Poker Project Requirements Document

1. **Introduction**

1.1. Purpose

This project requirements document is for our Multiplayer Poker Tournament project. Users will be able to register and create an account with us so that their progress will be saved to the server. Each user will be able to connect to a server room so that they will be able to compete with other users. This Requirements Document will outline all the specifications needed to reach our goal such as Functional requirements, non-functional requirements, and User requirements.

1.2. Scope

The scope for the Multiplayer Poker Tournament is to generate a client-server program that will connect multiple users to play against each other.

This project will be java based that utilizes object-oriented design and will use client-server programming to provide the multiplayer experience. A GUI will be created to provide the user ease of use. A profile registration system will be provided for the user to be able to return to their account with their most current progression. The Server side will store the users information and will also host the games for the users. The server should be able to show the users real-time changes within the game for example, if a player were to raise the bet, all players will be notified immediately after the bet has been raised.

1.3. Definitions, acronyms

Most, if not all of the Definitions that will be provided is based off of basic Poker terminology:

Buy-in: The cost the player must pay to enter a game.

Call: Player will match the bet, and doesn’t raise the bet. Allowing the player to continue.

Check: Player decides to not raise the bet, and continues on with the game. However if there is a bet in place, the user has to *call* or *raise* to continue.

Raise: increasing or placing a bet for the current round, which makes the other players decide if they want to *call*

Fold: Player opting out for the rest of the round.

Dealer: The person that will shuffle the deck and deal the cards to the other player, but for this project the dealer will be automated, and will not be a player within the game.

Hand: This will be referring to the players cards and will consist of only 2 cards

Stake: is a players buy-in amount for a round, *Big-blind* and *small-blind* are forced to put in their amount for the stake.

Big-Blind: Player that will put in the full amount for the *stake*, Each player will take turns being the *big-blind* and it will rotate clockwise after each turn.

Small-Blind: player to the right of the *Big-Blind*, the Small-Blind player will put in half of the stake initially

High Card: If nobody gets a pair, the player with the highest card wins the round.

One Pair: Having two cards of the same rank

Two Pair: Having two different pairs: KK

Three of a kind: Having three cards of the same rank: JJJ

Straight: Having 5 consecutive cards for example: 2, 3, 4, 5, 6

Flush: Having 5 cards that share the same suit, for example: 5 of hearts

Full-House: having a pair of one rank, and a three of a kind: 55QQQ

Four of a Kind: having 4 cards of the same rank: 4444

Straight Flush: Having 5 cards of the same suit, and consecutive too: 2, 3, 4, 5, 6(All Spades)

Royal Flush: Having 10, J, Q, K, A of the same suit. Highest ranking hand.

1.4. References

<https://palapoker.com/poker-terms/>

1.5 Overview

-Functional requirements: The services that the project provides is an easy to use GUI that allows players to register for an account, add funds to their virtual wallet, and connect with other players to play a poker game.

-Non-Functional requirements: Due to constraints, some restrictions for this project is that we can’t fully launch the application to the public due to not being able to constantly run the application on the server side. Another constraint is that creating a profile will not be fully secure as we will be storing the information onto a text file for demonstration purposes.

2. **Overall description**

2.1. Product perspective

The software will be implementing a client-server functionality for multiplayer poker game

The Main Services that is project will prove are the following:

* Account Registration: When there is a new player, the log in menu will prompt the user if they are a returning player, if yes, they could log in with their information. If they’re a new user, they will be redirected to create an account.
* Main Menu: This menu will provide the user with different options to choose from. They have the option to start a game, edit their profile, view their winnings, and their virtual cash.
* During the game: When all the participating players connect to the server, the game will start and each player will wait their turn so they could choose whether they want to Raise, Check, Fold, or Call. The player with the highest ranking hand wins the pot.

2.2. User characteristics

The user is to be expected to know the basic rules of Poker, and be familiar with a simple interface.

3. **Specific requirements**

3.1 User Requirements:

The user should expect ease of use from the GUI. They will be prompted if they would like to register for an account, or if they’re a returning player. The main menu should provide the user with all the information that they will be needing and also choices to accommodate their needs such as starting a game or changing their profile information.

3.2 System Architecture:

The program will be using java programming language, and will be utilizing client-server implementation. A lot of the programming will be object orientated so that we can store as much information about specific things such as the players information, the cards, the deck of cards, the cards, etc.

3.3 System Requirements

The system should be able to receive and send data to the server, as the server will be the backbone of the project. The server will store the users information and will be hosting the games for the players. The client side will contain most of the interface that the user will interact with. The system should also be able to read and write files as this will be our primary source of storing the players information.