**Jefferson Zhumi**

# OnlineChess

# Software Requirements Specification

**SE210**

**12/10/2017**

# Software Requirements Specification

## For

# OnlineChess

## Requirements for version 1.0

## Prepared by Jefferson

# Table of Contents

# 1. Introduction

## 1.1. Purpose

The purpose of this document is to specify the requirements and preview some elements of the analysis model of the program OnlineChess.

OnlineChess is an online chess application that offers unique features for the users to enjoy the game in many different aspects. The user can choose to login to play or to play as a guest. By default, the player will be randomly matched for the game. However, if the user gets to a certain rank range, she or he can request the game to match with someone who is in the same rank range. Being able to watch a game as a spectator is another feature that our application will support. The interesting part of this application is that users have the option to gamble. Users can enroll and place a bet in a game. In addition, the application provides strategic analysis to improve user's chess skills. The user has the feature to record his or her matches. By watching his or her own game, the user will be able to analyze their games strategically.

More details about application's requirements and features are described in the following sections.

## 1.2. Intended Audience and Reading Suggestions

This document is intended for any individual user,developer,tester,project manager or documentation writer that needs to understand the basic system architecture and its specifications. Here are the potential uses for each one of the reader types:

**Developer:**The developer who wants to read,change,modify or add new requirements into the existing program,must firstly consult this document and update the requirements with appropriate manner so as to not destroy the actual meaning of them and pass the information correctly to the next phases of the development process.

**User**:The user of this program reviews the diagrams and the specifications presented in this document and determines if the software has all the suitable requirements and if the software developer has implemented all of them.

**Tester:**The tester needs this document to validate that the initial requirements of this programs actually corresponds to the executable program correctly.

This document contains the necessary requirement and some aspects of the analysis of the requirements and is organized based on the IEEE Standard for Software Requirements Specification (IEEE 830-1993).

**Overview**

**1.Introduction:**Provide an overview of the application, describe the document structure and point the individual objectives.

**2.Overall Description:**Provide the specification of the system model, the classes model, the main constraints and the list any assumed factors that used within this document.

**3.Specific Requirements :** Provide the analysis of the requirements by feature.

**4.External Interface Requirements:**Provide the visualization of the program and the requirements that are related with hardware, software and networking.

**5.Other Nonfunctional Requirements:**Provide some other constraints that apply to factors such as performance, safety and security.

## 1.3. Definitions, acronyms, and abbreviations

1.3.1. **User/Player**: The person, or persons, who operate or interact directly with the product.

1.3.2. **Customer**: The person, or persons, who pays for the product and usually (but not necessarily) decide the requirements.

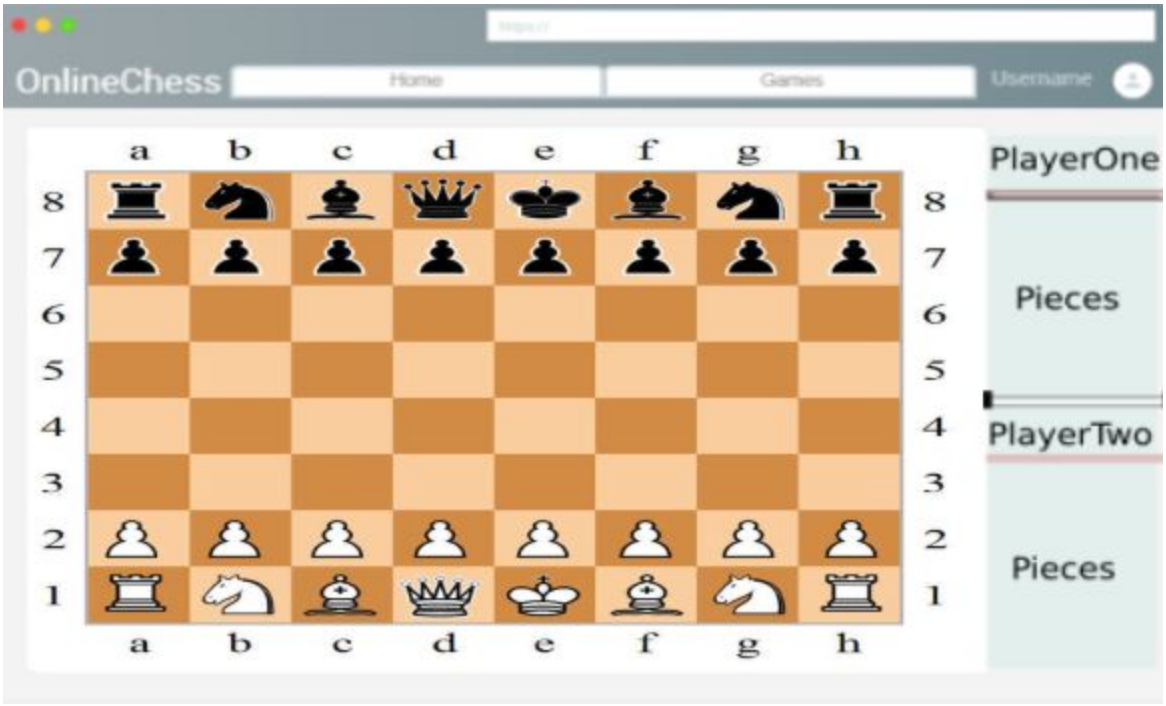
## 1.4. Project Scope

This web application OnlineChess offers a quality, portable, and friendly-user online experience for chess lovers.

Its main object is to differentiate itself from other online chess web applications by providing features that other applications do not.

The goal is to provide the user with a unique chess experience.

Here is a screenshot of the gameplay application:




## 1.5. References

1.5.1. Lamsweerde, A. van. *Requirements Engineering: from System Goals to UML Models to Software Specifications*. John Wiley, 2009.

1.5.2. *Kim Chan. "Writing a Software Requirements Specification Document." OneDesk, OneDesk Software Inc., 9 Nov. 2016, www.onedesk.com/writing-a-software-requirements-specification-document/.*

1.5.3. *"Minimum and Recommended System Requirements." Riot Games Support, support.riotgames.com/hc/en-us/articles/201752654-Minimum-and-Recommended-System-Requirements.*

1.5.4. *"Get Started Developing Web Applications in Linux." Linux.com | The Source for Linux Information, LINUX.COM, 12 June 2007, www.linux.com/news/get-started-developing-web-applications-linux.*

1.5.5. http://www.ics.uci.edu/~andre/ics228s2006/dartellisonfeilerhabermann.pdf

1.5.6. https://www.cde.ca.gov/re/di/ws/appdevstandards.asp

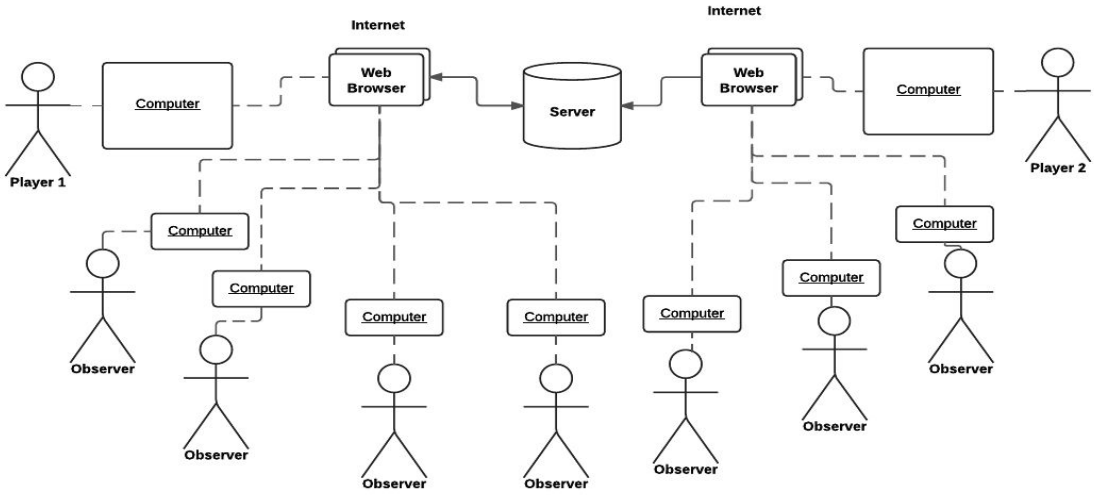
1.5.7. https://lichess.org/
1.5.8. http://dau.dodlive.mil/2015/09/21/requirements-engineering-in-an-agile-software-development-environment/
1.5.9. http://www.modernanalyst.com/Careers/InterviewQuestions/tabid/128/ID/804/What-is-the-difference-between-a-use-case-alternative-flow-and-an-exception-flow.asp
1.5.10. https://www.fluidui.com/
1.5.11. http://sequencediagram.org/

