

# Software Requirements Document

Team 2, The Triple Grobs  
Jonathan Cels (celsj)  
Pesara Amarasekera (amarasep)  
Rupinder Nagra (nagrar5)

February 11th, 2021

## 1 Introduction

### 1.1 Purpose & Stakeholders

The purpose of this document is to specify the system requirements and how the system interacts with the end user. Intended stakeholders for this document include the development team, the professor of the course, and the TAs of the course.

### 1.2 Scope

The software product will provide the user with an implementation of a chess game. The product aims to provide an interactive and entertaining game-play experience utilizing web programming. The user will be able to play with other users online.

### 1.3 Definitions, Acronyms, and Abbreviations

- a) The Software Requirements Specification (SRS) is a detailed document which describes a software system to be developed.
- b) The Web Content Accessibility Guidelines (WCAG) [1] are guidelines on making web content more accessible to people with disabilities.
- c) Artificial Intelligence (AI) in this document refers to a computer-controlled player.

### 1.4 References

- [1] Web content accessibility guide. <https://www.w3.org/WAI/standards-guidelines/wcag/>. Accessed: 2021-02-10.
- [2] Rules of chess. [https://en.wikipedia.org/wiki/Rules\\_of\\_chess](https://en.wikipedia.org/wiki/Rules_of_chess). Accessed: 2021-02-10.
- [3] History of chess sets and symbols. <https://www.chesscentral.com/pages/chess-sets-pieces-boards/a-history-of-chess-pieces-and-chess-sets.html>. Accessed: 2021-02-10.
- [4] Airbnb javascript style guide. <https://airbnb.io/javascript/react/>. Accessed: 2021-02-10.

### 1.5 Overview

The SRS document will go into depth on the description of the product in Section 2, describing functions, characteristics, and constraints on the system. In Section 3 this document will detail different use cases of the system and illustrate a use case diagram, providing an overview of the intended product behaviour. This document will list out functional and non-functional requirements that the stakeholders aim to achieve in the

final product. The functional requirements will be presented in Section 4, and non-functional requirements will be in Section 5. Finally, this document will include an overview of the different development phases and possible risks in the development process.

## 2 Overall Description

This section of the SRS describes the general factors such as the functions and user characteristics that affect the product and its requirements. Also, it provides a background for these requirements to make them easier to understand.

### 2.1 Product Functions

- a) The user can start a game with a chosen player or AI opponent. During the game users can choose to resign a game, draw a game, make a move on the board, or use the chat window to interact with the other player.
- b) The following are the functions explained in more detail:
  - The host player is able to start the game against an AI or another user. The timer will then begin for whichever player has the current move.
  - Either player can choose to resign the game, terminating the game and causing their opponent to win.
  - Either player can choose to draw the game. If this is selected, the other player must accept a draw as well. If both players accept, the game terminates and the game is a tie.
  - Either player can choose to make a move on the board using a drag-and-drop mechanism for a piece. The move will only be made if it is a legal move.
  - Either player can also interact using the chat window.

### 2.2 Constraints

- a) Time constraint: The final product must be fully implemented by the end of the semester.
- b) Resource constraint: The developers will have limited access to additional resources such as more developers, money, and software management tools.

### 2.3 Facts, Assumptions, and Dependencies

- a) It is assumed that the user has access to the required hardware and internet connection required to access and play the game without the hindrance of lag or other issues.
- b) It is assumed that the user has access to I/O devices, such as a keyboard/mouse/monitor for a computer, or a touchscreen for a mobile device.
- c) Based on the browser used for the web application, the **HTML** and **CSS** might be affected. This project will use the assumption that a user will be using the latest version of Google Chrome.
- d) It is assumed that the user has proficiency in the English language.

## 3 Use Cases

This section will include a description of intended use cases, accompanied by an illustration of the use case diagram for the system.

### UC1. Select Opponent

**Related Requirements**

- i. FR1, FR2 stated in section 4, UC1.
- ii. FR3, FR4, FR5, FR6, FR7, FR8, FR9, FR10 stated in section 4, UC2.
- iii. FR11, FR12, FR13 stated in section 4, UC3.

