CS414: Object-Oriented Design - Fall 2018

Project 1:
Banqi Game Use Cases
Updated 10/7/2018

Team:
ByteMe

Members:
Joshua Henry
Alex Hinterlong
Brian Martin
Stephen Porsche
Evan Salzman

Identified Use Cases:
- Create User Profile
- New Game
- Play Game/Move
- Forfeit game
- View Profile
- View Match History
- Respond to invitation
- Login
- Logout
- Unregister User Profile

Fully-dressed Use Cases:

Use case id:	1
Use case name:	Create User Profile
Overview:	User creates a user profile with an email address, password and the nickname they will use during play in order be a registered User.
Primary Actors:	User
Secondary Actors:	None
Pre-conditions:	1. User has a valid email, not associated with a registered account 2. The User's nickname is not already in use by another registered User 3. The User does not already have a registered profile
Main flow:	1. The User enters the email, password and nickname they will use for their created account 2. The system checks to see that the submitted email address and nickname are not in use by any other registered Users. 3. The system determines that the credentials are not in use by another registered User. 4. The system creates the User's profile and thus, the User is registered
Post-conditions:	The User has a registered User Profile
Alternative flows:	The User enters an invalid email when creating a profile: 1. The system reports back to the User that the provided email is not a syntactically valid email (e.g. no @ symbol in the input). 2. The system does not create a profile for the User.

	The User enters an email, password and nickname for their new account, but the email and/or nickname is/are already taken.
	1. The system reports back that account was not created, because either or both of the email and nickname are associated with an existent User.
	2. The system does not create a profile for the User.

Use case id:	2
Use case name:	New Game
Overview:	User creates a new game and is required to invite a second user
Primary Actors:	User - Host
Secondary Actors:	User - Player 2
Pre-conditions:	Both User - Host, and User - Player 2 have a registered profile
Main flow:	1. User creates a new game
	2. User inputs nickname of a User to invite to play
	3. Invite is sent to (other) User
Post-conditions:	A game is created and an invite is sent out to the player, which the hosting User invited.
Alternative flows:	The user inputs the nickname of another User with whom to play, but the system finds that the provided nickname is not associated with a registered User.
	1. The system reports the nonexistence of the chosen opponent (invited User) back to the hosting User.
	2. The system does not send the invite.

Use case id:	3
Use case name:	Play Game/Move
Overview:	The game is running and users take turns moving, with system checking conditions each move
Primary Actors:	User - Host, User - 2nd Player

Secondary Actors:	None
Pre-conditions:	The game exists, both Users are in the game

Main flow:	1. System notifies User it is their move (on game creation, User that sent invite is first to move)
	2. The User clicks a view game board button.
	3. The system opens the current game board for the User, which the User assesses before making a move.
	4. User attempts to move, System checks if move is valid
	5. Move is made if valid, System updates game status
	6. System checks if game is over
	    a. If yes, game is over, record match history
	    b. If no, other user gets notified it is their turn, repeats

Post-conditions:	The User has moved and it is now the other User's move
Alternative flows:	The system prompted the user about it being their turn, but the other User since forfeited the game. 1. The User tries to view the game board, but the system says the game is forfeited. 2. The system provides the User with an option to see the effect it has on their match history. The User who is currently eligible to take a turn, selects an invalid move. 1. The system notifies the User that the move could not be completed. 2. The User reassess the game board for a valid move.

Use case id:	4
Use case name:	Forfeit Game
Overview:	A user who is in the game can cancel the game
Primary Actors:	User
Secondary Actors:	None
Pre-conditions:	The game exists, is active and is not cancelled
Main flow:	1. User who wants to quit, clicks on the forfeit button 2. The system prompts that User to confirm they actually want to forfeit. 3. That User confirms they want to forfeit the game.

	4. The other User in the game is notified that the User they were playing against forfeited the game. 5. The game is recorded in the match history of both Users: a loss for the User who forfeited and a win for the other User.
Post-conditions:	The game is cancelled
Alternative flows:	The User had selected to forfeit the game, but then changes their mind when the system asks them if they really want to do so. 1. The system cancels the process of forfeiting the game. 2. No changes are made to the match history of either player. 3. The system ensures that the game continues as normal.

Use case id:	5
Use case name:	View Profile
Overview:	User can view profile of their own account, and can view the accounts of any other user
Primary Actors:	User
Secondary Actors:	None
Pre-conditions:	User must have created a profile of their own. Other users must have created their own profiles
Main flow:	1. User clicks on view own profile button to view their personal profile 2. User clicks view profile of a different user to view that users profile
Post-conditions:	User has viewed own profile, or user has viewed another users profile
Alternative flows:	

Use case id:	6
Use case name:	View Match History
Overview:	A user can view the results of the previous Banqi games they have

	played.
Primary Actors:	User
Secondary Actors:	None
Pre-conditions:	A user has a valid profile to play Banqi games and the user has played at least one previous game
Main flow:	1. User clicks on their own profile or to view that of another user. 2. Once on the profile, the User then selects an option to view the match history of that user.
Post-conditions:	User must exist to view their profile and history.
Alternative flows:	A user selects another user's profile but, once there, finds that user does not have a match history.

Use case id:	7
Use case name:	Respond to Invitation
Overview:	A user is invited to play a game can either accept or reject that invitation.
Primary Actors:	User
Secondary Actors:	None
Pre-conditions:	The user of a new game sends an invitation.
Main flow:	1. A user receives an invitation to join a created game 2. That user either accepts or rejects the game invite 3. The system begins the game between the User who sent the invite and the User who accepted the invite.
Post-conditions:	The user joins a game if the invitation is accepted, or the user has rejected the invitation.
Alternative flows:	An invited User (1) accepts the invitation, but a different User (2) already accepted the game invite before User (1): 1. The system tells User (1) that the someone else already accepted an invitation to play. 2. User (1) no longer sees the invitation for or anything about that game.

Use case id:	8
Use case name:	Login
Overview:	A user with a registered account can log into the system.
Primary Actors:	User
Secondary Actors:	None
Pre-conditions:	A registered user is currently logged out of the system.
Main flow:	1. The User enters a username/email and a password associated with their registered profile. 2. The system authenticates the user. 3. The system logs the user in to the game interface.
Post-conditions:	A registered user is logged into the system.
Alternative flows:	The User enters a username/email and a password, but the system finds that there is no profile registered with either or both those credentials. 1. The User is not logged in. 2. The system prompts the User, stating the email and/or password entered were invalid or unregistered.

Use case id:	9
Use case name:	Logout
Overview:	A User who is currently logged into the system can choose to log out of the system, but will be able to log back in at any time.
Primary Actors:	User
Secondary Actors:	None
Pre-conditions:	The User intending log out of the system is a registered User and is currently logged into the system.
Main flow:	1. The User selects an option to log out of the system 2. The system logs the User out
Post-conditions:	A registered User is logged out of the system.

Alternative flows:	

Use case id:	10
Use case name:	Unregister User Profile
Overview:	A User is able to unregister their account on the system
Primary Actors:	User
Secondary Actors:	None
Pre-conditions:	The User choosing to unregister themself from the system has a registered account on the system.
Main flow:	1. The User chooses the option to unregister their account. 2. The system confirms with the User that they want to unregister with a prompt on the screen. 3. The User confirms they want their account unregistered. 4. The account is removed from the system database of registered accounts. 5. The system unregisters the User.
Post-conditions:	The specified account is no longer active and the User associated with that account is unregistered.
Alternative flows:	When the system prompts the User to confirm they want to unregister their account, they select cancel. 1. The system cancels the unregistering process. 2. The User's profile remains registered in the system.