Chess Game
Requirements Specification

Group Members	Armaan Bhasin, Brandin Bulicki, Aparna Mishra, Tumaresi Yalikun, Briana Schuetz
Faculty Advisor	Dr. Filippos Vokolos, Ph. D.
Project Stakeholder	Dr. Filippos Vokolos, Ph. D.

Revision History

Name	Date	Reason for Change	Revision
Armaan Bhasin, Brandin Bulicki, Aparna Mishra, Tumaresi Yalikun, Briana Schuetz	07/15/2020	First Draft of the Requirements Document	1.0
Armaan Bhasin, Brandin Bulicki, Aparna Mishra, Tumaresi Yalikun, Briana Schuetz	7/20/2020	Revision and Submission of Requirements Document	1.5

Table of Contents

1. Introduction

1.1. Purpose of Document

This document will provide all of the requirements for the Chess Game. It will serve as a reference for the developers and the customers for developing the final version of the game.

1.2. Scope of Document

This document will contain enough information such that a developer will be able to easily translate the requirements to code without ambiguity.

1.3. Overview of Document

This document will contain functional and nonfunctional requirements for the Chess Game, as well as use cases, diagrams, and UI mock-ups. The game contains a component that will run on a web browser as well as different servers. Henceforth the web browser component will be called the client side and the servers will be called the server-side. The requirements will be provided for both the web browser component and the server. This document will also contain mock-ups for the graphical user interface and use cases.

2. Description

2.1. Product Perspective

Chess Game is a two-player game that will be hosted on an online platform. It allows a single player vs single player game over the internet on supported web browsers. The game is intended to recreate a Chess Game that is as close to the physical experience as possible.

Once the user is on the Chess Game web page, they will be asked to type a name and given the option to share the game link with other friends or invite friends through search to play the game together. The invited user can either decline or accept the game. The Chess game can be a timed or not timed game. If the opponent accepts the game, the match will start between two end users via the Chess Game platform. If the opponent rejects the game, the user will be notified, and the user can look for other opponents to challenge.

2.2. Product Functions

2.2.1. Client-Side Functionality

The client-side will have the following functionality:

- Prompt to ask for user's name
- Ability to send messages through a chatting feature
- Ability to display user with friends list
- Ability to search for other users
- Available game manual
- Reject/Accept the moves

2.2.2. Server-Side Functionality

The server-side will have the following functionality:

- Process Messages
- Move Validation

2.3. User Description

The ideal user for Chess Game would be two players for each game in a 1vs. 1 setup.

2.4. Assumptions and Dependencies

2.4.1. Technology 1 (Front-end Design Framework)

2.4.2. Technology 2 (Back-end Server)

2.4.3. Technology 3 (Platform to Host Game)

2.4.4. Technology 4 (Communication?)

Commented [BB1]: Needs To Be Done

2.5. Requirement Apportioning

Priority Level	Description
1	**Priority 1 -** requirements are essential to the product and must be in the final build.

	These requirements must be tested and verified to ensure proper functionality.
2	**Priority 2 -** requirements are not required for the final build but will be provided if there is sufficient time. The system will be designed such that it is extendable to easily incorporate these requirements at a later time.
3	**Priority 3 -** requirements are not required and will be considered in the design of the system. If sufficient time remains the requirement will be incorporated.

3. Functional Requirements

3.1. Board Layout

R3.1.1 Board Setup

The game board will have the shape of a quadrant consisting of 64 squares that will be arranged in 8 horizontal rows called files and 8 vertical columns called ranks. The squares will be in two alternate colors, light and dark. The board will hold 32 game pieces. (Priority 1)

R3.1.2 Board Piece

The game will include two sets of 16 board pieces where one set will be black and the other will be white. The pieces will be set up in the following order on the first and last rows of the chess board: Rook, Knight, Bishop, Queen, Bishop, Knight and Rook. The second row and the seventh row will have pawns. (Priority 1)

R3.1.3 Game Option

The game will feature an option to either play Bullet (1 minute), Blitz (3 minute), Standard (15 minutes) or default (no time limit) for the users. (Priority 2)

