SE 3XA3: Software Requirements Specification
PokerProject

Team #12
Safwan Hossain, hossam18
Eamon Earl, earle2
Tyler Magarelli, magarelt

April 13, 2022

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	Version	Notes
Feb 11, 2022	1.0	Initial Draft
April 12, 2022	2.0	Revision 1

1 Introduction

This document describes the requirements for a client-server Poker project that will be programmed using Java.

2 Project Drivers

2.1 The Purpose of the Project

Poker is a world renowned card game in which players must use their understanding of probability, wit and deceptive strategy to best their opponents and win money. While poker is very commonly played face-to-face on a table, the goal of this project is to make a variety of poker game-types easily accessible from anywhere, without betting, and for all ages. The current software we begin with sets the framework for randomizing poker hands and analyzing them, and by expanding upon this foundation we will be able to develop a complete poker experience.

2.2 The Stakeholders

2.2.1 The Developers

As our project will be created in a Java environment, the software developers are provided the correct environment to be able to easily adapt and tailor the software towards a simple yet effective interface with high cohesion and low coupling, while providing the user an easy way to play whenever desired with very little initial setup. The developers are also responsible for testing the program and internally managing the project.

2.2.2 The Customers

The primary stakeholder to consider during the development process is the subgroup that will be directly using and interfacing with the finished product; the customer. Our main goal for this relationship is to cater to them a full poker experience, and an easy way to engage in this neglected sport with no financial cash flow required from them for the purposes of betting. We will also be providing online play, via private online lobbies, so that the customer can play with their friends. Random match-making could be a

later implementation to further improve the user experience, but this goal will lay outside the scope of the project, and will be discussed later on in this document.

2.2.3 Other Stakeholders

For the purpose of this project, the publisher is considered to be the TAs and professor from the Department of Computing and Software at McMaster University, wherein their input will be queried and provided throughout the documentation and development process, and will most notably affect this SRS at revision 1, which will be finalized by April 12th.

2.3 Mandated Constraints

The scope of this project will, to some degree, be inherently described by the constraints, and as such they will be listed and discussed first.

Description: Our software will be able to run on any machine that is capable of running JVM on their system.

Rationale: Any semi-modern local computing device has the capability to run java programs, and as such we will be keeping our demographic of eligible customers as large as possible.

Fit Criterion: Our program will be entirely enclosed within java software and architecture, including the server implementation. Our UI will be text based and represented in the console.

Description: Our program will not involve gambling using real-world currency.

Rationale: Including this feature would conflict with the constraints specified by our publishers, and would greatly extend the scope of this project, to a degree that would require far more resources to complete.

Fit Criterion: Poker chips will be rewarded equally to all players for any game, and will merely be represented in data; we wish to focus on the practical and mental benefits of the game of poker, not the financial deficits.

2.4 Naming Conventions and Terminology

2.4.1 Definitions

- **Chips** - The currency of the game.

- **Pot** - The collective pool of chips that all the players bet on.
- **Hand** - A combination of a player's pair of cards and the cards on a table.
- **Bet** - To increase the amount of chips needed for the round to end.
- **Check** - To not bet without folding.
- **Call** - A bet that is the equal amount to the bet made prior. A player who calls frequently without raising or folding is known as a "calling station".
- **Raise** - To bet higher than an existing bet.
- **Fold** - To forfeit any bets made and surrender your cards within a hand.
- **All-In** - The act of risking all of your chips on one single hand.

2.5 Environment

Above, we see the general architecture of the online aspect of the game, where each user will access a shared server via their respective internet connections. Once again, this server will be low fidelity and will be created entirely within Java.

2.6 Architecture

Model-View-Controller (MVC) architecture will be employed for this project, as it is well geared towards playable games. It also gears itself towards designing for change in the sense of the generality required to implement other game modes in the future; they would use easily use the same model, as they are all played with one deck of cards and one hand per player, and would only require a specific controller to script the game (and possibly some additional methods in the view). This allows future developers to leave the entirety of the data untouched - an exceptionally valuable trait for developing pre-existing software.

