OnlineChess

Software Requirements Specification

SE 210: Software Specification and Design I

Joseph Mulray

3 December, 2016

Table of Contents:

PAGE #

1. Introduction

1.1 Purpose

The purpose of this document is to specify all the requirements for an *OnlineChess* system. These requirements satisfy the performance, capabilities, constraints, system interface, and application structure for a given system.

1.2 Project Scope

OnlineChess is an online chess playing platform that allows chess players to square off in friendly or competitive games through the web browser. The benefits of this is, unlike traditional, chess there are various skill levels and game modes offered to a User in *OnlineChess*. This allows for greater competition and user experience. Its main objective is to connect a User to a server, and with a selected opponent and time range, pair them with someone of their skill level. The goal of *Online Chess* is to offer an interactive, competitive game that offers more features and benefits then your average chess match.

OnlineChess is an online chess game is played between two players, either a computer or another user, on opposite sides of a chess board containing 64 squares of alternating colors. Each player has 16 pieces: 1 king, 1 queen, 2 rooks, 2 bishops, 2 knights, and 8 pawns. A reference to the layout of a chess board is listed below.

The goal of a game of *OnlineChess* is to checkmate the other king. Checkmate happens when the king is in a position to be captured by the other opponent and cannot escape from capture. Note that you cannot put your king in a position to be checked. Play continues until a King is checkmated or a draw occurs. The scenarios of a draw are the following.

1. Stalemate
   The king is not in check but cannot move anywhere, and there are no other pieces left
2. Threefold repetition
   A position has been repeated three times, such as players moving pieces back and forth
3. Mutual Agreement
   A player may offer a draw to his opponent at any stage of a game. If the opponent accepts, the game is a draw.

Upon Starting a game, the user will be prompted to select an opponent and a time range of each turn. Default time range is two minutes. If a computer opponent is selected the User will have the ability to select a difficulty of either Easy, Medium, or Hard. An *OnlineChess* match will then be created. Each player will have the time limit selected to make a move.

OnlineChess will randomly select each player's chess piece colors at the beginning of a match. The player with white chess pieces moves first. If a User is playing against a computer opponent, there is an option to save the current match to play at a later time or date. A user can have up to 3 saved progress matches to which they can play from. Play continues til there is a winner selected, or a User concedes or forfeits the match.

1.3 *OnlineChess* Definitions:

Check: To make a move that puts the opponent's King under direct attack.

Checkmate: a situation in which an opponent's king is in check and it cannot avoid being captured. This then brings the game to a victorious result.

En Passant: a method by which a pawn that is moved two squares can be captured by an opponent's pawn commanding the square that was passed

Stalemate: A situation in which a player's king is not in check, but that player can make no move. This then results is a stalemate, which is a draw.

Promotion: When a pawn reaches the other end of the board it can be changed for any other piece of its own colour, except the King

Castling: is a move in the game of chess involving a player's king and either of the player's original rooks. It is the only move in chess in which a player moves two pieces in the same move, and it is the only move aside from the knight's move where a piece can jump over another.

Reference:

By Using the International Rule: How Many Queens Are in Board? "What Are the Few Rules of Chess Which Are Not Commonly Known?" - *Quora*. N.p., n.d. Web. 09 Oct. 2016.

Blksheep. "How to Play Chess." *Instructables.com*. N.p., 03 Mar. 2008. Web. 07 Oct. 2016.

"Rules of Chess." *Wikipedia*. Wikimedia Foundation, n.d. Web. 07 Oct. 2016.

"Chess.com." *How to Play Chess: Rules & Basics* -. N.p., n.d. Web. 07 Oct. 2016.

"Castling." *Wikipedia*. Wikimedia Foundation, n.d. Web. 09 Oct. 2016.

"The Best Free, Adless Chess Server." *Lichess.org*. N.p., n.d. Web. 09 Oct. 2016.

2. Overall Description

2.1 Product Perspective:

OnlineChess software brings a user friendly chess environment to anyone with internet access. With an established internet connection, *OnlineChess* is a quick, no hassle, free software that eliminates all the hassle of traditional chess into a game open for anyone to try.

2.2 Product Features:

Saved Sessions: Allows user to save current match with a computer opponent to complete, load at a later time.

Cross Browser Support: Offers browser support for the following web browsers: Google Chrome, Safari, Opera and Firefox.

Mobile Integration: Offers support for mobile platform making functionality the same as if playing on a Desktop device.

