Software Requirements Specification

for

Black Jack

Version 1.0 approved

Prepared by

Atul Jadhav

Lokesh Ramdin

Western Illinois University

03/06/2017

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Project Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Features 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. System Features 3

3.1 Show……………………………………………………………………………………………….3

3.2 Hide 3

3.3 Draw……………………………………………………………………………………….3

3.4 Flop………………………………………………………………………………………..3

3.5 Play new game…………………………………………………………………………….3

4. External Interface Requirements 3

4.1 User Interfaces 3

4.2 Hardware Interfaces 4

4.3 Software Interfaces 4

4.4 Communications Interfaces 5

5. Other Nonfunctional Requirements 5

5.1 Performance Requirements 5

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

6. Other Requirements 5

Appendix A: Glossary……………………………………………………………………………6

Appendix B: Analysis Model……………………………………………………………………6

Usecase diagram…………………………………………………………………..6

Activity diagram…………………………………………………………………..7

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The goal for developing this project is to provide an easy and user friendly GUI for all the players who want to play Black Jack game.

## Document Conventions

* Every requirement especially functional requirement has its own priority regardless of other

stated requirements.

* We have documented this SRS very simply and the other requirements section has been underlined for highlighting them.

## Intended Audience and Reading Suggestions

* This Software Requirements Specifications makes it easy for all the audience to get a clear idea of how to play the Black Jack game and what are the standard rules and regulations.
* First comes the purpose of this project, followed by scope and on what operating environments will it work, the system features, external interface requirements and all other non-functional requirements.
* Every audience can easily understand the project intention if they read the SRS as specified.

## Project Scope

Scope of this project is to build a GUI where the player at the beginner level can play the Black Jack game easily. The Player can select the level from beginner to expert and Play the game.

## References

* <http://www.bicyclecards.com/how-to-play/blackjack/>
* <https://wizardofodds.com/games/blackjack/basics/>

# Overall Description

## Product Perspective

The project is deployed only on a single system. The game has a user-friendly GUI. The game is designed to be easily understood and played by the players who are new to game i.e. for the beginner’s. The number of the players are confined to 2 Players.

## Product Features

Blackjack is fundamentally a two-player game. In blackjack, players don't play against each other. The only competition is the dealer. The aim of the game is to accumulate a higher point total than the dealer, but without going over 21. User Classes and Characteristics

Every player who needs to play this game does not need any registration or login. A player can directly play the game. The player must know the basic terms of the Black Jack game to play the game. The user must have a basic knowledge of English to play.

## Operating Environment

This application can run on any versions of Windows Operating Systems.

## Design and Implementation Constraints

* The project design is simple and user friendly.
* The names of the players’ while being displayed on the screen must be up to 8 characters.
* As we are implementing the code in Java and Using JFrame to develop user friendly GUI, it needs to be run on a platform where there is JRE environment available.

## User Documentation

* Black Jack is a well-known game, so no user documentation is required for this project.
* We have provided a Instructions Tab at the start of the Game which explains the rules of game to new players.

## Assumptions and Dependencies

We mainly assume here that the dealer is the AI agent in the Balck Jack game. Apart from this, any other implementation of this game needs the code to be changed. We also assume that the user is aware of the technical terms used in this game and to have a minimum command on the English language.

## 3. System Features

**3.1 HIT or STAND**

Blackjack is fundamentally a two-player game. In blackjack, players don't play against each other. The only competition is the dealer. The aim of the game is to accumulate a higher point total than the dealer, but without going over 21. You compute your score by adding the values of your individual cards. The cards 2 through 10 have their face value, J, Q, and K are worth 10 points each, and the Ace is worth either 1 or 11 points (player's choice). When the player makes decision to HIT all the available cards value is added/computed and if it is higher than the total of the Dealer and less than 21 then the player wins. The player has an option to STAND which allows the user to take another card from the deck and place it. The player should Just make sure the cumulative doesn’t go over 21 points.

## 3.2 Insurance

When the dealer's face-up card is an ace, each player gets the chance to bet on whether the dealer has a blackjack or not. This is done before any other player actions. The insurance wager equals your original bet and is used to cancel out the likely loss of this bet. A winning insurance bet will be paid at odds of 2:1, and since you lose your original bet, you'll break even on the hand.

**3.3 Surrender**

If you have a bad hand compared to the dealer's hand (judging from what you can see of it,) you can give up the hand and reclaim half your bet. You need a really bad hand match-up for a surrender to be profitable, such as 16 vs the dealer showing a 10.

