# *Software Engineering Project Report*

## *Chess Of Champions For All Devices*

*Prepared by*

*Eldin Vujic, Victor Fong, Ethan Luong and Saude Chaudhry*

*for use in* **CS 440**

**at the**

**University of Illinois Chicago**

**September 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............................................................................................................... 9

3 The Scope of the Work................................................................................................ 9

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 ............................................................................................ 12 4a
Scenario Diagram(s) ................................................................................................... 12 4b
Product Scenario List .................................................................................................. 12 4c
Individual Product Scenarios....................................................................................... 12

5 Stakeholders ................................................................................................................ 12

5a The Client.................................................................................................................. 12
5b The Customer............................................................................................................ 13 5c
Hands-On Users of the Product ................................................................................... 13 5d
Maintenance Users and Service Technicians............................................................... 13 5e
Other Stakeholders....................................................................................................... 13 5f
User Participation......................................................................................................... 13 5g
Priorities Assigned to Users.......................................................................................... 13

6 Mandated Constraints.................................................................................................... 14

6a Solution Constraints..................................................................................................... 14
6b Implementation Environment of the Current System ................................................. 14 6c
Partner or Collaborative Applications .......................................................................... 14 6d
Off-the-Shelf Software.................................................................................................. 14 6e
Anticipated Workplace Environment ........................................................................... 14 6f
Schedule Constraints..................................................................................................... 14 6g
Budget Constraints........................................................................................................ 15

7 Naming Conventions and Definitions............................................................................ 15 7a
Definitions of Key Terms ................................................................................................ 15 7b UML
and Other Notation Used in This Document............................................ 15 7c Data Dictionary

for Any Included Models ........................................................... 15 3
8 Relevant Facts and Assumptions.................................................................................. 15
8a Facts................................................................................................................ 15
8b Assumptions.............................................................................................. 15

II Requirements........................................................................................................ 16

9 Product Use Cases.................................................................................................. 16
9a Use Case Diagrams........................................................................................ 16
9b Product Use Case List .................................................................................. 17 9c
Individual Product Use Cases................................................................................ 17

10 Functional Requirements........................................................................................ 19
11 Data Requirements .................................................................................................. 19
12 Performance Requirements ..................................................................................... 20
12a Speed and Latency Requirements.................................................................... 20
12b Precision or Accuracy Requirements................................................................ 20
12c Capacity Requirements ....................................................................................... 20

13 Dependability Requirements..................................................................................... 21
13a Reliability Requirements...................................................................................... 21
13b Availability Requirements.................................................................................... 21
13c Robustness or Fault-Tolerance Requirements..................................................... 21
13d Safety-Critical Requirements................................................................................ 22

14 Maintainability and Supportability Requirements......................................................... 22
14a Maintenance Requirements.................................................................................. 22
14b Supportability Requirements................................................................................. 22
14c Adaptability Requirements.................................................................................... 23
14d Scalability or Extensibility Requirements............................................................. 23
14e Longevity Requirements....................................................................................... 23

15 Security Requirements................................................................................................ 23
15a Access Requirements............................................................................................. 24
15b Integrity Requirements.......................................................................................... 24
15c Privacy Requirements ........................................................................................... 24
15d Audit Requirements............................................................................................... 24
15e Immunity Requirements......................................................................................... 25

16 Usability and Humanity Requirements ........................................................................ 25
16a Ease of Use Requirements.................................................................................... 25
16b Personalization and Internationalization Requirements.................................... 25
16c Learning Requirements......................................................................................... 26
16d Understandability and Politeness Requirements................................................. 26
16e Accessibility Requirements.................................................................................. 26

16f User Documentation Requirements .................................................................. 27
16g Training Requirements......................................................................................... 27

17 Look and Feel Requirements......................................................................................... 27
17a Appearance Requirements.................................................................................... 27
17b Style Requirements ............................................................................................... 28

18 Operational and Environmental Requirements ............................................................ 28
18a Expected Physical Environment........................................................................... 28
18b Requirements for Interfacing with Adjacent Systems......................................... 28
18c Productization Requirements .............................................................................. 29
18d Release Requirements............................................................................................ 29

19 Cultural and Political Requirements.............................................................................. 29
19a Cultural Requirements........................................................................................... 29
19b Political Requirements........................................................................................... 30

20 Legal Requirements.......................................................................................................... 30
20a Compliance Requirements.................................................................................... 30
20b Standards Requirements........................................................................................ 30