Move Checker: Shows highlighted list of available moves that a user can select from, saving time and assisting those who are not skilled in a game of chess.

Online Gameplay: Offers the ability to have a human opponent in an online chess match.

Teaching Mode: Offers the ability to learn the game of chess by step by step instructions on the overall goals of chess, and the available moves of a chess piece.

2.3 Operating Environment:

The *OnlineChess* client operating environment should be the web browser. This will allow anyone who visits the web site to play. The end user should be able to use all of the site's features as long as he or she has a modern web browser stated in section 2.2 Product Features. In order to make mobile usage easier, a mobile optimized version of the site should be made. In order to simplify login, OAth should be used to allow users to log in with their Google or Facebook. That way, users do not need to remember their passwords to use the site. The following will provide adequate operating system requirements for an *OnlineChess* system.

- The environment will support the development, test and integration of all components of the application.
- The supported user platforms will be any of the following web browsers: Google Chrome, Safari, Opera and Firefox.
- The application will be tested and be fully functional and compatible across all supported browsers (Google Chrome, Safari, Opera and Firefox).
- Any of the computers can function as web servers for integration testing.
- Source code control will be managed using "Git".
- Problem and issue tracking will be managed using Atlassian Jira.

There are a number of features needed for the server side to make it fit well into the operating environment of the customer. The customer owns several servers, which are used for hosting its sites. Since some of the servers are Windows, and some of them are Linux, the *OnlineChess* server application should be platform independent, written in a language that supports both Windows and Linux operating systems. Otherwise, the customer would be limited in where the server application could be run. Since the customer buying this software uses Oracle databases in many of its business functions, the server side application should be able to use the Oracle database as a backend. This will allow for easier system administration, since the customer already employs Oracle database experts.

2.4 Application Structure:

Application Structure - High Level Arch

The software will be divided into two components: client side and server side. The objective is to facilitate functional communication across clients as to enable a game of chess. The program will enable the creation of games across separate computers as well as against a bot.

Client Side:

The user will interface with a web GUI which interfaces with the program talking to the server hosting the chess program. A home screen will be the face of the program and allow a user to start: an offline game, random online game, request another account to play, or continue existing games. In addition, a user may send a request to replay moves of a past game. Once a game mode or reload request has been processed, the user is directed to a chess board interface. Either chosen or randomly decided, the user will appear in the back of either a set of black or white pieces. The user will have control of piece moves during their respective turns and can assign a preemptive move during their opponent's turn. At any point, a piece will highlight all possible legal moves on the board. Once a game has been decidedly ended or paused, the user is directed back to the home screen.

Server Side:

The server will contact the database at select instances throughout the running of the software. From the home screen the server will aggregate: the current paused games of the user, all past games of the user, the user's rank, and the available players. When a match is requested the server will create a game with the appropriate opponent for the user based on the game type. The server will check for the correctness of moves and determine the state of the game as either: unstarted, active, paused, or ended. After each move a record will be sent and stored in a database. If a request to the home screen has been requested the server will store current values of the game and aggregate all the necessary information for the home screen.

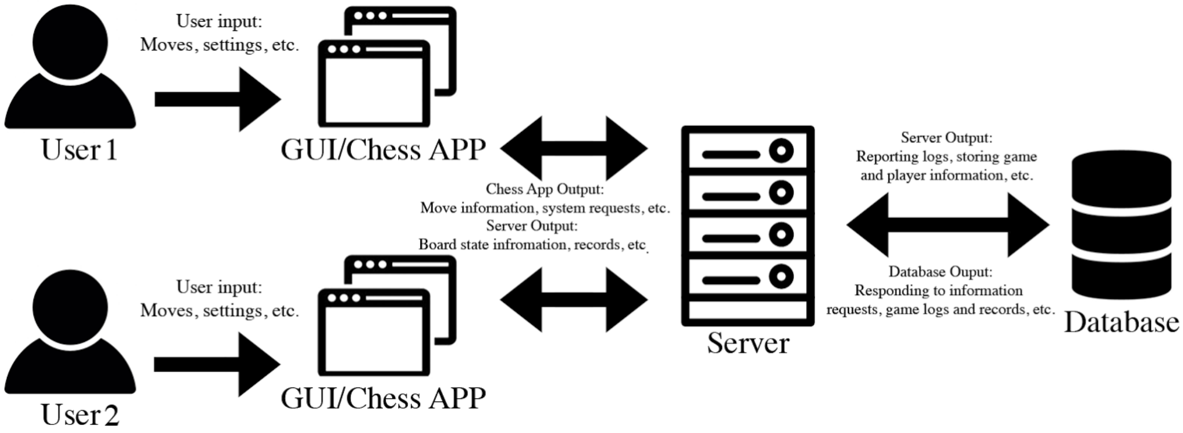
User input:
Moves, settings, etc.
User 1
GUI/Chess APP
Chess App Output:
Move information, system requests, etc.
Server Output:
Board state infromation, records, etc.
Server
Server Output:
Reporting logs, storing game and player information, etc.
Database Ouput:
Responding to information requests, game logs and records, etc.
Database
User input:
Moves, settings, etc.
User 2
GUI/Chess APP