**Initiating Actor:** User

**Actor's Goal:** To choose between playing against an AI or a human player.

**Participating Actors:** System

**Pre-conditions** The system displays the menu of available options. The choices are "AI" and "Online Player".

**Post-conditions** The "Start Game" use case is initiated.

**Flow of Events of Main Success Scenario:**

- i. Host player selects one of two player options, "AI" or "Online Player".
- ii. The application is initialized based on the choice made by the host. The game now starts.

**UC2. Start Game****Related Requirements**

- i. FR14, FR15, FR16, FR17, FR18 from section 4, UC4.

**Initiating Actor:** User

**Actor's Goal:** To start playing a game against the selected opponent.

**Participating Actors:** System

**Pre-conditions** An "AI" or "Online Player" was selected as the opponent.

**Post-conditions** The board is setup and one or two players are connected and able to play the game.

**Flow of Events of Main Success Scenario:**

- i. Either the "AI" or "Online Player" opponent option is selected.
- ii. The server connects the two players together, or connects one player to the AI.
- iii. The board is initialized to the starting state and piece colors (black or white) are arbitrarily selected between the players.
- iv. Control is passed to the user with the white pieces and the timers are initialized.

**UC3. Make Move****Related Requirements**

- i. FR19, FR20, FR21, FR22, FR23, FR24, FR25 stated in section 4, UC5.

**Initiating Actor:** User

**Actor's Goal:** To make a move in the game against the opponent.

**Participating Actors:** System

**Pre-conditions** The game has been started and the opponent has made their move.

**Post-conditions** The game may be terminated, or the opponent is now able to make their move.

**Flow of Events of Main Success Scenario:**

- i. The user makes a legal move.
- ii. The application registers that the user has made a move and then determines if the game must continue.
- iii. If the game has not ended, the opponent is now able to make their move.

**Flow of Events for Extensions (Alternate Scenarios):**

- i. The user may make a legal move.
- ii. The application will register that the user has made a move, where it must determine if the game must continue.
- iii. If the game has not ended, the opponent is now able to make their move.

#### UC4. Check Legal Move

##### **Related Requirements**

- i. FR26, FR27, FR28 stated in section 4, UC6.

**Initiating Actor:** User

**Actor's Goal:** To determine if the move made is valid.

**Participating Actors:** System

##### **Pre-conditions**

- i. The game has started.
- ii. The user has made a move.

**Post-conditions** The game may be terminated, or the opponent is now able to make their move.

##### **Flow of Events of Main Success Scenario:**

- i. The user makes a move.
- ii. The application will register that the user has made a move, where it must determine if the game must continue.
- iii. The application will check the game logic to determine if the move is legal.
- iv. If the game has not ended, the opponent is now able to make their move.

#### UC5. Initiate Chat

##### **Related Requirements**

- i. FR29, FR30, FR31, FR32 stated in section 4, UC7.

**Initiating Actor:** User

**Actor's Goal:** To initiate communication between players while concurrently playing a game.

**Participating Actors:** System

##### **Pre-conditions:**

- i. The player has started a game against another player.

##### **Post-conditions:**

- i. The player has communicated with the other player.
- ii. The game-play is continued asynchronously.

##### **Flow of Events of Main Success Scenario:**

- i. A game has started between two players.
- ii. One of the players selects the chat function and types a message.
- iii. The message is passed to the other player and made visible.
- iv. The opponent is able to asynchronously send messages and continue game-play.

#### UC6. End Chat

##### **Related Requirements**

- i. FR33, FR34, FR35, FR36 stated in section 4, UC8.

**Initiating Actor:** User

**Actor's Goal:** To stop communicating with one's live opponent.

**Participating Actors:** User

**Pre-conditions:**

- i. The player has selected an opponent.
- ii. The player has started a game.

**Post-conditions:**

- i. The player has made contact with the other player.
- ii. The game-play is continued asynchronously.

**Flow of Events of Main Success Scenario:**

- i. A player initiates chat with their opponent.
- ii. One of the players can at anytime choose to end the chat session.
- iii. One player ending the chat closes the chat for both players.

