

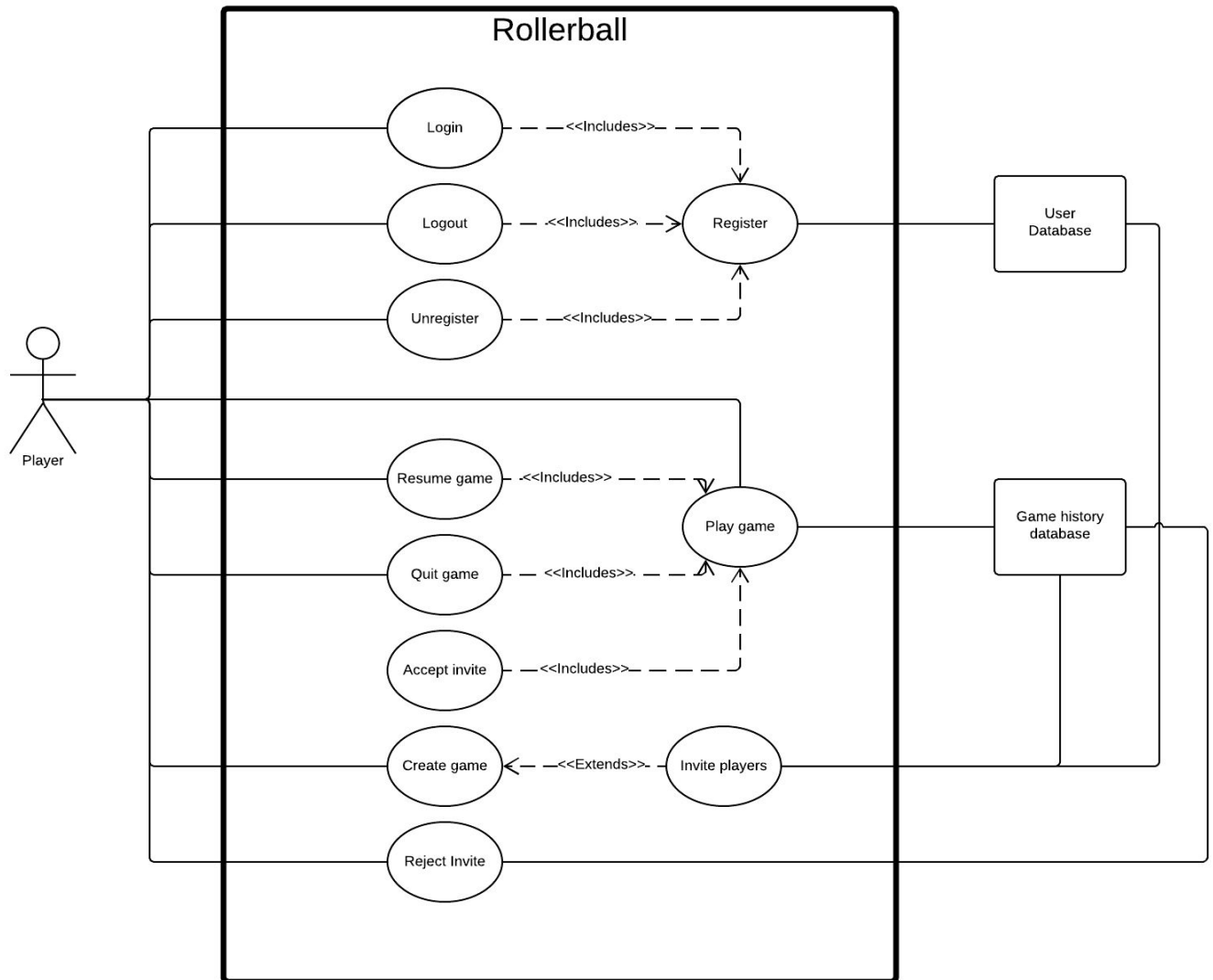
# Rollerball Use Case Document

The Soggy Zebras: Angela Root, Jonathan Cowles, Jordan Peterson, Courtney Torres, Dawson  
Canby