Figure 2.6.1 Interaction between client and server

2.5 Basic Architecture:

The user interface should be intuitive to use and familiar to the average chess player. In order to accomplish this, the client application should use conventions, like moving the white king to G2 to castle kingside. To enhance usability, the user should be able to drag and drop pieces to make moves, and submit moves without having to refresh the web page. A frontend framework like AngularJS or React should be used to accomplish this. There is currently a wide variety of mobile chess apps, so a mobile optimized version of the site should be made. A "mobile optimized" version of the site will make various tradeoffs to increase usability. In a mobile optimized version, the pieces can be controlled with taps rather than dragging and dropping, and in a "mobile optimized" version, the board will take up a greater portion of the screen.

Both the client side and the server side should enforce the standard chess rules on the end users. The client-side safeguards should help the users to know which moves are legal and illegal, and the server-side safeguards should prevent the clients from cheating on each other. The server side will need to handle the player matchup functionality so that, when users are entering into games with non-friends, they are matched with players of similar skill levels. There are a number of different chess ranking systems, but the most commonly used one is the Elo rating system. Chess.com, for example, uses the Elo rating system to set up matches.

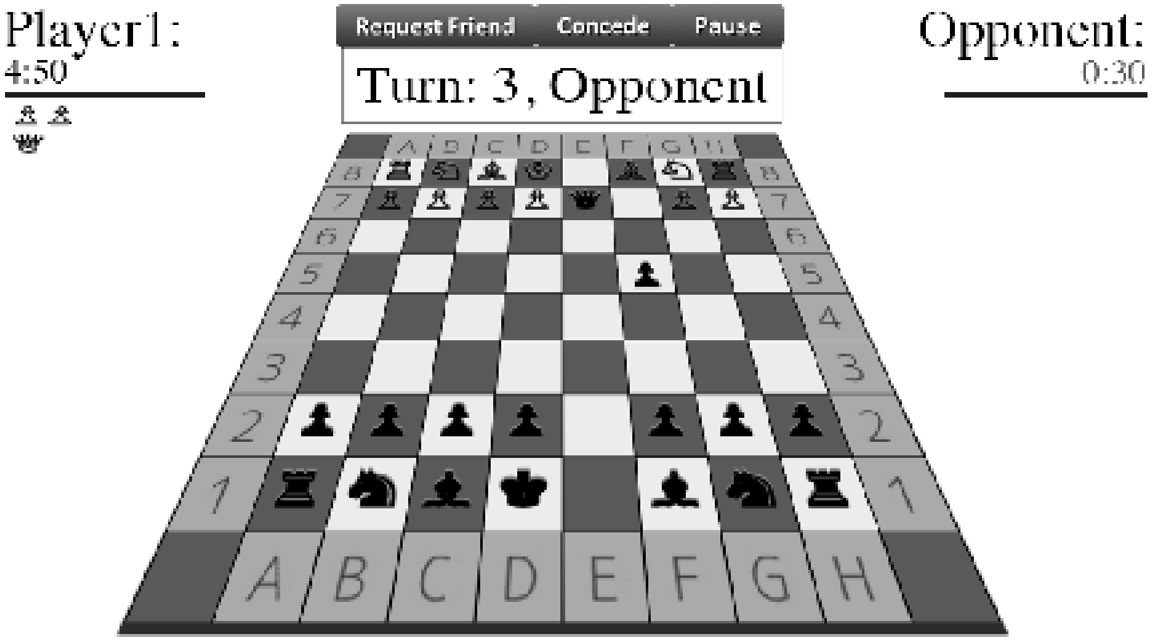

Figure 2.6.2 User Interface for a Online Chess Match

2.6 Usage Scenarios:

Scenario 1: Starting game against computer

Usage details

The following scenario describes a case when a user starts an *OnlineChess* game against a computer opponent and the pre and post conditions needed to satisfy that requirement.