**UC7. Start Timers****Related Requirements**

- i. FR37, FR38, FR39 stated in section 4, UC9.

**Initiating Actor:** System

**Actor's Goal:** To start the timer for each player that represents how long each player has to make their moves in the game.

**Participating Actors:** User

**Pre-conditions** The game has been started and the player with the white pieces has made a move.

**Post-conditions** The timers have begun to countdown from their original values.

**Flow of Events of Main Success Scenario:**

- i. The user with the white pieces makes a legal move.
- ii. The timer for the user with the black pieces begins to count down from its initial value.
- iii. The user with the black pieces makes a legal move, or that player's timer reaches 0.
- iv. If a legal move is made, that timer is paused and the other timer begins to count down from its initial value.
- v. If a timer reaches 0, the "Terminate Game" use case occurs and is overridden by "Timeout".

**UC8. Terminate Game****Related Requirements**

- i. FR40 stated in section 4, UC10.
- ii. FR41 stated in section 4, UC11.
- iii. FR42 stated in section 4, UC12.
- iv. FR43 stated in section 4, UC13.
- v. FR44 stated in section 4, UC14.
- vi. FR45 stated in section 4, UC15.

**Initiating Actor:** Any of: User, or System

**Actor's Goal:** To determine if the game must be ended.

**Participating Actors:** None

**Pre-conditions** The game has been started.

**Post-conditions** The game has ended with a definite result.

**Flow of Events of Main Success Scenario:**

- i. The system must start the game.

- ii. The application determines if a user has offered a decision to end the game in a resignation or draw. The other user can choose to accept or reject this offer.
- iii. The application determines if the user has won or lost the game through checkmate, stalemate or timeout. This signifies that one of the players can no longer make any moves.
- iv. The game will end in a definite result if any of the previous events are true.

Below is a visualization of the above use cases as part of a use case diagram.

