**MASTER USE-CASE DOCUMENT**

**Use Case: Sign-Up**

ID: 1

Brief Description: Player wants to create an account.

Primary Actor: Player

Secondary Actor: None

Type: Essential

Precondition(s):

-Player has an e-mail account.

Postcondition(s):

-Player has created an account for the site.

Mainflow:

1. Player clicks on “Create Account button”.

2. Player enters e-mail address.

4. Player enters their password making sure it is given the appropriate credentials.

5. Player re-enters their password and verifies that it is correct.

7. Once the information is correct the player hits submit and the account is created.

9. Player goes to their e-mail and clicks on a link in their e-mail to activate their account.

System Response

3. System checks whether e-mail address is valid or not.

6. System checks whether the two passwords match.

8. An e-mail is sent to the player for activating their account.

**Use Case: Login**

**ID: 1.1**

Brief Description: Player wants to log in to their account

Primary Actor: Player

Secondary Actor: None

Precondition(s):

-Player has an account

-Player is not logged in

Main Flow:

1. Main flow begins when the player accesses the login page.

2. Player enters their username and password.

3. Player hits "Log in" button.

4. System communicates with database to confirm information provided matches system records.

5. Session is created.

6. Player is taken to their account dashboard.

Post Condition(s):

-Player is logged into their account.

-Player is viewing their dashboard.

Alternate Flow(s):

-Incorrect password.

**Use Case Name: Failed Login**

**ID: 1.2**

Description: Player enters an invalid password (with regards to the username associated with it) on the login page

Primary Actor: Player

Secondary Actor: None

Preconditions:

-Player enters a password that does not match the given username.

-Player is currently not logged in.

-Player has attempted to log in with invalid information three (3) times.

Main Flow:

1.) The use case begins when the Player enters an invalid password wrong three times.

1.1.) An error message is displayed.

1.2.) Player’s account is locked out of the account for 10 hours.

1.3.) An email is sent to their email address, informing the player has been black listed from the poker site for 10 hours. This email also details what computer was used when attempting to access the player’s account.

Post condition:

-Player has been locked out from accessing their account for 10 hours.

-An email was sent to the owner of the account.

**Use Case Name: Reset Password**

**ID: 1.3**

Description: Player needs a way to login if he/she forgets password

Primary Actor: Player

Secondary Actor: SMS Respondent

Preconditions:

-Player is registered with a cell phone number.

-Player is not logged in.

-Player is on the login page.

Main Flow:

1. The Player hits “Forgot Password”
   1. The Player will type in their username
   2. Player will confirm their cell phone number
   3. Player will receive a generated verification code on cell phone
   4. Player enters the verification code
   5. Player will enter a new password

Post Condition: The Player has a new password

Alternate Flaws:

Invalid verification code

Verification code expire

Invalid password length

Invalid cell phone number confirmation

**Use Case Name: Transfer Funds to Online Poker Account**

**ID: 2**

**Purpose:** Let a customer electronically transfer funds into their account for using in online poker games.

**Primary Actor:** User

**Secondary Actor(s):** Bank, Credit Card Company

**Overview:** The customer clicks on an option from a drop-down menu for adding funds into their account. Then they select either taking money from their bank account or credit card. If a bank account is selected, then the customer provides the account number, routing number, account name, and amount that gets added into the system. If a credit card is selected, then the card number, card name, policy number, type of credit card, and the amount that the person wants added to their account. After the customer submits this information, the banking account or credit card is checked for having sufficient funds and if this check is successful than the funds are added into the account. If not the payment doesn’t go through and you return to the account page.

**Type:** Essential

**Preconditions:** The player has logged into their account successfully and owns a credit card or banking account.

**Postconditions:** Funds are added into the player’s accounts which they can use for the game.

**Special Requirements:** None

**Flow of Events**

Actor Action

1. This use case begins when a player selects “Manage Wallet” -> “Add Funds into Account” from a drop-down menu.
2. User selects their type of payment from either transferring funds via a credit card or a bank account.
3. The user proceeds to provide information from their credit card or their banking account. This information includes account name, (account number or routing number), (credit card company / bank branch), (cvv number or check number).
4. The user selects the amount of money they would like to withdraw from their account.
5. Once all the fields are entered, the user clicks the submit button.