3.2. Moves

R3.2.1 Pawn

Pawns can move directly forward one square or two squares (on their first move). They can move diagonally forward when they are capturing an opponent's chess piece. (Priority 1)

R3.2.2 Rook

Rooks can move anywhere for as long as they want in any direction (forward, backward, left or right) as long as they are not obstructed by any other pieces. (Priority 1)

R3.2.3 Knight

Knights can move forward, backward, left or right two squares and should then move one square in perpendicular direction. (Priority 1)

R3.2.4 Bishop

Bishops can move diagonally in any direction for as long as they want if they are not obstructed by any other pieces. (Priority 1)

R3.2.5 Queen

Queens can move in any direction (vertically, horizontally or diagonally) for as long as they want unless they are obstructed by another piece in their way. (Priority 1)

R3.2.6 King

Kings can move one single space in any direction (forward, backward, left, right or diagonal). This piece can also participate in a castling move where the piece can move up to three squares while exchanging places with a rook. (Priority 1)

3.3. Capturing

R3.3.1 Pawn Capturing

Pawns will have the ability to capture an opponent's chess piece by moving diagonally in the forward direction. (Priority 1)

R3.3.2 General Capturing

For the remaining pieces, the opponent’s chess piece can be captured if the player’s chess piece can move to that specific square. (Priority 1)

R3.3.3 En Passant

In this capturing move, a pawn captures an opposing pawn on the same rank right after the opposing pawn has moved forward two squares on its first move in the game. (Priority 1)

3.4. Promotion

R3.4.1 Promotion

A pawn upon reaching the end of the board will be able to choose to be any other piece, as desired by the user. (Priority 1)

3.5. Legality

R3.5.1 Illegal Move

Moves that haven't already been defined in the guidelines stated above will not be considered legal. Moreover, if the player's king is in check, then the player should either defend their king or move their king in order to avoid being in check.

Players cannot purposefully make any moves that allow their king to be captured. If the player's king is attacked and there are no squares that the king can move to without a threat, then the player will be in Checkmate. (Priority 1)

R3.5.2 Turn

Players, upon using their turn to move, cannot undo or redo their move. In order for them to move again, they must wait for their opponent to complete their turn. Each player will be given one move per turn. (Priority 1)

3.6. Game State

R3.6.1 Checkmate

If the king is threatened by the opponent's piece and cannot move in any direction to escape, then the game will end. The player who has been checkmated will lose the game resulting in its termination. (Priority 1)

R3.6.2 Stalemate

If the player has no more moves left to make apart from being in check then that position is counted as a draw and a Stalemate is declared. This will result in the termination of the game. (Priority 1)

R3.6.3 Voluntary Stalemate

If both players agree upon withdrawing from the game, then a Stalemate will be declared, and the game will be terminated. (Priority 2)

R3.6.4 Insufficient Material

If both players do not have a way to end the game in checkmate such as when both players are down to their kings, the game will be declared a draw. (Priority 3)

3.7. Servers

Commented [BB2]: Needs To Be Done

4. Non-Functional Requirements

4.1. Performance

R4.1.1 Latency Management

The application shall have minimum lag time between user interface response time, move validation, and board updates. The time to recognize a user's move on the board, validate the move, initiate a message request with the server, and update the opposing user's board shall be less than 5 seconds. (Priority 1)

R4.1.2 System Flexibility

Increases in the number of concurrent games and users will have minimum effect on application response time. (Priority 3)

4.2. Scalability

R4.2.1 Concurrency

The application shall handle at least 10 concurrent chess games between 20 users. (Priority 3)

4.3. Availability

R4.3.1 Application Uptime

The application shall be available 24 hours a day, every day. The overall availability shall be 99%. (Priority 1)

R4.3.2 Application Maintenance

Application downtime for development updates shall be minimal and during off-hours with less traffic. (Priority 1)

4.4. Reliability

R4.4.1 Critical Error

Upon a critical error, the application will log all error messages and user information. The application will restart. (Priority 1)