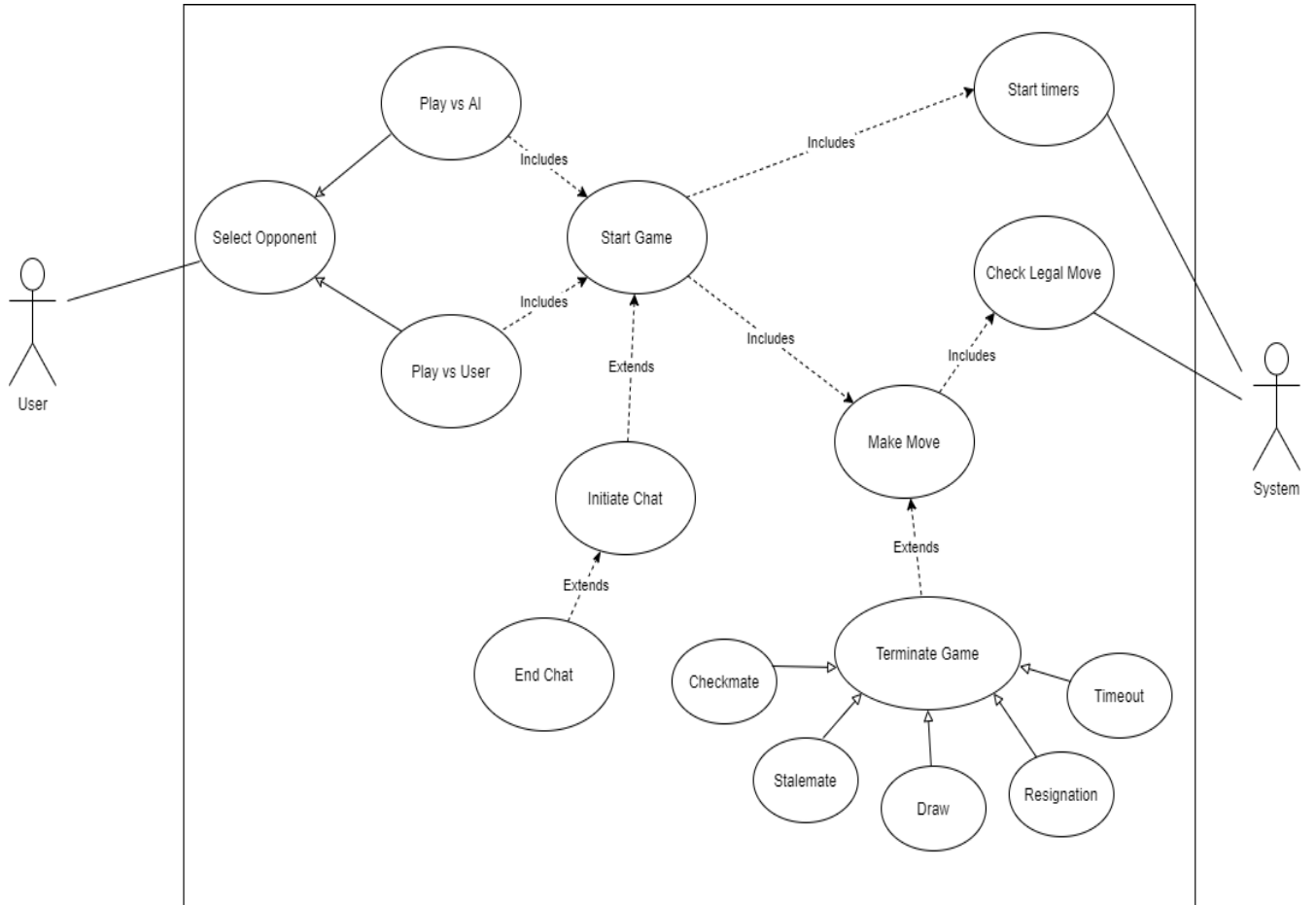


Figure 1: UML Use Case Diagram

## 4 Functional Requirements

This section of the SRS contains all of the functional software requirements. The functional requirements detailed here enable designers to design a system that satisfies these requirements, and enable testers to test that the system satisfies these requirements.

### UC1. Select Opponent

FR1. The user shall connect to the web application server.

FR2. The user shall be immediately presented with the option of playing with an “AI” or “Online Player” in the middle of their screen. They shall use the labeled buttons to choose between the two options in order to proceed.

## **UC2. Play Against Live Opponent**

- FR3. If a user decides to “Play Against a Live Opponent” the system shall pair them against a user with the same invite link.
- FR4. While the system is awaiting another player, the system shall be present the user with the options to “Quit”, or “Play Against AI”.
- FR5. If a user decides to “Quit” or “Play Against AI” the current game shall be terminated.
- FR6. If a user selects the “Quit” option, the system shall return them to the “Select Opponent” use case.
- FR7. If a user selects the “Play Against AI” option, the system shall send them to the “Play Against AI” use case.
- FR8. The system shall present the players with a loading screen while the application is initializing the game. The loading screen shall last no longer than 5 seconds.
- FR9. Once initialized, the system shall present the players with a view of the board and transfer to the “Start Game” use case.
- FR10. If a user requests to start a game on any server with 15 concurrent games, the system shall reject the request and return to the “Select Opponent” use case.

## **UC3. Play Against AI**

- FR11. If a user selects “Play Against AI” the system shall immediately transfer to the “Start Game” use case.
- FR12. While playing against the AI the system shall have no option to initiate chat.
- FR13. If a user requests to start a game on any server with 15 concurrent games, the system shall reject the request and return the user to the “Select Opponent” use case.

## **UC4. Start Game**

- FR14. If the user chooses “AI” from the “Select Opponent” use case, the server shall connect to an AI player and arbitrarily assign black or white to each player.
- FR15. If the user chooses “Online Player” from the “Select Opponent” use case, a shareable link shall be created with which a player can join the server. If a player joins, each player shall be arbitrarily assigned black or white.
- FR16. The board and pieces shall be initialized, and the orientation of the board will be determined based on the colour of the users pieces.
- FR17. The timer shall be initialized with a countdown of 10 minutes for each player, and the game control is dependent on the player with the white pieces.
- FR18. The system shall give exclusive control to the player with the white pieces and start that player’s timer.

