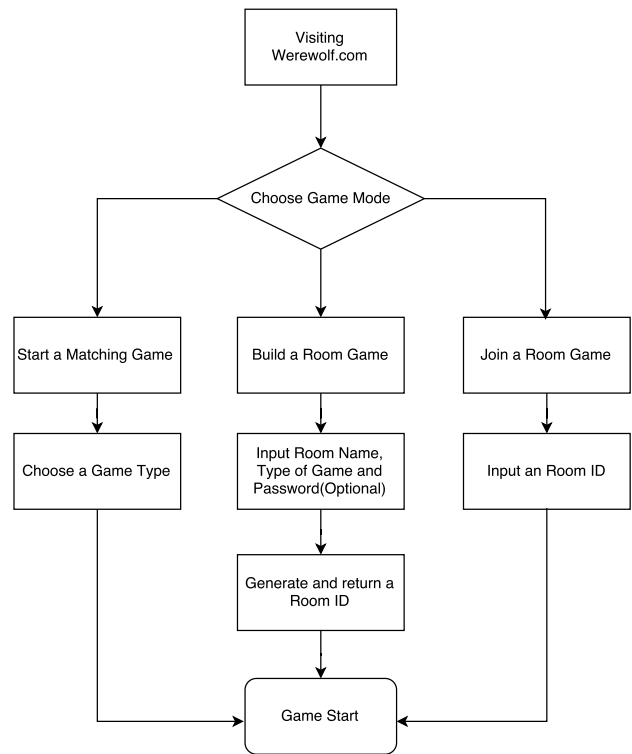


Figure 2: Functionality: Game Mode

1.1.4 Room Member

The main workflow is shown in Figure 3.

When Authenticated User join a Room Game, he/she automatically becomes a Room Member

1. Change to empty seat: Room Member may change the seat to a new one if it is empty
2. Leave the room: Room Member may leave the room by clicking the Leave Button
3. Chat: Room Member may chat in the room and the message can be seen by all the Room Members
4. Get ready: Room Member may get ready by clicking the Ready Button(when all the Room Member is ready, the Room Owner can Start the game)
5. Cancel ready: Room Member who is ready may cancel it by clicking the Cancel Button

1.1.5 Room Owner

The main workflow is shown in Figure 3.

Room Owner have the functionality of Room Member from 1 to 3. A Room Member can be selected as a Room Owner by the previous Room Owner. The original one is the user who build the room. A Room Game can only have one Room Owner. Room Owner has the extra functionalities as follow.

1. Set a Room Member as Room Owner: Room Owner may choose another user as the new Room Owner
2. Kickout Room Member: Room Owner may kickout Room member
3. Start Game: Room Owner may start the Room Game by clicking Start Button when all the Room Member is ready

1.1.6 Achievements & Tasks

The main workflow is shown in Figure 4.