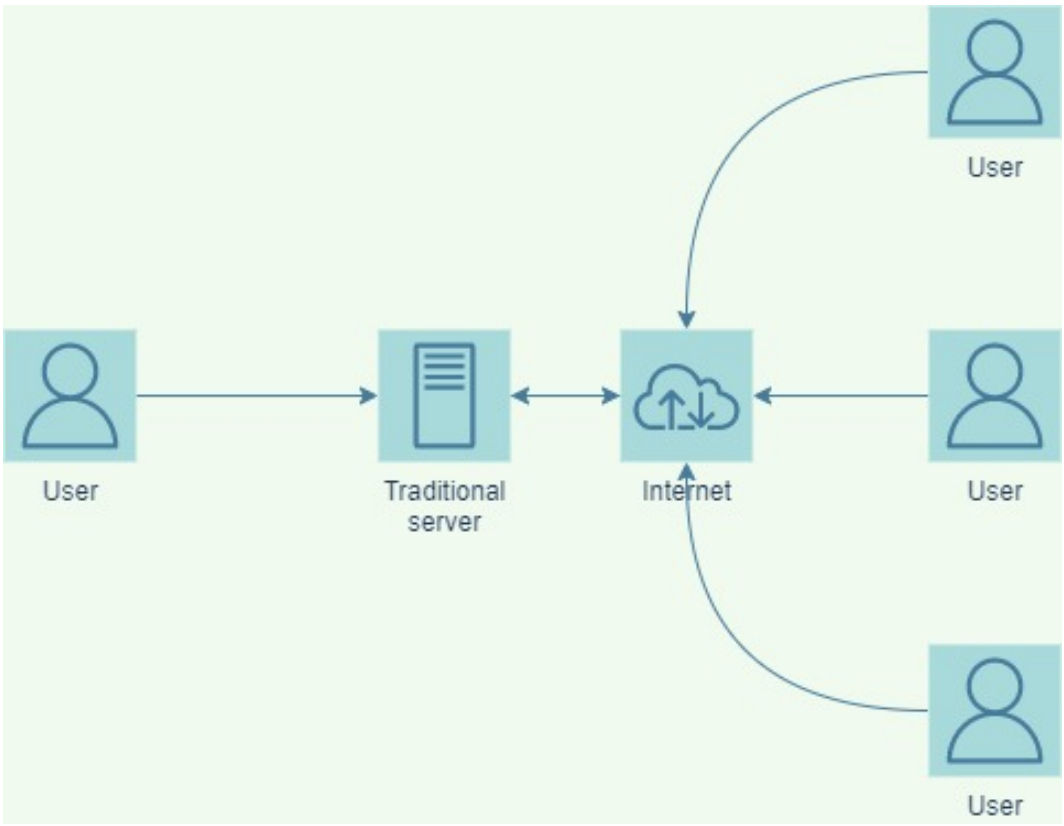

Figure 1: Server Architecture

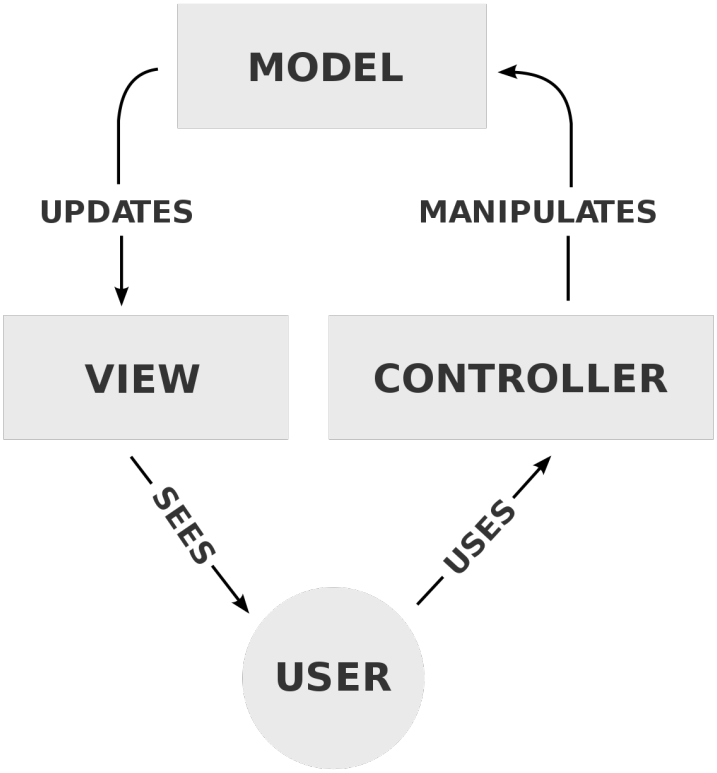

Figure 2: MVC Architecture

3 Functional Requirements

3.1 The Scope of the Work and the Product

This project is contained to being held within java framework, and is limited by the deadlines and relatively tight schedules of the students who take the roles of developers. We will implement one main poker game mode, and do so in such a way that adding more in future iterations of the project would be straightforward, and would not require reconfiguration of the existing architecture, merely additional controllers; thus designing for generality. As we will be implementing "Texas Hold'em" as the initial game mode, the program will be inherently a multiplayer experience, with offline local multiplayer and online multiplayer as well. The servers for online multiplayer will be implemented with low fidelity and bandwidth, as it is known by the project managers that at this point we are not designing for a large player base, and the project is not due to hit markets during the current project life cycle.

As previously discussed, using real currency for the betting process is outside of the scope of this project.

3.2 Functional Requirements

FR1. The user shall be able to generate a unique alphanumeric code that will allow other users to join their lobby.

FR2. The user shall be able to join lobbies by entering in a valid alphanumeric code.

FR3. The user shall be able to perform at most one of the following actions during their turn: fold, check, call, bet, raise.

FR4. The system shall automatically fold any player that does not make a move within the allotted period of time during their turn.

FR5. A game shall have a specified maximum amount of players.

FR6. A game shall have a specified minimum amount of players.

FR7. The system shall be able to identify the correct card ranking for a player's hand.

FR8. The system shall carry a virtual standard deck of cards, consisting of 52 unique cards.

FR9. The system shall shuffle its deck of cards before every game.

