# Assignment #2 – Multi-Player BlackJack

Software Requirements Specification

Ka Yee Ching

Daniel Jensen

Chun Yin Lee

Aastha Sharma

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** |
| 06/15/2022 | 1.0.0 | Initial Version | Daniel Jensen |
| 6/20/2022 | 1.0.1 | Updated Sections 1 & 2 & 4 | Daniel Jensen |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 multi-player blackjack game.

## Scope

This document will catalog the user and system requirements for the multi-player blackjack game. It will not, however, document how these requirements will be implemented.

## Definitions, Acronyms, Abbreviations

List any acronyms, terms etc. that need to be defined.

* + 1. Blackjack – A winning hand; where someone is dealt 2 cards, an ace (card value of 11) and a card with a value of 10 to hit a winning score of 21. The game gets its namesake from this winning hand.
    2. House – A term referring to the company running the game. The casino in most cases.

## 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 multi-player blackjack game will be designed to allow many users to play blackjack together on a game server. Users will be able to login through a client application to connect to the server.

# Overall Description

## Product Perspective

## Product Architecture

The system will be organized into \_\_\_ major modules: the \_\_\_ 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

* + 1. Since each game of blackjack will be limited to 5 players, multiple games will run at the same time.
    2. Since we cannot add new users to a game in progress, new users will be added to new games.
    3. Since players can leave (or disconnect) anytime they want, for any hand that they have bet on but not completed, that money will be forfeited to the house.

## Assumptions and Dependencies

* + 1. It is assumed that each game of blackjack will only be using one deck and said deck will get shuffled between each game.
    2. It is assumed that the game will be played according to the basic rules of blackjack. Extras like blackjack insurance will not be necessary.

# Specific Requirements

## Functional Requirements

### Common Requirements:

Provide requirements that apply to all components as appropriate. SR10

Example:

3.1.1.1 SR9 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 SR23 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. SR10

Example:

3.1.2.1 SR9 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. SR10

Example:

3.1.2.1 SR9 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. SR10

Example:

3.1.2.1 SR9 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. SR10

Example:

3.2.1 SR9 SR1 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. SR10

Example:

3.3.1 SR17 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 SR24 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

* + 1. The sSR8 system must only allow individual users and the system administrators to view balance and login credentials.

## Environmental Requirements

* + 1. The sSR8 system will require that a Java Runtime Environment (JRE) compatible with Java 17 be installed on the user’s computer.
    2. The system must be developed using Java 17.

## Performance Requirements

Example:

* + 1. System must render all UI pages in no more than 9 seconds for dynamic pages. Static pages (HTML-only) must be rendered in less than 3 seconds.