### 3.4 Splitting

When you get two starting cards of the same face value, you have the option to split the hand in two. You place another bet of the same size as the original bet and play on with two hands. When you've decided to split a hand, the dealer immediately deals a second card to each hand. When you're done splitting, each of your hands will be treated separately, meaning that you will take cards to your first hand until you stand or bust, and then carry on with the next hand.

**3.5 Play new game**

This is also implemented in the form of a button and the players upon clicking the button can start over a new game by exiting of the current game.

# 4. External Interface Requirements

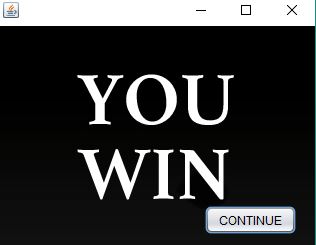
# 4.1 User Interfaces

 **Start the game:**

**Game Play Screen**



**Display of the winning the game output**



## 

## 4.2 Hardware Interfaces

The RAM must satisfy system minimum requirement and same for the hard drive. There must be an Intel Core processor (core i3) required.

## 4.3 Software Interfaces

Windows OS

JRE environment.

## 4.4 Communications Interfaces

As there is no need of any Internet requirements for the game to play. No communication interfaces are required in this project.

# 5. Other Nonfunctional Requirements

# 5.1 Performance Requirements

Minimum software and hardware requirements are enough for the project to run. So, the

performance will be good if the system satisfies minimum requirements.

## 5.2 Safety Requirements

As this is a basic standard game and can be played by the people of ages from 12 above, so any safety requirements are not needed for this project as it doesn’t contain any confidential matters.

## 5.3 Security Requirements

There isn’t any security concern to worry about, while developing this project.

**5.4 Software Quality Attributes**

**Usability:**

The user interface and GUI is designed such that any naïve user will be able operate them and can have access to the game.

**Portability:**

This can be portable to any other systems which are having an JRE environment as we are required to implement the code in JAVA programming language.

**Reliabilty:**

Availability of this game is 24/7 as the user can have access to it any time.

# 6. Other Requirements

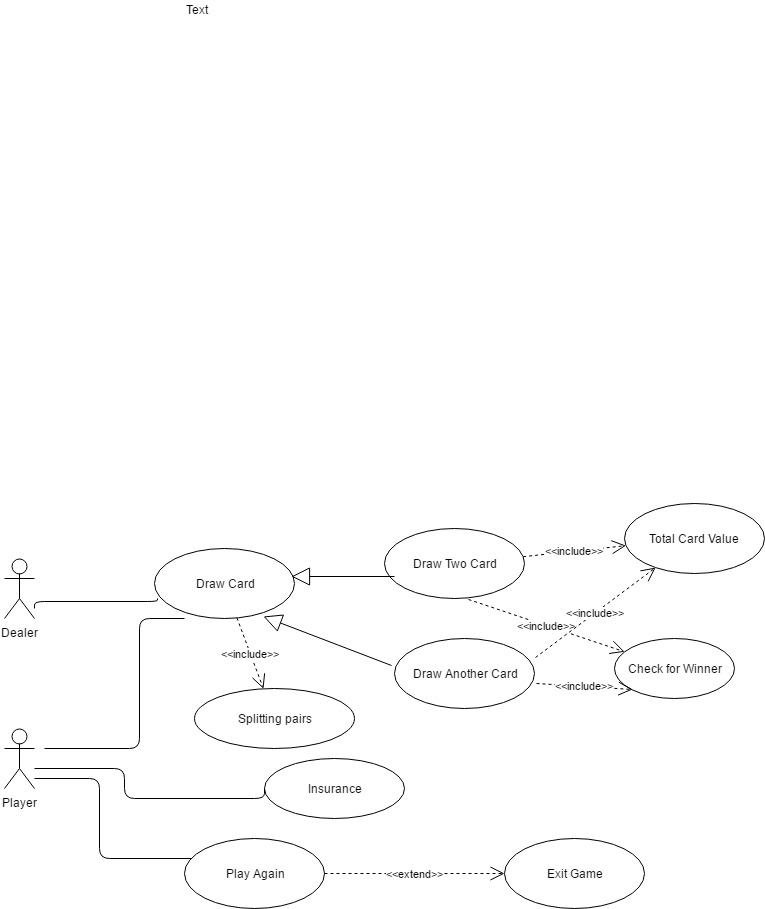
Reusability:

The functional requirements can be reused for other projects of Black Jack world as

They will be same in any Black Jack game but additional features varies from the type of game being designed.

Appendix A: Analysis Models

Use case diagram:



Activity diagram: