**Blackjack**

**Requirements Document**

*Initialized by: Isabell Ampon, Jian Ting Tan, Zhenwen Wang   
Finalized by: Ivan Peric, Avin Tiletile, Yew Mun Loon*

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** |
| 02/20/2020 | 1.0 | SRS, Update UML | Isabell, Zhen,  Andy |
| 04/10/2020 | 2.0 | Added current member names  to front page, grammar in  3.3. | Avin Tiletile |
| 04/10/2020 | 3.0 | Updated 2.2 for revised  UML. | Avin Tiletile |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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.......................................7

3.3. Internal Interface Requirements.......................................7

**4. NON-FUNCTIONAL EQUIREMENTS...............................................8**

4.1. Security and Privacy Requirements.....................................8

4.2. Environmental Requirements............................................8

4.3. Performance Requirements..............................................8

# **1. Purpose**

This document will cover all necessary information to carry out a successful execution of this project.

## **1.. Scope**

Specifications of how to execute Blackjack will be covered.

**1.2 Definitions, Acronyms, Abbreviations**

Draw – player receives a new card from deck

Bust – player have a hand value of over 21

Hand – the cards the player has

GUI – Graphical User Interface

## **1.3 References**

Minus the ads: <https://www.247blackjack.com/>

## **1.4 Overview**

This project will simulate the game Blackjack. The game

# **2. Overall Description**

## **2.1 Product Perspective**

This version of Blackjack will be a Java-based game with a website for user account management in which multiple users compete with each other.

## **2.2 Product Architecture // WILL BE UPDATED WHEN FINALIZED**

Card Object – Set suit, rank, and value of card.

Deck Object – Create array of 52 for deck, draw card, create new deck,

shuffle cards, display front, and display back.

User Object – Verify login with userID and password.

Dealer Object – Hold dealer ID.

Hand Object – Hold hand with value, draw card, check if BlackJack, check if Bust.

Game Object – Get player, reset game, do countdown.

Player Object – View profile and get balance.

Balance Object – Set balance, save balance.

Rules Database – Decide outcome of game.

Game Server Database – Allow user to sign up or login.

## **2.3 Product Functionality & Features**

* Will work for majority of devices.
* GUI we be smooth & easy to play on.

## **2.4 Constraints**

* Must access game through a Java GUI.
* At least an Intel Duo core processor must be used.
* At least 1mb/s fast Internet connection.

## **2.5 Assumptions and Dependencies**

We assume no more than 6 players at a time. We also assume stable

internet connection. Players will have 10 seconds to decide whether to deal, hit, or stand.

# **3. Specific Requirements**

## **3.1 Functional Requirements**

### **3.1.1** **Log in Requirements**

Users should be able to see an initial menu that has Login (Existing User) or Create an Account (New user). For creating a new account, the username can only contain letters from A-Z and 0-9, which will be taken as a string between 6 to 10 characters in length. New users will be given $2500 for the start of the game and existing users will have whichever amount they had previously. There will be a GUI where you can choose Login or Create an Account. For the login page, it will have you fill out your username & password. For Create Account, it will ask to create a new username, create a new password, and confirm password. Once information is filled, then it will be stored in the Authentication database.

### **3.1.2** **Deck Requirements**

The deck will contain 52 unique cards from 2 ~ 10 along with a Jack, Queen, King, and Ace.

Ace will have the value of 11 if total value of the hand is less

than or equal to 21 and a value of 1 if total value of hand increases to over 21.

Jack, Queen, and King all have a value of 10

Numbered cards will have its value be equal to it’s corresponding name.

Every card will have a front side and a back side

The face down card can only be seen by the user who has the card

The face up card can be seen by every player

### **3.1.3** **Gameplay Requirements // TO BE UPDATED**

The winner is designated to the player who has the highest value compared to the dealer and other players.

The loser will be the player who has a total hand value less than the winner or bust.

A timer for player to decide to play for the round and if time up and player hasn’t decide then default to not a player for that round of the game

Player decide amount of money to wager.

Total amount wager = amount wager of all player.

If win, then player is earned double their wager amount.

If lose, then player loses amount wagered.

If tied, then total amount wager / number of tied // CHECK THIS

Each player is given 2 cards, there will be one card face down and the other card face up.

All additional drawn cards will be face up.

A GUI for deal, hit, stand.

Deal function – player bets amount wagered.

Hit function – player receives 1 card and pick hit or stop

Stand function – player ends wager.

### **3.1.4. Save Requirements:**

Player’s money gets updated after losing or winning and stored in a database.

## **3.2 External Interface Requirements**

Provide GUI of the deal, hit, and stand functions.

In game, there’s a 10 second countdown timer.

GUI contains a main menu to login or create account.

In game representation of how much money you have through chips.

## **3.3 Internal Interface Requirements**

Money is calculated after every game and stored in a database.

Usernames, passwords, and money are all in a database.

All calculations to determine winner for each game is done internally.

# **4. Non-Functional Requirements**

## **4.1 Security and Privacy Requirements**

Any private account information won’t be available to the public.

## **4.2 Environmental Requirements**

Done through a Java GUI and played with a keyboard & mouse.

**4.3 Performance Requirements**

Every move player makes must be done by the CPU within 1 second.