Software Requirements Specification

Version 1.1

By: Tomer Shemesh, Kelly Shiptoski, Matthew Damore, Alex Marion

Revision History

Name	Date	Reason For Change	Version
Tomer, Kelly, Matt, Alex	01/13/16	Initial Version	1.0
Tomer, Kelly, Matt, Alex	02/03/16	Revised Version	1.1

Table of Contents:

1 Introduction ……4
1.1 Purpose 4
1.2 Scope 4
1.3 Overview 4
2 Description 4
2.1 Product Perspective 4
2.1.1 Java Client Interface 4
2.2 Product Functions 5
2.3 Requirements Apportioning 5
3 Specific Requirements 5
3.1 Functional Requirements 5
3.1.1 The Desktop Application 5
3.1.2 The Server-Side Application....6
3.2 Non Functional Requirements 6
3.2.1 Performance 7
3.2.2 Scalability 7
3.2.3 Usability 7
3.2.4 Reliability 7
3.2.5 Robustness 7
3.2.6 Portability 7
3.3 Use Case Modeling 8
3.3.1 Use Case Diagram 8
3.3.2 Use Case Specification 8
4 User Interface ……10
4.1 Main Menu 10
4.1.1 Buttons and Controls 10
4.2 Game Play Screen 11
4.2.1 Entering a Game 11
5 System Evolution ……12
6 Glossary ……13

1) Introduction

This section gives a description and overview of everything included in this requirements document.

1.1) Purpose

The purpose of this document is to give a detailed description of the requirements for the our networked chess software. It will show how the system will work, its constraints, and how communication will be made between the client and server. This document will be shown to the person who plans to buy this software, for them to look over and approve to allow the software developers to begin programming.

1.2) Scope

This document describes the software requirements for a networked chess game. This document is intended for the use of the developers to reference while programming and testers to use while seeing what the desired outcome of functions is.

1.3) Overview

This document will discuss information regarding a networked chess game and its requirements.

2) Description

2.1) Product Perspective

This networked chess game is an application intended to allow a user to use a client program to play chess with another random user over the internet, and keep track of their chess statistics over time.

2.1.1) Java Client Interface

The java client will be where the end user does all his interaction with the program. The main screen will be a menu that allows you to start a game, view your statistics, or exit. If you left while previously playing a game you will have an option to

resume the game. In the game you will be able to see your use make a move and then respond with another move until the game ends.

2.2) Product Functions

This chess program will have these functions:

- Setting a nickname that the other user will see
- View all playing statistics
- Play with a random person in chess over the internet
- Exit and resume playing later
- Forfeit a match that isn't going well

2.3) Requirements Apportioning

The priority levels for the requirements are:

Priority Level	Description
1	This is the highest priority level. All requirements with level 1 must be implemented for the program to function properly.
2	This is a lower priority level. These requirements are not required and are nice to have if time permits. These will be added once all priority 1s are complete.

3) Specific Requirements

3.1) Functional requirements

3.1.1) The Desktop Application

R1.1) The Login screen

R1.1.1) If no local user file is defined for the user, then the user should be prompted for a username input to be identified by when playing. **Priority 2**

R1.1.2) Once the user has been set up, the screen should take the user to the main menu for the application. **Priority 2**

R1.2) The Game Main Menu

R1.2.1) Display the main game functions to the user, including: play game, view stats, change username, and exit the game. **Priority 1**

R1.2.2) If there was already a game in progress, the server should alert the client upon login, notifying the user to continue the game or forfeit. **Priority 2**

R1.3) The Play Game Menu

R1.3.1) If the user selects to play a game, the user is taken to the lobby menu to select another user to play against. **Priority 1**

R1.4) The Lobby Menu

R1.4.1) The lobby menu displays all users looking for a new game. **Priority 1**

R1.4.2) The user can send a new game request to another user. **Priority 1**

R1.4.3) If the other user accepts the new game request, the user is taken to the game board screen. **Priority 1**

R1.4.4) If the other user denies the new game request, the user is notified and may send another request to a different player. **Priority 1**

R1.4.5) If another user sends a new game request, the user may accept or deny the request. **Priority 1**

R1.4.6) If the user accepts the game request, the user is taken to the game board screen. **Priority 1**

R1.4.7) If the user denies the game request, the user is directed back to the game lobby. **Priority 1**

R1.4.8) The user may also search the lobby for a specific nickname to send a request to play with their friend if they are also in the lobby. **Priority 2**

