Blackjack

Team 4

Greg Jacobus, Austin Jordan, Madison Dodd, Matt King

1. **Problem Statement**

The proposed system allows the user to simulate a casino game of Blackjack. Multiple users will be able to play against the dealer at once, competing with a simplified version of standard Blackjack rules. Each user’s balance will be stored between uses, so the user can pick up from where he/she stopped. This system will provide a new platform for users to enjoy a popular game of cards that they may not be able to play otherwise.

1. **Functional Requirements**

* The Player must log in with a valid username and password. (Login, LoginError)
* The Player can create a new account with a username and password. (CreateAccount, CreateAccountError)
* From the main menu, the Player can start a new game of Blackjack (NewGame)
* From the main menu, the Player can join an existing game of Blackjack (JoinGame)
* Before each hand, each player must place a minimum bet (PlaceBet, PlaceBetTimeoutException, InsufficientFundsException)
* When all players have placed their bets, each player is dealt two cards (DealCards)
* Starting with the player left of the dealer, each player gets to repeatedly hit (get another card) until they choose to stay (end turn) or bust (go over 21) (PlayHand)
* Either the user or the dealer goes over 21 (Bust)
* When the dealer card sum matches the player card sum, the player does not win or lose any money (Tie)
* The user wins double his bet if his card sum is higher than the dealer’s card sum and the player did not bust (UserWin)
* The user loses his bet if his card sum is lower than the dealer’s card sum, and the dealer did not bust (DealerWin)
* The player can forfeit the game in the middle of a hand (ForfeitGame)
* The player can cash out in between hands (CashOut)
* At any time, the player can access a help feature (Help)

1. **Use Case Descriptions**

Use Case: Login

Actors: Player

Steps:

1. The Player enters his or her username and password and chooses Login.
2. The System verifies the provided username and password.
3. The System logs the Player in.
4. The System displays a screen with options for starting a new game,

joining an existing game, or viewing player information.

Entry Condition:

The Player has started the game program.

Exit Condition:

The Player is logged in and is presented with the main menu screen.

Use Case: LoginError (Extension of Login at Step 2)

Actors: Player

Steps:

1. The System determines that the provided username and password are correct.
2. The System displays an error message
3. The player re-enters his or her username or password
4. The System goes back to step 2 of Login to check the username and password

again

Entry Condition:

The Player enters an incorrect username and password in Login.

Exit Condition:

The System checks the username and password again after the Player re-enters them and continues with Login.

Use Case: CreateAccount

Actors: Player

Steps:

1. The Player chooses the Create Account option on the login screen.
2. The System displays the account creation screen.
3. The Player enters a username and then enters a password twice.
4. The System validates the provided username and password. (The two  
    passwords must match, and the username cannot already be in use.)
5. The System creates the account and displays a success message.
6. The System returns the Player to the login screen.

Entry Condition:

The Player has started the game program.

Exit Condition:

A new account has been created, and the Player is returned to the login screen.

Use Case: CreateAccountError (extension of CreateAccount at step 4)

Actors: Player

Steps:

1. The System determines that the entered username or password was invalid
2. The System displays an error message explaining the problem.
3. The Player enters a new username and password for their account.
4. The System goes back to step 4 of CreateAccount to check the username and

password again.

Entry Condition:

The Player enters an invalid username or password in CreateAccount.

Exit Condition:

The System checks the username and password again after the Player re-enters them and continues with CreateAccount.

Use Case: NewGame

Actors: Player

Steps:

1. The Player chooses the New Game Option
2. The System prepares a new game
3. The System randomly assigns the player a seat at the Blackjack table
4. The System displays the client’s Blackjack Table GUI
5. The System displays the Bet popup GUI window

Entry Condition:

The user is logged in and not currently in a game

Exit Condition:

New game is created and the player is presented with the bet popup GUI window

Use Case: JoinGame

Actors: Player

Steps:

1. The Player chooses the Join Game Option
2. The System displays a list of available games to join
3. The Player chooses a game to join
4. The System adds the Player to the game
5. The System randomly assigns the player a seat at the Blackjack table
6. The System displays the client’s Blackjack Table GUI
7. The System displays the Bet popup GUI window when the current hand is

current is finished

Entry Condition:

The Player is logged in and not currently in a game

Exit Condition:

The player has joined a game and is presented with the bet popup GUI window

Use Case: PlaceBet

Actors: Player

Steps:

