# Group 2: Multi-Player **Blackjack**

Software Requirements Specification

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** |
| 02/27/2025 | 1.0 | Purpose | Mahlet Bekele |
| 03/01/2025 | 1.1 | Worked on definitions | Brandon Wall |
| 03/02/2025 | 1.2 | Worked on module requirements | Brandon Wall |
| 03/03/2025 | 1.3 | Section 2.4, 2.5, 3, 4 | Cristian Gonzales |
| 03/04/2025 | 1.4 | Add some Use cases | Mahlet Bekele |
| 03/04/2025 | 1.5 | Working on module requirements | Brandon Wall |
| 03/04/2025 | 1.6 | Revised some constraints, working on some requirements | Kenny Yu |
| 03/04/2025 | 1.7 | Adjusted some wordings and added some constraints and assumptions | Armand Tirado |
| 03/04/2025 | 1.8 | Worked on document formatting and edited module requirements | Brandon Wall |
| 03/04/2025 | 1.9 | Worked on requirements and formatting | Kenny Yu |
| 03/04/2025 | 1.10 | Worked on requirements and formatting  And class diagrams | Cristian Gonzales |
| 03/04/2025 | 1.11 | Worked on requirements and formatting | Brandon Wall |
| 03/04/2025 | 1.12 | Worked on requirements and formatting | Mahlet Bekele |
| 03/04/2025 | 1.13 | Worked on requirements and formatting | Armand Tirado |
| 03/05/2025 | 1.14 | Worked on the use case and sequence diagram | Kenny Yu |
| 03/05/2025 | 1.15 | Worked on the use case and sequence diagram | Mahlet Bekele |
| 03/05/2025 | 1.16 | Worked on the use case and sequence diagram | Brandon Wall |
| 03/05/2025 | 1.17 | Worked on the use case and sequence diagram | Cristian Gonzales |
| 03/05/2025 | 1.18 | Worked on the use case and sequence diagram | Armand Tirado |
| 03/05/2025 | 1.19 | Worked on use case specifications page formatting | Brandon Wall |

Table of Contents

1. Purpose [4](#__RefHeading___Toc19440719)

1.1. Scope [4](#__RefHeading___Toc19440720)

1.2. Definitions, Acronyms, Abbreviations [4](#__RefHeading___Toc19440721)

1.3. References [4](#__RefHeading___Toc19440722)

1.4. Overview [4](#__RefHeading___Toc19440723)

2. Overall Description [5](#__RefHeading___Toc19440724)

2.1. Product Perspective [5](#__RefHeading___Toc19440725)

2.2. Product Architecture [5](#__RefHeading___Toc19440726)

2.3. Product Functionality/Features [5](#__RefHeading___Toc19440727)

2.4. Constraints [5](#__RefHeading___Toc19440728)

2.5. Assumptions and Dependencies [5](#__RefHeading___Toc19440729)

3. Specific Requirements [6](#__RefHeading___Toc19440730)

3.1. Functional Requirements [6](#__RefHeading___Toc19440731)

3.2. External Interface Requirements [6](#__RefHeading___Toc19440736)

3.3. Internal Interface Requirements [7](#__RefHeading___Toc19440737)

4. Non-Functional Requirements [8](#__RefHeading___Toc19440738)

4.1. Security and Privacy Requirements [8](#__RefHeading___Toc19440739)

4.2. Environmental Requirements [8](#__RefHeading___Toc19440740)

4.3. Performance Requirements [8](#__RefHeading___Toc19440741)

# Purpose

This document outlines the requirements for the Multiplayer Blackjack game.

## Scope

This document will catalog the user, system, and hardware requirements for the Multiplayer Blackjack Gaming System. It will define what the system must accomplish but will not detail how these requirements will be implemented.

## **Definitions, Acronyms, Abbreviations**

* + 1. Hand: The collection of cards each player possesses during the game.
    2. Table: A game room consisting of one dealer and 0-6 players.
    3. Dealer: The person who distributes cards to active players and plays against them.
    4. Blackjack: A hand consisting of two cards whose sum equals 21.
    5. Bet: The amount of money the player wishes to wager that they’ll win against the dealer
    6. Table Limit: The minimum and maximum amount players can bet at a table.
    7. Hit: An action made by a player that will adds the top card of the deck into the player’s hand.
    8. Stand: An action made by the player when the player wishes to ends their turn.
    9. Fold: When a player decides to forfeit the game, losing all money they’ve bet for that game.
    10. Bust: When a player hand total is over 21 resulting in a loss.
    11. Card: An object that has a numerical value and a card symbol that is either a spade, diamond, heart, or club. Numerical values range from 1-11.
    12. Card Symbol: Each card has a card symbol, which is either a spade, diamond, heart, or club.
    13. Ace Card: A variant of a normal card that has a numerical value of either 1 or 11 by the player's choice and has a card symbol.
    14. Face Card: A variant of a normal card, but has a numerical value of 10 and is called either a Jack Queen or King. Face cards also have one of the four card symbols.
    15. Deck: A collection of 52 unique cards, where there are four versions of each card in the deck, each with one of the four card symbols.
    16. Shoe: A collection of multiple decks.
    17. Split: An action made by the player when a player's initial hand consists of 2 identical cards and wants to create two separate hands, each consisting of one of the identical cards from the initial hand and the next top card from the shoe.
    18. Double down is a move in blackjack where a player doubles their bet and receives only one more card.
    19. Push: When the sum of the player’s hand is equal to the sum of the dealer’s hand.
    20. TCP/IP (Transmission Control Protocol/Internet Protocol): a group of network protocols that let the client and the server communicate.
    21. GUI (Graphical User Interface): the game’s visual interface that allows players to interact with it.

