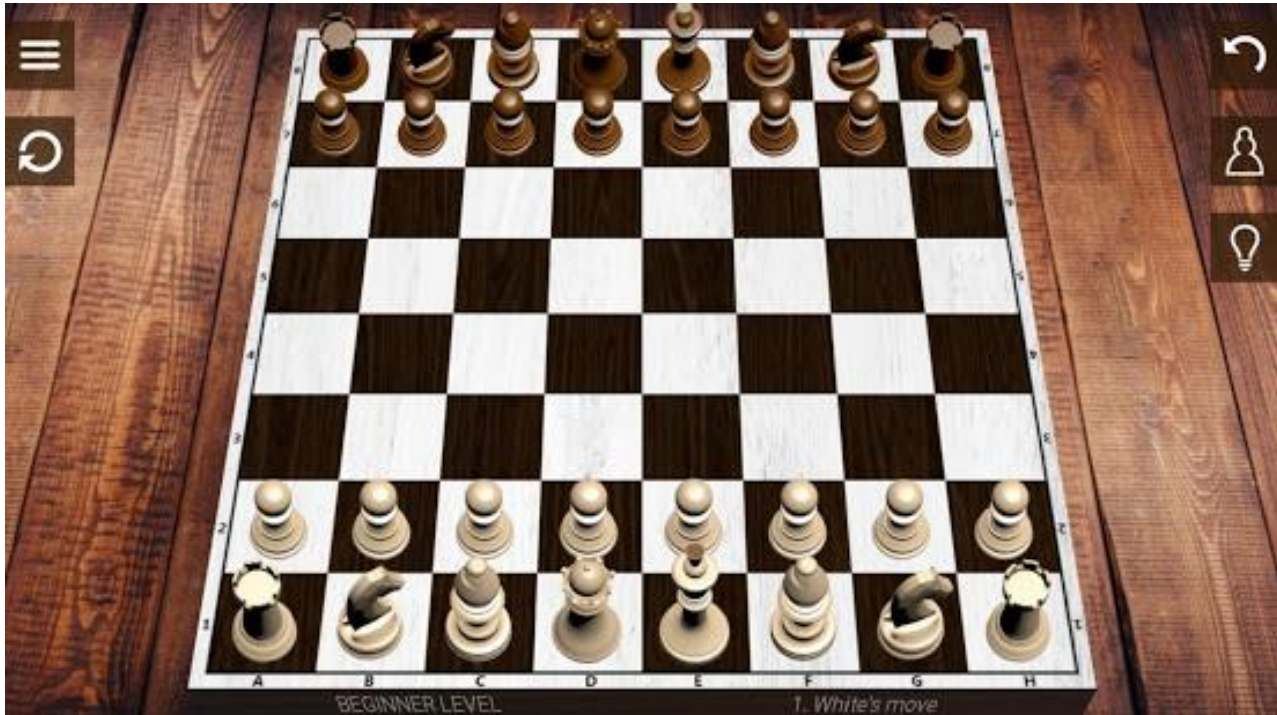


# Software Engineering Project Report



## Age Of Chess

Prepared by  
Jake Campbell, Onkar Dangi, Sergio Ruelas, and Steven Karall  
(GROUP # 15)  
for CS 440

at the  
**University of Illinois at Chicago**

Fall 2020

# Table of Contents

	List of Figures	7
	List of Tables	8
I	Project Description	9
1	Project Overview	9
2	The Purpose of the Project	9
	2a The User Business or Background of the Project Effort	9
	2b Goals of the Project	9
	2c Measurement	10
3	The Scope of the Work	10
	3a The Current Situation	10
	3b The Context of the Work	10
	3c Work Partitioning	11
	3d Competing Products	12
4	The Scope of the Product	13
	4a Scenario Diagram(s)	13
	4b Product Scenario List	14
	4c Individual Product Scenarios	14
5	Stakeholders	15
	5a The Client	15
	5b The Customer	15
	5c Hands-On Users of the Product	15
	5d Maintenance Users and Service Technicians	15
	5e Other Stakeholders	16
	5f User Participation	16
	5g Priorities Assigned to Users	16
6	Mandated Constraints	16
	6a Solution Constraints	16
	6b Implementation Environment of the Current System	16
	6c Partner or Collaborative Applications	17
	6d Off-the-Shelf Software	17

6e	Anticipated Workplace Environment	17
6f	Schedule Constraints	17
6g	Budget Constraints	17
7	Naming Conventions and Definitions	17
7a	Definitions of Key Terms	18
7b	UML and Other Notation Used in This Document	18
7c	Data Dictionary for Any Included Models	19
8	Relevant Facts and Assumptions	19
8a	Facts	19
8b	Assumptions	19
II	Requirements	20
9	Product Use Cases	20
9a	Use Case Diagrams	20
9b	Product Use Case List	21
9c	Individual Product Use Cases	21
10	Functional Requirements	24
11	Data Requirements	26
12	Performance Requirements	27
12a	Speed and Latency Requirements	27
12b	Precision or Accuracy Requirements	27
12c	Capacity Requirements	28
13	Dependability Requirements	28
13a	Reliability Requirements	28
13b	Availability Requirements	28
13c	Robustness or Fault-Tolerance Requirements	29
13d	Safety-Critical Requirements	29
14	Maintainability and Supportability Requirements	29
14a	Maintenance Requirements	29
14b	Supportability Requirements	29
14c	Adaptability Requirements	30
14d	Scalability or Extensibility Requirements	30

14e	Longevity Requirements	30
15	Security Requirements	31
15a	Access Requirements	31
15b	Integrity Requirements	31
15c	Privacy Requirements	31
15d	Audit Requirements	32
15e	Immunity Requirements	32
16	Usability and Humanity Requirements	32
16a	Ease of Use Requirements	32
16b	Personalization and Internationalization Requirements	33
16c	Learning Requirements	34
16d	Understandability and Politeness Requirements	34
16e	Accessibility Requirements	34
16f	User Documentation Requirements	35
16g	Training Requirements	35
17	Look and Feel Requirements	35
17a	Appearance Requirements	35
17b	Style Requirements	35
18	Operational and Environmental Requirements	36
18a	Expected Physical Environment	36
18b	Requirements for Interfacing with Adjacent Systems	37
18c	Productization Requirements	37
18d	Release Requirements	38
19	Cultural and Political Requirements	38
19a	Cultural Requirements	38
19b	Political Requirements	39
20	Legal Requirements	39
20a	Compliance Requirements	39
20b	Standards Requirements	39
21	Requirements Acceptance Tests	40
21a	Requirements – Test Correspondence Summary	40

	21b Acceptance Test Descriptions	41
III	Design	42
22	Design Goals	42
23	Current System Design	43
24	Proposed System Design	43
	24a Initial System Analysis and Class Identification	43
	24b Dynamic Modelling of Use-Cases	44
	24c Proposed System Architecture	44
	24d Initial Subsystem Decomposition	45
25	Additional Design Considerations	45
	25a Hardware / Software Mapping	45
	25b Persistent Data Management	46
	25c Access Control and Security	47
	25d Global Software Control	47
	25e Boundary Conditions	47
	25f User Interface	47
	25g Application of Design Patterns	47
26	Final System Design	49
27	Object Design	49
	27a Packages	49
	27b Subsystem I	49
	27c Subsystem II	49
	27d etc.	49
IV	Project Issues	50
28	Open Issues	50
29	Off-the-Shelf Solutions	50
	29a Ready-Made Products	50
	29b Reusable Components	50
	29c Products That Can Be Copied	50
30	New Problems	50

30a	Effects on the Current Environment	50
30b	Effects on the Installed Systems	50
30c	Potential User Problems	51
30d	Limitations in the Anticipated Implementation Environment That May Inhibit the	
30e	Follow-Up Problems	51
31	Migration to the New Product	51
31a	Requirements for Migration to the New Product	51
31b	Data That Has to Be Modified or Translated for the New System	52
32	Risks	52
33	Costs	52
34	Waiting Room	53
35	Ideas for Solutions	53
36	Project Retrospective	54
V	Glossary	54
VI	References / Bibliography	55
VII	Index	55

## **List of Figures**

Figure 1 - Context of the work Diagram	11
<u>Figure 1</u> - Context of the work Diagram	9
Figure 2 - Multiple Player Game Scenario	12
Figure 3 - Single Player Move Scenario	13
Figure 4 - Board Design	17
Figure 5 - 4 Player Game Use Case	19
Figure 6 - User and Gameplay UML Diagram	42
Figure 7 - Server Connection and Make Move Use Case	43
Figure 8 - Subsystem Decomposition UML	44
Figure 9 - Game Scenario Map	45
Figure 10 - User Interface	46
Figure 11 - Final System Design UML	47

## **List of Tables**

Table 1 - Event List

12



# **I Project Description**

Age of Chess will be a 3 to 4 player version extension of a regular chess game, with two additional additions in regards to a new game piece and safe spaces. This game will allow chess to be played by more than two players at a single time, which will allow for families and larger groups of two people to be able to play the game together vs taking turns with the two-player version. Thus, making your next get together way more fun.

## **1 Project Overview**

This project will expand the traditional 2 player versions of chess to allow 3 and 4 player versions of a chess game. this game will also make two changes to the original game board. there will now be a redesigned board to accommodate addition of two more players. This game will also include a new “monk” game piece which will replace one rook and one knight on each of the players' teams. and also, two spaces on each player's side of the board will have safe spaces.

## **2 The Purpose of the Project**

The purpose of this project is to create a fresh new version of a game that people already really enjoy. This game will also allow the people that enjoy chess to now play in bigger groups than the original version. Also, families and people in general seeking a game night will now be able to include chess in the selection because there's this version which will allow the family or people in general to play together. Any game is more enjoyable if there are more people playing it, hence a more fun version of the traditional chess game.

### **2a The User Business or Background of the Project Effort**

This project will create a very diverse and exciting way to play the game of chess. This application will allow for larger amounts of users to be able to play simultaneously and will be able to have a different and more fun experience than a traditional game of chess. This game will allow for various types of gameplay possibilities from the users. The different possibilities will be either a 2,3, or 4 player game. and this game will benefit any group of people that enjoy chess and want to be able to break away from the limitation of only being able to play against just one other player.

### **2b Goals of the Project**

the goals of this project are as follows:

- to provide an immersive and new experience to avid chess players
- to allow players to break that barrier of just playing this game with two people simultaneously and allow them to play with up to four players.
- to be able to give the players a more diverse game requiring very different strategies than a typical chess game would, with the auditions of the safe spaces and the game

piece called the “monk”.

- to offer an upgraded version of chess that can't be found anywhere else on the market.
- to be able to establish a better relationship for groups by giving them a very different game than is on the market, that they can now play together.
- to be able to offer a game that players will have an enjoyable enough experience in that they play frequently with other players.