R1.5) The Game Board Screen

R1.5.1) The game board screen allows the user to play chess against the player chosen from the lobby menu. **Priority 1**

R1.5.2) When it is the user's turn, they may select a piece. **Priority 1**

R1.5.3) After the user has selected a piece, the user may select a legal move from a list of legal moves displayed on the game board. **Priority 1**

R1.5.4) The user may not select an illegal move as these will not be displayed in the legal moves list. **Priority 1**

R1.5.5) Once the user selects a legal move, they may then either confirm the move, or cancel their movement. **Priority 1**

R1.5.6) If the user cancels their movement, they may select a different legal move to make. **Priority 1**

R1.5.7) If the user confirms their move, the move is sent to the server. **Priority 1**

R1.5.8) When the server receives the user's move, it updates the board of the opponent. **Priority 1**

R1.5.9) The server then updates the opponent's screen, notifying them that it is their turn to make a move. **Priority 1**

3.1.2) The Server-Side Application

R1.1) Data Store

R1.1.1) A SQL table to store a user's information, hiding their ip address and user statistics while displaying their chosen user ID. **Priority 2**

R1.1.2) An additional SQL table to store the current games that are running between players. Storing the game ID, as well as the two user IDs. **Priority 2**

R1.1.3) Requires a constantly open port 80 for rest requests from all clients. **Priority 2**

R1.1.4) Store the current game state and board for each currently active game running on the server. **Priority 2**

3.2) Non-functional requirements

3.2.1) Performance

R1.1) The system will respond within one second of the user pressing a button on the main screen, view stats screen, lobby screen, or game screen. **Priority 1**

R1.2) The system update the game board within two seconds of the opponent submitting their move. **Priority 2**

3.2.2) Scalability

R3.1) Any single chess game will never have to handle more than two players so individual games do not need to be scalable.

R3.2) The system will be able to handle more running games if given more server space.

R3.3) The program will be designed and built in such a way to be easily modified, expanded, or updated. Anticipated changes include support for other languages which would allow users to play other users from different countries. **Priority 2**

3.2.3) Usability

R3.1) This chess game should be easily usable by anyone who knows the rules of the game chess reasonably well.

R3.2) As mentioned in [3.2.1] the system response times will make for a usable chess program.

3.2.4) Reliability

R4.1) The system will have more resources than necessary to be reliable in order to create redundancies for reliability.

R4.2) If run for a year, the system should not have to be down for more than 12 hours in per year.

3.2.5) Robustness

R5.1) The system will not crash when given invalid input and will instead notify the user and prompt for valid input. **Priority 1**

R5.2) The system will notify the user if a server side error occurs and allow them to either quit the application or wait for a server response. **Priority 1**

3.2.6) Portability

R6.1) This software will run in any environment with Java 6 or higher. **Priority 1**

R6.2) An executable can be generated from the software to run or it can be imported into a compatible IDE and run from there. **Priority 1**