## References

Use Case Specification Document: Refer to Use Case Specification.docx

UML Use Case Diagrams Document: Refer to Use Case Diagram Blackjack.docx

Class Diagrams: Refer to UML Class Diagram Blackjack.docx

Sequence Diagrams: Refer to UML Sequence Diagrams Blackjack.docx

GitHub Repository: Refer to: <https://github.com/bwalldev/CS401_BlackJackGroupProject>

Gantt Chart: Refer to Gantt project planner.xlsx

## **Overview**

The multiplayer blackjack gaming system is a Java-based application with a GUI, enabling real players to join virtual tables, place bets, and play under standard blackjack rules. It operates over TCP/IP, ensuring real-time gameplay, fund transactions, fair play, and minimal cheating.

# Overall Description

## Product Perspective

The Online Multiplayer Blackjack Gaming System is a Java-based cross-platform program designed to provide a real-life blackjack gaming experience online. It uses a graphical user interface (GUI) for immersive player interactions with the system and a TCP/IP network for interactions between the players themselves.

## Project Architecture

The Blackjack game system will be organized into six major modules: the payouts and betting module, network module, game manager module, player module, dealer module, and table module.

## Project Functionality/Features

The high-level features of the system are as follows (see section 3 of this document for more detailed requirements that address these features):

## Constraints

* + 1. Verified users: The system must authenticate users by username and password; thus, if a user doesn’t have these valid credentials, the user is unable to play in the system.
    2. 6 Players Per Table: Only a maximum of 6 players can play at a table at any one time.
    3. Maximum 1 dealer per table: Only a maximum of 1 dealer can be at a table, and a dealer must be at a table for a game to start.
    4. Player Action: players can only stand, hit, double down, or split (if allowed)
    5. Player Leaves: If the player leaves the game, then the bet previously placed is to be forfeited.
    6. If a player’s hand exceeds 21 then they bust and lose the game automatically.
    7. Before receiving their hands, players must first place their bets.
    8. Player and Dealer must receive cards from a randomly shuffled deck.
    9. Players can only join a game lobby where a dealer is present.
    10. Game must follow the standard rules of blackjack.
    11. Players can only place bets that are within the amount of their funds.
    12. Players can only join a table if they have enough funds and if there are less than 6 players on the table.
    13. Players may only deposit money in their bank up to the amount of money in their account

## Assumptions and Dependencies

* + 1. It’s assumed that the players have a stable internet connection
    2. It’s assumed that players know how to play blackjack
    3. It’s assumed that each player has a unique account, such as unique username and password.
    4. It’s assumed that each player has enough or can obtain enough funds to play in a table.
    5. It’s assumed that each table works within the player number limit of 6 players per table.
    6. It’s assumed that each player makes a decision within a certain short amount of time or else be booted from the game to ensure smooth gameplay between all players.
    7. It’s assumed that the system will use a form of random number generator to shuffle the deck of cards to prevent card counting or cheating.
    8. It’s assumed that a game starts when the minimum number of players is met, being one player and they place a bet.
    9. It’s assumed that the player or players are all going to play their cards against the dealer.
    10. It’s assumed that any bet made is final once confirmed.
    11. The GUI design is uniform
    12. The GUI will adjust smoothly to real-time updates such as when cards are drawn, bets are placed, and stats are changed.
    13. It’s assumed that the dealer will use a “shoe” or 3 decks of cards together to shuffle with to minimize card counting and cheating.
    14. It is assumed that each player will only be logging in with one client at a given time.