## 2. Overall Description

### 2.1. Product Perspective

This online chess web application game is available to anyone that has an interest for chess. There are various reasons why should anyone use this program. First it's accessible everywhere in the world as long there is an internet connection. Second it provides a better user experience as it's more friendly and easily accessible unlike the traditional chess platform. And third it provides users to gamble. This feature allows users to showcase their chess skills to a different level.

**High level Architecture:**



### 2.2. Product Features

The major features this web application contains are the following:

- **Language support**: Offers multiple language support for global use
- **Betting system**: Offers system to gamble
- **Spectator view**: Spectator can view current matches
- **Record match**: Users are able to record their gameplay
- **View saved matches**: Users are able to access their gameplay recordings
- **Ranking system**: Offers a chess ranking system
- **Communication system**: Users are able to chat during gameplay
- **Auto-saving feature**: If the player loses internet connection, one can rejoin the game which was saved by auto-saving feature.
- **Report other players**: Users can report other players in a gameplay by filling out a complaint form

### 2.3. User Classes and Characteristics

**User/Player:** The User/Player that logs in or enter as guest in the Onlinechess and plays a game of chess.

**Client:** The client is the system that connects to the server and handles the tests based on the session and finally submits the information back to the server.

**Server:** The server is the system that accepts multiple connections from clients and saves the results.

**Database:** The database used to construct, modify, and save each move. The database used to construct, modify, and save each recorded match.

### 2.4. Operating Environment

This program will operate in the following operating environment for the client and the server GUI:

- Apple Mac OS X
- Linux/Unix
- Microsoft Windows

### 2.5. Development Environment Requirements

- **For the client side**
  - HTML5
  - CSS (Cascading Style Sheets)
  - JavaScript
- **For the server side:**
  - Java version 7 and above

- Microsoft ASP.Net 4.5
- Microsoft Visual Studio 2012 development environment
- Microsoft Visual Basic 2015 server-side programming language
- Microsoft IIS 8.0 Web server software
- Linux, Apache, MySQL and PHP, Perl or Python (LAMP) platform

- **For the testing:**
  - Firefox version 56.0a1 or higher
  - Google chrome version 61.0.3163 or higher
  - Safari version 11.0 or higher
  - Internet explorer 11.0 or higher
  - Opera version 48.0.2685.35 or higher

### 2.6. Constraints

#### 2.6.1. The application is web based and it can be run on any operating system that supports:

2.6.1.1. Firefox version 56.0a1 or higher
2.6.1.2. Google chrome version 61.0.3163 or higher
2.6.1.3. Safari version 11.0 or higher
2.6.1.4. Internet explorer 11.0 or higher
2.6.1.5. Opera version 48.0.2685.35 or higher

#### 2.6.2. Minimum Hardware specs:

2.6.2.1. Processor: 2.8 GHz or faster
2.6.2.2. Memory: 1.0 GB RAM
2.6.2.3. Graphics: 256 MB
2.6.2.4. DirectX Version: DirectX 9.0c

2.6.3. **The user needs to be connected to the internet.**

## 3. Specific Requirements

3.1. Section A (Use Cases)

**Use Case 1**: The users create accounts

**Pre-conditions**

- Users need to have a birthday
- Users must have unique username & password
- User has to have internet-connection
- The system has to have form which has requirement fields and non-requirements field for user to fill out

**Postconditions**

- System will notify the user that the account has been created
- Users are able to see their account username and chess rank
- System will ask the user to confirm the provided email that was used to register the account

**Base course of actions**

1. User clicks on the “create account” button
2. A system will display basic information such as name, birthday, username, password and gender.
3. User enters her/his information and system verifies for repeated data and correct format
4. System will confirm and store the data and allow the user to continue
5. Once the user clicks continue, the system will notify the user that the account has been created successfully

**Alternate actions**

- When the user does not enter a correct password based on the chess application password policy
    1. User will enter a password
    2. System will notify the user that the password does not meet the password policy
    3. User will enter the correct and appropriate details and click continue
    4. System will store the request and verify that the password meets our policy requirements
    5. System will allow the user to continue if password policy is met
    6. System stores the verified password to database

**Use Case 2**:Edit existing user information

**Precondition:**

- The user’s account has to exist on the database
- The system has to be able to retrieve data from the database
- The system has to have button which starts editing the account
- The system has to have page where summarizes user’s account to information

**Postcondition:**

- The system updates user’s information on ‘manage account’ page
- The system saves the changed information to the database
- The system displays alert saying that one’s information is successfully updated

**Base course of action:**

1. The user login to one’s account with right ID and password

2. The user goes to ‘manage account’ page and clicks ‘edit’ button on the page
3. The system lead the user where the user put the password again to verify the user one more time
4. The user puts right password
5. The system retrieve the information and displays old information the user filled out before (includes the two button at the end of the page: cancel and save changes)
6. The user changes required fields information, add non-required field information or both
7. The user clicks save changes
8. The system displays alert saying that the user’s account is successfully update (include two options: go back and continue)
9. The user clicks go back option
10. The system brings the user back to ‘manage account’ page
11. The system updates user’s account information and save to the database

**Alternate actions**

- The user cancel out from the update window
    1. If the user changed any field, the system asks whether the user wants to save the changes or not (include two buttons: save and discard)
    2. If the user clicks save button, go back to step 8
    3. If the user clicks discard button, the system brings the user back to one’s account summary page and send the original information back to database

- The user get wrong with one’s password on step 3 **(exception)**
    1. The system alerts that user’s password is wrong
    2. If the user keeps getting wrong up within 5 attempts, the system brings the user back to ‘manage account’ page

**Use Case 3**: Access and play recorded games
**Pre-condition**:

- User must be have an account
- User must be logged in
- User must have played a game before

**Postcondition**:

- Counter increasing showing how many times the match has been viewed by the user

**Base course of action**

- User wants to watch a game they played previously
    1. An user logs into the website with their account

2. From the main menu they chose to watch a game they played in a previous time
3. After clicking on watch reply they will be lead to a page where they can choose which game to watch
4. The user selects the game from a list of games
5. They click watch reply
6. The user clicks next to see the moves one by one
7. Counter at the top of the screen will indicate which move the game is on and how many are left
8. The user clicks back to rewatch a move
9. The user reaches the end of the match
10. A popup opens with options to rewatch the match, choose a new match to watch, or go back to main menu
11. The user goes back to the main menu

**Alternate Actions**

- User wants to check if they played a match against a user
    1. A user wants to see if they played a match against a certain user
    2. They go to the page with the archived matches
    3. Using the search feature they enter in the opponent's name
    4. They click search
    5. All of the matches with the opponent is displayed
    6. The user then goes back to the main menu