## **2c Measurement**

The way we will be able to measure when a goal has been met is by the experience that a group of players has with the game itself. if the group finds their experience enjoyable, immersive, is an upgrade in terms of a typical 2 player version of chess, and desires to play this game again. then we can say we met the goals of the project.

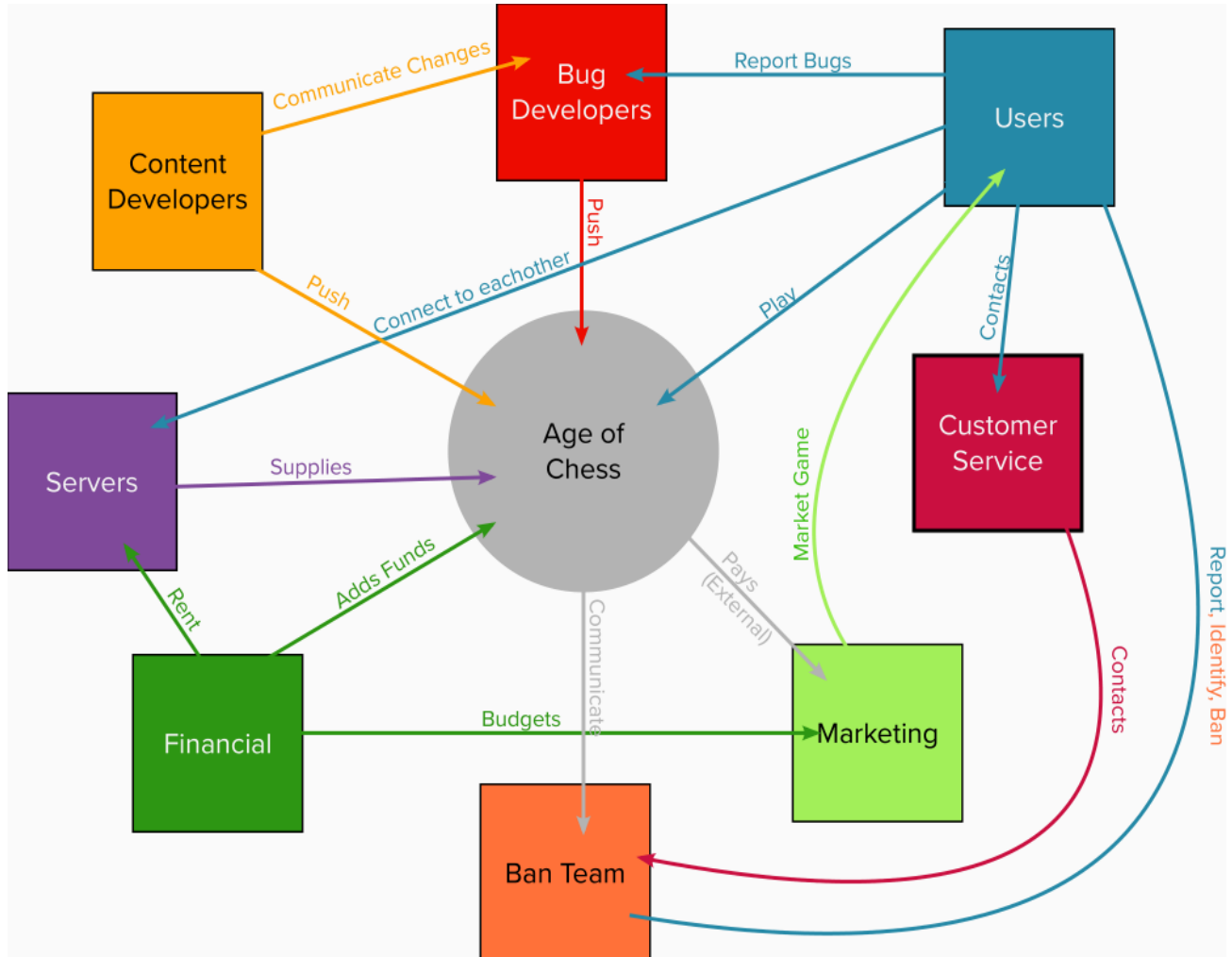
## **3 The Scope of the Work**

The enabling of a specific user base and monetization of said user base. In this case, the user base can be considered casual gamers and avid users of mobile technology.

### **3a The Current Situation**

The client currently exists in a very specific niche list of applications. Only a select few games in an equivalent market exists with respectable popularity.

### **3b The Context of the Work**



### 3c Work Partitioning

#### Business Event List

Event Name	Input and Output	Summary
1. Initial Release	Initial Release Customer interaction (out)	Monitor server load on initial release.
2. Low Server Load	Low Server Load (in)	Release rented servers to minimize cost.

3. High Server Load	High Number of Customers (in)	Purchase more servers to minimize performance issues with clients.
4. Bug Squashing	Bug Squashing (out)	Identify bugs reported or found by debugging team and mitigate.
5. Customer Service / Customer Complaint	Customer Complaints (in)	Manage customer complaints and mitigate if possible
6. Cheating	Identify cheating players (in)	Either by report or by software identification, ban or punish cheaters
7. Content Update	Time (in)	Identify small and meaningful content update paths and push to keep content fresh
8. Server Downtime	Disable Access (out)	Server downtime plan for middle of the night to update and reboot systems.
9. Community Events	Seasonal or temporary events (out)	Do community events ever so often to keep the community engaged with challenges

### 3d Competing Products

There exist several competing products. The constraining factors of this sector is how much time the client has to share as well as the disposable income allotted. Both are a limited resource and aside from similar games/products, there exist applications that can take time away from the market as well as allotted disposable income. Currently, as the market exists today, the most popular game and the most popular application currently dominate the market. The game, Among Us, is the most popular mobile “party” like game that exists with 800M monthly active users as of September of 2020, according

to a wear social report. The most popular application overall is Facebook with 2.5B monthly active users by the same report. The interesting thing about such a market is that it is very fluid. It changes much based on recent popularity aside from the very top performers such as Facebook. For example, Among Us was released in 2018 to a tiny audience. Just recently as a few months ago popularity skyrocketed by several factors.

What justifies a new product in the market is to take away popularity from other popular applications and drive them towards a new product. The inherent flaw in such a market is that such popularity does not last forever, and a new and exciting product can take a respectable hold in said market.

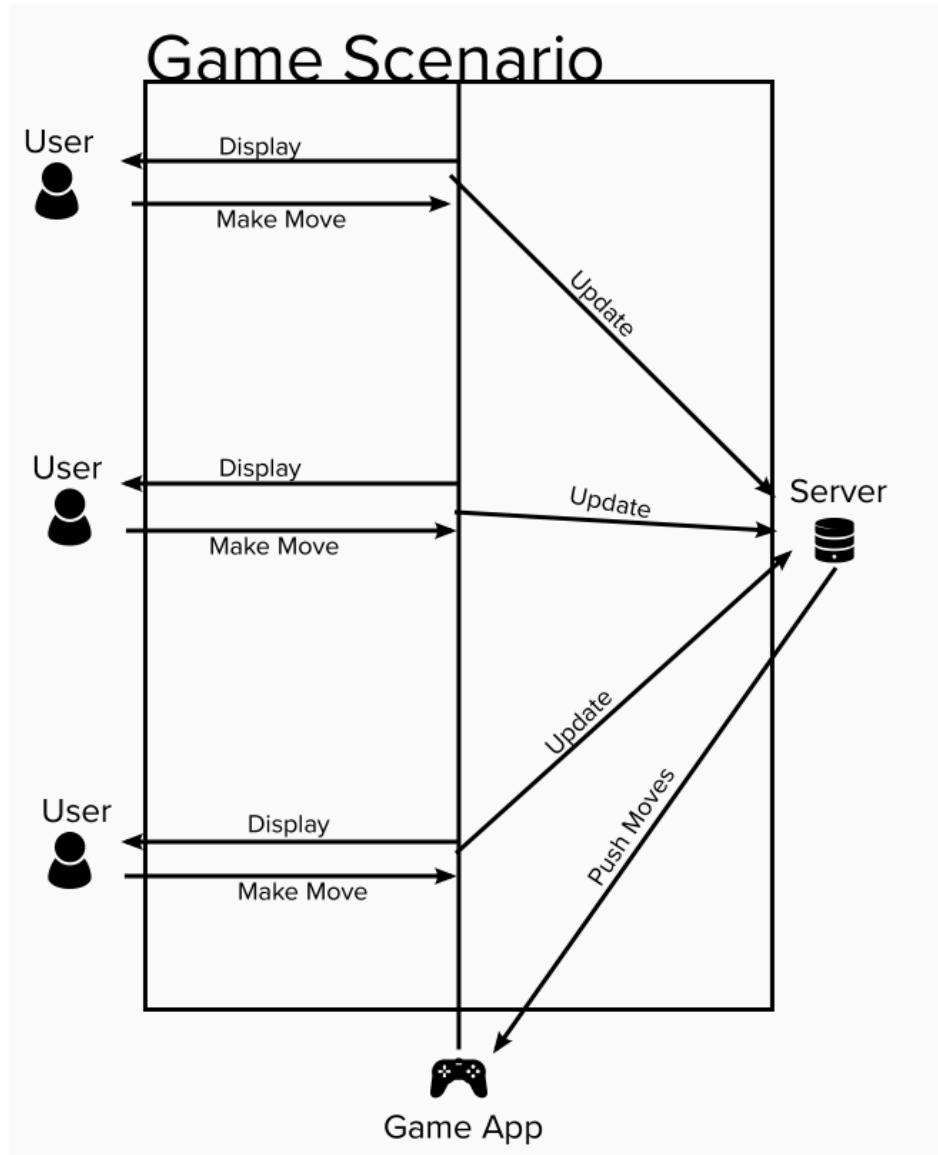
The most important motivation for driving such a following is to coalesce a community around the product. To help contribute to the “hype” or seemingly concrete popularity, quick moves encouraging social gaming is an absolute must.

Consider existing APIs that exist to help drive sociability. Connecting the game to various social media platforms, such as Facebook, can help drive that popularity. Considering the fact that Facebook has such a large user base, Facebook users can be considered a unique and important factor in driving popularity. Other APIs exist to help with sociability as well. APIs such as an embedded ability to communicate with other players. Communication via text or speech would be preferable, though the latter does introduce many issues that cannot be simply solved, such as limiting harassment or hate speech, especially to the target audience which will include minors. One other important consideration is how player lobbies are chosen, implementing these social media platforms and perhaps an existing API to randomly choose players would contribute to a much more efficient development experience.

#### **4 The Scope of the Product**

Briefly described, the work handled by the product includes player/user interaction with the game along with other players/users. The game is approximately 90 minutes and is designed to handle that allotted time; though not sequentially. Users may purchase additional content which the product entices. The product must also control interaction and limit undefined or unwarranted behavior (such as cheating).

