CS414 Team Project - F17

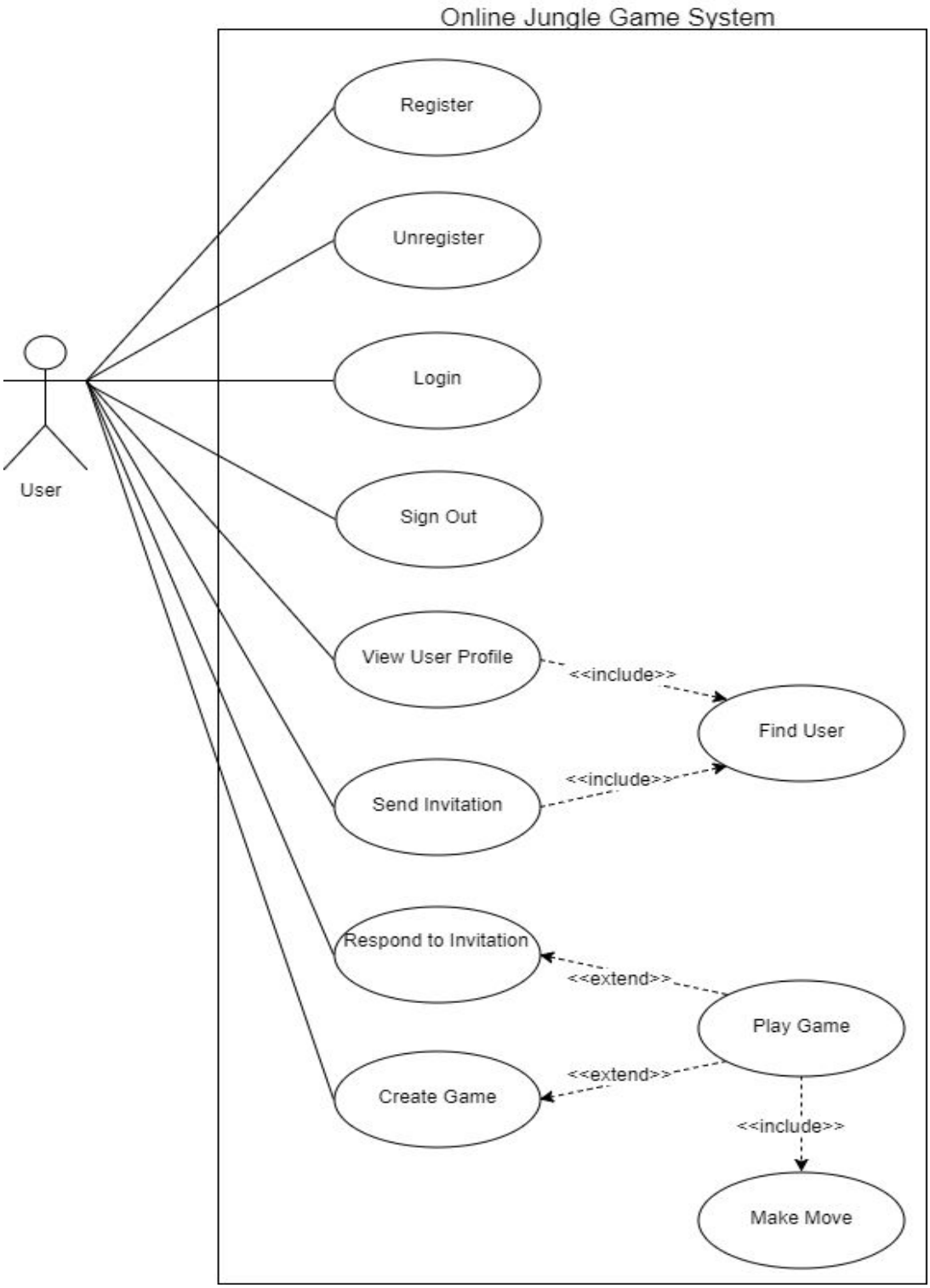
Team: And Yet It Compiles

Members: Laura South, Isacc Mauro, Duck Keun Yang, Brian Larson, Yan Wang

Core Requirements

1. Any person can register to the system. The registration requires an email (which is unique), a password, and a nickname (which is also unique).
2. A registered user can create a new game. The registered user becomes a player of the created game.
3. A registered user can invite another registered user (or set of registered users) to join a created game.
4. A registered user can accept or reject an invitation to join a game. If the user accepts the invitation, she becomes a player of the game.
5. A registered user can be part of different games at the same time.
6. A registered user only has access to the games she is a player of.
7. A player can quit a game at any time.
8. A registered user can unregister from the system.
9. The system must record the history of games played by a user. The record of a game includes the opponent, start date and time, end date and time, and the end result of the game (i.e., win, loss, tie, draw, abandoned, etc.)
10. A registered user has a profile, which consists of her nickname and history of played games. User profiles are only visible to other registered users.
11. A game cannot start until the minimum number of players required for the game have joined.
12. Once a game starts, new players cannot join.
13. The systems must determine which player starts the game according to the rules of the X game. If there are no specific rules, the user who has created the game is the one making the first move.
14. The system determines whose turn it is according to the game rules.
15. A player can only make moves in her active games.
16. A player can only make moves if it is her turn to play.
17. Players can only make allowed moves. Allowed moves are given by the game rules.
18. The system saves the state of active games. Players can play asynchronously but following the turn rules.
19. The system must determine when a game is over. The system must also determine who is the winner and the loser of each game, when there is a tie, or when there is a draw according the game rules.

Use Case Diagram



Use Case Descriptions

ID:	1
Use Case Name:	Register
Overview:	Enters user information into the system and allows them to access other features in the system..
Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	Actor: User
Stakeholders and Interests:	User: Wants to register his account, and play the jungle game System Administrator: Wants to store the new user's information in the Database System without an error. Wants to ensure that there is no duplicate users in the database.
Preconditions:	None
Success Guarantee:	1. User's information has been saved into the Database System. 2. User is registered in the system
Main Success Scenario:	1. System presents forms that are needed for registering a new user into the database. 2. User fills the forms, and submits it. 3. System saves User's information to the database. 4. User's information include Email and password. 5. System can check the user's information by format of the information. 6. User is redirected to the login screen.