**Use Case 4**: The user places a bet

**Pre-conditions**

- Both users need to have an existing account
- Both users need to have an existing balance of $x > 0
- Minimum of two players enrolled in the bet
- User must know password to very account identity for security purposes

**Postconditions**

- Players are enrolled in game
- System needs to have successful allocated money to bet field
- System needs to subtract wager from current balance
- System allocates money to winner player and updates balance

**Base course of actions**

1. User will click "bet" button and a bet placement screen will display
2. Bet placement screen lists add amount in dollars
3. System will give the user the option to bet in amount of one dollar up to five dollars
4. User enters amount to complete the bet
5. System will notify both players once bets have been placed
6. System stores data and updates balance

7. Game begins

**Alternate actions**

- When either player does not an existing balance
    1. The system will notify the user that the account has no balance
    2. The system list option to add balance or exit the placement bet screen
    3. User enters amount to complete the request
    4. System will display another screen to confirm user identity and link to bank account credentials
    5. User confirms amount to be transferred from bank account to application
    6. System will notify user that the request is complete

**User Case 5:** Start and play a game
**Pre-condition:**

- An account (optional)
- Two players

**Postcondition**:

- Match logged to the user's account
- A reply added to both of the user's archives

**Base course of action**

- User has an account and plays a game using their account
    1. A user logs in with his credentials
    2. From the match options the user selects play an unranked match
    3. After a few seconds of waiting an opponent joins
    4. The player with the white pieces makes the first move
    5. Both player plays until the end of the match
    6. The user check's the opponent and a popup displays showing the matches results
    7. Option is given to play again against the same opponent and to go back to main menu
    8. The user does not want to play again so they go back to the main menu

**Alternate actions**

- The user does not have an account and plays with a guest account
    1. The user selects play unranked match
    2. After a few seconds of waiting an opponent joins
    3. The player with the white pieces makes the first move
    4. Both player plays until the end of the match
    5. The user check's the opponent and a popup displays showing the matches results

6. Option is given to play again against the same opponent and to go back to main menu
7. The user wants to play again, and so does the opponent
8. Both player plays the match again
9. Seeing the match is a losing match, the user closes his browser
10. The match is over

**User Case 6**: User exits the ongoing game

**Pre-condition**

- The user has to be in game with other player
- The system has to have a functioning exit button
- The user has to be using compatible browser

**Post condition:**

- The system closes game window
- The user's rank is updated based on the user's record
- The user's record (win, lose, tied) changed

**Base course of action:**

- The user wants to exit the ongoing game by clicking the exit button
    1. The user clicks the button to exit the game
    2. The system displays the small alert asking the user that the game will not be saved and the user will lose the game and whether the user still wants to exit the game. (include system displays two buttons: yes and cancel.)
    3. The user chooses 'yes'
    4. The system closes the game window
    5. The system brings the user back to the page where the user can create or join the game.
    6. The system marks one lost on the user's record
    7. The system updates the user's rank based on the record

**Alternate actions**:

- User exits the game the ongoing game by closing the game window
    1. The user clicks the 'x' on one's browser
    2. The system displays alert asking the user that the game will not be saved and the user will lose the game and whether the user still wants to exit the game.
    3. The user choose 'yes'
    4. The system closes the game window
    5. The system brings the user back to the page where the user can create or join the game
    6. The system marks one lost on the user's record
    7. The system updates the user's rank based on the record

- User exits the game unintentionally due to bad internet connection or power down of the computer **(exception)**
    - If the user is not logged in, the user will lose data and the user on the other side will win.
    - If the user is logged-in
        1. The system holds the game for 5 minutes
        2. The system give the user on the other side notification saying that the user is temporarily disconnected (includes the system list this game to the "game to join" list for the user who exits accidently)
        3. If the user comes back and join the game before 5 minutes pass, the game resumes
        4. If the user doesn't come back within 5 minutes, the system notifies the other user that one won the game
        5. The system deletes the game from the "game to join" list
        6. The system marks the user's lost record

**User Case 7:** User watches the other player's game
**Pre-condition:**

- User has to be logged-in
- User has to have internet connection
- User has to have compatible browsers
- The system provide the list of ongoing game the users can watch
- The system provide the small side tab for 'watch game'
- The system provide a section which lists all ongoing game on the web-page

**Post condition:**

- The game appears on user's 'watch games' tab

**Base course of action:**

1. The user clicks the tab called 'ongoing game'
2. System lists the ongoing games in order of remaining play time
3. User clicks the watch icon on the game thread
4. The game that the user clicked appears on the 'watch game' tab

**Alternate actions:**

- The user wants to watch the game full screen
    1. The user clicks the tab called 'ongoing game'
    2. System lists the going game in order of remaining play time
    3. User clicks the watch icon on the game thread
    4. The game that user clicked appears on the 'watch game' tab
    5. The user click the game one wants to watch in full screen
    6. The system displays two options: close and full screen
    7. The user clicks the full screen option

8. The system opens the window for full screen game

- If there is no ongoing game (**exception)**
    1. The system will display “no ongoing game” on the list block

**User Case 8**: Move around pieces
**Pre-condition**:

- A chess piece must be selected
- The move must be a valid move
- It has to be the player's turn
- When castling no piece may be between the rook and king
- The king can not be in a check position if the user wants to castle
- When castling the king and rook must not have made a previous move

**Postcondition**:

- Piece moved to another location
- The move being logged and displayed
- Error message popping up if the move is an invalid move

**Base course of action**:

- User moves a piece
    1. The user selects which piece to move
    2. The user clicks on the piece and chooses the new location the piece to be at
    3. The movement is verified and the piece is moved
    4. If a piece is captured the piece is displayed on a side of the board permanently on the screen
    5. Game continues until the board reaches a state where a player wins or a draw occurs

**Alternate actions**:

- User is in check
    1. The opponent puts the user in check
    2. Every move that does not put the player out of check is seen as an invalid move
- User castles
    1. The user selects the king
    2. The king is moved to the king’s side
    3. The king moves two spaces
    4. The rook is moved past the king

**User Case 9** : The users and players send messages
**Pre-conditions**:

- Game between players exist
- User chat availability is set on
- 150 max characters

**Postconditions**:

- System must allow both players to exchange text messages
- When the game finishes the system must close the chats

**Base course of actions**:

- When the user sends a text message
    1. User clicks chat box
    2. The system will notify users to begin chatting
    3. The system will display max character available
    4. Users will be able to send text messages up to character limit
    5. User will close window when the game is finished

**Alternate actions**:

- When player one or player two availability chat is set off
    1. Either player request to turn chat availability on
    2. System will notify the the player of this request
    3. Player will have the option to accept or reject
    4. If the player clicks accept, then both users will be able to chat
    5. If the player clicks reject, the system will notify the requester that the player does not want to chat

**User Case 10**: Users have time limit to complete a game

**Pre-condition**

- A game must exist between two players
- Set time limit needed for each game
- Standard time limit of five minutes set for each game
- One minute countdown starts after both players enter game to allow preparation
- “End Turn” button for each player to stop timer and end their turn

**Postcondition**

- Game timer starts after one minute countdown ends
- Timer counts down for one player and ends once that player clicks “End Turn” button, turning on  counter of opposing player

**Base course of action:**

1. Two players search for and enter a new game from the lobby
2. System prompts both players that the game starts in one minute.
3. First player makes a move and stops their timer once “End Turn” button is pressed
4. Second player’s timer begins to countdown and stops once a move has been set and “End Turn” has been pressed
5. Game ends once one of the player’s timer is 0:00 or once a player wins or forfeits the game

**Alternate course of action:**

1. Player filters games by specified time limit in game lobby
2. System shows games with specified time limit
3. Player joins selected game

4. Start countdown begins once both players enter game
5. Game begins after countdown ends, game timer with specified time limit begins for first player
6. Timer alternates between players after each player presses “End Turn” button
7. Game ends once a player’s timer reaches 0:00 or a player wins or forfeits the game

## 3.2. Section B (Sequence Diagrams)

Below we can find the sequence diagrams for user create account case, replay menu case, bet placement case,  and move chess piece case.