##### **4a Scenario Diagram(s)**



#### 4b Product Scenario List

1. two player mode
2. three player mode
3. four player mode

#### 4c Individual Product Scenarios

game scenario 1: scenario one will be just a two-player mode with just two people playing against each other. This scenario will still contain the additions of two monk pieces and two safe spaces for each of the two players.

game scenario 2: scenario two will be a three-player mode with three people playing against each other. This scenario will still contain the additions of two monk pieces and two safe spaces for each of the three players

game scenario 3: scenario three will be the max number of players that can play against each other which will be four total players. This scenario will still contain the additions of two monk pieces and two safe spaces for each of the four players

## **5 Stakeholders**

### **5a The Client**

We will be guiding the development of the game ourselves. We all have interest in the project and are committed to the implementation of our new elements to the game, those being the new board that allows for 3 and 4 player chess, the safe space, and the Monk piece

### **5b The Customer**

The people intended to buy the product are extremely diverse. Chess is a beloved game in our society and people of all ages play the game. So the people who will be purchasing the game will fall in different categories. We will have kids who will be playing this game with their friends. Chess is a strategic game which enhances the thinking skills of a person so middle school might consider adding this version of the game in their curriculum. Other categories will include people buying the game for leisure activities. With the global pandemic, Age of chess is designed to enhance the quality of life of individuals without leaving the comfort and safety of their home and enjoy the classic game of chess with their friends. People who are retired often find themselves struggling with boredom. Age of chess will offer them to connect with their long-lost friends over a few games of chess.

### **5c Hands-On Users of the Product**

Hands on users for our game will mostly be the same people who will be purchasing our product. Schools might purchase the game for their students, so a large number of students will be using this game to enhance their strategic skills and play with their friends at the very same time. People hosting get together or small family events will be purchasing the game but the different people in there will be playing the game. This game could be a way to connect to a lot of your distant friends and bond over Age of chess.

### **5d Maintenance Users and Service Technicians**

The development team will be responsible for the maintenance of the product for the next five years. This period includes all the updates that will be released and the services for the product.

Once purchased, the product comes with an installation guide to help the user install the game on their devices. This guide will be designed so as to be used by anyone from novice to an expert in installation.

### **5e Other Stakeholders**

Websites that host online Chess games will want a version of our product that can be hosted on their site as another option for play.

### **5f User Participation**

Beta testers will be used to debug and fine tune the local and online multiplayer experience on mobile devices.

### **5g Priorities Assigned to Users**

Primary Users - Since this is a consumer-focused product, all users that play the game on our app will be primary users. Users will also have access to a forum where they can discuss strategies and share moments from play. The users may also voice their opinions on the products mechanics and report errors encountered during use.

Secondary Users - Users that use the forum, but play on other online sites or play physical versions of the game.

## **6 Mandated Constraints**

### **6a Solution Constraints**

Description: The general constraints set by the clients are that this version of the game needs to be 3-4-person game and significantly different from the traditional chess.

Rationale: The product should offer cross platform functionality while supporting the most primitive modern operating systems.

Fit Criterion: The product should be easy to understand/learn so that a majority of the population can play with ease

### **6b Implementation Environment of the Current System**

This product is designed to engage the majority of the population so the implementation



is designed in such a way that the game works on all the operating systems and even offers cross platform functionality.

### **6c Partner or Collaborative Applications**

This product will not use any type of partner or collaborative applications.

### **6d Off-the-Shelf Software**

This product will not use any type of Off-the-Shelf Software.

### **6e Anticipated Workplace Environment**

The user can connect to other devices both locally and over the internet, so the environment is entirely dependent on the user's preferences.

### **6f Schedule Constraints**

this application would get the biggest amount of people playing it during the winter months given in some areas it's too cold and unpleasant to do most or any activities outside. Having this application ready just before winter begins typically around the month of September would give it the best chance for success.

- If this application is not done by the winter time there is a possibility that a low number of people use it because of the available options of activities outside during the summer months, which could impact the initial revenue stream this game could produce.
- The financial impact of this product not being completed just before the winter months can reduce revenue for this game and hurt the possibility of having a growing player base

### **6g Budget Constraints**

There will be a team of 4 back end developers working on this project. there will also be a team of 4 front end developers that will also be the quality assurance tester as well. there will also be a network engineer that will take care of proper functionality between the server and client side of this project.

- Back-end developer:  $60,000 \times 4 = 240000$
- front-end developer:  $55,000 \times 4 = 220000$
- network engineer: 60,000
- project manager: 90,000

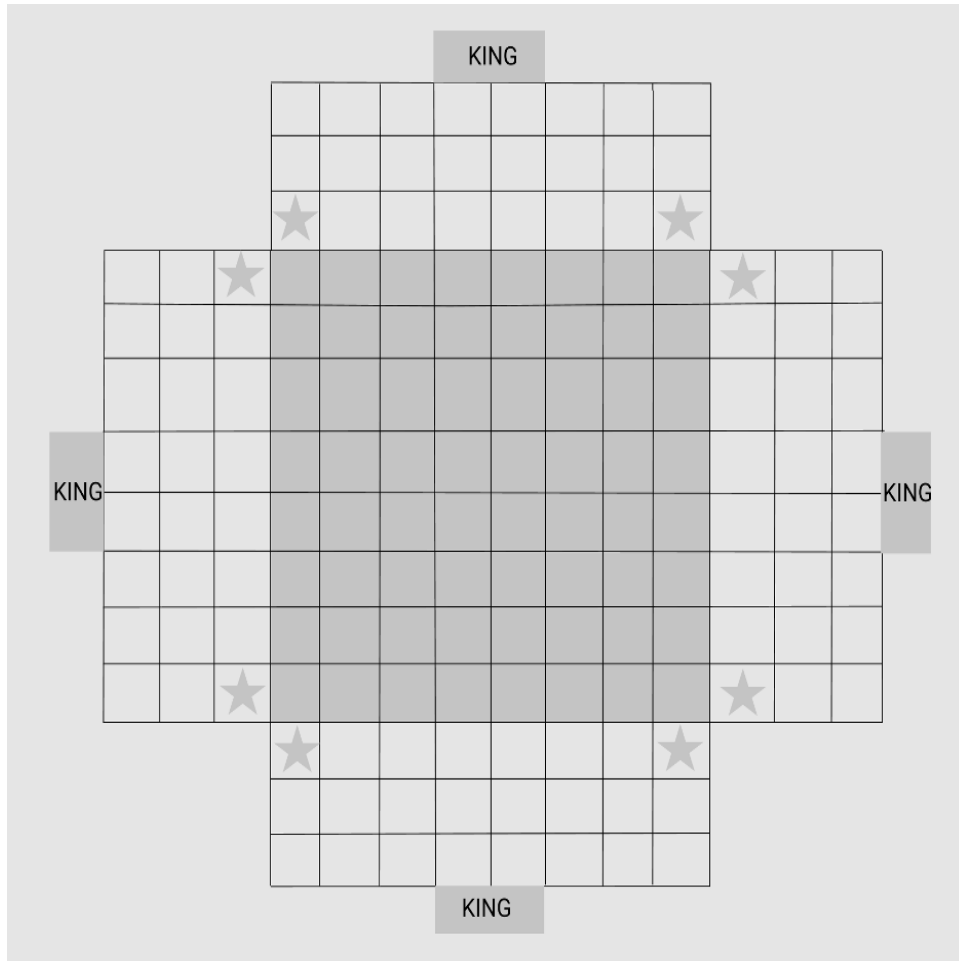
total estimated cost for this project is: 610,000

## **7 Naming Conventions and Definitions**

### **7a Definitions of Key Terms**

- The pawn: will be able move only one space at a time, except on its first move can move twice
- The Bishop: will be able to move as many spaces as the player desires diagonally
- The Knight: will be able to move in a L-shape with two squares left or right horizontally and then the last and third square either up or down.
- The Rook: will be able to move as many spaces as the payer desires vertically or horizontally
- The Queen: will be able to move as many spaces as the player desires either vertically, horizontally, or diagonally
- The King: will be able to move only on space at a time, but once defeated that player loses
- The Monk: this game piece will only move one space per turn, but it will have the ability to take the place of any defeated piece on the board. The monk has the ability to convert up to two opposing Pawns from the other team to be on your team. The rules for this are the opposing piece has to be at most two spaces away from the monk piece.
- Safe spaces: This game will have a total 8 spaces on the board that are safe spaces. And in these safe spaces, you can stay safe for a maximum of two moves. The main reason for the introduction of safe spaces was to add more strategic complexity to the game.

### **7b UML and Other Notation Used in This Document**



### 7c Data Dictionary for Any Included Models

Data Structures and Data properties related to this project include a grid that will simulate the spaces of the board, and other specific data variables to simulate the game pieces.

## 8 Relevant Facts and Assumptions

### 8a Facts

- On average, it will take approximately 90 minutes to complete one game of age of chess
- Each player will have two minutes to make a move to run the game efficiently and smoothly
- The game will function the best when every device connected has a stable internet connection

### 8b Assumptions

- The game is designed to operate on all platforms but it is assumed that the customer

- has the updated version of their devices.
- It is assumed that the devices connected to the game have a stable connection at all times.
- It is assumed that the player playing the game has a basic knowledge of the classic game of chess

## **II Requirements**

### **9 Product Use Cases**

#### **9a Use Case Diagrams**



Additional Actors: Minimum 2 additional players. Maximum 3 additional players.
<p>Sequence of Events:</p> <ol style="list-style-type: none"> <li>1. Player 1 clicks on "Join Game" <ul style="list-style-type: none"> <li>System verifies player's account</li> <li>System sends a find server request.</li> <li>System allocates new server if existing server marked as seeking Players does not exist.</li> <li>System marks server as full when Player 2, Player 3, and optionally Player 4 joins server.</li> </ul> </li> <li>2. Player attempts move. <ul style="list-style-type: none"> <li>System checks move logic.</li> <li>System updates server with move.</li> <li>System checks for win condition.</li> <li>System asks next player for move.</li> </ul> </li> <li>3. Player 2/3/4 makes move. <ul style="list-style-type: none"> <li>System checks move logic.</li> <li>System updates server with move.</li> <li>System checks for win condition.</li> <li>System announces Player wins.</li> </ul> </li> <li>4. Player exits game. <ul style="list-style-type: none"> <li>System deallocates server.</li> </ul> </li> </ol> <p>Alternatives: Player tries to join game with no account. In this case, player is prompted to create an account and then attempt to join a game.  Exceptions: Player disconnects from server while attempting to join server. All players end up in a tie state and none win. No servers exist to allocate. Player disconnects from server after being connected.</p>

<p>Use case ID: P2                      Name: Create Account</p> <p>pre-conditions: Player does not have an account. Player clicks on "Create Account"</p> <p>post-conditions: Player has an associated account on the server.</p> <p>Initiated by: Player</p>
--

Triggering Event: Player clicks on "Create Account" Additional Actors: None
Sequence of Events: <ol style="list-style-type: none"> <li>1. Player clicks on "Create Account"</li> <li>2. System shows create account screen asking for player information.</li> <li>3. Player enters information.</li> <li>4. System confirms valid input email and that email is not already in use.</li> <li>5. Player proceeds to log in with newly created account</li> <li>6. System logs the user in confirming a successful account creation.</li> </ol>
Alternatives: None Exceptions: Player enters invalid information. Player already has an account.