## **UC5. Make Move**

- FR19. The system shall not allow the player to make a move until their opponent has played a move, with the exception of the first move of the game.
- FR20. If a player clicks a piece, holds the mouse down, drags the piece to a square, and releases the mouse, the system shall register an attempted move.
- FR21. The application shall confirm that the move is legal as defined by the rules of the game.
- FR22. If the attempted move is legal, the move shall be sent to the server and the opposite player shall be given control.
- FR23. If the attempted move is not legal, the move shall not be sent to the server and the player shall retain control.

FR24. The ‘Terminate Game’ use case shall be checked after every move is made.

FR25. If the game is not terminated, the timer for the other player shall resume.

#### UC6. **Check Legal Move**

FR26. The application shall register the move the user has made, and check the program logic to determine if the move can be made.

FR27. The move is then reflected in the board state and the opponent may now make their move.

FR28. Only one player will be allowed to make a move at once.

#### UC7. **Initiate Chat**

FR29. The system shall present the user with an option to “Start Chat” to start a chat.

FR30. If “Start Chat” is selected, the system shall open a chat window for both players.

FR31. If both players click on the “Start Chat” button simultaneously then the system shall only register one of the selections.

FR32. The “Start Chat” button shall be replaced by an “End Chat” button.

#### UC8. **End Chat**

FR33. The system shall present the user with an option to “End Chat” at any time once a chat has started.

FR34. If the chat is ended by one player the chat shall be disabled for both players. The “End Chat” button shall be replaced by the “Start Chat” button.

FR35. If both players click on the “End Chat” button simultaneously then the system shall only register one of the selections.

FR36. If the “End Chat” option is selected, the system shall clear all previous chat data.

#### UC9. **Start Timers**

FR37. Once a game has been confirmed a timer shall be started by the system.

FR38. The timer shall be set for a total of 10 minutes per player.

FR39. The timer shall begin for a player once it is their turn to move and will pause on their opponents turn.

#### UC10. **Terminate Game**

FR40. The system will have checks to establish if the game shall be terminated with the appropriate use case possibilities. These include the Checkmate, Stalemate, Draw, Resignation, and Timeout use cases.

#### UC11. **Checkmate**

FR41. The system shall determine if the opposing player is in ‘Check’ [2] (state where the King is under attack) and has no legal moves for the King. This shall result in a victory with ‘Checkmate’ for the current player.

#### UC12. **Stalemate**

FR42. The system shall determine if the opponent player is not in ‘Check’ and has also no legal moves. This shall result in a draw with ‘Stalemate’ for both players.

#### UC13. **Draw**

FR43. The system shall present the user with the ‘Offer Draw’ button to offer a draw to the opponent. The draw must be accepted to end the game.



#### UC14. **Resignation**

FR44. The system shall present the user with a decision to end the game with the ‘Resign’ button. This shall immediately end the game.

#### UC15. **Timeout**

FR45. The system shall determine if the user is out of time. This causes that player to lose the game.

## **5 Non-Functional Requirements**

### **5.1 Look and Feel Requirements**

#### **5.1.1 Appearance Requirements**

LF1. The system shall use white, black, and brown as its primary colours.

LF2. The system shall use a minimalist design.

#### **5.1.2 Style Requirements**

LF3. The game shall have a calm and intuitive feel as users will be concentrated on making moves under a time constraint, making distractions undesirable.

LF4. The game shall include sound-effects to inform that a move has been made.

LF5. The game shall not include background music.

### **5.2 Usability and Humanity Requirements**

#### **5.2.1 Ease of Use Requirements**

UH1. The game shall not require any intricate keyboard presses or fast mouse movements.

UH2. The product shall be easy to use by 10 year old children. Ninety percent of a sample test panel of 10 year old children shall be able to successfully play a move within 5 minutes of starting a game.

#### **5.2.2 Personalization and Internationalization Requirements**

UH3. The game shall only display information in English.