Extensions:	1a. If the email or the nickname are not unique in the database: 1. System notifies user and do not save User's information to the database.
Special Requirements:	None
Technology and Data Variations List:	1a. User information entered by keyboard. 1b. The password must be at least 8 characters. 1c. The password must include at least one special character.

	1d. The username must consist of alphabets and numbers.
Frequency of Occurrence:	Could happen once when User executes the Online Jungle Game Software.
Miscellaneous:	

ID:	2
Use Case Name:	Unregister
Overview:	Removes an existing user from the system.
Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	Actor: User
Stakeholders and Interests:	User: Wants to remove his account. System Administrator: Wants to ensure that the removed user does not exist in the Database
Preconditions:	1. User is registered in the system. 2. User is logged on to the system.
Success Guarantee:	1. User's information has been removed from the system.
Main Success Scenario:	1. When system needs to update or there are some bugs in the game, system will compel User to sign out. 2. System asks User if User is going to unregister for sure. 3. User clicks 'Yes' button. 4. User is logged out from the system.
Extensions:	1a. If User clicks 'No' button: 1. User's information has remained in the database 2. The game will show a conclusion of this game that the leaving player will be a loser and the opponent is winner.
Special Requirements:	1. User's information should be deleted right after User clicks 'Yes' button.
Technology and Data Variations List:	1a. User's decision is made by clicking 'Yes' or 'No' button with mouse.
Frequency of	Could happen multiple times after User is registered and exists in the

Occurrence:	database
Miscellaneous:	None

ID:	3
Use Case Name:	Login
Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	Actor: User
Stakeholders and Interests:	User: Wants to log on to the system in order to play the game. System Administrator: Wants to make sure that User is not a malicious user. Wants to ensure that the information User submitted is same as the information stored in the database.
Preconditions:	1. User is registered in the system.
Success Guarantee:	1. User is logged on to the system.
Main Success Scenario:	1. System represents forms for the username and the password. 2. User enters the username and the password, and submits. 3. System checks given User's information is matching the one in the database. 4. User is logged on to the system, and can access the other features in the system.
Extensions:	3a. If the username does not exist: 1. System notifies User that the username submitted does not exist. 2. Return to the step 1 in Main Success Scenario. 3b. If the username and the password does not match: 1. System notifies User to make sure the password is correct. 2. Return to the step 1 in Main Success Scenario.
Special Requirements:	None
Technology and Data Variations List:	2a. The username and the password entered by keyboard.
Frequency of Occurrence:	Could be nearly continuous.