4.5. Usability

R4.5.1 Game Rules Guide

Users shall be able to access a How-To-Play guide with the complete rules to chess. (Priority 2)

R4.5.2 Help Guide

Users shall be able to access a Help guide on how to navigate the application. (Priority 2)

R4.5.3 Ease of Use

Users shall be able to start a game against an opponent, play a game using the interactive chess board, learn the rules, and exit the game with relative ease and speed. (Priority 1)

R4.5.4 Mouse Support

Users shall be able to navigate the application and make moves on the board through mouse clicks and movements. (Priority 1)

R4.5.5 Touchscreen Support

Users shall be able to navigate the application and make moves on the board through touch events on devices with touchscreens. (Priority 2)

R4.5.6 Web Browser Support

The application shall run on all major Internet web browsers, including Google Chrome v84.x and Mozilla Firefox v78.x. (Priority 1)

R4.5.7 Operating System Support

The application shall run on all major operating systems, including Windows OS and Mac OS. (Priority 1)

4.6. Security

R4.6.1 Message Encryption

The application shall (Priority 3)

4.7. Reports & Logs

R4.7.1 Logging

The application shall log all users and user moves and store them for at least a year. (Priority 2)

R4.7.2 System Messages

The application shall log and store all system, socket, and error messages for at least a year. (Priority 2)

R4.7.3 User Reports

The application shall log and store all user reports for at least a year. (Priority 2)

4.8. Development Constraints

R4.8.1 Front-End

The application front end shall be developed using React version xxx. (Priority 1)

R4.8.2 Back-End

The application back end shall be developed using Node.JS v12.18. (Priority _)

R4.8.3 Communication

The real-time communication between the front and back ends shall be handled by the Socket.IO NPM package v2.3. (Priority _)

Commented [BB3]: Need Priorities

R4.8.4 First Release

The first release shall be available on August 25, 2020. (Priority 1)

5. User Interface

5.1. System Workflow

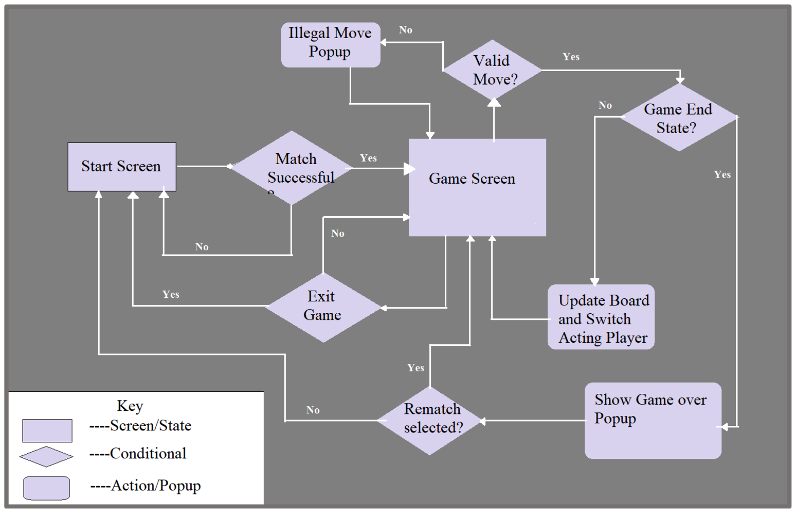


Figure 5a: A Diagram Displaying the Chess Game Workflow

5.2. Start Screen

Start Screen will Contain Text Box for users to enter their name, As well as a List of all potential chess partners online. Player will select a partner from the List (currently selected partner being indicated by the darker shading of their option, as shown in figure 5b), and press the button to request a game, which will cause a notification to popup on the other user's screen notifying them of the invite and giving them the option to either accept or reject the request. If the other user rejects the game request, a popup will appear on the original user's screen notifying them that their request has been rejected. If the other user accepts the invite, both users will be taken to the Chess Board Screen.

