# The Other Alex

Devin Kadillak, Alex Malott, Alex Shellum, Aaron Thompson, Lucas Wilson

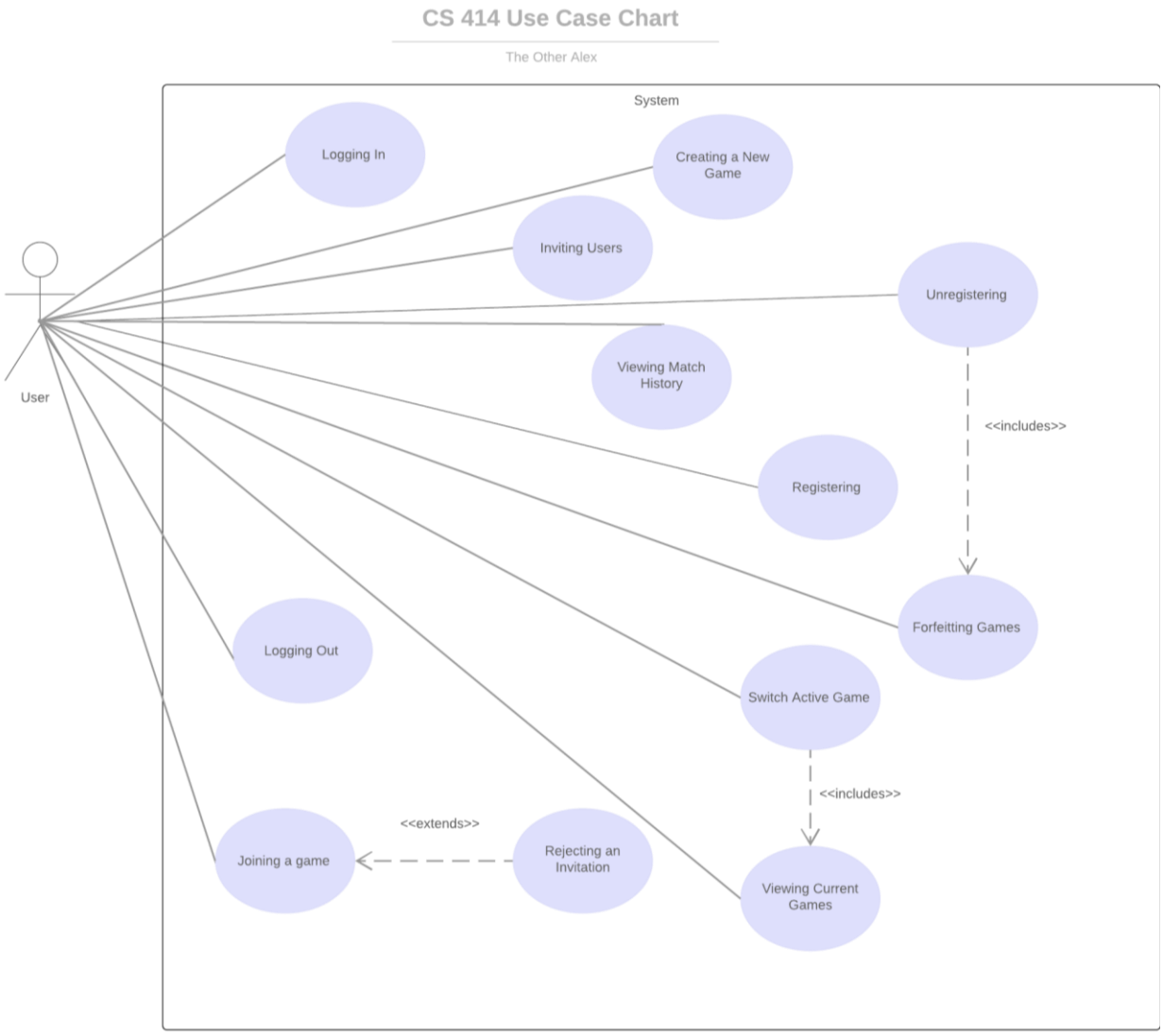
# Requirements

1. User should be able to register an account using email, nickname, and password.
2. User should be able to create a new match but must wait for another player (bot or invited player) to join before playing.
3. The user should be able to invite one or more people to join their game.
4. The first user to accept the invitation to a game should be chosen to join the game.
5. User should be able to reject invitation, and the sender should be notified of either acceptance or rejection.
6. User should be able to play multiple games at once.
7. Users should also be able to unregister their account.
8. Requirement: System should store user match history information (see below) Could wrap this and 10 into player information.
9. Requirement: system should store information about games like: players, start and end dates and times, and whether the game ended in a loss, win, or withdraw.
10. System should keep info about abandoned games.
11. Registered users should be able to view the match history of other players.
12. While playing, the rules of the game must be followed.
13. The invite invitation remains open only until another player joins.
14. The first player to move is the more experienced; if experience is equal, first move goes to the creator of the match.
15. The system knows whose turn it is.
16. The players can only make a legal move during their turn.
17. The state of matches should be saved after user logs out, so that the user can come back to a game and play whenever they want. The other player should be able to make moves while the other player is logged out.
18. Users should be informed of if they won or lost, according to the rules of the game.