Miscellaneous:	Open Issues: 1. How many times User can submit incorrect password before blocked? 5 times in a row 2. How long should User wait to login again after blocked? 1 day

ID:	4
Use Case Name:	Find User
Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	Actor: User
Stakeholders and Interests:	User: Wants to know if the username is valid and registered in the system. System Administrator: Wants to ensure that the username is in the database.
Preconditions:	1. User is logged on to the system.
Success Guarantee:	1. User is notified by the system that the submitted username is valid and registered in the system.
Main Success Scenario:	1. System represents a form asking the username to the User 2. User enters the username and submits it. 3. The system finds the other user's information, and notifies User that the username is valid or not.
Extensions:	all. At any time, if User closes Find User pop-up window: 1. User is returned to the main screen. 3a. If the username is valid: 1. User can send an invitation, or view user profile. 3b. If the username is not valid: 1. User can re-submit the username or cancel Find User
Special Requirements:	1. Answering User's request should be done in 2 seconds.
Technology and Data Variations List:	2a. The username entered by keyboard.
Frequency of Occurrence:	Could be nearly continuous.

Miscellaneous:	None

ID:	5
Use Case Name:	View User Profile
Overview:	An user accesses the other user's profile.
Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	Actor: User
Stakeholders and Interests:	User: Wants to know other user's profile.
Preconditions:	1. User is logged on to the system. 2. The other user that User wants to check is registered in the system.
Success Guarantee:	1. System represents the other user's profile to User who made a request.
Main Success Scenario:	1. include(Find User) 2. User clicks 'View User Profile' button. 3. System shows the other user's profile stored in the database to User who made a request. The user's profile is consists of nickname, and a history of played games. The record of a game includes the opponent, start date and time, end date and time, and the end result of the game (i.e., win, loss, tie, draw, abandoned, etc.).
Extensions:	all. At any time, if User closes user profile pop-up window: 1. User is returned to the main screen.
Special Requirements:	1. Answering User's request should be done in 2 seconds.
Technology and Data Variations List:	None
Frequency of Occurrence:	Could be nearly continuous.
Miscellaneous:	Open Issues:

	1. How many game records will be shown in the user profile? 2. How long should the system hold the game record after the game has finished?

ID:	6
Use Case Name:	Create Game
Overview:	Allows a registered user to create a new game. Play will not begin until the minimum number of players are added to the game.
Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	Primary Actor: User Secondary Actor: None
Stakeholders and Interests:	User: Wants to create a game so the other user can join, and play the game.
Preconditions:	1. User is logged on to the system.
Success Guarantee:	1. A game session has created. 2. System places a bot as an opponent or tries to find other user who is currently online.
Main Success Scenario:	1. User clicks ‘Create Game’ button. 2. System asks User to choose the type of opponent(AI or human). 3. If the opponent is AI, the User selects the type of opponent. 4. If the opponent is human System seeks for an opponent, and assigns it to User. System asks to the players that they are ready.
Extensions:	all. At any time, if User closes Create Game pop-up window: 1. System cancels current matchmaking process. 2. User is returned to the main screen. 1a. If both User and the other user are ready: 1. System creates a new game tab. 2. extend(Play Game) 1b. If one of the players is not ready: 1. System cancels current matchmaking process. 2. User is returned to the main screen.

Special Requirements:	1. Most of the times, matchmaking should be done in a timely manner(less than 60 seconds).
Technology and Data Variations List:	1. Opponent type is selected by mouse. 2. Opponent type is either AI or human.
Frequency of Occurrence:	Could be nearly continuous.
Miscellaneous:	Open Issues: 1. Should we consider User's rating when matchmaking?

ID:	7
Use Case Name:	Send Invitation
Overview:	Sends invitation to another user.
Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	Primary Actor: User Secondary Actor: None
Stakeholders and Interests:	User: Wants to send an invitation to the other user so both can play in the same game.
Preconditions:	1. User is logged on to the system 2. The other user is registered in the system. 3. The user create the game and game still need a player
Success Guarantee:	1. An invitation has successfully sent to the other user.
Main Success Scenario:	1. include(Find User) 2. User clicks 'Send Invitation' button. 3. System checks the other user's current status. 4. System sends an invitation to the other user.
Extensions:	all. At any time, if User closes Send Invitation pop-up window: 1. User is returned to the main screen. 2a. The other user is not online: 1. System notifies User that the other user is offline. 2a. The other user is already in game: 1. System notifies User that the other user is in game.

