**Multiplayer BlackJack Game**

Software Requirements Specification

Revision History

| **Date** | **Revision** | **Description** | **Author** |
| --- | --- | --- | --- |
| 09/13/2022 | 1.0 | Initial Version | Yang Liu |
| 09/24/2022 | 1.1 |  | YL, AL, EL |
| 9/24/2022 | 1.2 | Added 2.4.1, 4.1.1, 4.2.1, 4.3.1 | Yang Liu |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

**1.** **Purpose 4**

1.1. Scope 4

1.2. Definitions, Acronyms, Abbreviations 4

1.3. References 4

1.4. Overview 4

**2.** **Overall Description 5**

2.1. Product Perspective 5

2.2. Product Architecture 5

2.3. Product Functionality/Features 5

2.4. Constraints 5

2.5. Assumptions and Dependencies 5

**3.** **Specific Requirements 6**

3.1. Functional Requirements 6

3.2. External Interface Requirements 6

3.3. Internal Interface Requirements 7

**4.** **Non-Functional Requirements 8**

4.1. Security and Privacy Requirements 8

4.2. Environmental Requirements 8

4.3. Performance Requirements 8

# Purpose

This document outlines the requirements for the Multiplayer BlackJack Game (MBG).

## Scope

This document will catalog the user, system, and hardware requirements for the MBG. It will not, however, document how these requirements will be implemented.

## Definitions, Acronyms, Abbreviations

Sit - Refers to the player’s ability to take a seat at the table to join the game.

Hit - When the player opts to ask for another card.

Stand - When the player opts to *not* ask for another card.

Split - Refers to the player’s ability to ‘split’ their hand into two separate hands if they were dealt two cards of the same denomination. If choosing this route, the player must place a bet equal to the first amount, and then both hands are played separately, starting with the one on the left. Limitations of splitting: a pair of aces yields the ability to ‘hit’ just once after splitting. Additionally, if a ten-card is dealt to either hand with an ace, the payoff is 1:1, rather than the typical 1.5:1.

Double - Refers to the player’s ability to double their bet when the cards in their dealt hand total to 9, 10, or 11. The next card will be dealt to the player face down, and is turned up only after the bets are settled at the end of the hand.

## References

Use Case Specification Document – Step 2 in assignment description

UML Use Case Diagrams Document – Step 3 in assignment description

Class Diagrams – Step 5 in assignment description

Sequence Diagrams – Step 6 in assignment description

## Overview

The Multiplayer BlackJack Game allows multiple players to connect to a server to play BlackJack. It will allow players to place wagers, and deposit money into or cash out their account.

# Overall Description

## Product Perspective

## Product Architecture

The system will be organized into \_\_\_ major modules: the Player module, the \_\_\_ module, and the \_\_\_\_\_ module.

Note: System architecture should follow standard OO design practices.

## Product 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

List appropriate constraints.

Constraint example: Since users may use any web browser to access the system, no browser-specific code is to be used in the system.

2.4.1 Users must use an OS with GUI interface. Software not developed for CLI use.

## Assumptions and Dependencies

Assumption Example: It is assumed that the maximum number of users at a given time is 15,000.

Players have Java installed.

Players have a reliable internet connection.

Money transferred in and out of accounts is imaginary.

Players will know

# Specific Requirements

## Functional Requirements

### Common Requirements:

Provide requirements that apply to all components as appropriate.

Example:

3.1.1.1 Users should be allowed to log in using their issued id and pin, both of which are alphanumeric strings between 6 and 20 characters in length.

3.1.1.2 The system should provide HTML-based help pages on each screen that describe the purpose of each function within the system.

### \_\_\_\_\_ Module Requirements:

Provide module specific requirements as appropriate.

Example:

3.1.2.1 Users should be allowed to log in using their issued id and pin, both of which are alphanumeric strings between 6 and 20 characters in length.

### \_\_\_\_\_ Module Requirements:

Provide module specific requirements as appropriate.

Example:

3.1.2.1 Users should be allowed to log in using their issued id and pin, both of which are alphanumeric strings between 6 and 20 characters in length.

### \_\_\_\_\_ Module Requirements:

Provide module specific requirements as appropriate.

Example:

3.1.2.1 Users should be allowed to log in using their issued id and pin, both of which are alphanumeric strings between 6 and 20 characters in length.

## External Interface Requirements

Provide module specific requirements as appropriate.

Example:

3.2.1 The system must provide an interface to the University billing system administered by the Bursar’s office so that students can be automatically billed for the courses in which they have enrolled. The interface is to be in a comma-separated text file containing the following fields: student id, course id, term id, action. Where “action” is whether the student has added or dropped the course. The file will be exported nightly and will contain new transactions only.

## Internal Interface Requirements

Provide module specific requirements as appropriate.

Example:

3.3.1 The system must process a data-feed from the grading system such that student grades are stored along with the historical student course enrolments. Data feed will be in the form of a comma-separated interface file that is exported from the grading system nightly.

3.3.2 The system must process a data-feed from the University billing system that contains new student records. The feed will be in the form of a comma-separated text file and will be exported from the billing system nightly with new student records. The fields included in the file are student name, student id, and student pin number.

# Non-Functional Requirements

## Security and Privacy Requirements

4.1.1 User information will not be shared to third parties

## Environmental Requirements

4.2.1 Users must have Java virtual machine installed

## Performance Requirements

4.3.1 N/A performance is server and connection dependent.