Pre-conditions

- The user of the system has an account
- The user of the system has a rank (default rank is 1200)
- The user of the system has selected a time range
- The user of the system has selected the computer game mode

Post-conditions

- The user of system has a computer opponent set to the appropriate difficulty based on the user's rank

- A new game is logged under the user record

Base Course of Action

- User Logs On to the OnlineChess system with their user account.
- User selects “Start Game vs Computer”
- User’s system rank is retrieved from system.
- New game screen is displayed
- Game is initiated using a difficulty level appropriate for the user.
- System awaits user’s first move (user is white).

Scenario 3: Starting game against random opponent

Usage details

The following scenario describes the case when a user starts an *OnlineChess* match against a random opponent and the pre and post conditions needed to satisfy that requirement.

Pre-conditions

- The user of the system has an account
- The user of the system has a rank (default rank is 1200)
- The user of the system has selected a time range
- The user of the system has selected the random opponent game mode

Post-conditions

- The user of system has a game set against a random opponent with a similar rank
- A new game is logged under the user
- A new game is logged under the random opponent

Base Course of Action

- User Logs On to the OnlineChess system with their user account.
- User selects “Start Game vs Random Opponent”
- User’s system rank is retrieved from system.
- Random opponent is retrieved from the list of online users who have a similar system rank.
- Random opponent receives notification of game invitation.
- Random opponent selects “accept game invitation” to accept the invitation.
- User receives notification of acceptance.
- New game screen is displayed
- System randomly selects random opponent or user as white and black.
- White player is prompted for their first move.

Alternate Course of action

- User Logs On to the OnlineChess system with their user account.
- User selects “Start Game vs Random Opponent”
- User’s system rank is retrieved from system.
- Random opponent is retrieved from the list of online users and with a similar system rank.
- Random opponent receives notification of game invitation.
- Random opponent selects “reject game invitation” to reject the invitation.
- User receives notification of rejection.

2.7 Sequence Diagrams:

The goal of a game in *OnlineChess* is to checkmate the other king, the following sequence diagram illustrates how a match would be declared over.