FR10. At the start of the game, the system shall give a random player the role of the dealer. Note that this is just a title and the system will deal the cards, not the player.

FR11. Each round, the first player to the immediate left of the dealer shall be given the role of small blind.

FR12. Each round, the first player to the immediate left of the small blind shall be given the role of big blind.

FR13. The system shall take half of a specified amount of money from the small blind and add it to the money pool in the beginning of the round.

FR14. The system shall take a specified amount of money from the big blind as it did from the small blind and add it to the money pool in the beginning of the round.

FR15. The system shall give each player on the table 2 cards in the beginning of the first round.

FR16. The player to the immediate left of the big blind shall be given the first turn.

FR17. The system shall allow at most one user to have a turn any time during a game.

FR18. After a player is done their turn, the player to the immediate left of them shall be given a turn.

FR19. Once every player has either bet the same amount or decided to fold the round ends.

FR20. After the first round, the system shall present three cards from its deck face up to all of the players.

FR21. After the second round, the system shall present one card from its deck face up to all of the players.

FR22. After the third round, the system shall present one card from its deck face up to all of the players.

FR23. After the fourth round, each player shall be forced to present their cards face up to all the other players.

FR24. During the fifth round, the player with the highest hand ranking will win the pot.

FR25. If the highest hand ranking belongs to more than one player, then the pot will be split among the highest ranking players.

FR26. After the fifth round, all cards will be collected and the game will start again from round one.

FR27. The system shall give each player a specified amount of money when they join.

FR28. If a player loses all their money they will be kicked from the table.