7. The specific company verifies in their system if the account is up to date and funds are sufficient and if so the transaction is valid.

**System Response:**

1. The system contacts the respective company that the account from the payment method is registered under.

8. The user’s payment gets added into their account.

**Alternative Flow of Events**

Line 3:

-Fields are left blank. Return to Step #3.

Line 4:

-A user enters an amount of money that is less than $5. Return to Step #4.

Line 7:

-The account doesn’t exist in the system and adding funds to their account is locked (See Use Case 2.0.1)

-The user’s credit card is expired and adding funds to their account is locked. (See Use Case 2.0.1)

-There are insufficient funds in the user’s account and their account is locked. (See Use Case 2.0.1)

**Use Case Name: Lock Account from Making Transactions**

**ID: 2.0.1**

**Purpose:** Locks the user account from making transactions to their account in the instance a previous payment method fails.

**Primary Actor:** Company that the user’s account is from.

**Secondary Actor(s):** None

**Overview:** The company’s that the account is from goes through their database and performs a series of test verifying whether the information a user entered is legitimate. There are three tests performed. The first test is making sure the account exists in the system, by searching through the account number and checking if the other fields match up. If that test succeeds then a second test is performed if the payment is from a credit card checking whether the card is expired or not. Lastly, if both tests succeed then the final test is checking whether that are insufficient funds in the account. If one of these tests fail, the user’s account for the online poker application is locked from making transactions to their credit card or bank account.

**Type:** Essential

**Preconditions:** The player has successfully logged into their account, click the “Add Funds To Account” button and filled out the fields for withdrawing a payment.

**Postconditions:** The player’s account is locked and must manually unlock the account before they can make account transactions.

**Special Requirements:** None

**Flow of Events**

**Actor Action:**

1. The company receives a request from the online poker application for withdrawing a payment from the account.
2. The company checks the account number in the system and verifies whether it’s a valid account.
3. If the company is a credit card company, then it checks whether the card has expired or not.

6. The company finally checks whether the account has enough funds in the account for withdrawing the amount specified.

**System Response:**

1. If the account found is invalid then the account gets locked, if not the process continues.

**5.** If the payment option is from a credit card and it is expired, then the account gets locked. If not the process continues.

7. If there are insufficient funds in the account, then the user’s online poker account gets locked.

**Alternative Flow of Events**

**Line 6:**

-The company tests whether there are sufficient funds in the account and all tests succeeded. (see use case 2, step #8).

**Use Case Name: Deposit Winnings into Bank Account**

**ID: 2.1**

**Purpose:** Lets the user request funds from their PokerTable account to get put into their credit card or bank account.

**Primary Actor:** User

**Secondary Actor(s):** Bank, Credit Card Company

**Overview:** The user clicks on a button for requesting funds getting transferred into their account. First the player selects an amount they want added to their account, the player must make a transaction of at least $1. Once this is selected the player enters the account information for either a credit card or bank account. If a bank account is selected, then the customer provides the account number, routing number, and account name. If a credit card is selected, then the card number, card name, policy number and type of credit card. After the transaction is made, the banking company or credit card company checks whether the account exists and then accepts the request for putting funds into the account.

**Type:** Essential

**Preconditions:** The user has successfully logged into their account, click the “Make A Transaction” button, has at least $1 in their account, and owns a credit card or bank account.

**Postconditions:** The transaction is processed and the user receives the money in their account.

**Special Requirements:** None

**Flow of Events**

**Actor Action:**

1. This use case begins when a player selects “Manage Wallet” -> “Add Funds into Account” from a drop-down menu.
2. The user selects the amount of money they want transferred into their account.
3. The user selects whether they want the funds transferred to their bank account or credit card.
4. The user proceeds to provide information from their credit card or their banking account. This information includes account name, (account number or routing number), (credit card company / bank branch), (cvv number or check number).
5. Once all the fields are entered the user clicks the submit button.