21 Requirements Acceptance Tests...................................................................................... 31
21a Requirements – Test Correspondence Summary ................................................ 31
21b Acceptance Test Descriptions............................................................................... 31

III Design.............................................................................................................................. 32 22
Design Goals ......................................................................................................................... 32 23
Current System Design........................................................................................................ 32
24 Proposed System Design ................................................................................................... 32
24a Initial System Analysis and Class Identification................................................. 32
24b Dynamic Modelling of Use-Cases ....................................................................... 32

24c Proposed System Architecture .......................................................................... 32
24d Initial Subsystem Decomposition..................................................................... 33

25 Additional Design Considerations................................................................... 33

25a Hardware / Software Mapping ............................................................................ 33
25b Persistent Data Management................................................................................ 33
25c Access Control and Security ................................................................................ 33
25d Global Software Control ...................................................................................... 33
25e Boundary Conditions............................................................................................ 34
25f User Interface ....................................................................................................... 34
25g Application of Design Patterns ......................................................................... 34

26 Final System Design........................................................................................... 34

27 Object Design ....................................................................................................... 34

27a Packages.................................................................................................................. 35
27b Subsystem I ........................................................................................................... 35
27c Subsystem II........................................................................................................... 35
27d etc. ......................................................................................................................... 35

IV Project Issues................................................................................................. 35 28
Open Issues...................................................................................................... 35 29
Off-the-Shelf Solutions ......................................................................................... 35

29a Ready-Made Products............................................................................................ 35
29b Reusable Components............................................................................................ 35
29c Products That Can Be Copied ............................................................................ 36

30 New Problems ....................................................................................................... 36

30a Effects on the Current Environment.................................................................... 36
30b Effects on the Installed Systems.......................................................................... 36
30c Potential User Problems....................................................................................... 36
30d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product ................................................................................................................ 36
30e Follow-Up Problems............................................................................................. 36

31 Migration to the New Product ............................................................................ 37
31a Requirements for Migration to the New Product................................................ 37
31b Data That Has to Be Modified or Translated for the New System ..................... 37

32 Risks........................................................................................................ 37 33
Costs........................................................................................................ 37 34
Waiting Room ........................................................................................... 37 35
Ideas for Solutions..................................................................................... 37 36
Project Retrospective.................................................................................. 38
V Glossary..................................................................................................... 38 VI
References / Bibliography ............................................................................ 38 VII
Index ............................................................................................................ 38

## List of Figures

Figure 2 - Sample Use Case Diagram from Bruegge & DuToit ( modified )............................. 16
Figure 3 - Sample Use Case Diagram from Robertson and Robertson ....................................... 17

## List of Tables

*Table 2 - Requirements - Acceptance Tests Correspondence*...................................................... 31

# I Project Description

## 1 Project Overview

The traditional game of Chess is a common game that is still played today that was believed to be created before the 7th century which was derived from the Indian game chaturanga. So what better way then extending this traditional game into a more modern type game. To extend the traditional game of chess, this game will use concepts from a major modern day genre known as MOBAs that includes games such as Dota 2 and League of Legends. Essentially what this game does is it extends chess into a 4 player game while using the concepts of League of Legends / Dota 2, creating this game known as *Chess Of Champions.*

## 2 The Purpose of the Project

The reason this game is being done is so that players who play chess or any MOBA genre can interact with each other by playing a game that combines these two genres of a game. Ultimately what our development team hopes to achieve from this is a unified game of veteran players from both of these genres that helps drive this game into a more competitive manner which can lead to tournaments, and eSport teams. *Chess of Champions* is a free competitive and casual type game that helps drive the competitiveness from all types of players, whether it's players who never played a MOBA or Chess or veteran players who want to test their skill.

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

The business this game has is an entertainment-strategy game. Since this game primarily uses the concepts of two of these major strategy games played today while at the same time adding custom games and new features to provide an entertaining game for casual players as well.

### 2b Goals of the Project

The reason this project is being carried out is so it can fulfill a player who plays either one of these games. So if a player has been playing Chess for sometime but they always wanted to play a more modern game without giving up too much of Chess, this game would be a great alternative for them (This applies to a MOBA player as well). So ultimately we want to help players that come from both of these genres play another game with similar style but at the same time also being completely different.

### 2c Measurement

Some measurable results that we can point to and say that the goal has been met is by looking at the product the development team has created. And see if this feels like Chess and a MOBA at the same time. If it feels like neither one of these games then the goal we tried to achieve will fail, but if it feels and plays like these two games and is enjoyable for the player, we can successfully say that the goal has been met.