3.3) Use Case Modeling

3.3.1) Use Case Diagram

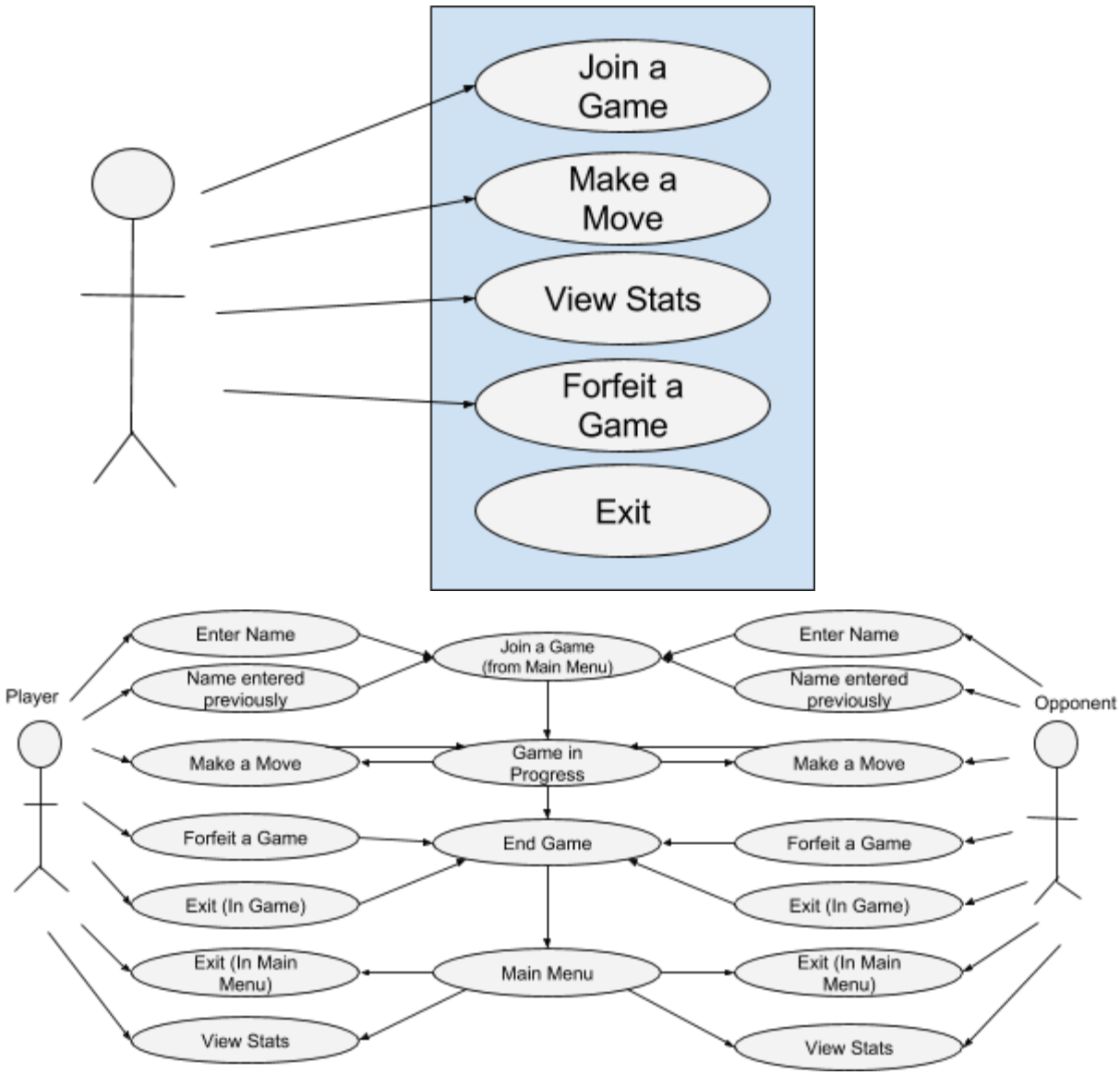

Figure 1: Use case diagram for a user of the chess application

3.3.2) Use Case Specification

R1.1) Join a Game

1.1.1) Preconditions

- The player has launched the chess application and is at the main menu
- The username has been validated

1.1.2) Main Flow

- The player initiates a game with the opponent

1.1.3) Subflows

- The player chooses an opponent from the list of opponents
- The chosen opponent agrees to play the game
- The game begins when a connection between the players is established

1.1.4) Alternative Flows

- The player chooses requests a random opponent

- The random opponent is generated by the game
- The random opponent agrees to play the game
- The game begins when a connection between the players is established

R1.2) Make a Move

1.2.1) Preconditions

- The connection between players was established
- Chessboard is visible in the application on both players' screens, all pieces are in correct positions (original positions if a new game, previous positions if a resumed game)
- It is the turn of the player who is trying to make a move
- A game is in progress

1.2.2) Main Flow

- The player makes a move in the chess game

1.2.3) Subflows

- The player plays a move by clicking on the piece they want to move
- Valid moves are highlighted in yellow, valid attacks are highlighted in red
- Player can click "confirm" to confirm move or "cancel" to cancel move
- Control is given to opponent so they may make their move

1.2.4) Alternative Flows

- The player does not make a move within 120 seconds
- The player is forced to forfeit the game
- Both players are notified and are brought back to the main screen of the game

R1.3) Forfeiting the Game

1.3.1) Preconditions

- The connection between players was established
- Chessboard is visible in the application on both players' screens, all pieces are in correct positions (original positions if a new game, previous positions if a resumed game)
- A game is in progress

1.3.2) Main Flow

- The player chooses to forfeit the game

1.3.3) Subflows

- The player clicks on the "forfeit" button during gameplay
- The connection between players ends
- The players are returned to the main menu
- Game progress is not saved
- The opponent's screen displays a message stating the player has forfeited

