

## Use cases

Use case ID: 1

Use case name: Login

Relevant Requirements: clients/ Message

Primary Actor: Client

Pre-conditions: Username must have a minimum of 3 characters

Post-conditions: Direct the user to input their password

Basic Flow or Main Scenario:

1. Request the user to input a username that is a minimum of 3 characters or more.
2. The system will ask the user to input their password.
3. The system redirects to the login page.

Extensions or alternate Flows: If a username is already taken, the system will prompt the user to create another name.

Exceptions:

1. If there are less than 3 characters, prompt the user to correctly place 3 characters or more.

Related use cases: Ask the user if they are comfortable with their Login UN and PW after signup was properly successful. Login will shortly be displayed as follows.

Use Case ID: 2

Use Case Name: Login/ Client

Relevant Requirements: Signup client, Player/Dealer=(user)

Primary Actor: Client

Pre-Conditions: A valid (in format) username and password.

Post-Conditions: Display the user information on the client side, in format: (Username, balance)

Basic Flow or main Scenario:

1. Request the user to input their username and password.
2. Return their account information after successfully verified.
3. Display to the User of a successful login as well as their balance amount and their role.

Extensions or Alternate Flows: If the user enters a UN/PW more than 5 times, the system will prompt to create a new account.

Exceptions:

1. If the username is found but the wrong password, retry until valid info has been entered.

Related Use Cases: Signup to create an Account to enter Lobby view

Use Case ID: 3

Use case Name: Server

Relevant Requirements: Clients and Lobby manager

Primary Actor: Client

Pre-Conditions: User will connect through Client and Server

Post-Conditions: Server will have communications with Client throughout the program

Basic Flow or Alternate Flows:

1. After a User has been successfully logged in through the Client

2. The Server will verify that the User has a proper account
3. Allow communication throughout the process of the program

**Extensions or Alternate Flows:** The Message class will relay information from the server to the client for simplicity of the program.

**Exceptions:** If the server is unable to verify the Users account the program will not communicate through different clients

**Related Use Cases:** Client, Message

**Use Case ID:** 4

**Use Case Name:** Lobby Manager

**Relevant Requirements:** Login for account creation

**Primary Actor:** Account/User

**Pre-Conditions:** Must have an account with a balance

**Post-Conditions:** Display account name and balance

**Basic Flow or Alternate Flows:**

1. Program will direct the User to GUI of Lobbies.
2. The User will be given a display of various lobbies and their current balance.
3. There are several options to choose from when in LobbyManager: Join a specific lobby or Bank(Reload)
4. Prompt user of their current balance and how much they would like to reload.
5. Show user several lobbies(tables) and the amount of how many players are at a table.
6. System will show which tables are available to join.

**Extensions or Alternate flows:** Users are able to access the "bank" to increase their balance, removing a lobby that already exists

**Exceptions:**

1. If the balance of the account is less than 100 then the user is unable to join a table.
2. If the number of players at a table is full and the user tries to join, then the system will prompt the user to select another table.

**Related Use Cases:** Account and balance overview to get into a table for gameplay

**Use Case ID:** 5

**Use case Name:** Account Balance

**Relevant Requirements:** Client

**Primary Actor:** User

**Pre-Conditions:** Must have an account of with a balance

**Post-Conditions:** Have a new balance amount

**Basic Flow or Alternate Flows:**

1. Button to add set amount of funds
2. Successfully add funds into their account.
3. Update balance to server

**Extensions or Alternate Flows:** There will be a set limit to how many times a user can add funds to their account

**Exceptions:** If a User has a balance of 0 then they will not be allowed to select any of the lobby options

Related Use Cases: **Lobby Manager**

Use Case ID: **6**

Use case Name: **Join**

Relevant Requirements: **Lobby Manager**

Primary Actor: **User**

Pre-Conditions: **User has an Account w/ balance and specific role**

Post-Conditions: **Seated into a lobby**

Basic Flow or Alternate Flows:

1. First, the user can select their preferred lobby
2. Then, a button will allow a user to join their desired lobby

Extensions or Alternate Flows: **There can only be one dealer per lobby, and if the lobby contains one already then jump to exceptions**

Exceptions: **If there are a max amount of players in a lobby, the message will notify the User to select a different one.**

Related Use Cases: **Server, Lobby Manager**

Use Case ID: **7**

Use Case Name: **Gameplay**

Relevant Requirements: **Account can link to a player/dealer**

Primary Actor: **Player/Dealer/ User**

Pre-Conditions: **Account must have a minimum balance of \$100**

Post-Conditions: **Players and Dealers have the option to opt-out at the end of a round.**

Basic Flow or Alternate Flows:

1. After successfully joining a lobby the system will assign seats.
2. System will prompt users of how much they want to bet, but must be more than minimum (\$5).
3. Dealers will have the start round button.
4. Start the round: Players receive 2 cards face-up, dealer has 1 face-down and 1 face-up card
5. Going clockwise after the dealer, taking turns, players will have several options: double down, split, stay, or hit.
6. If the sum of cards is over 21, players lose their bet immediately to the dealer.
7. Dealer plays, if dealer has 17-21 automatic stay, If over 21 everyone who hasn't 'busted' gets a payout.
8. Payout is done based on each player's hand compared to the dealer.
9. If balance reaches 0, Automatically send the player to the lobby manager.

Extensions or alternate flows: **The double-down, split, stay, and hit will be a separate, and simple function within the gameplay. Dealer is playing for the house, they will have no currency to bet**

Exceptions:

1. If the player has less than the minimum amount to bet, the system will prompt the user to reload back at lobby view.
2. If a player exceeds their balance to bet, the system will input the user's entire balance.

3. If a player has insufficient funds to double down, the option to double down will not be available for the player.

Related Use Cases: Account balance, Server

Use Case ID: 8

Use case Name: Deck (Hand out cards)

Relevant Requirements: Gameplay is able to run

Primary Actor: Server

Pre-Conditions: Properly Loads 52 cards within a deck

Post-Conditions: Reshuffle all the cards that have been used back into the deck

Basic Flow or Alternate Flows:

1. Cards will be passed out based on an array of 52 unique cards
2. As each card gets taken out, that card cannot be called again
3. After the round has finished all the cards that have been taken out get placed back into the deck, until the next round.

Extensions or Alternate Flows: There will be a total of 13 different ranks but each will have a total of 4 different suits per rank

Exceptions: If there is no deck then the user will have to return to the lobby manager

Related Use Cases: Gameplay

Use Case ID: 9

Use case Name: Message

Relevant Requirements: Clear directions for Users to follow

Primary Actor: Client

Pre-Conditions: Server and Client have a live connection throughout process

Post-Conditions: The message class will terminate along with the program.

Basic Flow or Alternate Flows:

1. Dialogue/ text box
2. Will inform the user of each instance of the program's process
3. Will also show the gameplay dialogue

Extensions or Alternate Flows: Message is required for the program to be successfully run.

Exceptions: If there is no message the user would have to restart the entire program

Related Use Cases: Client and Server