## 3 The Scope of the Work

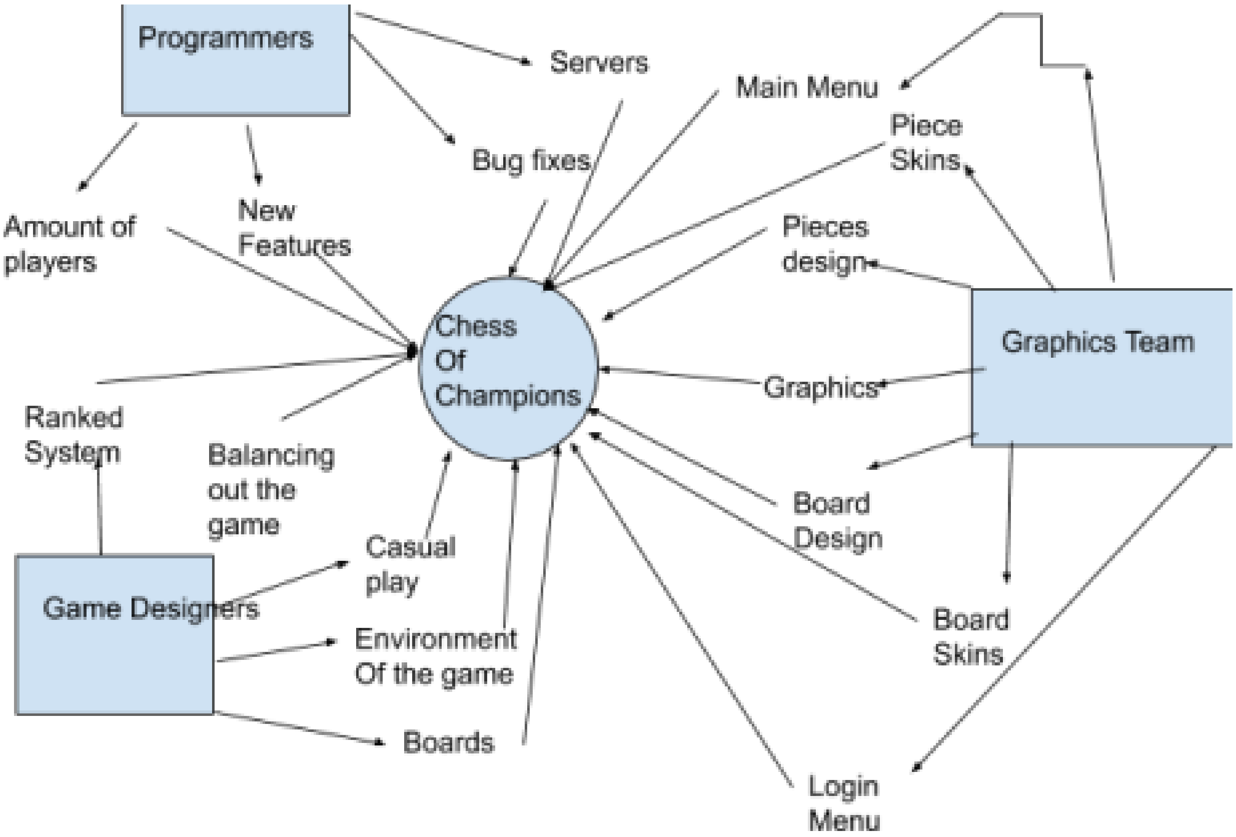
Essentially Chess of Champions is aimed to make the game as competitive as possible while at the same time making the game for casual players as well. This game will be driven with these two play-styles which will hopefully provide the players entertainment while playing a strategy based game.

### 3a The Current Situation

Since Chess is a game where you can usually predict when a game is about to be over, this also goes for MOBAs as well, but at the same time they both have a huge comeback factor. Also since both of these games are both free we decided to follow this as well making it available for all players. This project will be aimed for players who seek this thrill of being able to play the early game however they want, but still have a huge impact on the game. This project will be aimed for players who want this competitive drive and for players who want this competitive feel but still want to play the game casually.

### 3b The Context of the Work

This game will have a couple of different teams that help with the flow and the execution of the development aspect. These teams will include the programmers, graphics team, and game designers.