FR29. If a player does not have enough money to call a bet they will be allowed to go all in.

FR30. If a player does not have enough money to pay for big or small blind, they will be allowed to go all in.

4 Non-functional Requirements

4.1 Look and Feel Requirements

Because the system will be currently be using the console and optionally in the future use a GUI this requirement is not applicable.

4.2 Usability and Humanity Requirements

4.2.1 Ease of Use Requirements

UH1. The user should be able to perform their desired action in no more than 3 clicks/menus.

4.3 Performance Requirements

4.3.1 Speed and Latency Requirements

PR1. Any interface between a user and the system shall have a maximum response time of X ms.

PR2. The maximum time to boot the game on any system shall be less than t minutes.

4.3.2 Reliability and Availability Requirements

PR1. The system shall run whenever the user desires to open the application.

4.3.3 Robustness or Fault-Tolerance Requirements

PR1. The system shall be designed to operate on a highly robust level.

4.3.4 Capacity Requirements

PR1. Each lobby / local game can have at most 6 individual players.

4.3.5 Scalability or Extensibility Requirements

PR1. The system shall be able to update through software updates that are issued by the developers.

4.3.6 Longevity Requirements

PR1. The product will be developed under the heuristics of designing for change via generality.

4.4 Operational and Environmental Requirements

4.4.1 Expected Physical Environment

OE1. The system will be expected to be installed onto the user's local machine.

4.5 Maintainability and Support Requirements

4.5.1 Supportability Requirements

MS1. The system should be able to be update through software patches issued by the developers.

4.5.2 Adaptability Requirements

MS1. The system should be able to be extended by adding more features by the developers.

4.6 Security Requirements

4.6.1 Access Requirements

SR1. The alphanumeric codes for lobby creation and access will be encrypted / generated by some means such that unwanted players cannot easily join.

4.6.2 Privacy Requirements

SR1. The lobby creators will have the option to kick players out of a lobby before the game commences.

4.7 Cultural Requirements

N/A.

4.8 Legal Requirements

N/A.

4.9 Health and Safety Requirements

N/A.

5 Project Issues

N/A.

5.1 Open Issues

N/A.

5.2 Off-the-Shelf Solutions

Some of these off the shelf solutions for the risks may include:

1. Hostinger - a cheap cloud based virtual server with varying payment plans for different requirements
2. InterServer - a fairly priced server for hosting powerful Java applications (could be more sophisticated than we need)
3. Hostwinds - requires more knowledgeable server developer, but grants more in-depth control and customization, with great service almost internationally

5.3 New Problems

N/A.

5.4 Tasks

To view the tasks and scheduling see our schedule: Gantt Chart

5.5 Migration to the New Product

N/A.

5.6 Risks

Some technical risks that need to be weighed out are the implications of running a low fidelity java server off of one of the developer's local machines. This will be something that needs to be tested during the proof of concept such that we can see if our intended number of concurrent online players is feasible under such conditions, else we may wish to consider some off-the-shelf solutions for our server.

5.7 Costs

There are currently no costs, but once a global server is implemented (waiting room) there will be the cost of keeping the server running through an online service.

5.8 User Documentation and Training

N/A.

5.9 Waiting Room

1. Game Settings - Right now the game has a default setting preset used when running. This means things such as maximum number of players, minimum number of players and maximum turn timer are set to preset values (which is set to its least restrictive form e.g. infinite turn timer). In the future the host will be allowed to set these settings before starting the game

2. GUI - Right now the game uses Java console to display its properties. In the future there will be a GUI for the game that will display the elements of the game visually instead of printing text to a console.

3. Global Server - Currently the game can only connect through LAN servers (every device must be connected to the same network). In the future the game will have the ability to connect through any network using a global network.

4. Spectator - Currently if you lose, you will be kicked out of the game. In the future you will be allowed to spectate the match.

5.10 Ideas for Solutions

N/A.

6 Appendix

6.1 Symbolic Parameters

$X = 1000$ ms

$t = 2$ minutes