Use case ID: S1                      Name: Ban pre-conditions: Account associated with a Player has been flagged for cheating. post-conditions: Account is no longer allowed to play. Initiated by: System, Player Triggering Event: System received cheating complaint. Additional Actors: None
Sequence of Events: <ol style="list-style-type: none"> <li>1. Player is reported for cheating.             <ul style="list-style-type: none"> <li>System checks game in question.</li> <li>System verifies if cheating occurred.</li> <li>System disallows account from logging in.</li> </ul> </li> <li>2. Player attempts to log in to play.             <ul style="list-style-type: none"> <li>System reports error "due to a ban..."</li> </ul> </li> </ol>
Alternatives: System acknowledges cheating report, but no cheating was found. Exceptions: System flags account for cheating, but no cheating occurred.

Use case ID: S2                      Name: Servers Full
---

pre-conditions: All servers are full. post-conditions: Estimated wait time is broadcasted. Initiated by: Player Triggering Event: Player initiated P2. Player is attempting to join a game. Additional Actors: System Administrators.
Sequence of Events: <ol style="list-style-type: none"> <li>1. Player attempts to join a new game. <ul style="list-style-type: none"> <li>System attempts to allocate server.</li> <li>Severs are all busy.</li> <li>System reports occurrence to a log.</li> <li>System finds game with the most moves currently played.</li> <li>System sends estimated wait time to Player.</li> </ul> </li> <li>2. Player waits for new server. <ul style="list-style-type: none"> <li>System finds a new server.</li> <li>System allocates server.</li> </ul> </li> <li>3. Player joins new server.</li> <li>4. System administrators look over the occurrence log.</li> </ol>
Alternatives: Player leaves while waiting for a new server. Exceptions: Player waits for a much longer time than estimated.

## 10 Functional Requirements

### **Requirement F1 - Allocated Servers**

**Description:** Servers from which to allocate from must exist.

**Rationale:** Players connect to a server from which to play from.

**Fit Criterion:** Players can play the game and interact with the server updating status constantly.

**Acceptance Tests:** Server Ping Test.

### **Requirement F2 - Server Updates Moves**



**Description:** Servers update the game as it is played.

**Rationale:** Player makes move and other players need to see the game update accordingly.

**Fit Criterion:** Player makes move. Other players see what move the player made.

**Acceptance Tests:** Game UI Test.

### **Requirement F3 - Server Maintains Connection**

**Description:** Servers allocated to 3-4 players must maintain connection with those players throughout the game.

**Rationale:** Players that disconnect from the server can no longer play.

**Fit Criterion:** Throughout the course of the game, a connection between the server and the players is maintained until an explicit disconnect is initiated by a win/lose/tie condition.

**Acceptance Tests:** Server Ping Test (continuously)

### **Requirement F4 - User Interface**

**Description:** A user interface is presented to the player.

**Rationale:** Players use this interface to initiate commands such as “sign in”, “create account”, and “play game”.

**Fit Criterion:** Players open the game and are presented with an interface from which they can interact with.

**Acceptance Tests:** Front-End UI Test

### **Requirement F5 - Server Ban**

**Description:** Servers must not allow a banned account to access game servers.

**Rationale:** Unwanted players can be a nuisance and possibly corrupt the integrity of the game or systems therein.

**Fit Criterion:** Player’s associated account has been banned. On attempting to log in, the system denies access and updates the player of their ban.

**Acceptance Tests:** Server Access Test

## 11 Data Requirements

### **Requirement D1 - Account Information**

**Description:** Account information is encrypted and stored for each player. User entered "Account Information" includes: name, username, and email address. System entered information includes: ID, Games played, and is banned flag.

**Rationale:** This information is needed to limit the number of duplicate players since they must enter an email address. Name is used for system or system administrators to refer to when communicating with players. Username is used as a unique identifier for players amongst themselves. ID is created by the system to provide easier reference. Games played is used as a reference for other systems.

**Fit Criterion:** New user creates an account and enters the required information. Name is limited to characters only and not numbers. Name is limited in length to 20. Username must be unique to other players. Username may include numbers as well as characters. Special characters are limited. Cannot be longer than 10 in length. Email address must include '@' symbol and end with known domains such as .com/.org/.io/etc. Banned flag is a Boolean created by system. ID must be numerical only dictated by system. Games played is numeral only typically not allowed to overflow.

**Acceptance Tests:** Account Creation Test.

### **Requirement D2 - Encrypted Storage**

**Description:** Account information must be encrypted and stored in a hash table.

**Rationale:** Bad actors may infiltrate systems and gain access to player information. Email address is valuable.

**Fit Criterion:** Pen testing results in unusable data to attackers.

**Acceptance Tests:** Server Pen Test.

### **Requirement D3 - Redundant Storage**

**Description:** Servers which host account information must be backed up weekly.

**Rationale:** Data corruption, server failure, etc can occur leading to a loss in account information.

**Fit Criterion:** A server with account information hosted fails and the backup restores the data.

**Acceptance Tests:** Data Corruption Test.

## **12 Performance Requirements**

### **12a Speed and Latency Requirements**

#### **Requirement P1**

**Description:** Player moves are updated at most 5 seconds after a player confirms their move.

**Rationale:** In order to minimize the length of time required to play a four person game, the system needs to provide players with the most current data about the board and the pieces, so they can make informed decisions in a timely manner.

**Fit Criterion:** Most legal moves are updated within 3 seconds of them being confirmed.

#### **Requirement P2**

**Description:** In game players private messages are updated at most 4 seconds after they are sent.

**Rationale:** In order to maintain player interaction and strategy, messages between players must be exchanged with priority so as to not hinder players' deal making and planning.

**Fit Criterion:** Players will receive most messages within 2 seconds of them being sent.

### **12b Precision or Accuracy Requirements**

#### **Requirement P3**

**Description:** Position of pieces must be exact.

**Rationale:** The position of pieces on the board is critical information that players need to know to strategize.

**Fit Criterion:** Pieces will be in the center of the square they occupy and will only be placed there when moved by a player.

#### **Requirement P4**

**Description:** Pieces are properly represented.

**Rationale:** Pieces on the board must represent the roles described earlier. A piece can then be identified, and a player will be able to make legal moves.

**Fit Criterion:** A piece will only be able to make the legal moves assigned to pieces like it.

## 12c Capacity Requirements

### Requirement P5

**Description:** The system should allow for 10,000,000 new users every hour during the time frame of 9am - 11pm.

**Rationale:** Most traffic will be from users during this time frame, so it should be able to support that number of users.

**Fit Criterion:** When the number of hourly users is less than 10,000,000 they should experience no lag time.

## 13 Dependability Requirements

### 13a Reliability Requirements

#### Requirement R1

**Description:** The system must support up to 4 players at a time.

**Rationale:** In order to play 4-player the system should keep all four human players in the same game. Players that disconnect will be able to reconnect to a game. If a player disconnects and does not rejoin the game a computer will take over playing until the rest of the game unless the player reconnects.

**Fit Criterion:** A game will always have the same number of players as it did when the game started.

### 13b Availability Requirements

#### Requirement R2

**Description:** The system will be hosting games at all times. Both versus human and computers.

**Rationale:** This allows players to play at any time they wish. Either against other humans or AI. In the case that players wish to have several players higher than they have humans to play, the AI may play as any missing players.

**Fit Criterion:** A player may play at any time the wish and always have up to three other players.

### **13c Robustness or Fault-Tolerance Requirements**

#### **Requirement R3**

**Description:** The game will only work if the user is connected to the internet.

**Rationale:** Since the system hosting all the games requires the player to connect to the internet to find or join a game, any player that cannot connect cannot play.

**Fit Criterion:** A player cannot play without the internet.

### **13d Safety-Critical Requirements**

#### **Requirement R4**

**Description:** Players will be warned not to disclose any personal information in their online interactions.

**Rationale:** In order to protect the users from any potential abuse of player interactions, players will be warned to not use personal information in their usernames and passwords as well as not to share any information when using the private messenger.

**Fit Criterion:** When a player creates an account and when they open their private messenger for the first time in a game, they will be warned against sharing private information.

## **14 Maintainability and Supportability Requirements**

### **14a Maintenance Requirements**

#### **Requirement M1**

**Description:** The system can be accessed by members of the development team and any persons given special permissions.

**Rationale:** Only developers and personal given status should be allowed to have access to the system to perform any type of maintenance that has been identified as a priority to handle.

**Fit Criterion:** Unless given permission, no one aside from the developers may have access to modify the system.

### **14b Supportability Requirements**

### **Requirement M2**

**Description:** The system will be supported by a forum.

**Rationale:** The system has linked forum pages that are used for discussion, bug-reporting, social interaction, tutorials, constructive feedback, and FAQ's.

**Fit Criterion:** If a player wants to find something out about the system, they can look for an answer in the forum or post a question.

## **14c Adaptability Requirements**

### **Requirement M3**

**Description:** The system will be able to be played on any device that has a web browser.

**Rationale:** The system can be played on any device that can connect to the internet, because the game is hosted online, not through an application.

**Fit Criterion:** All devices that can connect to the internet will be able to play Age of Chess.

## **14d Scalability or Extensibility Requirements**

### **Requirement M4**

**Description:** There is room for new custom pieces to be added to the game.

**Rationale:** Age of Chess will launch with the introduction of one new piece to the game, but the developers have the freedom to introduce new pieces into the game.

**Fit Criterion:** New pieces may be added to Age of Chess if the developers decide to add new pieces.

## **14e Longevity Requirements**

### **Requirement M5**

**Description:** With proper maintenance the system can be run for at least another 5 years.

**Rationale:** With support from the developers, once Age of Chess is launched it only needs assistance if an error is made. It should be rather autonomous.

**Fit Criterion:** Age of Chess will be able to be accessed for 4 years with little to no upkeep from the developers.

## 15 Security Requirements

### 15a Access Requirements

#### **Requirement S1**

**Description:** Only the software developers and the maintenance team should have the access to the database, the source code of the game.

**Rationale:** Sensitive information should be kept private about the game.

**Fit Criterion:** Only the software developers and the maintenance team will have the access to the database, the source code of the game so as to release new versions of the game and fix the bugs regularly.

#### **Requirement S2**

**Description:** The users should have no control over the implementation of the game.

**Rationale:** The game should not be modified by the users.

**Fit Criterion:** The game will not provide any option to change the implementation of the game thus preserving the original game.

### 15b Integrity Requirements

#### **Requirement S3**

**Description:** The game shall report for bugs as soon as the users run into one so that the maintenance team can fix it.