#### **5.2.3 Learning Requirements**

UH4. The product shall be able to be used by members of the public over 10 years of age with no previous training. Ninety percent of a sample test panel of the public over 10 years of age shall be able to start a game and move a piece within 5 minutes of accessing the product.

UH5. Instructions shall be in place to teach the user the rules of the game. Eighty percent of a sample test panel of the public over 12 years of age shall understand how to move each piece within 1 hour of accessing the product.

#### **5.2.4 Understandability and Politeness Requirements**

UH6. All symbols and words shall be naturally understandable by the user community and shall be similar to historically used Chess symbols [3].

#### **5.2.5 Accessibility Requirements**

UH7. System shall follow guidelines for correct colour contrast ratio for text to the background as stated in the WCAG [1].

## **5.3 Performance Requirements**

### **5.3.1 Speed and Latency Requirements**

- PR1. The average time between player input and visual game response shall be less than 0.25 seconds.
- PR2. The maximum time between player input and visual game response shall be less than 2 seconds.
- PR3. The maximum time between a player input and that player's timer pausing shall be less than 0.1 seconds.

### **5.3.2 Safety-Critical Requirements**

- N/A. The system shall have no safety-critical components or hazards.

### **5.3.3 Precision or Accuracy Requirements**

- PR4. All timer values shall be accurate to one decimal place.
- PR5. All material score values will be accurate to the nearest integer.

### **5.3.4 Reliability and Availability Requirements**

- PR6. The product shall be available for use 24 hours per day, 360 days per year on average.

### **5.3.5 Robustness or Fault-Tolerance Requirements**

- PR7. The product shall reconnect the player to the game they were playing if they lose connection to the server.

### **5.3.6 Capacity Requirements**

- PR8. The product shall require a minimum amount of 2GB of RAM to function effectively.

## **5.4 Operational and Environmental Requirements**

### **5.4.1 Installability Requirements**

- N/A. The system shall require no installation for access and use.

### **5.4.2 Requirements for Interfacing with Adjacent Systems**

- OE1. The application shall interface with an external server to make requests to the back-end of the application.

### **5.4.3 Productization Requirements**

- OE2. The product shall be deployed to a public website where users shall have access to it.

### **5.4.4 Release Requirements**

- OE3. The product shall be tested for possible bugs and issues, and will accordingly be fixed and redeployed to the product's website on a case-by-case basis.

## **5.5 Maintainability and Support Requirements**

### **5.5.1 Maintenance Requirements**

- MS1. The product shall be maintained actively until the end of the semester.

### 5.5.2 Supportability Requirements

- MS2. The product shall come with an accessible set of instructions on the web-application and in the repository for rules of the game.
- MS3. The product shall provide users with a way of contacting the developers in order to report any bugs or issues they encounter.

### 5.5.3 Adaptability Requirements

- MS4. The program shall be accessible from any internet browser on a computer or mobile device.
- MS5. The product shall be deployed onto an external platform.

## 5.6 Security Requirements

### 5.6.1 Access Requirements

- SR1. Any user shall be able to access and play the game.
- SR2. The product repository shall only be accessible by the developers, TAs, and the professor.
- SR3. Only the developers of the project shall be able to modify the files, domain, and the database of the product.

### 5.6.2 Privacy Requirements

- SR4. There shall be no data collected on users outside of the moves they input.

## 5.7 Legal Requirements

### 5.7.1 Standards Requirements

- LR1. The product shall be developed using the Airbnb JavaScript Style Guide [4].

# 6 Project Development Phases and Risks

## 6.1 Project Schedule

A Gantt chart detailing the project schedule can be found at the [following link](#).

## 6.2 Phases of System Development

This section will describe the different phases of system development, including the content of each phase, motivation behind it, and the form that the development phase will take.

### Problem Statement

1. The problem statement includes developing a description of a problem that the system aims to address. This should include context around the problem as well as why the problem is relevant and important.
2. The problem statement is the motivation behind the development of the system.
3. The problem statement includes 3 main sections: Problem definition, problem importance, and problem context.