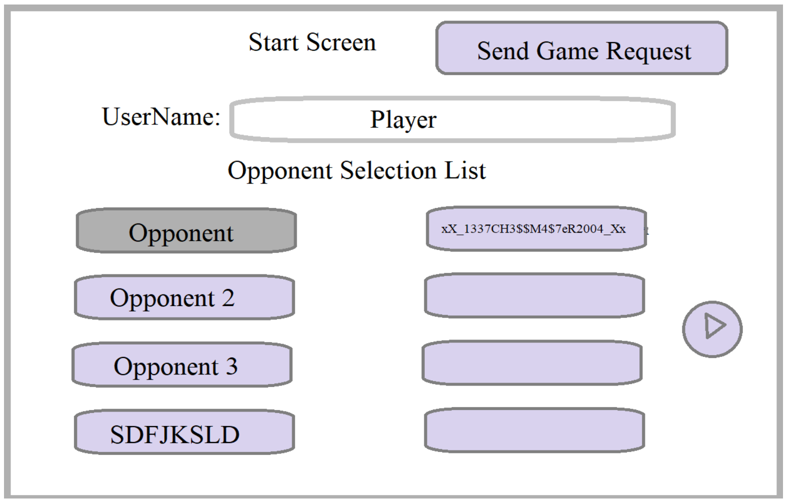


Figure 5b: An Example Start Screen Design

5.3. Game Screen

The Game Screen is where the Users will be to see the current state of and interact with the Chess Game Board. It will be a standard Chess Board, 8x8 squares with each player starting with 8 pawns, 2 knights, 2 rooks, 2 bishops, 1 queen, and 1 king, as in Figure 5c. The side of the Screen will display whose turn in the game it currently is.

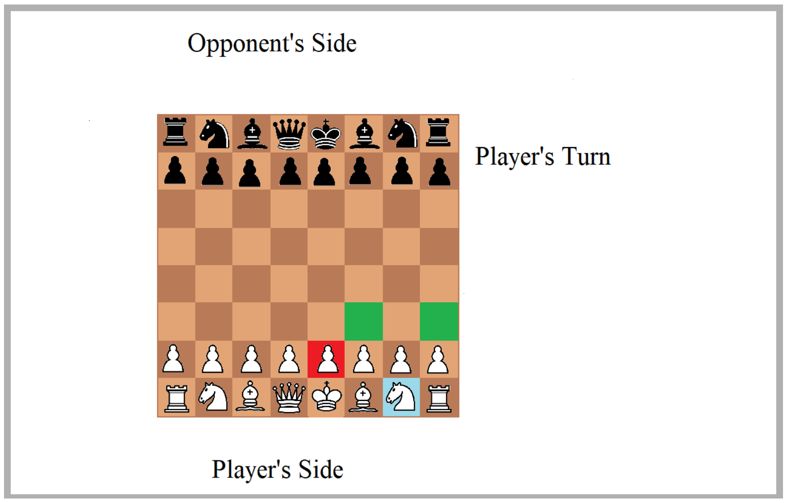
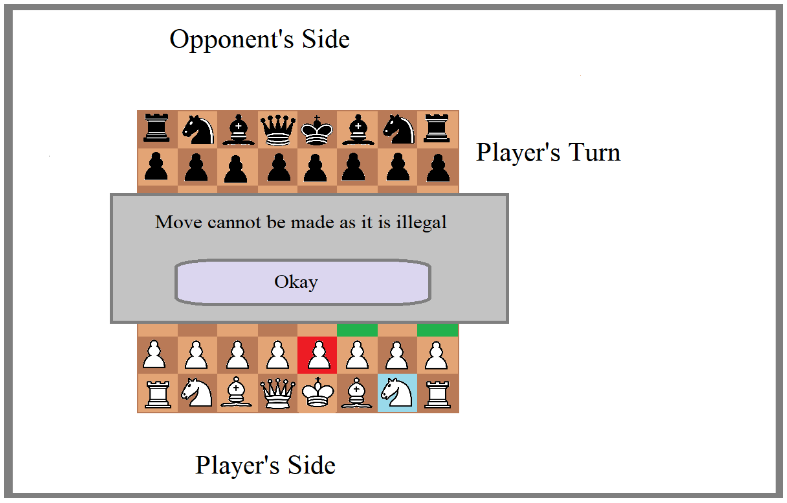