# Specific Requirements

## Functional Requirements

* + 1. **Common Requirements:**
       1. Users should be allowed to log in to an account using an existing username and password.
    2. **Payouts and Betting Module Requirements:**
       1. The system must allow players to wager money from their account funds when they are in an active game.
       2. The system must add game winnings from each game to each winning players funds
       3. The system must be able to remove funds from players accounts when placing a bet if they have funds available.
       4. Players must place a minimum bet of $1 per hand.
       5. Players must place bets using integer values only, decimal values are not allowed.
    3. **Network Module Requirements:**
       1. The system shall allow users to connect to the game server
       2. The system shall update game data in real-time
       3. The system shall allow players to disconnect at any time. Dealers must wait until the end of the current game before they are able to disconnect.
       4. If a player loses connection, they will be dropped from the game and all their current bets will be forfeited
    4. **Game Manager Module Requirements:**
       1. The system must track and update the game’s current state in between transitions of betting phase, card dealing, turns, and round end.
       2. A time limit will be implemented for each player’s turn to ensure smooth gameplay.
       3. If a player fails to act within the time limit (1 minute) the system will automatically stand for them.
       4. The game system must compare against the final hand of each player's hand and dealer to determine winners.
       5. The system will remove a table from the list of available tables once there is no dealer present at a table.
    5. **Player Module Requirements:**
       1. Players are allowed to withdraw and deposit money into their account. Winnings are credited, and players can withdraw their funds at any time.
       2. The player user account will include session management and player balance.
       3. Players can only join a table if it doesn’t exceed the maximum number of players of 6 and must have available funds in their account.
       4. Player must wait in the lobby until invited to join a table by dealer
    6. **Dealer Module Requirements:**
       1. The dealer must play only after all players have completed their acts first.
       2. The dealer must hand out two cards for the players and themselves initially.
       3. For a fair play, the dealer must not make any out-of-game rule decisions.
       4. Dealers will not have the option of leaving a table while there is an active game in session.
    7. **Table Module Requirements:**
       1. In order for a game to start, there must be at least one player and exactly one dealer, with a maximum of six players.
       2. A table will remain active as long as there is exactly one dealer present.
       3. Each table will have a designated table limit, which includes a minimum and maximum betting limit per game.
  1. **External Interface Requirements:**
     1. The system must provide an easy-to-navigate graphical user interface for the players and dealers to interact with.
     2. The GUI must display real-time game data, which includes player actions, card draws, player bets, game payouts, game results, standing, hitting, busting, splitting, doubling down, and all other game actions.
     3. The system must display a unique dashboard for dealers and a unique dashboard for players.
     4. The system must have an immersive and visually appealing GUI for players to use throughout the game to keep them enthralled.
     5. The GUI should have animations for shuffling cards, card dealing, and chip movements to enhance player immersion.
     6. Players must be given an option to place bets, hit, stand, double down, split, and leave game
     7. Before playing a game, the system must display game lobbies in order to view any available tables for the player to join.
  2. **Internal Interface Requirements:**
     1. The game must verify the user’s username and password in order to login through internal authentication of the system.
     2. The game must generate player funds upon login.
     3. If the user is timed out, quits, or loses connection, then all in-game interactions must be immediately terminated.
     4. The system must track all game status changes and alert the player in real time such as who won the hand and all other player and dealer hands.
     5. The internal interface must support all possible player actions such as; hit, stand, double down, split, and leave game.
     6. The system must use a Random Number Generator to ensure a fair deck shuffling.
     7. The system must notify when the player doesn’t have sufficient funds that their betting request is rejected before the game starts
     8. The deck must take the place of a “shoe” or 3 decks together and reshuffling must occur halfway through the game.
     9. The systems graphic user interface must receive data from the game engine in real time and display it for the user to see.

# Non-Functional Requirements

## Security and Privacy Requirements

* + 1. **Secure Authentication:** The player must log into the system using a valid username and password.
    2. **Role-Based Controls:** Users of certain roles are to have limited access to functions that are strictly limited to those of their roles (Player & Dealer).
    3. **Anti-Cheating:** The system must include anti-cheating in the game by using 3 decks of cards or a “shoe” for card counters and even reshuffle the deck when it is halfway also.

## Environmental Requirements

* + 1. The system must be developed using the Java programming language that must be compatible with all platforms, including Windows, MacOS, and Linux.
    2. The system must have consistent UI design from all screens, from the lobby menu to checking balance to the actual blackjack gameplay.
    3. The system must operate over a TCP/IP, ensuring real-time gameplay between more than one player on different devices in the same network.
    4. The system must be Java-based with a GUI which allows for the players to join virtual tables in order to play on the system
    5. The system needs a stable internet connection to maintain real-life gameplay immersion; otherwise, there will be lag.

## Performance Requirements

* + 1. **Immediate Response Time:** Any actions from the player in the lobby or in the game must be processed within 500 milliseconds so as to not break immersion.
    2. **Table Users:** The system must be able to support 6 concurrent players in each table in maintaining real-time gameplay without performance issues.
    3. **Casino Users:** The system must be able to support a lobby or casino of multiple 6-person tables maintaining real-time gameplay without performance issues.
    4. **Disconnecting Players:** In the event where a player loses connection to the system, willingly quits the game, or connection is timed out due to inactivity, then the game