**Rationale:** Users should be provided with an option to report for bugs.

**Fit Criterion:** The maintenance team should be notified of the bug and the team should work to fix it as soon as possible.

### 15c Privacy Requirements

#### **Requirement S4**

**Description:** The users must consent to the user agreement while installing the game.

**Rationale:** The user agreement will encapsulate everything to protect the users and the stakeholders.

**Fit Criterion:** Each user process for the game will not have access about other processes on the device, thus protecting the user.

### **Requirement S5**

**Description:** The users must consent to allowing the use of microphone and camera while starting the game.

**Rationale:** The camera and microphone should only be accessible if the user permits it.

**Fit Criterion:** Each user process for the game will not have access about other processes on the device, thus protecting the user.

## **15d Audit Requirements**

There is no requirement for transaction auditing in this version of the game.

## **15e Immunity Requirements**

### **Requirement S6**

**Description:** The game has no requirement for saving extensive personal data of the user playing the game.

**Rationale:** address, contact info, SSN should not be asked besides an email.

**Fit Criterion:** all the users will have the same privileges, so an email is enough to construct an online multiplayer version of the game.

## **16 Usability and Humanity Requirements**

### **16a Ease of Use Requirements**

#### **Requirement U1**

**Description:** The product must be designed to have an easy user interface for the users.

**Rationale:** The game should only provide the extremely necessary options right at the beginning of the screen to easily transit into the game with required settings



**Fit Criterion:** The interface will be really easy to use and will provide with an option of a tutorial.

### **Requirement U2**

**Description:** The product must be designed to have a good amount of resemblance to the original version of chess

**Rationale:** The game should be readable, and users should no difficulty initiating and playing the game

**Fit Criterion:** The interface will be extremely clear about the information and options it displays to minimize the clutter.

### **Requirement U3**

**Description:** The product must be designed to not crash when encountering errors.

**Rationale:** The game should be capable of detecting and handling the errors.

**Fit Criterion:** The application will provide a detailed report of errors and bugs encountered while runtime to the maintenance team.

## **16b Personalization and Internationalization Requirements**

### **Requirement U4**

**Description:** The product should offer language configuration for the game to incorporate more users.

**Rationale:** The game should offer most spoken languages as an option.

**Fit Criterion:** The game will offer top 10 most spoken languages as an option for the language for the game.

### **Requirement U5**

**Description:** The game should be able to work in any modern device and on both modes (landscape and portrait)

**Rationale:** When rotating between different modes of screen, the game should work perfectly and in the same way.

**Fit Criterion:** The game will be compatible in different modes and will offer an option to stay in one configuration based on user's preference.

## 16c Learning Requirements

### Requirement U6

**Description:** The application should be easily learnable, and user should not have difficulty navigating throughout the game.

**Rationale:** After the tutorial, the user should be equipped with all the knowledge and tools to play the game efficiently.

**Fit Criterion:** All the users will have no difficulty learning to play the game if they already know how to play the original version of chess. For new users, the tutorial will be sufficient to get started with their friends.

### Requirement U7

**Description:** The product should offer feedback based on the invalid moves during the game.

**Rationale:** If the user wants to go through an invalid move, the game should provide the user with feedback on why it is invalid and what is the valid move for that game piece.

**Fit Criterion:** The game will have feedback and error handling for invalid moves.

## 16d Understandability and Politeness Requirements

### Requirement U8

**Description:** The game should use simple grammar for the ease of understanding for the users

**Rationale:** Simple grammar is easier to understand by all kinds of users

**Fit Criterion:** The game will be using very basic grammar for all kinds of languages.

## 16e Accessibility Requirements

### **Requirement U9**

There are no accessibility requirements in this version of the product. This product is not designed to be accessible but people with disabilities can play this game with the help of an aid. Later versions will be incorporated with voice control to handle this constraint of the product.

## **16f User Documentation Requirements**

### **Requirement U10**

**Description:** The product will have a direct link to the user documentation in the help section of the game as well as in the app store description of the product.

**Rationale:** The user documentation will provide answers to all types of generic questions the user might have about the product.

**Fit Criterion:** A website will be created for the user documentation and will be available to access/download when the users download the product.

## **16g Training Requirements**

### **Requirement U11**

**Description:** The training eBook should be accessible/downloadable when the user downloads the game.

**Rationale:** The training eBook will have all FAQs and the tutorial for the product

**Fit Criterion:** The eBook will be available in simple English for the ease of understanding by all types of users.

## **17 Look and Feel Requirements**

### **17a Appearance Requirements**

#### **Requirement L1**

**Description:** The product should have clear instructions on how to navigate through the game.

**Rationale:** The product should be user friendly so as to have a wide variety of users.

**Fit Criterion:** Any new user will be able to completely play the game with no difficulty within the first few minutes of the tutorial.

### **17b Style Requirements**

### **Requirement L2**

**Description:** The icon for the application should display the special features of the game making it distinct.

**Rationale:** The application should be attractive and not dull to look at for hours with the option to customize the background.

**Fit Criterion:** After the tutorial and a couple rounds with friends, the game should be easily accessible and fun for the user.

## **18 Operational and Environmental Requirements**

### **18a Expected Physical Environment**

#### **Requirement L3**

**Description:** The product shall be used for predominantly players playing at home or are stationary in general.

**Rationale:** Players will need to focus on the screen given this game involves attention.

**Fit Criterion:** the application should display a message warning the users to play while stationary.

#### **Requirement L4**

**Description:** The product shall allow for a 2, 3, and 4 player versions of the game.

**Rationale:** This feature will depend on the number of users that want to play with a maximum of 4 users.

**Fit Criterion:** The interface should seamlessly allow for users to select between the three different modes.

#### **Requirement L5**

**Description:** The product shall offer two background modes either positive contrast or negative contrast.

**Rationale:** Depending on the user's preferences and possible sensitivity to light. This application will contain these two modes.

**Fit Criterion:** This application will be tested in both modes to ensure that the modes successfully make the experience acceptable for users of various types of sensitivity to light.

## **18b Requirements for Interfacing with Adjacent Systems**

### **Requirement L6**

**Description:** The product will function with a client and server

**Rationale:** Users will log in on the client side to the server where the game will be played.

**Fit Criterion:** This must be compatible with both mobile and general computer systems.

### **Requirement L7**

**Description:** The product will use a database for specific information.

**Rationale:** The users and their scores will be recorded in a database for other players to see.

**Fit Criterion:** There will be a system that automatically records and stores the scores.

## **18c Productization Requirements**

### **Requirement L8**

**Description:** This product will be compatible with both the Android, iOS, and windows operating systems.

**Rationale:** These are the only and most popular options for applications on mobile and computer operating systems.

**Fit Criterion:** The application will be thoroughly tested to ensure the application is compatible with these systems and there are no performance issues.

### **Requirement L9**

**Description:** The product shall be available for download on the google play store, apple application store, and the Microsoft store for windows.

**Rationale:** These application stores are the corresponding stores for all of the operating systems compatible with this application.

**Fit Criterion:** This requirement will be met when this application is successfully available to download from all three of the mentioned stores.

## **18d Release Requirements**

### **Requirement L10**

**Description:** Each release should update the user of any new features and bug fixes that have occurred.

**Rationale:** The Users will be informed of any changes and improvements made to the application.

**Fit Criterion:** When the user opens the application after an update a message will be displayed notifying the user of all the changes made.

### **Requirement L11**

**Description:** Smaller patches should be conducted weekly or immediately if there is a more serious issue.

**Rationale:** The weekly update will designate a specific time the application will be not available to implement the updates.

**Fit Criterion:** A message on the main screen of the application will notify the user of the time and date of the update. which the update will take place at 4:00 am every Monday.

## **19 Cultural and Political Requirements**

### **19a Cultural Requirements**

#### **Requirement C1**

**Description:** This product will be designed in a way to not offend any race, gender, religion, disability, and sexual orientation.

**Rationale:** The Content for this application will strictly be related to playing a game of chess with other users.

**Fit Criterion:** There will be a list of banned words for this application that won't be able to be used for players names or any other user filled in text field.

### **Requirement C2**

**Description:** This application will not discriminate against any language they will be using for this game.

**Rationale:** There will be an option for each user to choose the language they speak to be used throughout their side of the game.

**Fit Criterion:** When the user begins the game there will be a question that shows up on the screen asking the user which language they prefer to use.

## **19b Political Requirements**

### **Requirement C3**

**Description:** This application will be developed through only one company and not hire any outside sources to assist in the development process.

**Rationale:** For this process the management prefers creating this application with no outside help, to avoid any delays and confusion in the development process.

**Fit Criterion:** All of this application will be developed through the same company with no outside help.

## **20 Legal Requirements**

### **20a Compliance Requirements**

#### **Requirement C4**

**Description:** This product will comply and follow any local, state, or federal laws.

**Rationale:** This application will be available in all 50 states in the United States.

**Fit Criterion:** A Legal team that the company hires, will ensure this application is in compliance of all the local, state, or federal laws.

## 20b Standards Requirements

### Requirement C5

**Description:** This application will comply with all of the (FCC)federal Communications Commission's rules. and will also comply with (COPPA) Children's Online Privacy Protection Rule.

**Rationale:** This application will meet all of the rules and regulations that the FCC enforces. and also since children are part of our target audience for this game following the COPPA regulations are necessary as well to protect the children's privacy.

**Fit Criterion:** A legal team that the company hires, will ensure this application complies of all rules and regulations imposed by the FCC.

## 21 Requirements Acceptance Tests

### 21a Requirements – Test Correspondence Summary

Test	Requirements																			
	Req 1	Req 2	Req 3	Req 4	Req 5	Req 6	Req 7	Req 8	Req 9	Req 10	Req 11	Req 12	Req 13	Req 14	Req 15	Req 16	Req 17	Req 18	Req 19	Req 20
Test 1	X																			
Test 2		X				X														
Test 3			X	X																
Test 4					X	X			X											
Test 5								X												
Test 6							X													
Test 7																				
Test 8																				
Test 9										X										
Test 10											X									
Test 11												X								
Test 12													X							
Test 13														X						



Test 14															X					
Test 15																X				

## 21b Acceptance Test Descriptions

### **ID #1 - Transition Functionality Events**

**Description:** This application will ensure that the transition through the main screen to the game is seamless and easy for the user to accomplish. inthe event an internet outage occurs for the user the game will automatically pause and only continue when all users have a good intended connection established again.

### **ID #2 - Accessibility and Ease of use**

**Description:** This application will ensure that the users has a seamless experience accessing all functions requiring a complete playthrough of the game. and also an easy system that allows the user to select a language of choice for their versions of this application.

### **ID #3 - Permissions and Privacy**

**Description:** Once the application is in the main screen there will be an option for the user to use their real name and display their location. in the interest of privacy, the user will be able to use a different or created name and turn off their location if so choose. This test will ensure that the user is allowed to do these features ensuring that their privacy is remaining at the utmost importance.