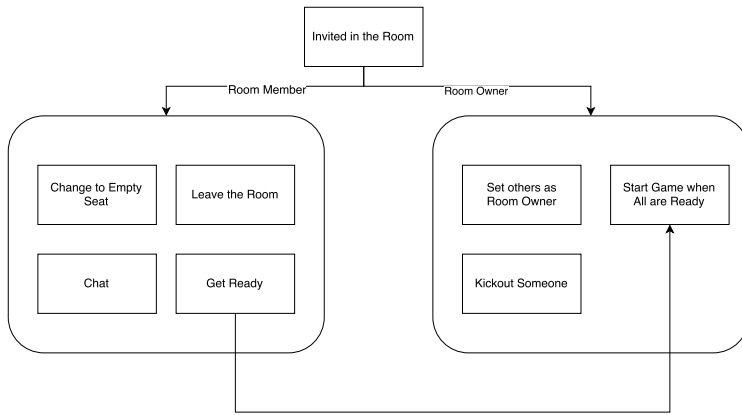


Figure 3: Functionality: Room Member & Room Owner

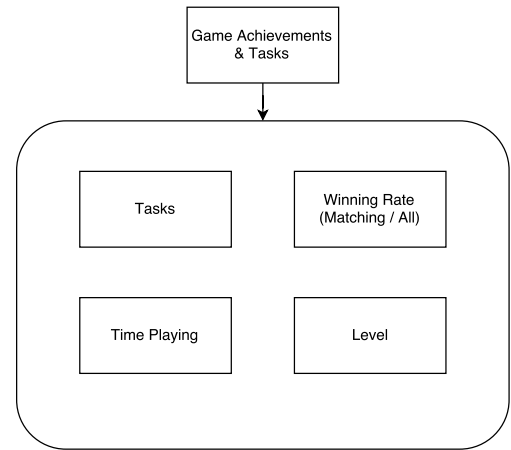


Figure 4: Functionality: Achievements & Tasks

1. Tasks: Authenticated User may receive one task a day automatically in the Achievements & Tasks Board, and get awards by completing it. One user can hold three tasks at max
2. Exchange task: Authenticated User may exchange a task one day, the new one will be different from the old one
3. Winning Rate: Authenticated User may check the winning rate in the Achievements & Tasks Board
4. Time Played: Authenticated User may check the time played in the Achievements & Tasks Board
5. Level: Authenticated User may check the level in the Achievements & Tasks Board, which is mainly based on the winning rate and time played. The level is the main dependency of Matching Game

1.2 In a Game

When a Game starts, a Room Member automatically becomes a Player

1.2.1 Get Identity Card

Before the actual game starts, player can get a identity card which indicates different ability and winning condition in this game. All the identity is shown as follow.

1. Villager
2. Werewolf
3. Seer
4. Witch
5. Hunter
6. Idiot

1.2.2 Day Phase

The main workflow is shown in Figure 5.

1. Vote: Player may vote for sheriff or exiling a player by dragging the Hand Button in the menu to the player
2. Speech: Player may give a speech in his turn. The time of the speech is 1 minute, and player can always end speech in advance

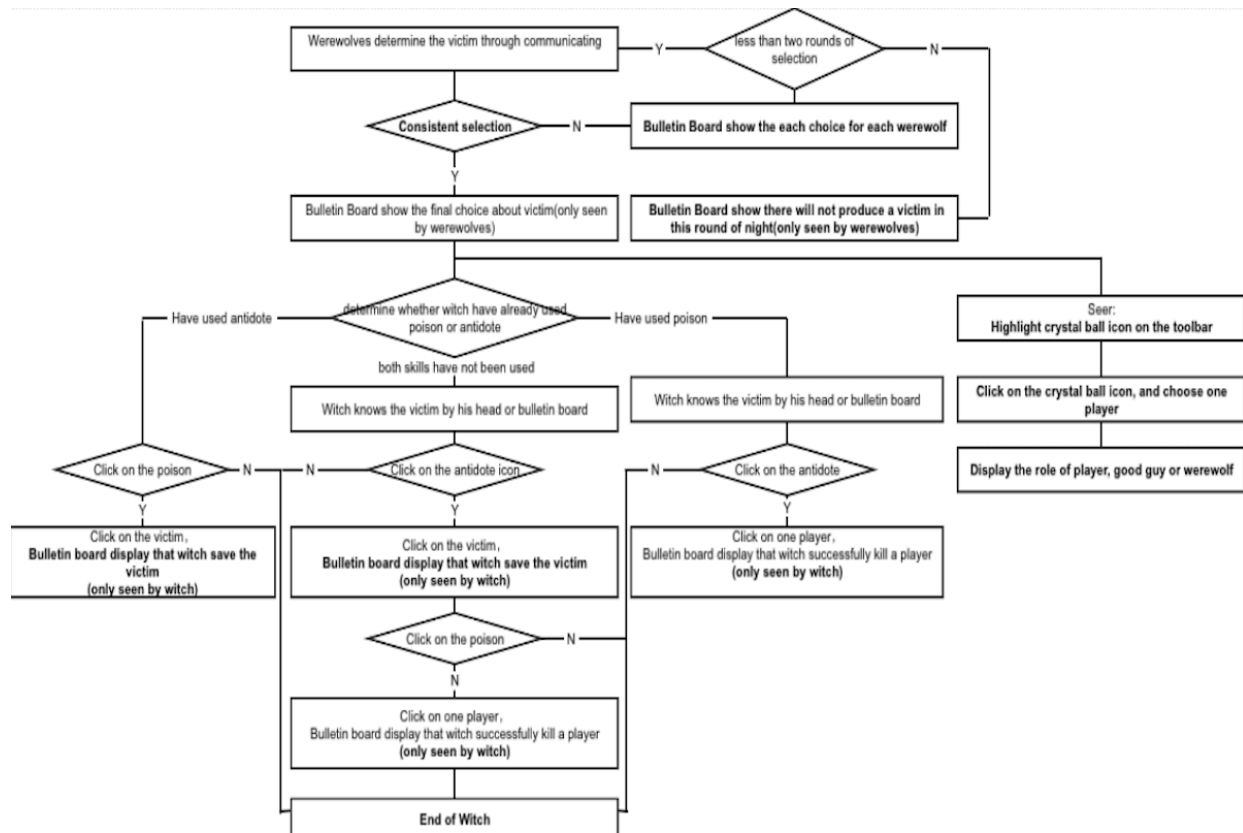


Figure 6: Functionality: Night Phase

All werewolf must agree on one victim. If not agreed in two rounds, there will be no victims.