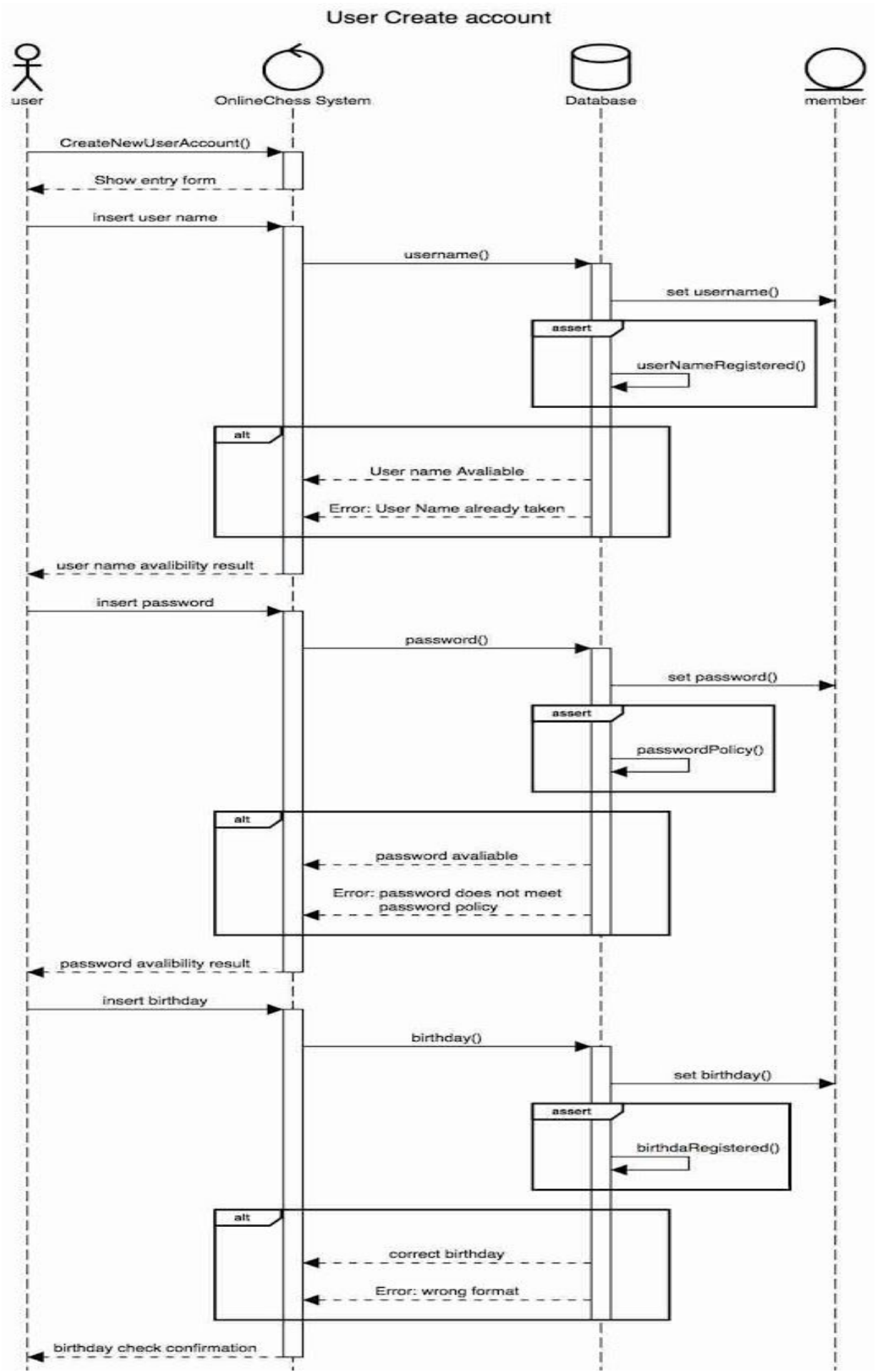

User Create account
user
OnlineChess System
Database
member
CreateNewUserAccount()
Show entry form
insert user name
username()
set username()
assert
userNameRegistered()
alt
User name Avaliable
Error: User Name already taken
user name availibility result
insert password
password()
set password()
assert
passwordPolicy()
alt
password avaliable
Error: password does not meet password policy
password availibility result
insert birthday
birthday()
set birthday()
assert
birthdaRegistered()
alt
correct birthday
Error: wrong format
birthday check confirmation