### **ID #4 - Errors and Safety Requirements**

**Description:** This application will ensure that any action taken by the user does not break or crash the game. There will be a message that pops up if the user decides to turn off their location and change their name to ensure they wanted to do that action and not did it by accident.

### **ID #5 - Database**

**Description:** The database for this application will ensure that the data that is stored will store players' scores but allows the users to choose whether their real name and location is sent to the database. There will be a test to ensure this functionality is correct and accurate. This check will also ensure that the information that is sent to the database is accurate and correct, as well as making sure that the basic functionality of the database is also functional as well.

### **ID #6 - Legal Requirements**

**Description:** This application will be developed by a single company with no assistance from any other companies. There will be a legal team employed by the company that develops this application to uphold and follow any local, state, and federal laws that pertain to his application.

### **ID #7 - Usability Requirements**

**Description:** This application will be available on Android, iOS, and windows operating systems. Also ensure that users are able to download this application from those stores with no issue. And ensure that the application is functional and working on every user's device after being downloaded from the stores.

### **ID #8 - Audio and Background Music**

**Description:** This application will have some background music for the Main screen only and not for the game segment. Ensure that the background music for the Main screen is fully functional and working properly.

### **ID #9 - Speed and Optimization**

**Description:** This requirement will ensure that the application takes no longer than 5 seconds to load and a user can start the game in no longer than 10 seconds given other users are ready to play immediately as well.

## **III Design**

### **22 Design Goals**

Cross-platform online play is the overall base of the project. Taking this into account, the speediness of interpreting a player move and updating the server as quickly as possible takes on the utmost precedence. At the cost of visual fidelity, and even simplicity of the development of the project, interpreting player moves must be handled as quickly as possible. When considering the efficiency of algorithm implementation, a quicker approach is to take priority, even at the expense of increased complexity of the code itself.

The design goal of server communication and data storage is the most flexible aspect as far as implementation goes. Initial release of the product will be to a limited number of individuals, and as such, security is not a priority. There are minimums to security as highlighted in the requirements section. Past a certain threshold of users, and with

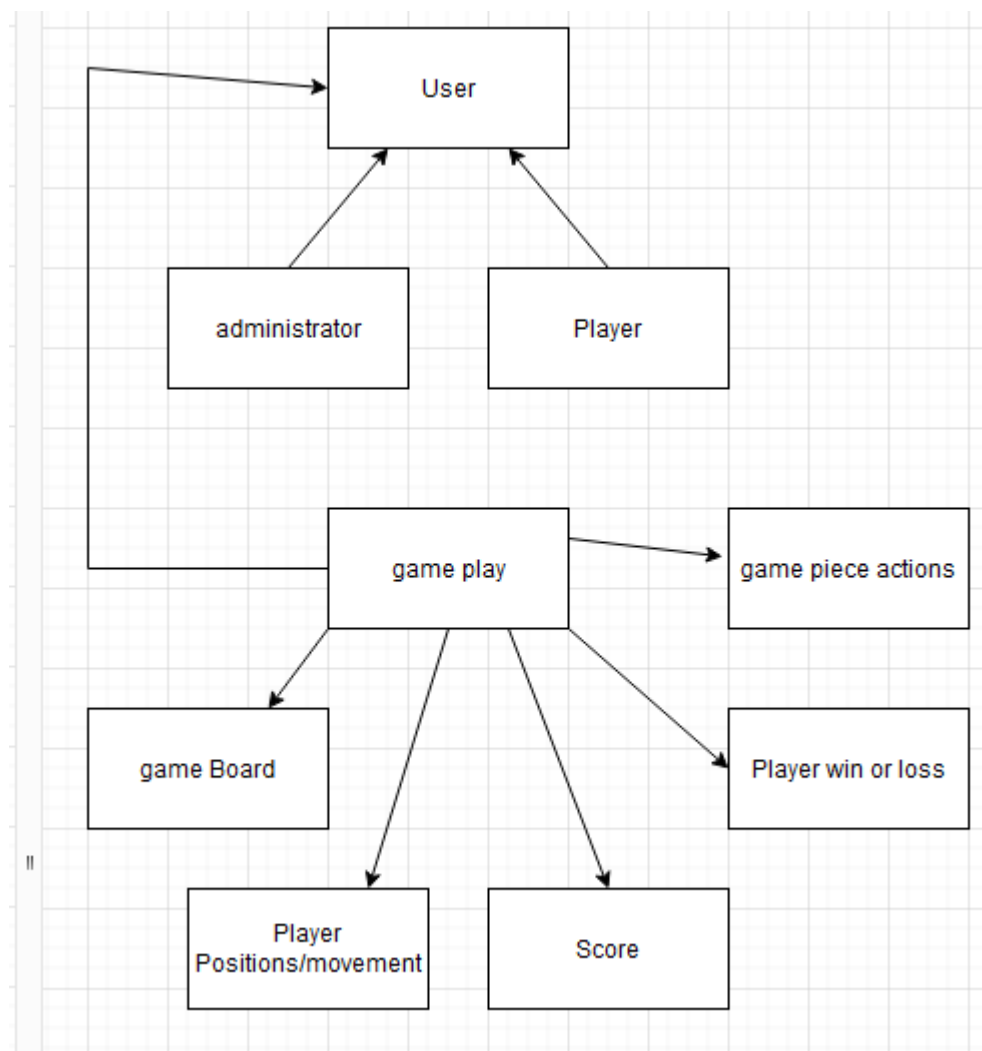
increasing and perhaps even exponential user base growth, security will increase with priority. Server communication and the encapsulation and protection of transmitted and stored data will need to be met with increased scrutiny. Using these principles for design, the summarized priority list will be as follows: Speedy Move Interpretation, Speedy Server Communication, Visual Fidelity, and Security.

### 23 Current System Design

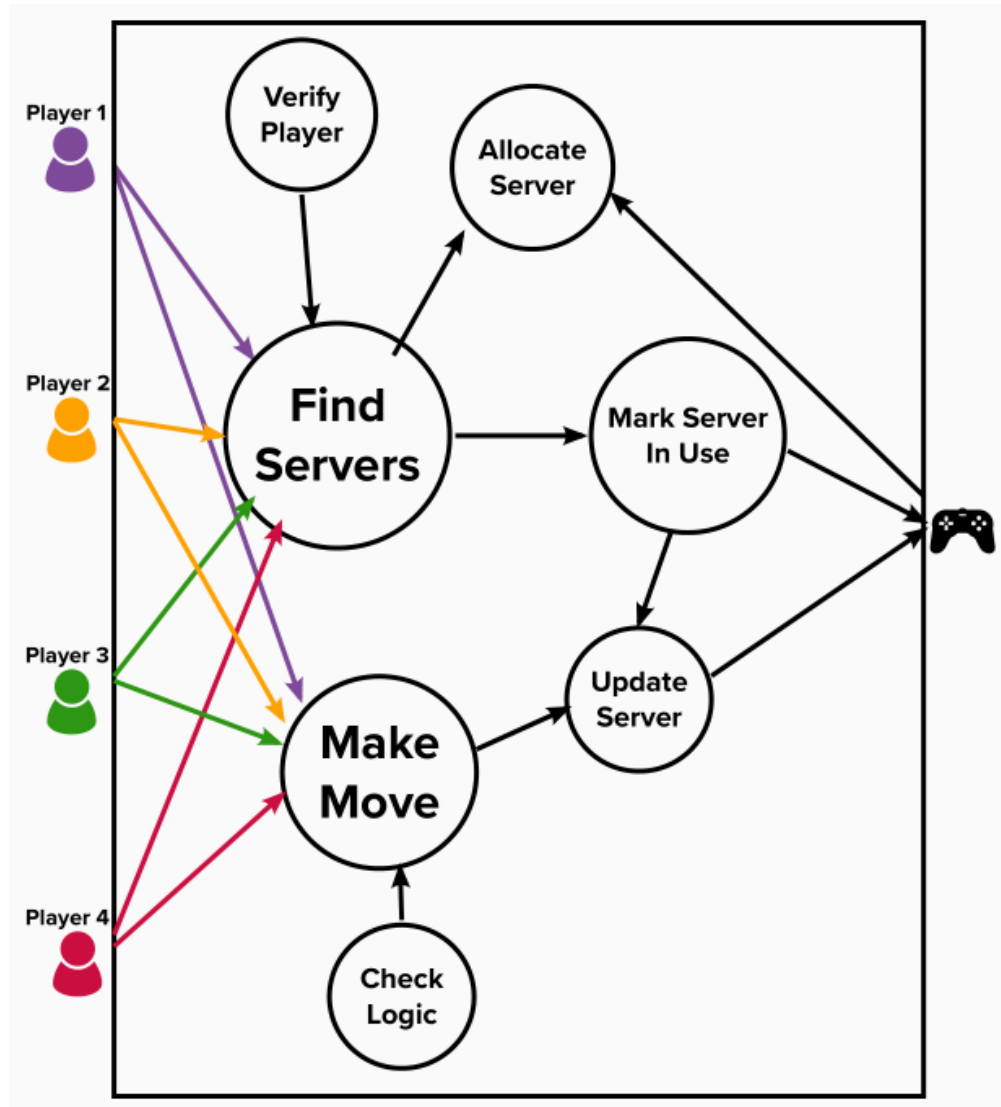
There currently exists no current system or implementation.