## **Table of Contents:**

Use case diagram	3
Logging in	4
Logging out	5
Unregistering	6
Registering	7
Resuming a game	8
Quitting a game	9
Accepting a game invite	10
Playing the game	11
Creating a new game	12
Inviting players to a new game	13
Rejecting a game invite	14

## Use case diagram



## Logging in

Use case id:	1
Use case name:	Login
Brief description:	The player logs into the system.
Type:	User Goal
Primary actors:	Player
Secondary actors:	User database
Pre-conditions:	1. Player must be registered
Main flow:	<ol style="list-style-type: none"><li>1. Player enters username and password.</li><li>2. System verifies credentials<ol style="list-style-type: none"><li>2a. System checks if player is in the user database<ol style="list-style-type: none"><li>2a1. Extension point Register</li></ol></li><li>2b. Player enters incorrect username or password<ol style="list-style-type: none"><li>2b1. System notifies user of invalid credentials</li><li>2b2. Return to 1</li></ol></li></ol></li></ol>
Alternate flow:	None
Post-conditions:	<ol style="list-style-type: none"><li>1. The player is logged in</li></ol>

## Logging out

Use case id:	2
Use case name:	Logout
Brief description:	The player logs out of the system
Type:	User Goal
Primary actors:	Player
Secondary actors:	User database
Pre-conditions:	1. The player is registered in the system 2. The player must be logged in
Main flow:	1. The player chooses to logout.
Alternate flow:	none
Post-conditions:	1. The player has been logged out of the system

## Unregistering

Use case id:	3
Use case name:	Unregister
Brief description:	The player removes registration from user database
Type:	User Goal
Primary actors:	Player
Secondary actors:	User database
Pre-conditions:	1. Player is currently registered in the user database 2. Player is currently logged in to the system
Main flow:	1. Player chooses to unregister account 2. User database is updated removes player entry
Alternate flow:	1.a. Player declines to unregister during the process b. User database is not updated to remove player entry c. Unregister process is cancelled
Post-conditions:	1. Player is unregistered and account details are no longer in the user database.

## Registering

Use case id:	4
Use case name:	Register
Brief description:	The player creates an account with a unique username, email, and password.
Type:	User Goal
Primary actors:	Player
Secondary actors:	User database
Pre-conditions:	1. Username must be unique. 2. Email must be unique. 3. Password must fulfill requirements TBD at a later time.
Main flow:	1. Player provides email, username, and password. 2. System checks for unique username and email. 3. System updates user database. 4. Player is registered
Alternate flow:	2.a. Username/Email is not unique. b. Restart registration process
Post-conditions:	1. Player is registered 2. Player's login information is stored in the user database.

## Resuming a game

Use case id:	5
Use case name:	Resume game
Brief description:	The player may resume a game that has already been started.
Type:	User Goal
Primary actors:	Player
Secondary actors:	Game History Database
Pre-conditions:	1. A game must already be started 2. A player must be logged in
Main flow:	1. The player resumes the game 2. The game history database updates the state of the game
Alternate flow:	1. The game does not exist
Post-conditions:	1. The game will be resumed



## Quitting a game

Use case id:	6
Use case name:	Quit game
Brief description:	Quit the game that the player is currently in.
Type:	User Goal
Primary actors:	Player
Secondary actors:	Game history database
Pre-conditions:	1. The player is in a game
Main flow:	1. The player quits the game 2. The game history database is updated to show that the player left 3. The other player is notified that the player quit
Alternate flow:	1. The player does not to quit the game
Post-conditions:	1. The player can no longer play the game 2. The game history database shows that the player quit 3. The other player has been notified that the player quit