1. The user selects an amount to bet
2. The user submits the bet amount when ready, within a given amount of time
3. The System verifies that the user has sufficient money to place the bet
4. The System subtracts the bet from the user’s account
5. The System removes the bet popup GUI window and returns the player to

the main game screen

Entry Condition:

The player is presented with the bet popup GUI window

Exit Condition:

The player is presented with the main game GUI window, a bet is placed, and the bet is subtracted from the user’s account

Use Case: InsufficientFundsException (extension of PlaceBet at step 3)

Steps:

1. The System displays an error message
2. The System returns the user to the bet popup GUI window
3. The user makes a new bet
4. The System verifies that the user has sufficient funds. Otherwise step 3 is

repeated

Entry Condition:

The user tries to place a bet greater than his/her current balance

Exit Condition:

The system checks the bet again after player re-enters them and continues with PlaceBet

Use Case: PlaceBetTimeoutException (extension of PlaceBet at Step 2)

Actors: Player

Steps:

1. The System displays an error message
2. The System returns the user to the main game GUI window
3. The Player watches the hand without particpating

Entry Condition:

The user failed to place a bet within a given amount of time

Exit Condition:

The hand has finished without the player participating

Use Case: DealCards

Actors: Player

Steps:

1. The System randomly gives each player two cards
2. The System updates the GUI so the player can the cards
3. The System presents the user with the option to hit or stay

Entry Condition:

All users have placed bets

Exit Condition:

Each player has two cards and is presented with the option to hit or stay

Use Case: PlayHand

Actors: Player

Steps:

1. The User can choose to hit (get another card) until they want to stay(finish turn) or go over 21
2. The System verifies that the score is under 21 after each hit

Entry Condition:

Each user has been dealt two cards and is presented with options to hit or stay

Exit Condition:

The player has chosen to stay or gone over 21 (bust)

Use Case: Bust (Extension of PlayHand at Step 2, DealerPlay at Step )

Actors: Player

Steps:

1. The System determines that the user’s score is above 21
2. The System does not return the user’s money
3. The user observes until the next hand

Entry Condition:

The System has determined that the score is above 21

Exit Condition:

The actor is not returned any money

Use Case: Tie

Actors: Player

Steps:

1. The System returns the user’s bet to the user
2. The System displays a message telling the user of the result
3. The Player acknowledges the result
4. The System updates the player’s account info
5. The System starts the next hand

Entry Condition:

The System has determined that the player card sum is equal to the dealer card sum

Exit Condition:

User is returned his bet and moves on to next hand in bet popup GUI window

Use Case: UserWin

Actors: Player

Steps:

1. The System returns the user’s bet to the user
2. The System displays a message telling the user of the result
3. The Player acknowledges the result
4. The System updates the player’s account info
5. The System starts the next hand

Entry Condition:

The System has determined that the player card sum is greater than the dealer card sum or the dealer busts

Exit Condition:

User is returned double his bet and moves on to next hand in bet popup GUI window

Use Case: DealerWin

Actors: Player

Steps:

1. The System returns the user’s bet to the user
2. The System displays a message telling the user of the result
3. The Player acknowledges the result
4. The System updates the player’s account info
5. The System starts the next hand

Entry Condition:

The System has determined that the player card sum is less than the dealer card sum or the player busts

Exit Condition:

User is not returned the money he bet and moves on to next hand in bet popup GUI window

Use Case: ForfeitGame

Actors: Player

Steps:

1. At any time during a hand, the user clicks the button to forfeit the game
2. The System removes the user from the game
3. The System does not return any money the user has already bet
4. The System returns the user to the main menu

Entry Condition:

The user is in the middle of a hand

Exit Condition:

The user is removed from the game and returned to the main menu

Use Case: CashOut

Actors: Player

Steps:

1. The user clicks the “Cash Out” button
2. The System removes the user from the game
3. The System updates the user’s bank account
4. The System returns the user to the main menu

Entry Condition:

The user is presented with the bet popup GUI window

Exit Condition:

User has left the game and returned to main menu

Use Case: Help

Actors: Player

Steps:

1. The user clicks the help button
2. The System presents the user with a help menu
3. The user can continue playing the game while the help window is displayed
4. The user can close the help window at any time

Entry Condition:

User is logged in

Exit Condition:

The user has viewed the rules and the help window is closed

1. **Use Case Diagram**