7. The specific company verifies in their system whether the specific account exists in their system.

**System Response:**

1. The system contacts the respective company that the account from the payment method is registered under.

8. The user’s transaction gets added into their account.

**Alternative Flow of Events**

Line #2:

-The user tries making a transaction under $1 or over the amount currently in their account. Return to Step #2.

Line #7:

-The user is informed that their transaction has failed. Return to Step #1.

Use Case: Game Setup

ID: 3

Brief Description: Player creates a game session.

Primary Actor: Player

Secondary Actor: None

Precondition(s):

-Player has an account.

-Player is logged in.

-Player has sufficient funds to use in the game.

Main Flow:

1. The main flow begins when the player hits the "Create Game" button on [page here].

2. Player fills out the following information:

-Game title

-[Game type?]

-[Friends only/Public/Private?]

-Max number of players (2-7)

-Starting ante

-Pot growth rate

-Game Money (Amount of money Game Creator brings into the game)

2.1 As these fields get updated, minimum and maximum entry funds are displayed.

-Minimum is three (3) times the starting ante.

-Maximum is five (5) times the starting ante.

3. Player hits the submit button.

3.1 System confirms that the player has sufficient funds to start the game.

3.2 System confirms that the pot growth rate field is equal to or less than the starting pot field.

4. Game instance is created.

4.1 Player is taken to the waiting lobby.

4.2 Game instance is posted on the "Search for Games" page.

Post Condition(s):

-Game instance is created

-Player is in the game lobby for the game instance they created

Alternate Flow(s):

-Invalid fields

Use Case: Game Lobby

ID: 3.1

Brief Description: Game creator is waiting in the lobby for players to join. Game starts when requirements are met.

Primary Actor: Player (Game creator) [See Use Case 3]

Secondary Actors: Players

Precondition(s):

-All players are logged in.

-A game session has been created (but not yet started) by player (Game creator).

Main Flow:

1. The main flow begins when the player (Game creator) has entered the lobby for their created game.

2. Wait for players to join.

3. When player attempts to join, refuse if:

-They have insufficient funds.

-The number of players in the lobby exceeds the maximum.

3.1 If a player (not the game creator) hits the "Exit Game Session" button, see Use Case 3.1.1

4. When the number of players in the lobby is exceeds 1, the "Start Game" button becomes active.

4.1 When the number of players in the lobby drops to 1, the "Start Game" button becomes inactive.

5. The game creator hits the "Start Game" button.

6. The game session is set up with the game creator's specifications.

7. All players in the lobby are taken to the game session.

Post Condition(s):

-The game creator's custom game session begins.

-All players in the lobby are taken to the newly created game session.

Alternate Flow(s):

-Game creator hits the "Exit Game Session" button (See Use Case 3.1.2)

Use Case: Player leaves game session

ID: 3.1.1

Brief Description: Player in lobby of a game that is not their own chooses to leave the session.

Primary Actor: Player

Secondary Actor: None

Precondition(s):

-Player is logged in.

-Player is in a game lobby (i.e. the game has not started yet).

Main Flow:

1. The main flow begins when the player hits the "Exit Game Session" button.

2. The player's name is removed from the game lobby.

3. The player is taken to the "Search for Games" page.

Post Condition(s):

-Player is no longer in any game lobby.

Use Case: Game creator leaves game lobby

ID: 3.1.1

Brief Description: Game creator their own game lobby leaves their session, which ends the whole session.

Primary Actor: Player (Game creator)

Secondary Actor: None

Precondition(s):

-Player is logged in.

-Player is in a game lobby (i.e. the game has not started yet).

-Player created the game session.