## Extra Requirements

19. Users should be able to play against and AI “bot” that can determine what moves to make.
20. Users should be able to make a tournament that other users can join.

# Use Case Diagram



# Use Cases

| | |
|---|---|
| Use case id: | EU-1 |
| Use case name: | Registering |
| Overview: | The user sets up an account. |
| Type: | Primary |
| Actors: | User {Initiator} {primary} |
| Pre-conditions: | The user has a valid email address. |
| Main Flow: | 1) The system prompts the user to enter email<br>2) The user enters email<br>3) The system validates email<br>4) The system prompts user to enter username and password<br>5) The system validates the username/password.<br>6) The system stores the user's information.<br>6) The system notifies the user that the account has been successfully created. |
| Alternative flows: | 3a) Invalid email.<br>3a1) The system finds that the email is invalid.<br>3a2) The system notifies the user about the invalid email.<br>3a3) Return to 1.<br>5a) Invalid password.<br>5a1) The system finds that the username/password is invalid.<br>5a2) The system notifies the user about the invalid username / password.<br>5a3) Return to 4. |
| Post-conditions: | The system contains the provided user data and the user can log in. |

| | |
|---|---|
| Use case id: | EU-3 |
| Use case name: | Inviting Users |
| Overview: | The user invites one or more additional users to play a game. |
| Type: | Primary |
| Actors: | User {Initiator} {Primary} |
| Pre-conditions: | EU-12 |
| Main Flow: | 1) The user enters another user's username into the system.<br>2) The system validates the user's username.<br>3) The system sends invitation link to the invited user.<br>4) Return to 2 if the user wishes to invite more users. |
| Alternative flows: | 3a1) System finds user isn't valid.<br>3a2) Return to 1. |
| Post-conditions: | The provided users have outstanding invites links. |

| | |
|---|---|
| Use case id: | EU-4 |
| Use case name: | Creating a Game |
| Overview: | The game is created when an invited user accepts an invitation. |
| Type: | Primary |
| Actors: | User {Initiator} {Primary} |
| Pre-conditions: | EU-3 |
| Main Flow: | 1) The invited user receives invitation.<br>2) The invited user accepts invitation.<br>3) The system checks validity of invitation and finds it valid.<br>4) The system creates the game. |
| Alternative flows: | 3a) Invalid invitation. |

| | |
|---|---|
| | 3a1) The system checks validity of invitation and finds it invalid.<br>3a2) The system notifies the user about the invalidity of the invitation and removes it from the invited player dashboard.<br>3a3) Use case ends. |
| Post-conditions: | The invite sender and invited user join the game. |

| | |
|---|---|
| Use case id: | EU-5 |
| Use case name: | Rejecting Invitation |
| Overview: | The invited user rejects an invitation. |
| Type: | Primary |
| Actors: | User {Primary} {Initiator} |
| Pre-conditions: | EU-3, EU-12, The logged in user has a pending invitation |
| Main Flow: | 1) The system sends an invitation to the invited user.<br>2) The invited user rejects the invitation.<br>3) The user is notified of invited user's rejection. |
| Post-conditions: | The invited user does not join user's game. The user's invitations to other invited users are still active. |

| | |
|---|---|
| Use case id: | EU-6 |
| Use case name: | Switch Active Game |
| Overview: | The user can switch between instances of games they are participating in |
| Type: | Primary |
| Actors: | User {Initiator} {Primary} |
| Pre-conditions: | EU-4, EU-2 |
| Main Flow: | 1) The user selects one of their game instances to play.<br>2) The system loads the game's current state.<br>3) The user takes their turn.<br>4) The user leaves the game instance.<br>5) The system stores the game state.<br>6) return to step 1. |
| Post-conditions: | The user's move has registered in the loaded game instance(s). |
| Cross references: | EU-4 |

| | |
|---|---|
| Use case id: | EU-7 |
| Use case name: | Forfeiting Game |
| Overview: | A user leaves a game by forfeiting. |
| Type: | Primary |
| Actors: | User {Initiator} {Primary} |
| Pre-conditions: | EU-2 |
| Main Flow: | 1) The initiating user opts to leave an in-progress game by forfeiting.<br>2) The system records the game as a loss for the initiating user and a win for the other user.<br>3) The system notifies both users the game has been successfully forfeited.<br>4) The system deletes the game data. |
| Post-conditions: | The system has recorded a loss for the initiating user and a win for the other user. The system has deleted the game data. |

| | |
|---|---|
| Use case id: | EU-8 |
| Use case name: | Unregistering |
| Overview: | The user unregisters their account. |

| | |
|---|---|
| Type: | Primary |
| Actors: | User {Initiator} {Primary} |
| Pre-conditions: | EU-12 |
| Main Flow: | 1) The user chooses to unregister their account.<br>2) The system logs out the user.<br>3) The system forfeits the user's active games.<br>4) The system deletes the user's account. |
| Post-conditions: | The user is logged out. The system no longer contains the user's data. |
| Cross references: | EU-10 EU-7 |

| | |
|---|---|
| Use case id: | EU-9 |
| Use case name: | Viewing match history |
| Overview: | The user can view the match history of a selected user. |
| Type: | Primary |
| Actors: | User {Initiator}{Primary} |
| Pre-conditions: | EU-12 |
| Main Flow: | 1) The user provides the name of a user.<br>2) A list of previously played games and aggregate statistics (W/L ratio, ELO) connected to that name is provided. |
| Alternative flows: | 1a) The provided user name is not found – an error message is presented. |
| Post-conditions: | The user has a list of their match history |
| Cross references: | EU-10 |

| | |
|---|---|
| Use case id: | EU-10 |
| Use case name: | Viewing Active Games |
| Overview: | The user views all active games in which they participate. |
| Type: | Primary |
| Actors: | User {Initiator}{Primary} |
| Properties: | Performance:<br>Security:<br>Other: |
| Pre-conditions: | EU-12 |
| Main Flow: | 1) The user requests their active game information.<br>2) The system sends the user their active list of games. |
| Alternative flows: | 2a) The system finds that the user has no active games.<br>2a1) The system sends the user an empty list. |
| Post-conditions: | The user has a list, which can be empty, of their active games. |
| Cross references: | EU-9 |

| | |
|---|---|
| Use case id: | EU-11 |
| Use case name: | Moving |
| Overview: | The user makes a legal move during their turn. |
| Type: | Primary |
| Actors: | User {Initiator} {Primary} |
| Pre-conditions: | EU-6, The user has selected the active game |
| Main Flow: | 1) User moves a piece on their turn.<br>2) System checks if user has made a legal move and finds it legal.<br>3) System checks if the user won the game and finds that the user has not.<br>4) System notifies opponent that it is their turn. |
| Alternative flows: | 2a) Illegal move |

| | |
|---|---|
| | 2a1) System checks if user has made a legal move and finds it illegal.<br>2a2) System notifies the user that the move the user tried to make is illegal.<br>2a3) Use case ends<br>3a) Winning move<br>3a1) System checks if the user won the game and finds the user has.<br>3a2) System notifies the user that the user has won.<br>3a3) Use case ends. |
| Post-conditions: | The user's piece moved from its old position to a new one. |
| Cross references: | EU-9 |

| | |
|---|---|
| Use case id: | EU-12 |
| Use case name: | Logging in |
| Overview: | The user logs in to the system |
| Type: | Primary |
| Actors: | User {Initiator} {Primary} |
| Pre-conditions: | EU-1 |
| Main Flow: | 1) User makes a legal move on their turn<br>2) System checks if user has made a legal move<br>3) System checks if the move wins the game |
| Subflows: | 1a) For first move, user with most games played goes first.<br>2a) Illegal move<br>2a1) System checks if the move the user makes is legal and finds it illegal<br>2a2) System notifies the user that the move the user tried to make is illegal<br>2a3) Use case ends |
| Post-conditions: | The user's piece moved from its old position to a new one. |
| Cross references: | EU-9 |

| | |
|---|---|
| Use case id: | EU-13 |
| Use case name: | Logging out |
| Overview: | The user logs out of the system |
| Type: | Primary |
| Actors: | User {Initiator} {Primary} |
| Pre-conditions: | EU-12 |
| Main Flow: | 1) The user initiates the log-out sequence<br>2) The system saves the state on any active games<br>3) The system logs the user out |

| | |
|---|---|
| Post-conditions: | The user logged out |

Notes:

-EU-2 became obsolete, so we merged EU-4 ‘Joining A Game’ with EU-2 ‘Creating A Game’ into the new EU-4 called ‘Creating A Game’ to better represent how games are created.

-EU-7, ‘Forfeiting A Game’ was never implemented. We have an enum recording the state change in our backend, but we have no functionality to do anything with that state of the game.