*Figure1 above. User create account*

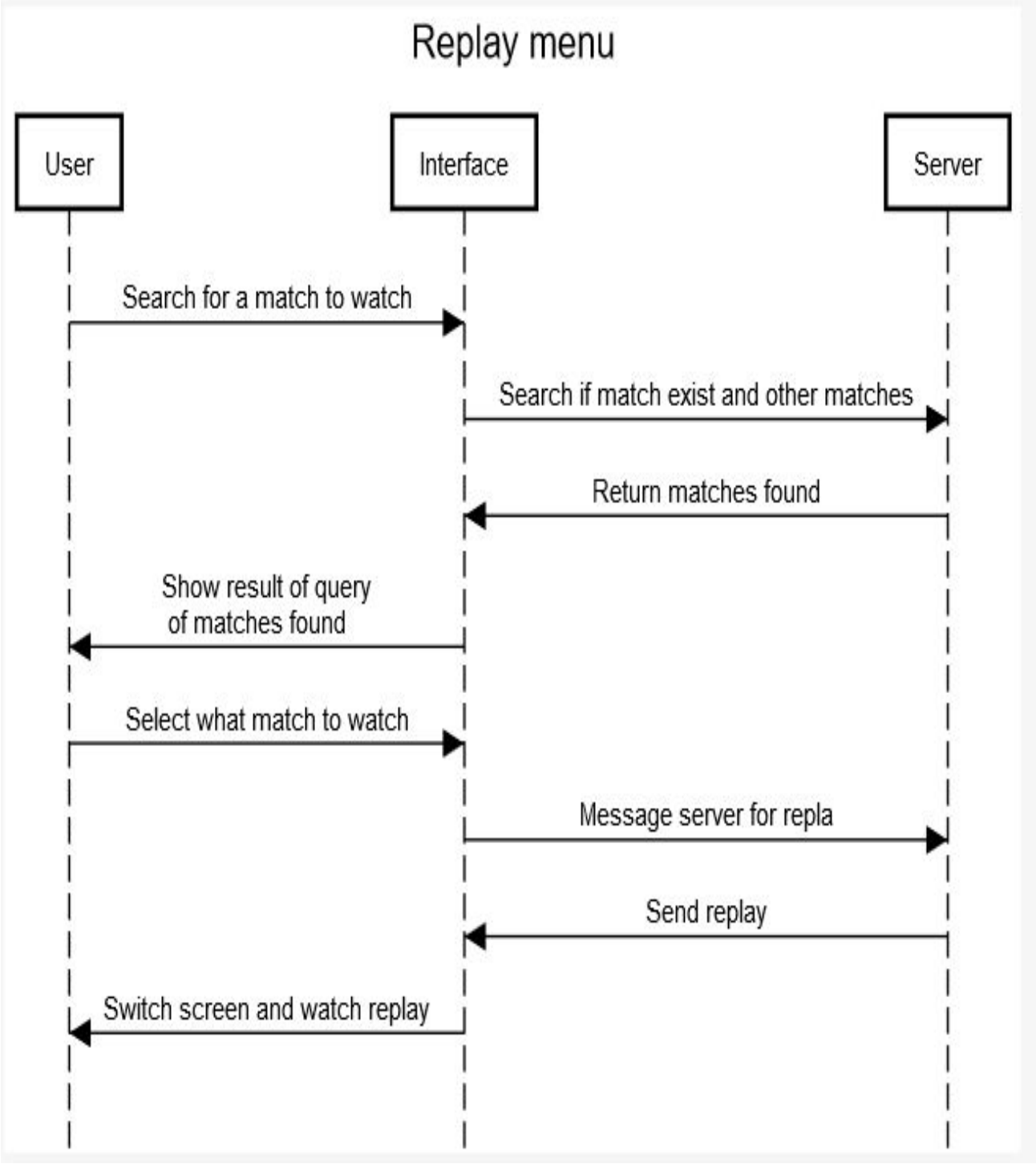



*Figure2. Sequence Diagram for replay menu*

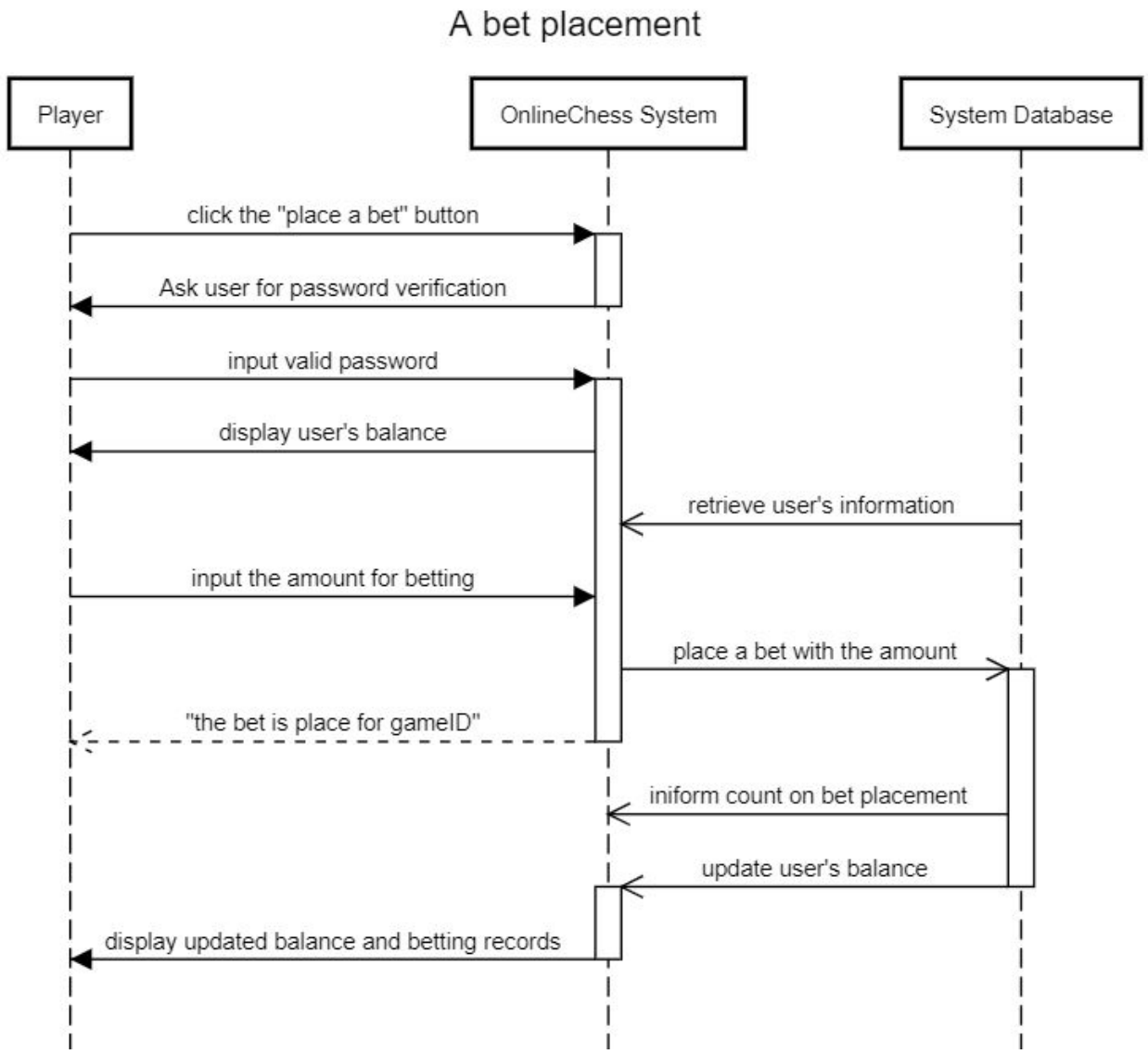



*Figure3. Sequence Diagram for a betting system*

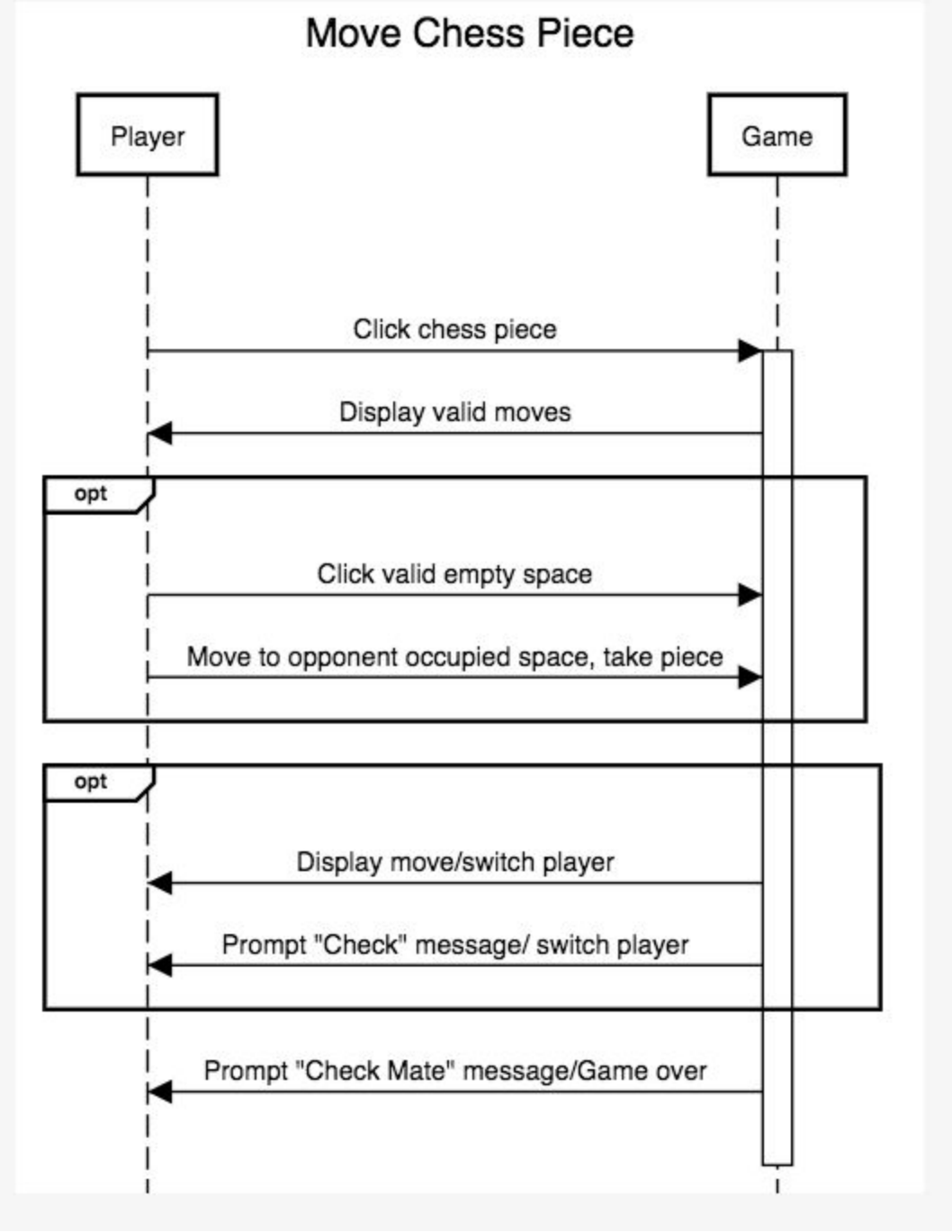



*Figure4. Sequence Diagram for moving a chess piece*

## 4. External Interface requirements

### 4.1. User Interfaces

Below we will find user interfaces for login page, game main page, bet placement page, game platform, user account summary page, replay page1 and replay page2.

