

Cookie Byte Use Cases:

Use Case id:	EU-001
Use Case Name:	Register to System
Overview:	A new, unregistered user will enter his/her email, userID,, and password to register to the system.
Primary Actors:	User [Initiator][Primary][Unregistered]
Secondary Actors:	None
Properties:	<p>Performance: Registration must happen immediately and allow user to start a new game right away.</p> <p>Security: Email address entered by user and user-set password must both be secure and inaccessible to the public.</p>
Preconditions:	User must already have an email address that has not yet been used. User must have a nickname that has not yet been used.
Main Flow:	A user will enter their email address, a nickname, and a password and click a button to sign up for an account. If the email and nickname are not already in use, the account will be created and user will be notified.
Postconditions:	A user will have an account and will be able to play games and will become a registered user.
Alternative Flows:	If a user's selected email or nickname have been previously used, the user will be notified and asked to use a different email or nickname depending on if one or both have been previously taken. The account will not be created.

Use Case id:	EU-002
Use Case Name:	Logging into System
Overview:	A registered user can enter the system by using their username and password.
Primary Actors:	User [Initiator][Primary][Registered]
Secondary Actors:	None
Properties:	Performance: Once user enters correct username and password they

	<p>are connected to the corresponding account.</p> <p>Security: All userID and passwords must be both be secure and inaccessible to the public.</p>
Preconditions:	User must have already be registered to the game system with unique userID and password.
Main Flow:	A user will enter their userID and a password and click the sign in button. Database will authenticate and confirm User. Message will be sent to user once process is finished.
Postconditions:	The user then will be able to access their account and create/play games.
Alternative Flows:	If a user's selected an incorrect nickname or password, the user will be notified and asked to retype their username and password. No account will be accessed.

Use Case id:	EU-003
Use Case Name:	Create Game
Overview:	A user creates a new game and becomes a player of the game.
Primary Actors:	User [Initiator][Registered]
Secondary Actors:	User [Registered]
Properties:	<p>Performance: This process will be done via invitations. (See use case EU-004)</p> <p>Security: Games can only be created if the Secondary user responds to the Primary users invitation. No other users should be able to access this game.</p>
Preconditions:	User is registered and logged into their account
Main Flow:	A user initiates the "create game" function. The game is created and the user that initiated the create game is added as a player of this game.
Postconditions:	The game is created with the user that created the game as a player.
Alternative Flows:	The game is not created and the user is notified that the game cannot

	be created.
--	-------------

Use Case id:	EU-004
Use Case Name:	Invite to Game
Overview:	A registered user can invite another registered user (or set of users) to join a created game or set of created games.
Primary Actors:	User [Receiver][Primary][Registered]
Secondary Actors:	User [Initiator][Secondary][Registered]
Properties:	<p>Performance: The receiver should get an invitation. And once accepted the initiator should get a notification. And both players should appear in the game.</p> <p>Security: Only the Initiator can invite other users to the instance of the game they created. Only 2 players can enter the game instance and play at the same time.</p>
Preconditions:	<p>-Both users have to be registered.</p> <p>-The Initiator can invite multiple users at the same time.</p> <p>-A game instance should be created for each invitation that is sent.</p>
Main Flow:	The secondary player sends an invite and the primary player receives a notification of that game.
Postconditions:	The primary player is a part of the game instance that the secondary player created.
Alternative Flows:	The primary Player never responds to the invitation and the invite is just stored as pending under that user's account.

Use Case id:	EU-005
Use Case Name:	Respond to Invitation
Overview:	A registered user can accept or reject an invitation to join a game. If the user accepts the invitation, she/he becomes a player of the game.
Primary Actors:	User [Primary] [Receiver]

Secondary Actors:	User [Initiator]
Properties:	<p>Performance: User is invited without large delay.</p> <p>Security: Only the intended user is invited to the game. Other users do not see the invitation.</p>
Preconditions:	Both users are registered with the system.
Main Flow:	User accepts the invitation to join game. The system associates that user with the specific game. The user is now playing the game from which the invitation specified.
Postconditions:	User is placed into game with initiating user.
Alternative Flows:	User rejects the game invitation. The system does not associate user with the game the invitation specified.