### 24 Proposed System Design

#### 24a Initial System Analysis and Class Identification



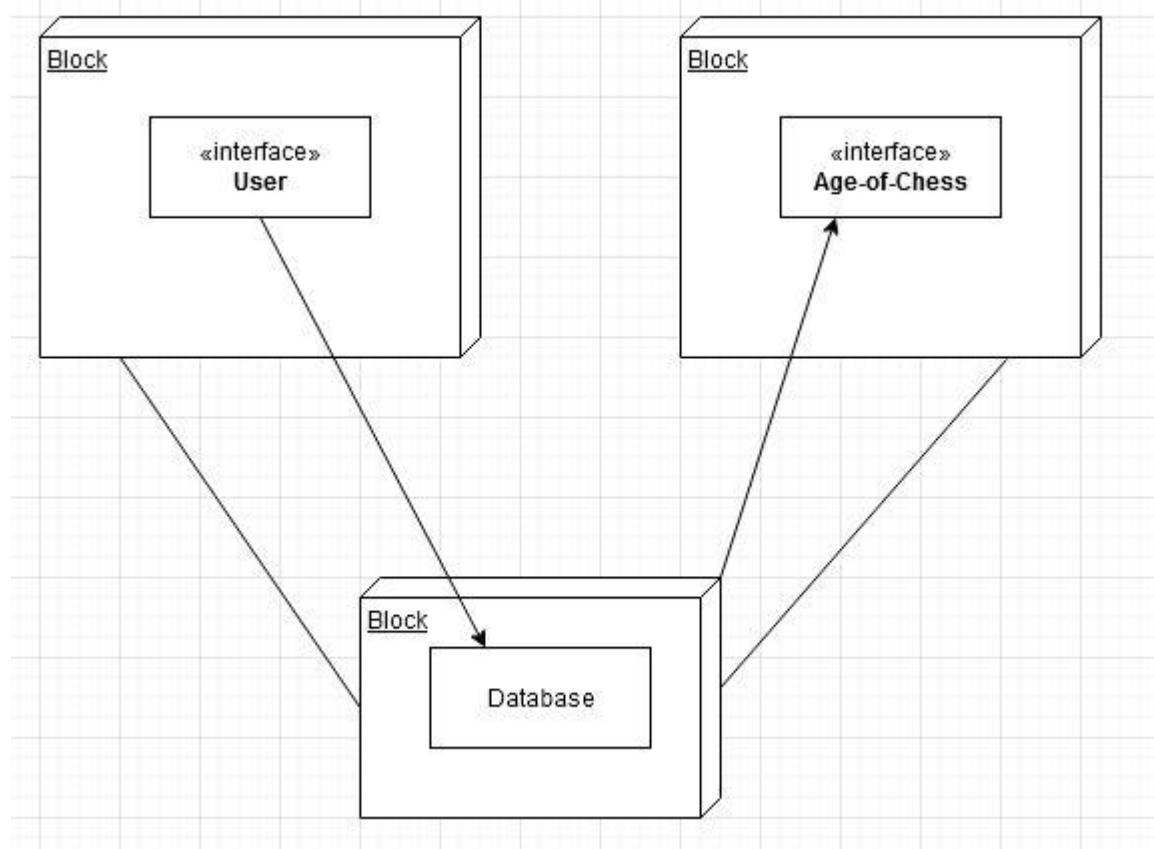
#### 24b Dynamic Modelling of Use-Cases



#### 24c Proposed System Architecture

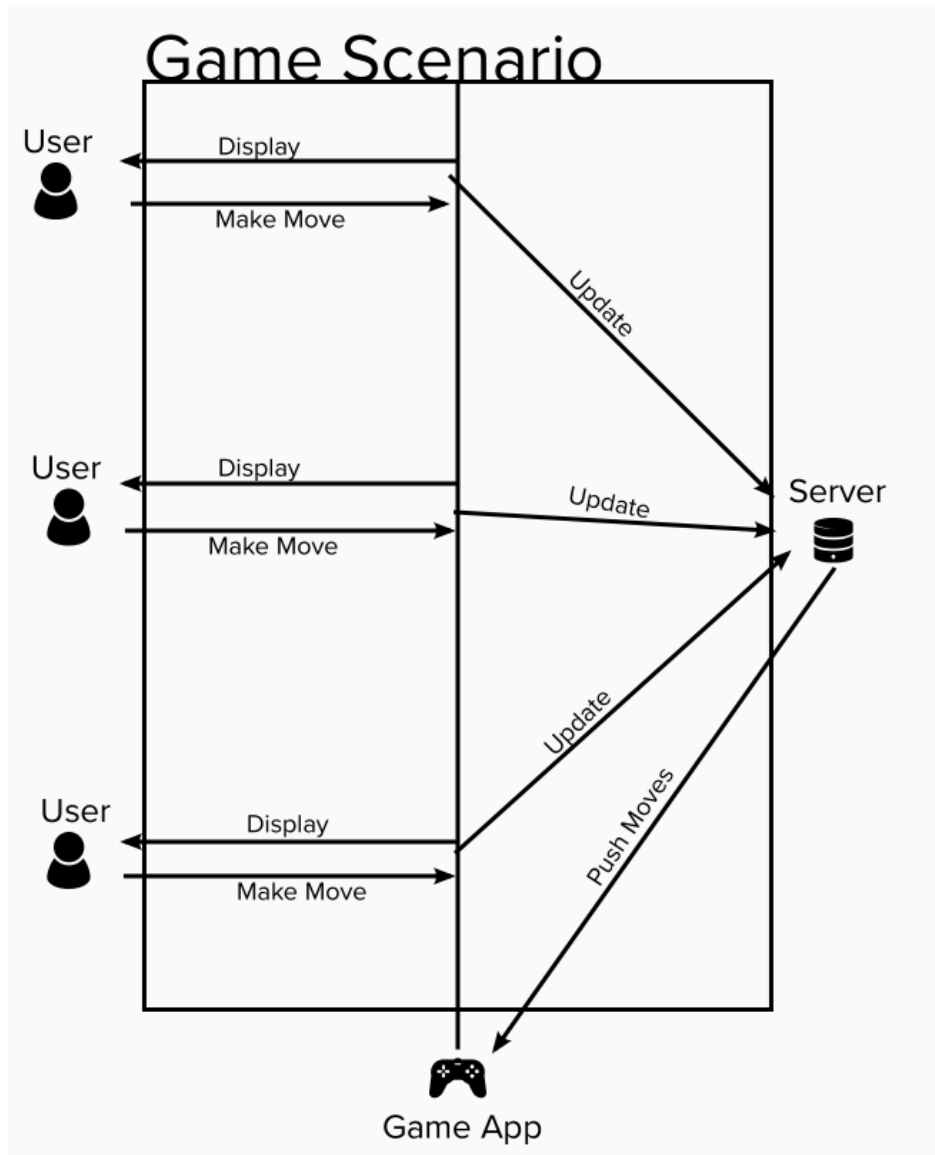
This application will have a client-server systems architecture. each user will have their own client side which holds their name and location if they so choose to display them. and the game itself will be on a server which the clients will enter from their individual client sides of the application.

#### 24d Initial Subsystem Decomposition



## 25 Additional Design Considerations

### 25a Hardware / Software Mapping



### 25b Persistent Data Management

Data for the database will be updated in a consistent timed manner. Users location and name will be an option for the user to choose to display or not. but if the user decides to display their location that will be stored and displayed in the score records in the database. Once the game is shutdown/closed then the application will no longer have locational services.

There will be one database that saves and displays all the users scores. This database will have different categories of scores. One of the categories will be all the scores based on the person's region they played the game in. and if the user chooses to not display their name or location then their scores will not be sent to the database. the user will at least need to input their name for their scores to be put in the database.

## 25c Access Control and Security

**Location:** One of the Main security concerns for this application is the user's privacy. the users will be able to display their location or not and regardless of that decision they soul have the same experience.

**Name:** Another security concern in regards to the user's privacy is allowing them the decision to use their real name or not. This decision also won't change the experience in any way for the user.

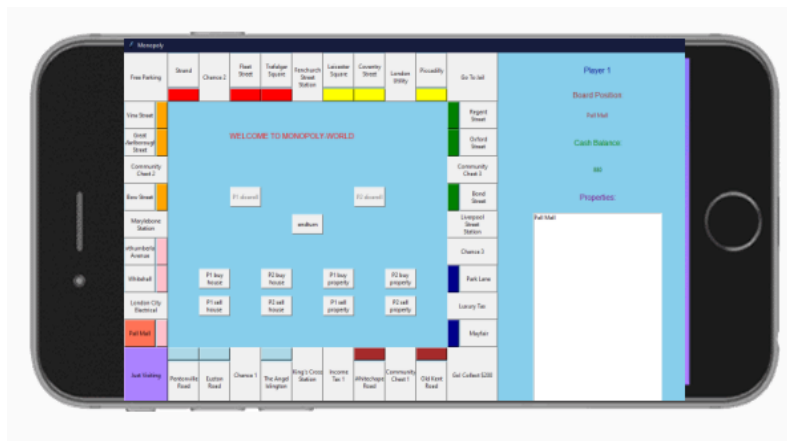
## 25d Global Software Control

Users will be required to make an account stored on a local server or cloud server and authentication would ensure controlled access to application, besides the login screen, to the potential player.

## 25e Boundary Conditions

Server load must be checked at all times. The class that supports a player attempting to find a game must check that the threshold for users has not been exceeded. Reject a server connection if the threshold has been met. Create another class of a queue to allow waiting.

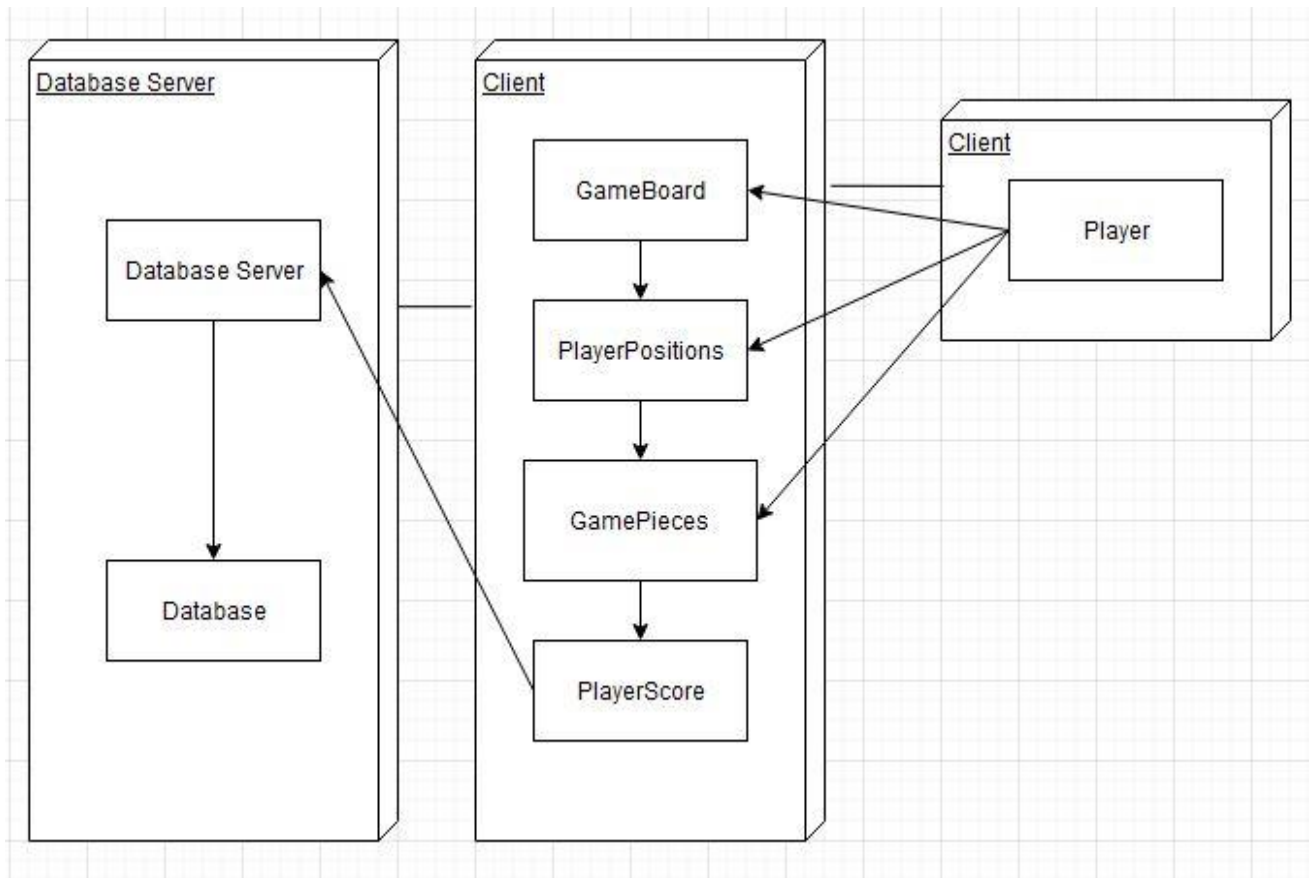
## 25f User Interface



## 25g Application of Design Patterns

All design patterns have been referenced.