2. Poison Potion: Witch can poison a player to death through dragging the Poison Button in the menu. The skill can only be used once and can't be used with Revive Potion in the same night
3. Revive Potion: Witch can save the victim tonight by clicking the Antidote Button in the menu. The skill can only be used once and can't be used with Poison Potion in the same night
4. Foresee: Seer can check whether one player is a werewolf by dragging on crystal ball in the menu to the player

2 Data Model

2.1 User

User class is Django basic authenticate model. See Class Diagram in Figure 7 for detail

1. first_name: First Name
2. last_name: Last Name
3. username: Username for login
4. password: Password for login
5. email: Email for reset password

2.2 Profile

Profile class store the user data for the website. See Class Diagram in Figure 8 for detail

User
+ username: CharField
+ password: CharField
+ first_name: CharField
+ last_name: CharField
+ email: CharField

Figure 7: Data Model: User

Profile
+ user: OneToOneField(User)
+ avatar: ImageField
+ no_game: PositiveIntegerField
+ no_win: PositiveIntegerField
+ level: PositiveIntegerField
+ tasks: OneToManyField(Task)
+ preferred_game: ChoiceField(Werewolf/Avalon)
+ waiting_time: PositiveIntegerField
+ log_verbosity: PositiveIntegerField

Figure 8: Data Model: Profile

Game
+ state: PositiveIntegerField
+ role1...12: Role

Figure 9: Data Model: Game

Role
+ role: ChoiceField(Villager/Seer/Witch/Hunter/Idiot)
+ user: OneToOneField(Profile)
+ skill_state1: IntegerField
+ skill_state2: IntegerField

Figure 10: Data Model: Role

1. user: user relation to Django Auth User
2. avatar: user avatar
3. num_game: number of games that user played
4. num_win: number of games that user won
5. level: user level showing the user strength
6. tasks: tasks assigned to user
7. preferred_game: user preferred game to play, shortcuts for play
8. waiting_time: the time user wait before his speech
9. log_verbosity: the log verbosity for the game state

2.3 Game

Game class store the data for each game. See Class Diagram in Figure 9 for detail

1. state: different states in a game, such as when state equal 0, it means the first night round
2. role1...12: foreign key relate to Role

2.4 Role

Role class store the role information. See Class Diagram in Figure 10 for detail

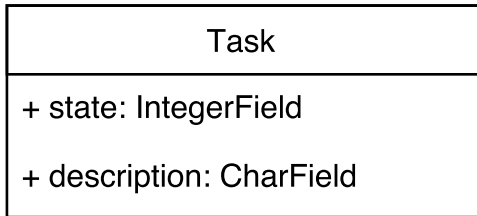


Figure 11: Data Model: Task

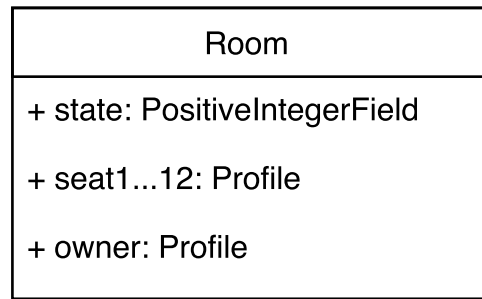


Figure 12: Data Model: Room

1. role: particular role
2. user: foreign key relate to a Profile
3. skill_stater: whether role have used its first skill
4. skill_state2: whether role have used its second skill

2.5 Task

Task class store the particular task information. See Class Diagram in Figure 11 for detail

1. state: if the task is in which process
2. description: task description for user

2.6 Room

Room class store the room information. See Class Diagram in Figure 12 for detail

1. state: if the room is in ready, start or end state
2. seat1...12: foreign key to Profile, indicates which user the seat belongs to
3. owner: foreign key to Profile, indicates which user is the room owner