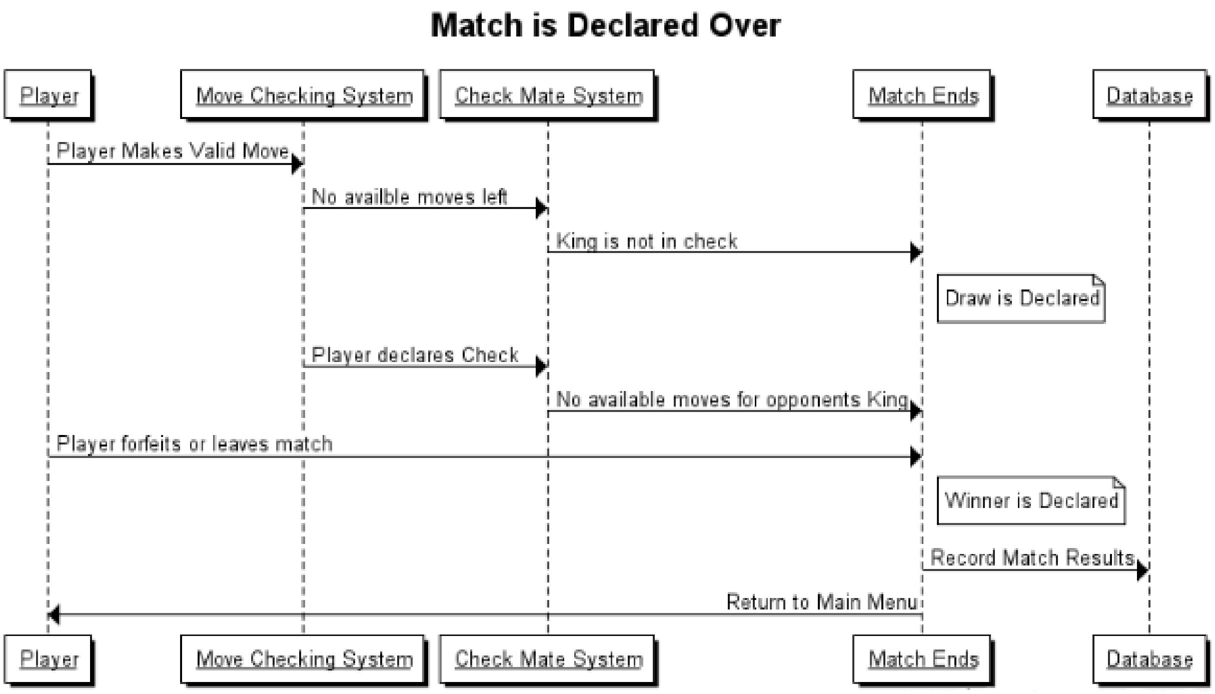

Figure 2.7.1 Match is Declared Over

There are several ways a match can be declared over. The first instance is a draw being declared where a player makes a move and there is no available moves left and the king is also not in check. This will result in a draw recording for the user’s record. A second scenario is a player declaring checkmate where there are no available moves for an opponent's King, and has already declared check. The final scenario is a player forfeits or leaves the match. Each scenario results in a match being declared over, and the results of the winner/loser or draw is recorded in the database.

Starting game with random user:

This Diagram shows the interaction between the users, the pairing system, and the game server.

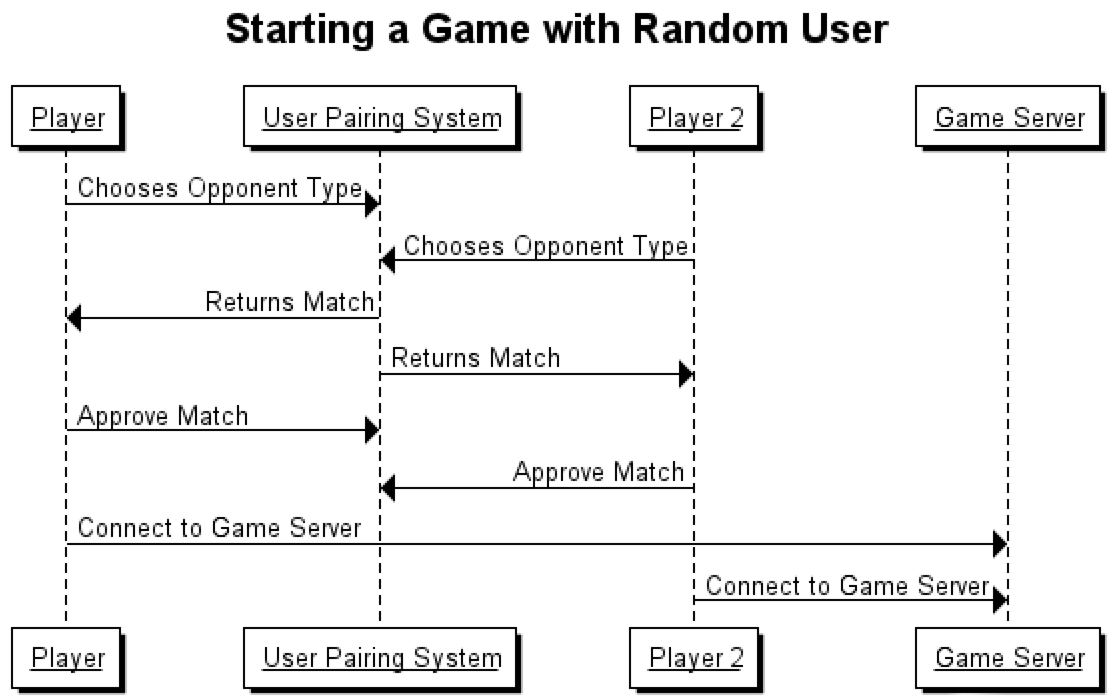

Figure 2.7.2 Starting a Game with a Random User

This system matches two users who are looking for games against random opponents. After one user checks in, the system finds another user who has also checked in, and it returns the matches to both of the users. Once the user's approve the matches, they are connected to the game server, and they can begin their game.

User Playing a Move:

While in-game, the user will move their pieces in order to win the game, the following sequence diagram illustrates a move of a piece.