Use Case id:	EU-006
Use Case Name:	Quit game
Overview:	A player can quit a game at any time.
Primary Actors:	User [Initiator][Primary][Registered]
Secondary Actors:	
Properties:	<p>Performance: No impact.</p> <p>Security: Only the user can quit games he/she is a part of. Other users should not be able to quit a game for another user.</p>
Preconditions:	User must be a part of game.
Main Flow:	User quits a game, loss is saved to user profile.
Postconditions:	User exits the game. The other player is given a win and the initiator is given a loss. All game history is then updated.
Alternative Flows:	User chooses not to quit game and remains a player.

Use Case id:	EU-007
Use Case Name:	Unregister
Overview:	A registered user can unregister from the system.
Primary Actors:	User [Initiator] [Primary][Registered]
Secondary Actors:	
Properties:	Performance: No impact. Security: Only the primary actor can unregister him/herself from the system.
Preconditions:	User must be registered.
Main Flow:	User tries to unregister from the system and is asked to confirm they want to remove their account.
Postconditions:	The user no longer has an account. All invites sent and data attach to the user are also deleted.
Alternative Flows:	User does not unregister and the account remains active.

Use Case id:	EU-008
Use Case Name:	Record History
Overview:	The system must record a history of games played by a user. The record must include the opponent and result of a game. Also a user's overall playing record must be stored.
Primary Actors:	System
Secondary Actors:	
Properties:	Performance: No impact.
Preconditions:	User is registered, Game has finished.
Main Flow:	Each time a game is played, the system records the opponent's name,

	and the result of the game under both users.
Postconditions:	The results of the game are saved to the system.
Alternative Flows:	User has not played any games and therefore has a blank history or game is unfinished/pending and nothing needs to be recorded.

Use Case id:	EU-09
Use Case Name:	View Game History
Overview:	A registered user has a profile which consists of his/her userID and history of played games that can be viewed by the owner of the account.
Primary Actors:	User [Initiator] [Primary] [Registered]
Secondary Actors:	
Properties:	<p>Performance: A registered user that is not in a game making a move should be able to look at their own account to see their record (wins, losses). Once a game instance is finished both players accounts must be updated accordingly.</p> <p>Security: Only the owner of the account can view game history.</p>
Preconditions:	The User must be registered in the game system. A player should have at least played and finished one game in the past to have a history. That game data is stored under the player's account.
Main Flow:	User opens a window under their account and all of that player's history including wins and loses are shown.
Postconditions:	The player sees their current and most recent game history and can exit the display to start a game instance.
Alternative Flows:	If the player searched on does not have history yet and hasn't played on game yet a default profile must be set.

Use Case id:	EU-010
Use Case Name:	Play game

Overview:	A player begins a new game
Primary Actors:	User [registered][Invited]
Secondary Actors:	
Properties:	Performance: None. Security: Players can not make moves in games they are not apart of.
Preconditions:	The player is part of the game.
Main Flow:	A player makes a move in a game they are in.
Postconditions:	The games a player moves in is only a part of their game list.
Alternative Flows:	N/A

Use Case id:	EU-011
Use Case Name:	Make move
Overview:	User can make a move within the game they are a part of.
Primary Actors:	User [Initiator] [Primary] [Registered]
Secondary Actors:	
Properties:	N/A
Preconditions:	A player has a game and it is his/hers turn.
Main Flow:	A player tries to make a move. The only moves he can make are game-allowed. The player makes an allowed move.
Postconditions:	The move made is saved and it adheres to the game rules.
Alternative Flows:	Turn can't be done because game is over and winning conditions are met or the move is not valid.