Figure 5c: An Example Chess Board Starting State with selected piece and move highlights

5.4. Player Move Notifications

During their turn, Players will be able to move their pieces first by clicking on the piece they wish to move. This will highlight the space which the piece is on in Blue and will also highlight all other places that that piece can reach according to its movement rules in the game. If a board space cannot be reached by the selected piece, either because it is blocked or the movement required to get there is not allowed by the movement rules for the piece (like a Rook trying to move diagonally), then the board space will not be highlighted. If a board Space can be reached by the selected piece, BUT for some reason isn't a valid move, either because that space is already occupied by a friendly unit or because moving there would put their king in Check or fail to get them out of Check, that that board space will be highlighted in red, as shown above in Figure 5c. If the Player still chooses to click on a red highlighted Space, A popup will appear, informing the player as to exactly why that move is invalid, and telling the player to make a different one (as shown in Figure 5d). If a board space can be reached by the selected piece AND is a valid move, that board space will be highlighted in green. If the Player chooses a valid move AND a Game End state is not detected, then the board will be updated for both the Player who made the move and the opponent and will be the opponent's turn to make a move. If a Game End State is detected, then the Game will end for both players



Move cannot be made as it is illegal

Figure 5d: An Example of an Illegal Move Notification

5.5. Exit Confirmation

If a player chooses to quit the game, a notification will appear, asking the player to confirm that they wish to leave the game. If the player clicks ‘confirm’, the game will end for both players, and they will both be taken back to the start screen.

5.6. Game End Screen

If a Game End State is detected, then a popup will appear for both Players indicating the end of the Game, the Reason for its end (Victory, Stalemate, etc.), and buttons for either returning to the start menu, or requesting a rematch (as shown in Figure 5e). If both Users vote to Rematch, then they will be able to engage in another game on the Game Screen reset to a starting state, without having to go back to the Start Screen. If both players choose to go back to the start screen, then they will both return to the start screen with no more interruptions. If One player votes to rematch but the other doesn’t, then Player didn’t vote to rematch will be taken back to the Start Screen straightaway, while the Player who voted to rematch will receive a notification saying that the other player did not vote to rematch and will be returned to the Start Screen as well after that Notification is Closed.

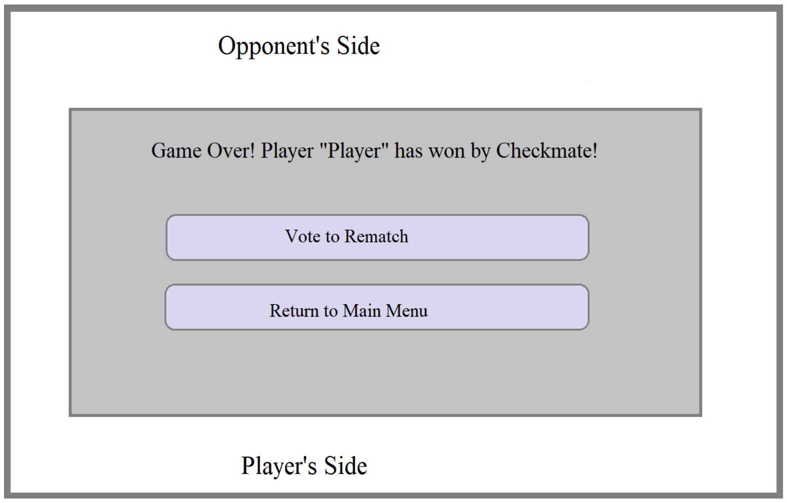


Figure 5e: An Example of an "End of Game" Notification

6. Use Cases

6.1. Use Case Flow

6.1.1. Starting a game (User)

User views menu and starts a game.

Precondition: The user is on the menu screen and is not already playing a game.

Action: The user selects a player and starts the game.