Main Flow:

1. The main flow begins when the player hits the "Exit Game Session" button.

2. The game creator is taken to the "Search for Games" page.

3. Every player that is not the game creator is sent to an error page which reads, "Game Creator has ended the game session. Sorry about that!"

3.1 Those players can then hit the "Search for other games" button, which will take them to the "Search for Games" page.

4. The game session is then terminated.

Post Condition(s):

-Players are no longer in the game lobby.

-The game instance no longer exists.

-Players are now on the "Search for Games" page.

Use Case: Player joins game session

ID: 3.1.3

Brief Description: Player wants to join a registered game session.

Primary Actor: Player

Secondary Actor: None

Precondition(s):

-Player is logged in.

-Game session the player wants to join exists.

Main Flow:

1. The main flow begins when the player hits the "Join Game" button on the "Search for Games" page.

2. System checks if the player fits the requirements for the game.

3. System confirms that the game has not started.

4. Player is taken to the game session's lobby page.

Post Condition(s):

-Player has joined the game.

-Player is now viewing the game session's lobby page.

Alternate Flow(s): None

Use Case: Search for game

ID: 3.2

Brief Description: Player wants to search for games.

Primary Actor: Player

Secondary Actor: None

Precondition(s):

-Player is logged in.

-Player is on their main dashboard.

Main Flow:

1. The main flow begins when the player hits the "Search for Games" button on their dashboard.

2. The system takes them to the "Search for Games" page.

3. Results on page only show if the player can possibly join the session.

3.1 If they check the "Show unavailable games" option, they can see other game sessions (only sessions hosted on servers with ping of 100 or less).

3.2 The "Join Game" button on any game session entry on the page is inactive if the player does not fit the requirements for joining the game.

3.3 If the game has already started, the game entry states "Game in progress" and the "Join Game" button is inactive.

4. Game page refreshes whenever the status of any game changes (where ping to game server is 100 or less).

Post Condition(s):

-Player sees a list of available games that updates whenever there is a change.

Alternate Flow(s): None

Use Case: Game force-quit

ID: 3.3.1

Brief Description: Player leaves in the middle of the game.

Primary Actor: Player

Secondary Actor: None

Precondition(s):

-Player is logged in

-Player has entered a game session

-Entered game session is in progress

Main Flow:

1. Process starts when the player leaves in the middle of the game.

2. When the player leaves the game, a flag is set to where the round began.

2.1 A counter representing the number of times a player has left mid game is incremented.

3. When the player logs back in, he or she is given a a pop-up in the application notifying them the outcome.

- "You have logged out of your last match in the middle of the game. If this is done after a few times, there will be consquences [link to consequences]."

3.1. Player presses 'Okay' to remove the pop-up.

4. Player then receives another pop-up notifying him what the outcome of the match was.

- "Your received \*Blank\* and your opponent received \*Blank\*."

5. Player them presses 'Okay' to remove the pop-up.

Post-Condition(s):

-Player is notified he left in the middle of the game.

-Player is notfied what the outcome of the game was.

-Player can now setup matches or search for them again.

Alternative Flow(s):

-Player has left in the middle of the game 5 times.

Use Case: Error Handling Part 2

ID: 4.1

Brief Description: Reached a count of 5 times quitting leaving mid game.

Primary Actor: Player

Secondary Actor: None

Precondition(s):

-Player has already left the game 4 times

-Player is in another game session

Main Flow:

1. Process starts when the player leaves the middle of the game.

2. The player should have now reached a count of 5 for leaving in the middle of the game.

3. Now when the user tries logging in, the information will be entered but the account will not log in to the home page.

4. A pop-up appears instead. It will in form the user what will happened and what the consequence is.

-"You have decided to leave mid-game 5 times. This may seem not like a lot but it is unfair for the players in the match.

So your account will not be locked for 12 hours. An email will be sent claiming your account is locked."

5. The player presses 'Okay' to remove the pop-up.

Post-Condition(s):

-The player is notified that he or she left mid-game 5 times.

-Player is notified the account is locked for 12 hours.