**Assignment #2 – Template**

Software Requirements Specification

Revision History

| **Date** | **Revision** | **Description** | **Author** |
| --- | --- | --- | --- |
| 09/23/2022 | 1.0 | Initial Version | Nathaniel Gong  Anshul Gupta  Armando Ramirez  Joshua Jefson Romulo  Marlon Trejo |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 Blackjack gaming system.

## Scope

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

## Definitions, Acronyms, Abbreviations

**GUI**: graphical user interface

**TCP/IP**: Transmission control protocol/internet protocol - a set of communication rules

that allows computers to communicate via networks

**Hit:** Take another card.

**Stand:** Take no more cards; also known as "stand pat", "sit", "stick", or "stay".

**Double-down:** Increase the initial bet by 100% and take exactly one more card.

## 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 Blackjack gaming system is designed to allow players to play online using a Java application over TCP/IP, and are able to access their password-protected user accounts to deposit and withdraw funds as needed for their games. The Blackjack gaming system is able to host many games with the capacity of 6 players per game, starting off by giving said players 2 cards, and are given the ability to ask the dealer to hit (draw) or stand (pass). If a player obtains a combined card value of more than 21 (bust), then said player loses; otherwise, if the player obtains a combined card value of 21, then he/she wins over other players who have less a value of 21.

## Product Perspective

## Product Architecture

The system will be organized into 6 major modules:the account module, the player module, the game module, the card module, the deck module, the dealer module

## 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):

Banking feature that allows storage and withdrawal of funds

A turn feature that gives players options during their turns

An account feature that holds player information

A stand feature that allows players to maintain their hand

A hit feature that allows players to draw cards

A feature that makes the dealer draw up to the value of 17

A timing feature to prevent players from stalling

A random shuffle feature

## 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.

The user (within an acceptable age range)

Internet connection

A minimum amount of money in your account.

They have to have java installed

Code needs to be written in java

Code cannot have web/HTML components

Turns are 30 seconds long (max)

## Assumptions and Dependencies

List appropriate assumptions

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

The game is dependent on the the maximum number of players being 6

it’s assumed that each player has an account

It’s assumed that each player has a stable connection to the internet

It’s assumed that players will play fairly (No anti-cheating features will be implemented)

It’s assumed that the

# Specific Requirements

## Functional Requirements

### Account Module Requirements:

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

2.1.1.2 Users should be allowed to deposit/withdraw money from their account.

2.1.1.3 Users should only have one account.

### Player Module Requirements:

2.1.2.1 Players should be able to hit (ask for a card)

2.1.2.2 Players should be able to stand (end their turn without taking a card)

2.1.2.3 Players should have a state checking whether they’re still in the game.

2.1.2.4 Players should be able to double down only after their two original cards total up to 9, 10, or 11 (double their wager and take a card)

2.1.2.5 Players should be able to surrender (give up half their bet and retire from the game).

2.1.2.6 Players’ cards should all be face up on the table.

### Game Module Requirements:

2.1.3.1 Game class should reference the Account class to check and display the balance of the player’s funds.

2.1.3.2 If the player’s funds are insufficient, prompt the user regarding their insufficient funds, and ask if they want to add money to their account.

2.1.3.3 If the player stalls (within a 2-3 minute period), the player will automatically lose, and the amount that they wagered during that stall phase will be deducted.

2.1.3.4 The order of the players in which they are hitting/passing is determined by when they have entered a game/table.

2.1.3.5 After a game finishes a message is prompted whether a player(s) would like to play again or exit the table/game.

### Card Module Requirements:

2.1.4.1 Cards from 1 to 10 have a face value, Ace cards have a value of 1 or 11, and King, Queen, Jack have a value of 10.

2.1.4.2 The card class should return the color, value, and shape of the drawn card.

### Deck Module Requirements:

2.1.5.1 The deck class/module should have 52 cards.

2.1.5.2 The deck should have randomized cards, taking in from the Card class (making objects).

### Dealer Module Requirements:

2.1.6.1 The dealer needs (required) to hand out cards when the user requests for a card.

2.1.6.2 The dealer needs (required) to draw cards if the value is 17 and under.

2.1.6.3 The dealer needs (required) to have one face up card and one facedown card for their starting hand.

2.1.6.4 The dealer needs (required) to go last and reveal his cards after the players do.

## External Interface Requirements

2.2.1 The account interface should have a text box for an email and password

2.2.2 There should be a login button to verify the user and give access to the account

2.2.3 There should be three button options displayed once the user logs in: Play and Add Funds and Withdraw Funds

2.2.4 The withdraw funds interface should have displayed current account balance and a text box to specify withdrawal amount. A button will verify and withdraw.

2.2.5 The Add Funds interface should display a current account balance and a text box to specify the amount to be added. A button will verify and add..

2.2.6 The Game Interface will have a betting text box, along with a button to verify and place the bet.

2.2.7 The Game Interface will have the player’s cards visible to the player, along with button options for hit/stand/doubledown.

2.2.8 Game will have an exit button to exit the button.

2.2.9 Username will be displayed while the player is in-game.

## Internal Interface Requirements

2.3.1 The system must keep a log of a player’s wins/losses

2.3.2 The cards must be randomized via java’s random class.

2.3.3 The system must check if the player has enough funds to withdraw the specified amount of money

2.3.4 The system must check if the user has enough funds necessary to place the player’s bet

# Non-Functional Requirements

## Security and Privacy Requirements

3.1.1 Email sent to recover account if player forgets password

3.1.2  Players funds will be kept private (other users won’t be able to see it)

3.1.3 Player’s email, age, and banking information will be kept private (other users can’t see it)

3.1.4 Age will be checked before User can create account

3.1.5 System log for admin purposes

3.1.5 The user can have 3 maximum attempts before the account is locked

## Environmental Requirements

3.2.1 System will run on any OS

3.2.2 System just needs java installed

## Performance Requirements

3.3.1 Front-end load time will have a load time no less than 10 seconds when the user accesses the client.

3.3.2 Workload will be able to keep up with 100 users

3.3.3 Platform : will be able to run on windows OS /linux OS /apple OS

3.3.4 Scalability : the blackjack application will support up to 200 players simultaneously