1.3.4) Alternative Flows

- None

R1.4) View Stats

1.4.1) Preconditions

- Name has been authenticated
- The player is at the main menu

1.4.2) Main Flow

- The player chooses to view their stats

1.4.3) Subflows

- The player clicks on the “view stats” button in the main menu
- The player is brought to a screen that lists total wins, losses, forfeits, and total time played
- The player can return to the main menu by clicking the “exit” button in the “view stats” menu

1.4.4) Alternative Flows

- None

R1.5) Exit from Main Menu

1.5.1) Preconditions

- The player is running the chess application

1.5.2) Main Flow

- The player wants to exit the application

1.5.3) Subflows

- The player is in the main menu of the application
- The player clicks the “exit” button
- The application closes

1.5.4) Alternative Flows

- None

R1.6) Exit from Game

1.6.1) Preconditions

- Players are in the process of playing a game of chess

1.6.2) Main Flow

- The player chooses to exit the game

1.6.3) Subflows

- The player clicks the “exit” button in the game window
- The player is returned to the main menu
- Game progress is saved
- The opponent is returned to the main menu and notified that the player has exited the game

1.6.4) Alternative Flows

- Either player has the option to resume the game after exiting
- If another game is initiated between the same two players, the initiator is given the option to resume the last game where it left off

4) User Interface

4.1) Main Menu

4.1.1) Buttons and Controls

R1.1) The main menu appears when the user first opens the chess application. It contains the following buttons: new game, view stats, and exit (as shown in figure 2).

R1.2) The “new game” button takes the user to a list of possible opponents so that he can choose who he would like to play (provided his name had been authenticated).

R1.3) The “view stats” button takes the user to a screen so that he may view his total wins, losses, forfeits, and total time played.

R1.4) The “exit” button closes the application. If the user had previously exited a game, the “resume game” dialog would also appear in in the main menu (as shown in figure 3). If it the first time the user has used the application, a dialog will prompt the user to enter a username (as shown in figure 4).

4.2) Game Play Screen

4.2.1) Entering a Game

R1.1) When the user chooses to create a new game, chooses an opponent, and the opponent agrees to play, both players are brought to the game play screen.

R1.2) If the user chooses to resume the game they are brought to the game play screen

R1.3) The game play screen contains the chess board and all chess pieces, information about whose turn it is, and the name of the opponent, and forfeit and exit buttons (as shown in figure 6).

Figure 2: Main Menu

Figure 3: Resume game dialog

Figure 4: Enter username dialog

Figure 5: View stats screen

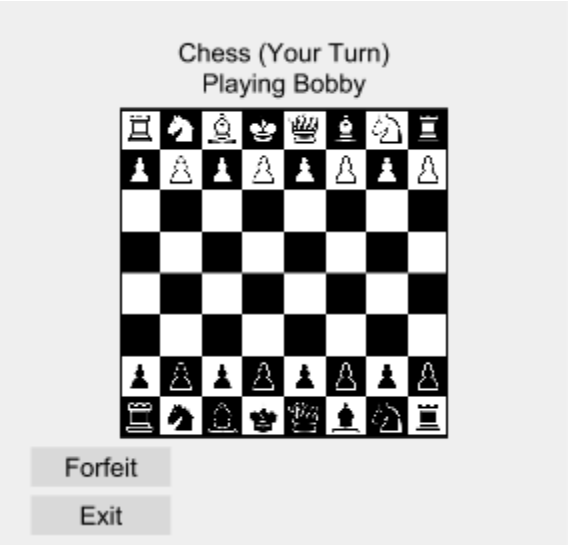

Figure 6: Game play screen

5) System Evolution

The first version of this chess game will be written using the java programming language. It will use Java swing for the front end and a Jersey Server for the backend linked to MYSQL. The end user will need Java 8 in order to run the program. The program will change as time goes on based on user needs and requirements. As the program gains popularity the backend may need to be updated to handle a higher traffic load.

As new functionality and changes are added to the backend and frontend new use cases will need to be added and existing ones modified. Currently this version of chess does not include a way to transfer your profile from computer to computer, so if you change computers you have to start your profile over, but as the system evolves we can add this to the backend so that the user will be able to login from another computer. Maintaining the program will include code changes incase the frontend or backend become broken when the computer or Java is updated.

6) Glossary

Game Play - The in game activity of the chess game

Statistics - The statistics refer to the number of wins, losses, forfeits, and the amount of time played
User ID - The chosen username of a player. This information is stored for further use.