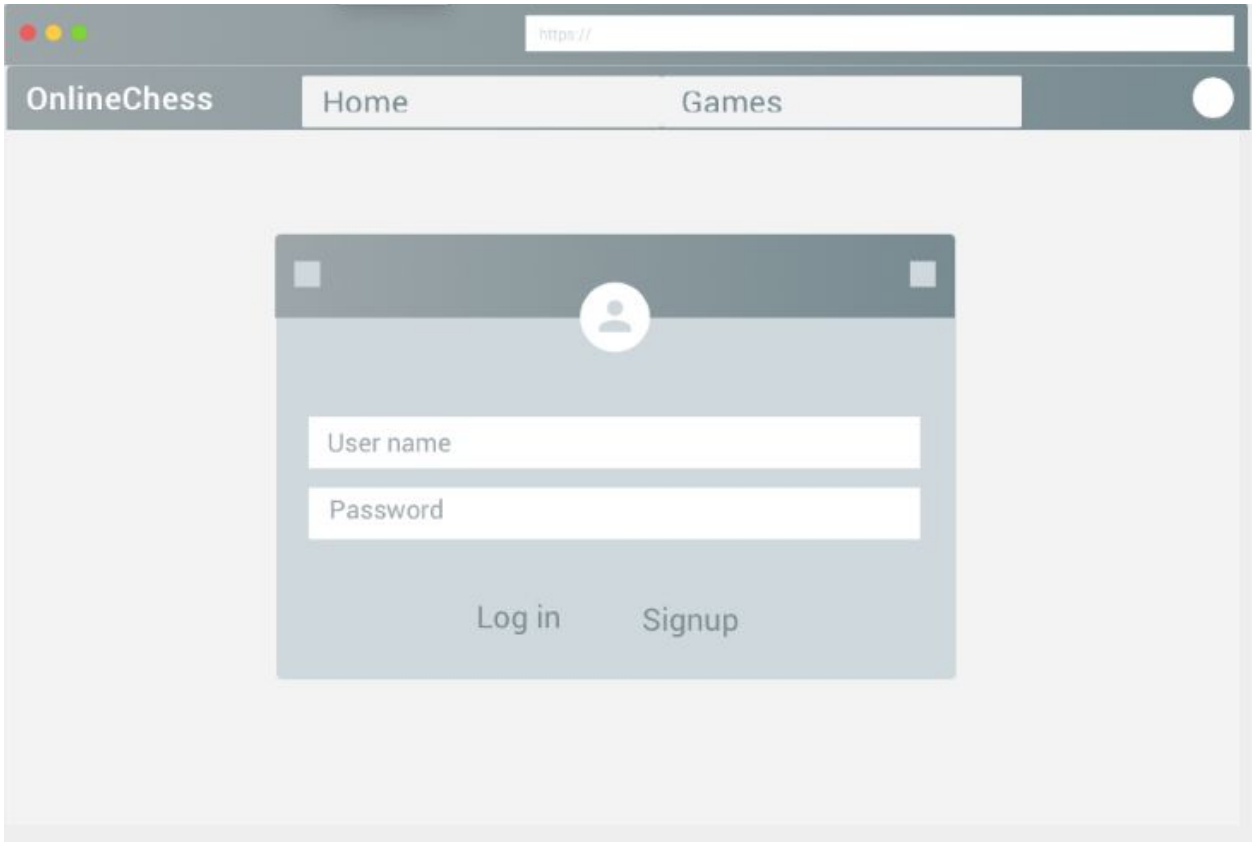



*Figure1. Login Page*

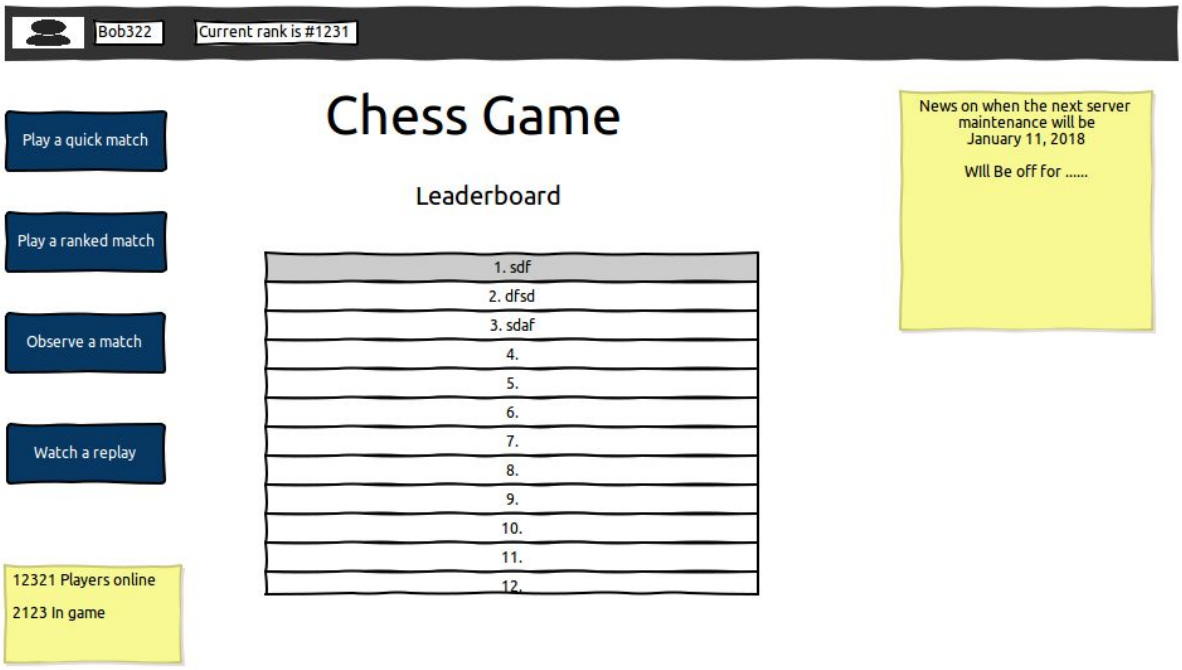



*Figure2. Game Main Page*

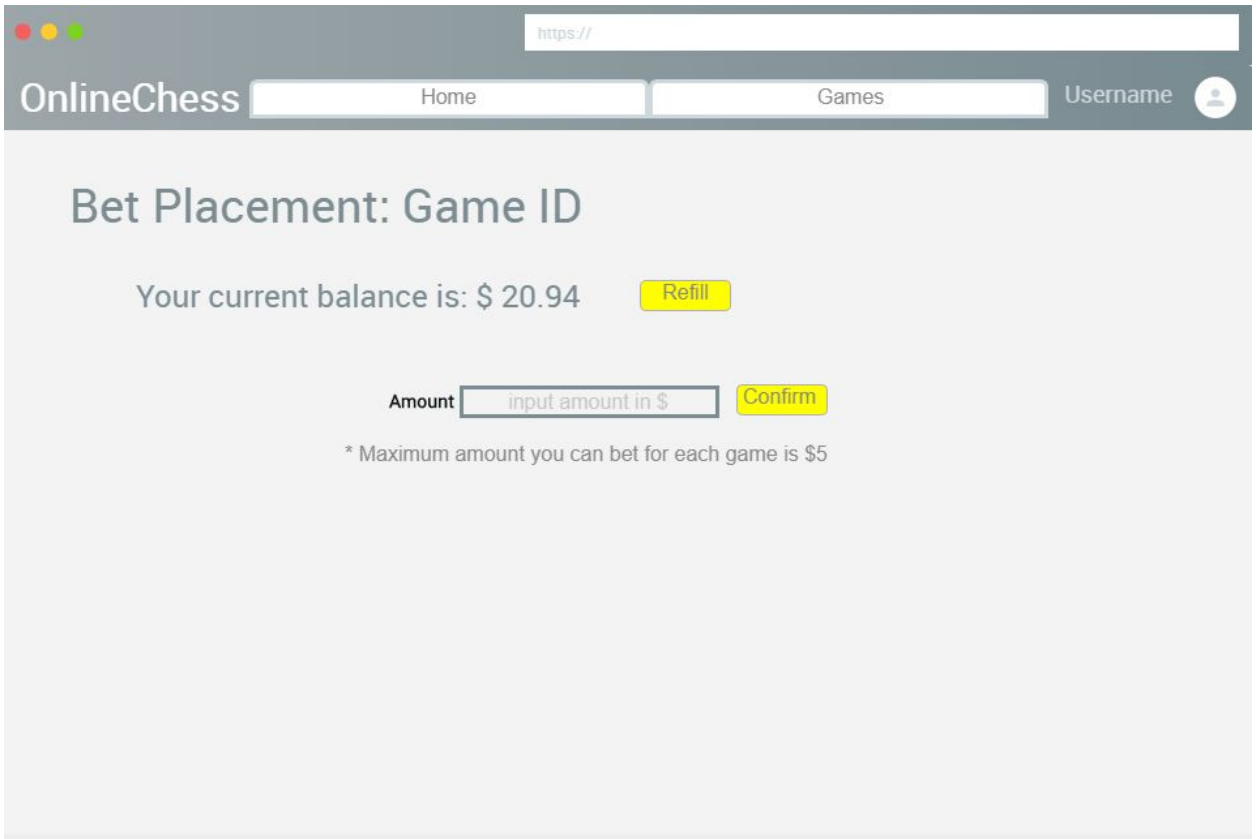



*Figure3. Bet Placement page*

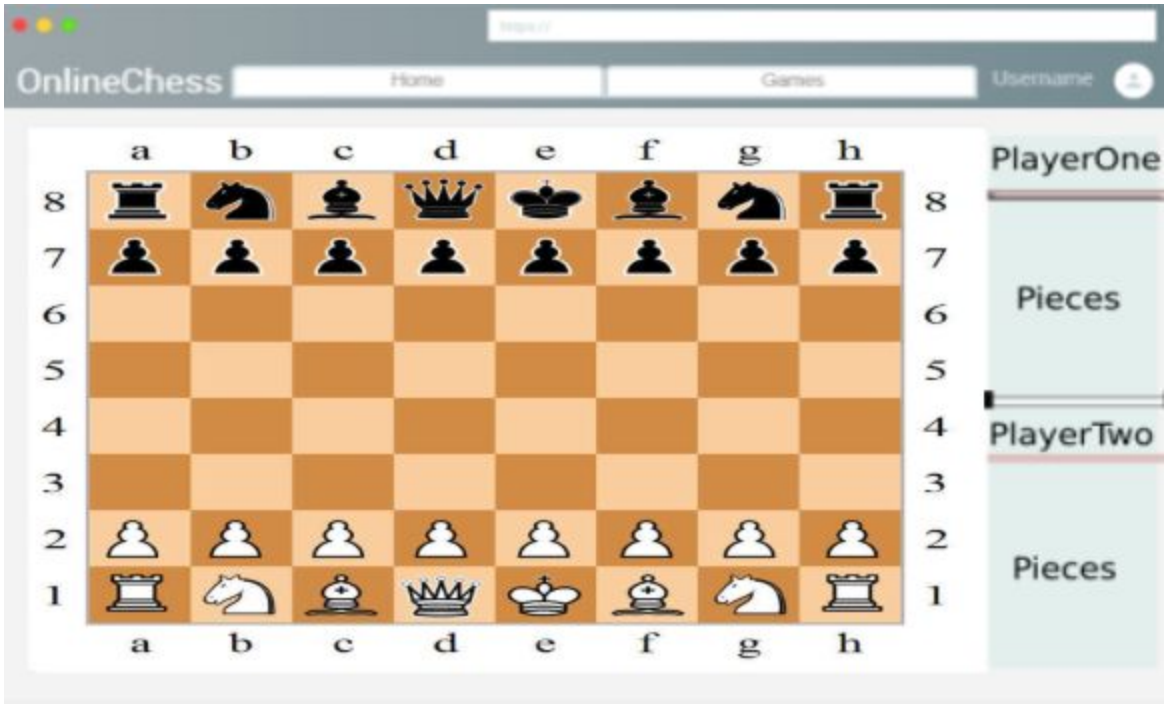



*Figure4. Game platform*


*Figure5. User Account Summary Page*

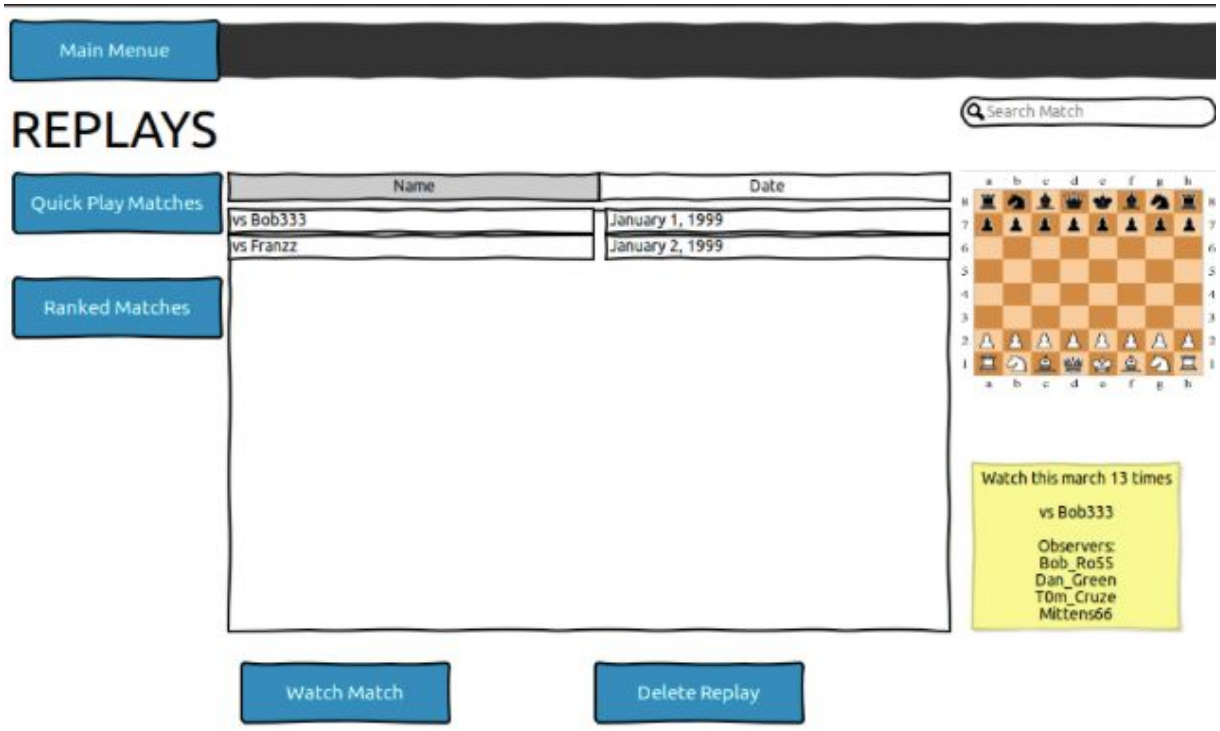



*Figure6. Replay Page1*

