CS 401 Project Phase 2

Software Requirements Specification

Isabell Ampon, Jian Ting Tan, Zhenwen Wang

Revision History

Date	Revision	Description	Author
02/20/2020	1.0	SRS, Update UML	Isabell, Zhen, Andy

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

1. Purpose

This document outlines the requirements for our 21 Game. Our game is a twist on the classic card game of BlackJack.

1.1. Scope

This document will show the system and user requirements for the 21 Game.

1.2. Definitions, Acronyms, Abbreviations

Draw – player receives a new card from deck
Bust – player have card total value of over 21 and loses
Hand – the card the player has
Stop and wait – network strategy. Sends first data, wait for acknowledgement and send second data.

1.3. References

https://www.247blackjack.com/

1.4. Overview

This project will simulate a gambling game. Our card game 21 is designed for up to 6 players to play against one another and the dealer. It is based on the classic card game, Blackjack, but with a twist in the different rules. Players will be able to create an account to track their money and winnings so if they decide to play later on, they will have the amount of money from the previous play time.

2. Overall Description

2.1. Product Perspective

An entertainment website in which multiple users compete with each other and whoever obtains the highest total value wins. A thrilling game in which users risk their wager in hopes of doubling their wager or losing it all.

2.2. Product Architecture

- Card Object - makes card class objects, initialize a private integer, cardValue, and a private string variable, suit, and contains method for setting the values of the variables.

- Deck Object - creates an array of 52 card objects, has a draw function to randomly pass out two cards to each player.
- Player Object – creates a player object that handles all user containers.
- Money Object – Contain the Initial amount of chips for each user. Creates an accumulating money object for each user.
- Rule Database – Contains all the rules of the game and decides which user wins.
- Authentication Database - Prompts users to either create a new account or authenticate account, load previous money object if returning player or creates new player object.

2.3. 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):

2.4. Constraints

Access game through a web browser.
Good network Traffic. Implements stop and wait.

2.5. Assumptions and Dependencies

It is assumed that there will be no more than 6 players playing for a round.
Players will have 10 seconds to decide whether to play the round, draw a card, stop drawing a card.

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 (Exisitng 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 $200 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 and 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 cards, there will be card values of 1-10 and string name Ace, King, Queen, Jack.

Ace will have the value of 11 if total value of the hand is less than 21
King, Queen, Jack all have a value of 10
All the other numbered cards will have its value same as their 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:

The winner is defined to the player who has the highest value compared to the dealer and other players or 21.
The loser will be the player who has the total card value less than the winner or bust(above 21).
Play function – player plays for a round
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 – double amount wager
If lose – lose amount wager
If tied – total amount wager / number of tied

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

A GUI for draw, double down, split, stop function
Draw function – player receives 1 card and pick draw or stop
Stop function – player stop drawing and they can't pick draw
Double down- player draw only once and double the wager amount
When bust player lose and auto stop

3.1.4. Save Requirements:

Player's money gets updated after losing or winning
Quite&save function - Player's money gets stored in a database

3.2. External Interface Requirements

Provide GUI of the draw, stop, quit&save, and play.
Countdown timer 10 seconds to make decision
GUI main menu login or create account
A GUI representation of how much money you have through chips

3.3. Internal Interface Requirements

The calculations of money of when you and win
The storing of information username, password, and money in a database
Getting total value of cards
Comparing value of cards with players and dealer that decides who wins or loses

4. Non-Functional Requirements

4.1. Security and Privacy Requirements

Account information won't be share
Changing amount of money in account can't happen outside the game

4.2. Environmental Requirements

Web Browser
Keyboard and Mouse

4.3. Performance Requirements

For GUI functions, it should be able to perform within 3 seconds.