## Accepting a game invite

Use case id:	7
Use case name:	Accept Invite
Brief description:	A player will be able to accept an invite sent from another player
Type:	User Goal
Primary actors:	Player
Secondary actors:	None
Pre-conditions:	<ol style="list-style-type: none"><li>1. The player must be logged in</li><li>2. The invite must not be accepted by anyone else</li></ol>
Main flow:	<ol style="list-style-type: none"><li>1. System validates the invite</li><li>2. The player accepts invite</li><li>3. System cancels all other game invites</li></ol>
Alternate flow:	<ol style="list-style-type: none"><li>1. Invite is not valid<ol style="list-style-type: none"><li>a. The game is not accepted</li><li>b. The game is not started</li></ol></li><li>2. The player does not accept the invite<ol style="list-style-type: none"><li>a. Reject Invite.</li></ol></li></ol>
Post-conditions:	<ol style="list-style-type: none"><li>1. The game was accepted</li><li>2. The game is started</li></ol>

## Playing the game

Use case id:	8
Use case name:	Play game
Brief description:	Two players take turns taking actions. The game ends when a win condition is met for one of the players
Type:	User Goal
Primary actors:	Player
Secondary actors:	User database, Game history database
Pre-conditions:	<ol style="list-style-type: none"> <li>1. Player is logged in</li> <li>2. Both players have joined a game against each other</li> </ol>
Main flow:	<ol style="list-style-type: none"> <li>1. System checks if it is the player's turn <ol style="list-style-type: none"> <li>a. The system notifies the player that it can move</li> </ol> </li> <li>2. Player takes a move</li> <li>3. System checks impact of the move <ol style="list-style-type: none"> <li>a. Win condition is not met, next players turn, repeat step 1 and 2</li> <li>b. Win condition is met</li> </ol> </li> <li>4. Game ends</li> </ol>
Alternate flow:	<ol style="list-style-type: none"> <li>1. Player does not make a move <ol style="list-style-type: none"> <li>a. The system waits for the player to move</li> </ol> </li> <li>2. During player's move, player leaves the game <ol style="list-style-type: none"> <li>a. The game is over</li> </ol> </li> <li>3. During player's move, other player leaves the game <ol style="list-style-type: none"> <li>a. The game is over</li> </ol> </li> <li>4. Move is invalid <ol style="list-style-type: none"> <li>a. Player is notified of invalid move</li> </ol> </li> </ol>
Post-conditions:	The game is over and the game history database records the results

### Creating a new game

Use case id:	9
Use case name:	Create game
Brief description:	Create a new rollerball game that the player can invite others to.
Type:	User Goal
Primary actors:	Player
Secondary actors:	None
Pre-conditions:	1. Player must be registered.
Main flow:	1. Player chooses to create game. 2. Extension point Invite players
Alternate flow:	1. Player cancels creation of game. a. Created game is erased.
Post-conditions:	1. A new game is created that invited players can join.

### Inviting players to a new game

Use case id:	10
Extension use case:	Invite players
Brief description:	When a match is created, player may invite other players to join.
Type:	User Goal
Primary actors:	Player
Secondary actors:	User database
Pre-conditions:	1. Player has created a new game. 2. Player is logged in. 3. Invited player(s) exists.
Main flow:	1. Player selects one or more players to invite to play. 2. Invite is sent to other players immediately.
Alternate flow:	1. Player cancels create game. a. Created game is erased.
Post-conditions:	1. A new game has been created with invites sent to other players.

### Rejecting a game invite

Use case id:	11
Use case name:	Reject Invite
Brief description:	Player rejects the invite from another user to play the game
Type:	User Goal
Primary actors:	Player
Secondary actors:	None
Pre-conditions:	1. Player must have been invited.
Main flow:	1. Player gets new game invite notification. 2. Player rejects the invite. 3. Player that sent the invite is notified of rejection.
Alternate flow:	2.a. Player ignores the game invite. 2.b. Player accepts game invite. 1. Accept Invite.
Post-conditions:	1. Invitation has been rejected