### Development Plan

1. The development plan consists of the method in which the project development will be carried out by the group. This includes information that would be pertinent for the development team.

2. The development plan provides an overview of the project life cycle. This is important to ensure that the development process is successful.
3. The development plan consists of a document with overviews of the following: team meeting plan, communication plan, member roles, workflow, proof of concept demonstration, technology, coding style, project schedule, and project review.

#### **Requirements Document Revision 0**

1. The purpose of the SRS phase is to create an artifact which specifies the system requirements and how the system interacts with the end user.
2. The development of the SRS document will identify use cases, requirements, constraints, and other important design considerations.
3. The SRS includes a description of the project scope, stakeholders, use cases, functional and non-functional requirements, project plan and issues.

#### **Proof of Concept Demonstration**

1. This is the first implementation phase of the project, where a low-functionality version of the product will be developed and demonstrated.
2. The proof of concept demonstration is important to ensure that the client is able to see visual progress on the product, and proves that the idea behind the system is sound.
3. The proof of concept demonstration includes implementation of the system at a low level. This includes the base game logic, some basic display format, and connection between players.

#### **Test Plan Revision 0**

1. The test plan includes developing a plan for testing the system against the requirements and to ensure robustness and proper functionality.
2. Testing will detect errors and fix bugs in the program, as well as ensure that requirements are being properly met.
3. The test plan will include a structure for how different testing will be performed on the system. This includes a traceability matrix to test against requirements, as well as descriptions of and identification of different test cases using various testing methods.

#### **Design & Document Revision 0**

1. The Design & Document Revision 0 will provide the information around the design of the system.
2. The Design & Document Revision 0 is required to make sure that the project conforms to a proper development process, and the designed product is of high quality. This initial revision will provide a measure of how much of the project was done properly, and how much was done improperly.
3. The revision will be a presentation of the developed artifacts and would consist of a discussion with the group members that address the contributions of the individuals.

#### **Revision 0 Demonstration**

1. The revision 0 demonstration phase includes an additional design and functional implementation of the project description.
2. The revision 0 demonstration is the motivation behind guaranteeing that the project is slowly getting towards the final product, after creating the skeletal version of the project in the proof of concept phase.
3. The revision 0 demonstration includes the following sections: a basic front-end design, and a majority of the game functionality.

#### **Final Demonstration**

1. The final demonstration phase includes the visual and functional implementation of the project description.
2. The final demonstration is the motivation behind ensuring that the implementation functions in the desired manner based on the previous phases.
3. The final demonstration includes the following sections: the front-end design, and the game functionality. The project is planned to be designed with the core functionality of the game, such as the ability to play with another player. Additional features such as including a chat window the interact with the other player, and the ability to play with an AI, are features that will only be included if the project is not under a strict time constraint.

#### **Peer Evaluation of Final Demonstration**

1. The peer evaluation phase includes judging the productivity of the group members and review overall project contribution.
2. The peer evaluation is the motivation behind ensuring that the all group work that was assigned and completed up to expectations.
3. The peer evaluation includes each group members' contributions, and how important they were to the success of the project.

#### **Final Documentation**

1. The final documentation phase includes the detailed description of the problem and its solution based on the final implementation of the project.
2. The final documentation is the motivation behind addressing the final results of the development of the system.
3. The final documentation includes the following sections: Problem definition, problem statement, the design process, the solution, and the future considerations of the project.

### **6.3 Risks**

The following risks are most likely to occur in the development of our project.

1. **Inadequate measurement:** There is a risk that certain capacities will be measured inaccurately, such as server capacity. This will likely be due to a lack of resources and a lack of knowledge around best practices.
2. **Cancelled features:** There is a risk that intended features of the product may not be completed within the time-frame for the project. This could cause the design of the system to be worse than if the product was designed without those features in mind. This risk is likely to occur, specifically with the AI feature of the product.
3. **Error prone:** A lack of testing in the project might lead to more errors and unexpected results. It is important to discover as many bugs as possible before the final demonstration, which causes the software to be more reliable and easy to use. There is a risk that there may be uncaught errors and bugs in the program that could negatively affect the user experience.