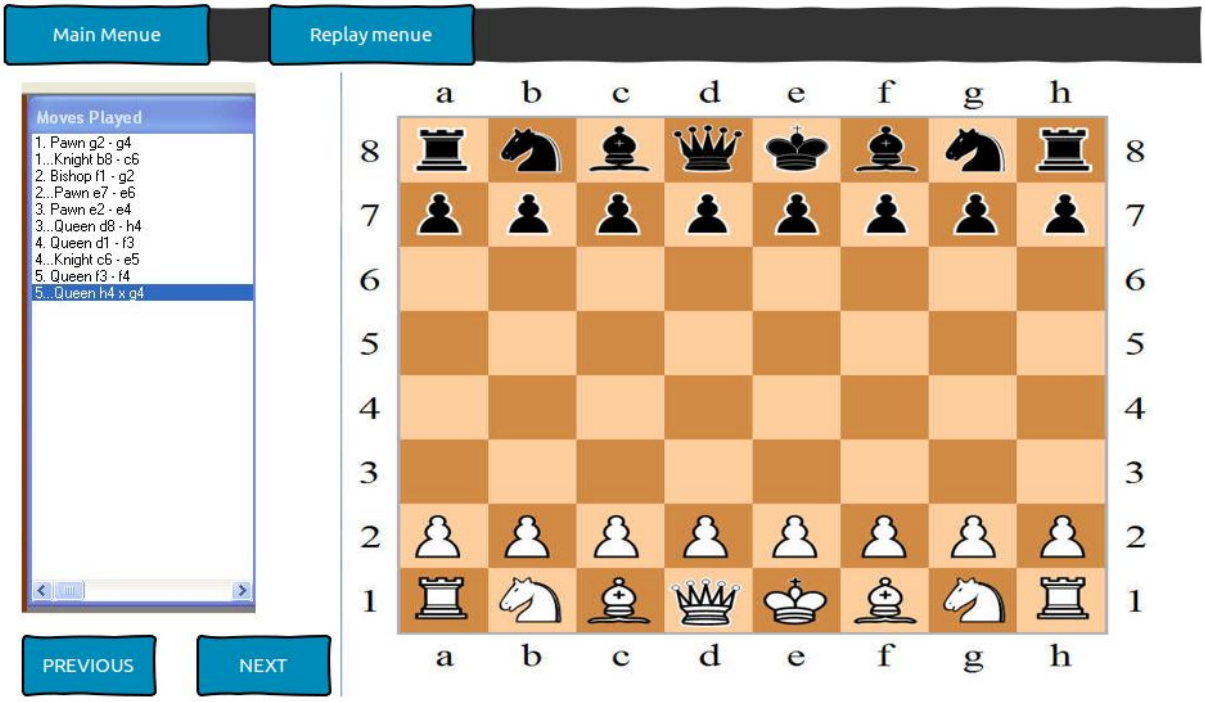



*Figure7. Replay page2*

### 4.2. Software interfaces

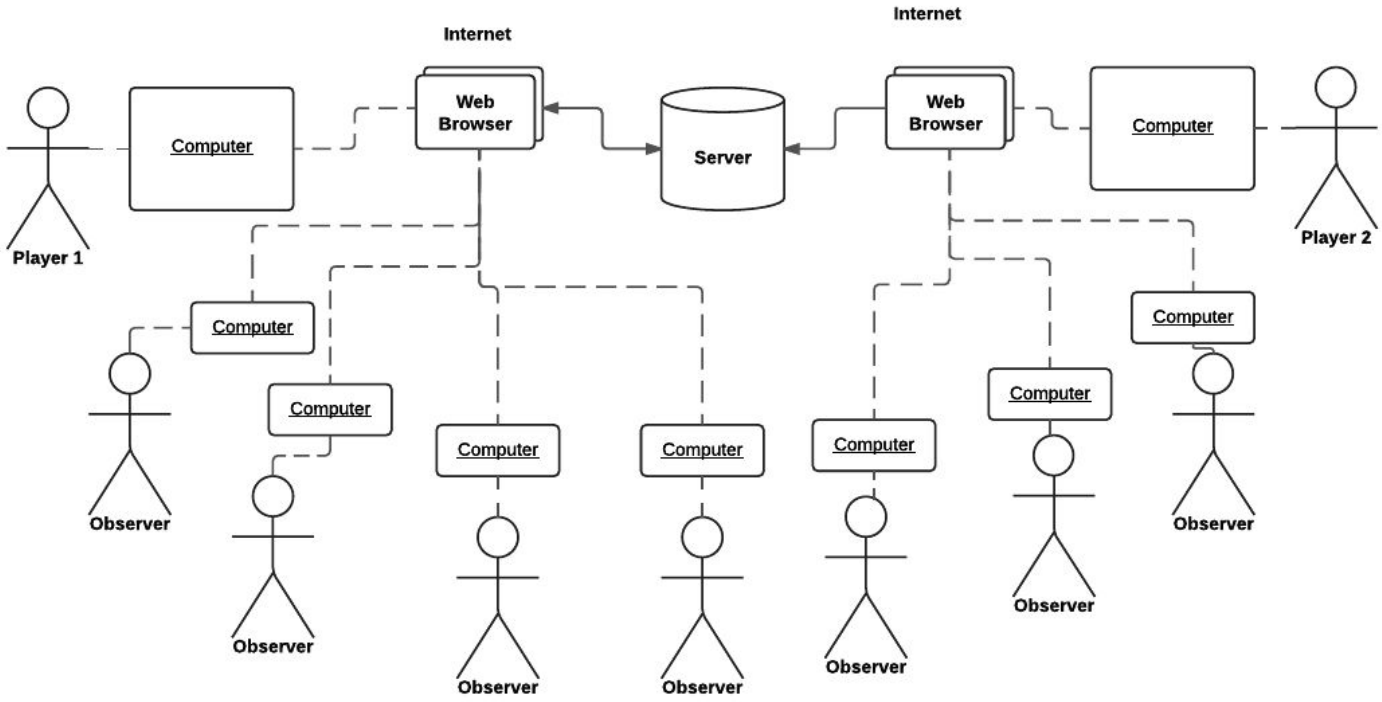



Figure 1 - Relationship between software

## 5. Other nonfunctional requirements

5.1. **Security** - Payment process for betting should be all encrypted with most recent and complicated algorithm.

5.2. **Usability** - Application can easily navigate through the game and game menus. All buttons in the system will adhere to established button convention.

5.3. **Response time** - The response time when players choose their moves should be less than 5 ms.

5.4. **Stability** - The application should have low frequency of crashing.

5.5. **Availability** - Uptime rate for the game should be over 98%.

5.6. **Backup** - All data from users should be saved in the cloud.