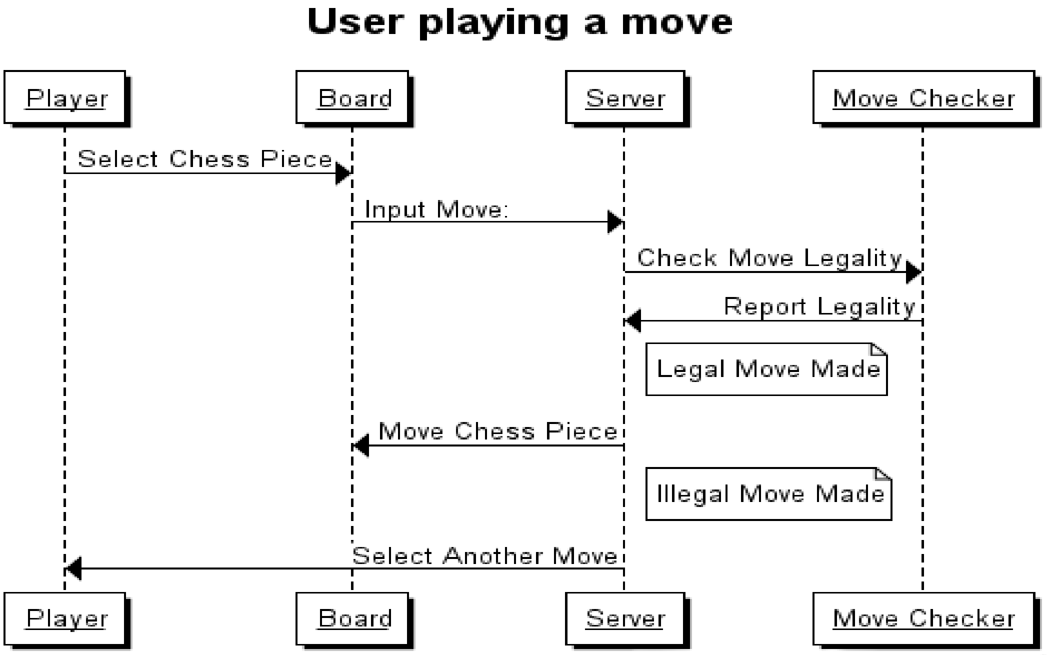

Figure 2.7.3 User Playing a Move

A user will select a piece and then input where the piece will request to move. The server will communicate with the database to check for move legality. If the move reports turns out illegal, the move will not be made. If the move is legal, the server will store the move in the database under the game's record. The move is then made on the board.

3 Specific Requirements:

3.1 Functional Requirements

- Will offer a computer opponent with a ranging skill type, easy, medium, hard.
- A Move Checker System, that checks for validity of moves and positions of chess pieces.
- A User Pairing System that pairs a user with a chosen opponent type.
- When a chess piece is selected, the legal moves will be highlighted that the user can select from.
- Players color pieces are chosen at random.
- User can save and load a previous match against a computer opponent only.
- Captured pieces will display below the user's name.
- Limit how many times player or opponent can request for a draw.
- Quitting or conceding a game will return you to the main menu.

- A Game isn't declared over until a checkmate,draw is declared, or user concedes.
- User is able to set several settings for their game which include: minutes per turn, time per player, opponent type, and or a previous game.
- Board is displayed with all the active pieces in the game in their respective positions

3.2 Non-Functional Requirements

- Amount of games a user can save
- The supported user platforms will be any of the following web browsers: Google Chrome, Safari, Opera and Firefox.
- Both the client side and the server side should enforce the standard chess rules on the end users
- The server side will need to handle the player matchup functionality so that, when users are entering into games with non-friends, they are matched with players of similar skill levels.
- Inability to find a match, would result in playing a user not in that skill range.
- User ability to drag and drop pieces to make moves, and submit moves without having to refresh the web page
- System must reliably handle errors for a given circumstance.

3.3 Security Requirements:

Ensure clients are connected to server, securely. That all data transferred from matches and information regarding accounts is protected. Log files should be pertinent to achieve maximum understanding of occurrences of the system.

3.4 Hardware Requirements:

- Stable internet connection to transfer data, and match User with an opponent.
- Internet Browser to run the *OnlineChess* software
- Mouse or Touchpad to move Chess Pieces on the UI
- Keyboard for interaction with User Interface

3.5 Software Quality Attributes:

Availability: Checking that the system always has a backup server and database so that in case of an error, or system malfunction, no data or functionality to the system should be

Usability: Checking that the *OnlineChess* system is easily navigated and handles performance with no visible delay. Ensure that system is running as smooth as possible.

Functionality: Checking that system provides the correct tools to allow all functionally for a given *OnlineChess* match. In this case ensure that the Checkmate System are tested a proven to work across a given system.