## 26 Final System Design



## 27 Object Design

### 27a Packages

- Game Pieces package: composed of classes representing each type of game piece and the moves it can make.
- Models package: composed of classes of UI game models which represent the game pieces.
- Game Board package: composed of a class for the UI model of the game board and classes that keep track of the game pieces location on the board.
- Game Rules/Logic package: composed of the controller classes which handle interactions between the other classes and run the game.
- Player package: composed of classes showing the players pieces and



### **27b Subsystem I**

Game Pieces package: contains a class for each piece in the game with get functions that return ownership of the instance of the piece and the move set that the specific piece can execute. Also, includes a get for the type of game piece which will be called to confirm that a move a piece tries to make is allowed. Finally any special actions a piece can make will have their own function only declared in that class. Ex. King: switch with rook, Pawn: exchanged for another piece upon reaching the other side of the board, Monk: convert enemy piece.

### **27c Subsystem II**

Game Board package: contains a class which handles the UI for the board, keeps track of all the pieces positions, and safe spaces. This class is constantly checked by the Game Rules class for the positions of pieces and updating the pieces' positions.

### **27d Subsystem III**

Game Rules/Logic package: contains the main game controller class that runs the game primarily through the setTurn, movePiece, and checkWinStatus. setTurn iterates through turn order. movePiece handles a pieces's movement by validating the move and updating the pieces position on the board. checkWinStatus checks if a king has been captured, if so that king's player's status is set to lose and they are removed from the turn order.

### **27e Subsystem IV**

Player package: contains the players' classes which handle the player's input into the game. Players may make a move by selecting a piece in the UI and moving it to the desired location, the legality of the moves is checked by the Game Rules class and a move cannot be taken unless the Game Rules class accepts the move and will then make the move. Also the player can only make a move if the isTurn method returns true which is also controlled by the Games Rules class. Finally the status method is checked at the end of each turn to see if a player loses and is removed from the game.

## **IV Project Issues**

### **28 Open Issues**

The overall assumption with this project is that there will exist enough players to warrant continued support. An unfortunate reality is that popularity is difficult to predict. Popularity may spike unexpectedly and break current systems in place. Popularity may dwindle or never take off and may break current systems in place. While artificial players may fill servers, not having a single player may very well be a reality and will lead to a financially unsustainable state.

## **29 Off-the-Shelf Solutions**

### **29a Ready-Made Products**

Servers may be rented/purchased on the fly such as Amazon Web Services.

### **29b Reusable Components**

Existing Development Platforms such as Unity exist for easier visual fidelity development.

### **29c Products That Can Be Copied**

None

## **30 New Problems**

### **30a Effects on the Current Environment**

since this application is for recreational fun and is meant to be played in an environment that is where the users focus is on the application. this application should not have any effect on and type of work environments.

### **30b Effects on the Installed Systems**

This application may use up a substantial portion of battery life, possibly deleting it if the user uses the application a substantial amount.

For this application to ensure it is a consistent experience for all users. this application must be compatible with both Android and IOS operating systems.

### **30c Potential User Problems**

Users may be so focused on the application that they may be unaware of events happening around them.

There may be some inaccuracies with the GPS location the user's application displays.

Users who don't have any experience playing chess may experience a learning curve with playing a game on this application.

### **30d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product**

Certain locations or cities can pose a signal issue with the connection or event GPS.

The use of this application while operating a vehicle or walking around in public can result in a dangerous situation.

The possible popularity of this game can cause lagging in the system.

### **30e Follow-Up Problems**

If a high number of users use this application at launch this could cause the servers to slow down or even crash. This will be taken into consideration when the application is released. and will have safeguards in place to prevent this issue from taking place.

In order to prevent inaccuracies on displaying the correct location the user is at, there will be a safeguard in place to check and prevent any inaccuracies.

Given that this application allows for the user to write in their own name there will be a system in place to prevent users from being able to enter derogatory or inappropriate words.

## **31 Migration to the New Product**

Not applicable

### **31a Requirements for Migration to the New Product**

Not applicable

### **31b Data That Has to Be Modified or Translated for the New System**

Not applicable

### **32 Risks**

The possibility exists that weekly maintenance or any other maintenance for that fact can take longer than expected.

For users that choose to display their location there does exist the possibility of inaccuracies GPS and issues with if their device has the location turned off.

Since this application is designed for a user that is stationary, there is the possibility a user decides to use this application while driving or walking on the streets.

### **33 Costs**

\$610,000 (*highlighted in section 6g*) is the estimated cost of salaries for the various engineers and project manager involved with the implementation of this project. However, it will not take an entire year to yield results.

Estimated time for a working prototype with 8 developers is about 4 weeks. After that sprint, a network engineer will take about 3 weeks, working alongside the developers, to integrate a working online prototype. At this stage, cross-platform is not needed. After that sprint, 16 independent contractors will be paid per hour at an acceptable rate of \$25/hr to test the prototype for 1 week (20 hours total each). After which, developers will spend an additional week implementing changes from this feedback, pushing a new version to the contractors who will repeat testing. This will be repeated 4 times totaling 8 weeks of prototyping and testing.

At this stage of development, 15 weeks have passed. Server testing with a target goal of 100 maximum concurrent players will be tested. Less specialized independent contractors will be contracted to test 100% capacity for 2 days total non-concurrently. These independent contractors will be paid \$15/hr for 8 hours total. Day 1 will consist of testing. After which, the 4 developers and 1 network engineer will fix any issues found for 1 week. One final push to the contractors will test loads.

At this stage of development, 17 weeks have passed and a product in alpha stages capable of 100 concurrent players will be in a relatively ready state. From now, the 8 developers will work with integrating either IOS or Android, whichever was not the

chosen initial development environment. At this stage, visual fidelity will take on a larger role in development and will lead to 4 weeks of additional development.

21 Weeks have passed and testing will need to be completed by independent contractors. 100 contractors at \$15/hr for 10 hours in one week will gather enough feedback for developers to make the final changes before release.

Developers will be allotted 4 weeks to reach an acceptable state passing most requirements. After which point, may be pushed to either a larger audience or to the mass market, depending on the state of the market. A marketing team will be hired to produce a campaign push for 2 weeks. Budget allocated will be around \$20,000.

- 28 weeks estimated total development cycle.
- 19 weeks of engineering work with a total yearly salary of \$610,000.
- \$59,000 for independent contractors.
- \$20,000 marketing campaign.

Total Estimated Cost: \$301,884.62

### **34 Waiting Room**

The World Chess Championships is renowned for its prestige. With the rise of E-Sports, pushing this version of Chess into online tournaments could prove to be beneficial. The beginning steps of such an endeavor could include integration into online streaming platforms such as Twitch, YouTube, etc.

A separate version designed to be played much quicker could attract a larger audience. Many existing games that tend to take extended periods of time have already implemented shortened versions, Monopoly, for example.

### **35 Ideas for Solutions**

If Unity Engine is the chosen major development platform, integrating the project into multiple platforms such as IOS, Android, and PC would be streamlined. Toolkits exist to integrate Unity into XCode for IOS, Java/Kotlin for Android, and C++/C# for Windows. Multiple languages will still need to be developed with, such as Swift, C++/C#, and Java/Kotlin.

A dedicated team or paid contractors to test alpha states of the product would prove to be beneficial. By avoiding releasing to the public initially, marketing can be better controlled. As with any new product push into the open market, public perception is critical.

Consideration for integrating into cloud-based servers would prove to be beneficial in the long run should local servers prove to be unable to scale to user growth. Testing with products such as Amazon Web Services from the beginning of development would lend itself to easier full integration if need be down the line.

## 36 Project Retrospective

Having an example of a good software engineering report was very helpful. Many of us had not seen one before and working on such a large-scale document over several weeks was certainly a new experience. The “checkpoints” we had throughout the semester were helpful, rather than having the entire document due at the end with no checkpoints.

Some aspects of the project that we did not like was the inconsistency of the grading. A more flushed out rubric of some kind would prove to be very helpful in this scenario. Another aspect that was brought up was the fact that this report was not tied in with our coding project. Considering the fact that we are actively developing the coding project, making a report on a scale such as this would certainly be easier, but the reasons for not tying them together are also apparent. Considering this is a report you would typically submit before development ever began, having that mindset to think critically about future development was a learning opportunity, but the learning curve was steep at times.

## V Glossary

- The pawn: will be able move only one space at a time, except on its first move can move twice
- The Bishop: will be able to move as many spaces as the player desires diagonally
- The Knight: will be able to move in a L-shape with two squares left or right horizontally and then the last and third square either up or down.
- The Rook: will be able to move as many spaces as the payer desires vertically or horizontally
- The Queen: will be able to move as many spaces as the player desires either vertically, horizontally, or diagonally
- The King: will be able to move only on space at a time, but once defeated that player loses
- The Monk: this game piece will only move one space per turn, but it will have the ability to take the place of any defeated piece on the board. The monk has the ability to convert up to two opposing Pawns from the other team to be on your team. The rules for this are the opposing piece has to be at most two spaces away from the monk piece.

- Safe spaces: This game will have a total 8 spaces on the board that are safe spaces. And in these safe spaces, you can stay safe for a maximum of two moves. The main reason for the introduction of safe spaces was to add more strategic complexity to the game.

## VI References / Bibliography

- [1] Robertson and Robertson, Mastering the Requirements Process.
- [2] A. Silberschatz, P. B. Galvin and G. Gagne, Operating System Concepts, Ninth ed., Wiley, 2013.
- [3] J. Bell, "Underwater Archaeological Survey Report Template: A Sample Document for Generating Consistent Professional Reports," Underwater Archaeological Society of Chicago, Chicago, 2012.
- [4] M. Fowler, UML Distilled, Third Edition, Boston: Pearson Education, 2004.
- [5] <https://wearesocial.com/blog/2020/10/social-media-users-pass-the-4-billion-mark-as-global-adoption-soars>
- [6] <https://wearesocial.com/blog/2020/07/digital-use-around-the-world-in-july-2020>

## VII Index

Design .....	60, 70
Requirements .....	28, 60
Project issues .....	70, 77
Project Description .....	09, 27