Special Requirements:	None
Technology and Data Variations List:	None
Frequency of Occurrence:	Could be nearly continuous.
Miscellaneous:	None

ID:	8
Use Case Name:	Respond to Invitation
Overview:	Receives invitation for user to accept or decline.
Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	Primary Actor: User Secondary Actor: None
Stakeholders and Interests:	User: Wants to respond to the invitation
Preconditions:	1. User is logged on to the system. 2. User is not in game. 3. User has received an invitation from another user.
Success Guarantee:	1. User accepts invitation.
Main Success Scenario:	1. System notifies that User has received invitation, and asks to select either ‘Accept’ or ‘Decline’. 2. User chooses one of the options.
Extensions:	User rejects Invitation: 1. System notifies the sender that User has declined the invitation. 2. User is returned to the main screen.
Special Requirements:	None
Technology and	None

Data Variations List:	
Frequency of Occurrence:	Could be once for each invitation.
Miscellaneous:	None

ID:	9
Use Case Name:	Play Game
Overview:	Main gameplay loop, contains sequences related to playing an active game.
Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	Primary: User(player) Secondary: Database System
Stakeholders and Interests:	User: Wants to interact with the system to play jungle game. Opponent: Same as User. System Administrator: Wants to ensure that the game state must be valid according to the game rules throughout a game.
Preconditions:	1. User is registered in the system. 2. User is logged on to the system. 3. User has created or joined an existing game. 4. An opponent has joined the game. 5. The initial game state has been setup by the system.
Success Guarantee:	1. A game has ended. 2. User won/lost/left. 3. System logs game record to the database.
Main Success Scenario:	1. System represents the jungle game board to both User and Opponent. 2. The system chooses the player to make the first move, based on the rules of Jungle, and that player is notified. 3. User/Opponent makes the move. 4. System updates the game state. 5. Repeat step 3 and step 4 until the win condition is met by one of the players. 6. When the win condition is met, the game is ended. User/Opponent receive a notification of if they won or lost.

	7. System records the game to the database. 8. User/Opponent are returned to the main screen.
Extensions:	all. At any time, if User decides to quit game: 1. The game is ended, and the game tab is deleted.. 2. System updates User's game history with an abandon. 3. User will be replaced by AI, and Opponent continues the game. all. At any time, if User selects different game tab: 1. The game is suspended, and Opponent waits for User to make move. 2. User is redirected to the another game session with different Opponent. all. At any time, if User's system fails: 1. The game is ended, and Opponent is returned to the main screen. 2. System updates User's game history with an abandon. 3. System updates Opponent's game history with a win. all. At any time, if System fails: 1. The game is ended, and User/Opponent are notified that System failed. 2. System does not update the record of the game. 3. User/Opponent are returned to the main screen. 3a. If it's User's turn: 1. include(Make Move). 3b. If it's Opponent's turn: 1. User waits for Opponent to include(Make Move). 4a. If User made a move: 1. System prints User's move to both User and Opponent. 2. System notifies that it's now Opponent's turn. 4b. If User made a move: 1. System prints Opponent's move to both User and Opponent. 2. System notifies that it's now Opponent's turn.
Special Requirements:	None
Technology and Data Variations List:	None
Frequency of Occurrence:	Could be nearly continuous.
Miscellaneous:	Open Issues: 1. How long System should keep the state of suspended game?

ID:	10
Use Case Name:	Make Move
Overview:	During an active game, a player makes a valid move in the game, according to the game rules.
Scope:	Online Jungle Game Software
Level:	User Goal
Actors:	Primary Actor: User Secondary Actor: None
Stakeholders and Interests:	User: Wants to make a move in the jungle game. Opponent: Same as User. System Administrator: Wants to ensure that the game state must be valid according to the game rules throughout a game.
Preconditions:	1. The User is currently in an active game. 2. It is the User's turn. 3. The game is in a valid state.
Success Guarantee:	1. The game is in a valid state according to the Jungle game rule. 2. System updates the current state of the game. 3. It is now Opponent's turn.
Main Success Scenario:	1. User chooses a piece to move. 2. User chooses a game tile to move the piece to. 3. The game state is updated for both User and Opponent, and Opponent is notified that it is now the opponent's turn.
Extensions:	2a. If the move is invalid according to the jungle game rule: 1. User is notified and asked to make a valid move. 2. System does not update the game state
Special Requirements:	None
Technology and Data Variations List:	1. A piece is selected by mouse. 2. A tile is selected by mouse.
Frequency of Occurrence:	Could be nearly continuous.
Miscellaneous:	None.