Postcondition: The client will have a real-time connection to the server, with a socket open that checks for any updates.

6.1.2. Starting a game (Server)

User views menu and starts a game.

Precondition: The user is on the menu screen and is not already playing a game.

Action: The user selects a player and starts the game.

Postcondition: The server will start a session of a new game with the two users. The board will be initiated in memory to allow for updates to the game to be made.

6.1.3. Taking a Turn

The user selects a piece and moves it to a new location.

Precondition: A game is already in progress, and it is the player's turn.

Action: The player selects the chess piece which they would like to move and selects the new tile they would like to move to.

Postcondition: If the move is legal, the piece will be moved on both user's screens and in memory. If the move is invalid, the player making the move will be notified and asked to make a different move.

6.1.4. Capturing a Piece

A piece of the opposite color is moved to the same location as another piece, the piece that is covered is captured.

Precondition: A game is in progress and a player is taking their turn.

Action: A player moves a piece to a tile that is already occupied by the opponent's piece.

Postcondition: The opponent's piece will be removed from gameplay. This will be seen on both user's screens and in memory.

6.1.5. Placing in Check

A piece of the opposite color is able to capture the king on their next move, the player in check must move out of check.

Precondition: A game is in progress and a player is taking their turn.

Action: A player moves their piece to a tile that will allow the capture of their opponent's king on their next move.

Postcondition: A notification is sent to the player whose king is at risk of capture. The player must move out of check on their next move.

6.1.6. Placing in Checkmate

A piece of the opposite color is able to capture the king on their next move, the king in check and is unable to move out of check.

Precondition: A game is in progress and a player is taking their turn.

Action: A player moves their piece to a tile that will allow the capture of their opponent's king on their next move and that their opponent cannot move out of check.

Postcondition: The screen will be cleared for both users and a notification will inform both users that there has been a checkmate. The winner's name will be displayed. The instance of the game on the server will be cleared.

6.1.7. Reaching Stalemate

The player taking a turn is not in check but has no legal moves, resulting in a draw.

Precondition: A game is in progress and a player is taking their turn.

Action: A player moves their piece in such a way that their opponent is not in check but is unable to make a legal move.

Postcondition: The screen will be cleared for both users and a notification will inform both users that there has been a stalemate. The instance of the game on the server will be cleared.

6.1.8. Quitting a Game

The player does not want to finish the game and quits the match.

Precondition: A game is in progress.

Action: One user exits the game.

Postcondition: The screen will be cleared for the remaining users and a notification will inform the users that their opponent has left the game. The instance of the game on the server will be cleared.

6.2. Activity Diagram

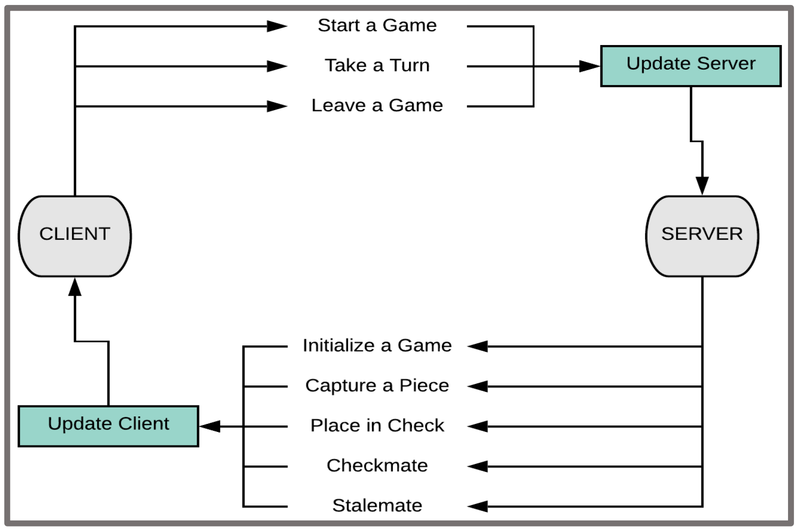


Figure 6a: An activity diagram displaying the use cases from the perspective of one user.