### 3c Work Partitioning

Programmers will help the game function and make sure there are no bugs the players run into since one bug can completely ruin the game experience for the player. The game design team will help balance out some features of the game and also help create the game environment. It will also have the team dedicated to helping create the graphics of the game which is an essential part of making the game look both like a MOBA and chess.

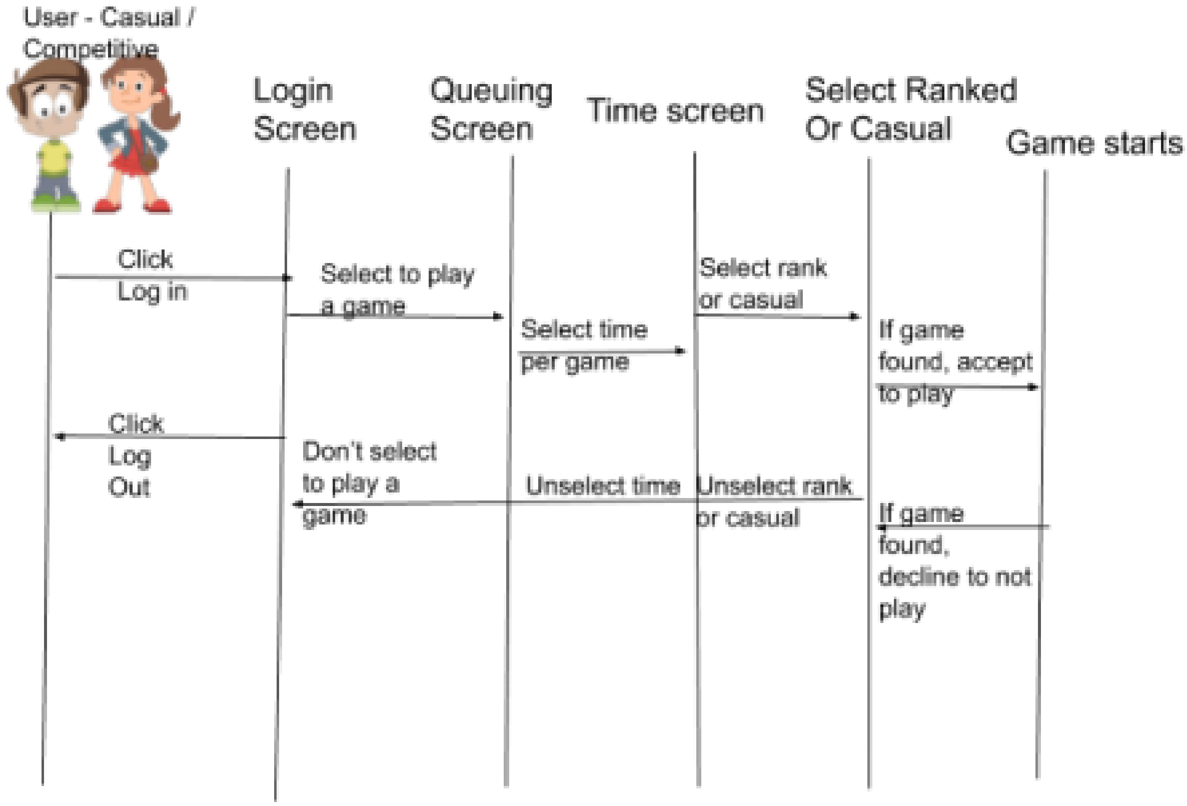
### 3d Competing Products

Currently there are no other existing products that combine the two concepts of these two genres. If there were to be a competitor it would be the game Chess and any MOBA due to this game being a direct expansion of these two games. The reason this product is still needed is to the major purpose of colliding players from both of these genres so they can compete against each other due to the fact that both of these genres require a lot of strategy when playing. So a combination of these skilled players is a reason why this product is beneficial due to it making these players strategize even more from before and it also is a free browser based game.

## 4 The Scope of the Product

*Chess Of Champions* will require secure databases that help maintain every user information that has ever logged in and played this game (This helps with maintaining the users rank, friends list, and the skins they have achieved). It will also require a lot of dedicated servers and since this is based in the U.S we are looking to have 2 different server locations that will help players maintain a decent amount of ping so they could be able to play the game smoothly without running into lag issues. This game will also require keeping track of players who have left a match without surrendering or any login errors the players may have had.

### 4a Scenario Diagram(s)



### 4b Product Scenario List

Scenario External Actor

Casual play This will be for players such as Casual Users.

Ranked Play This will be for players such as Competitive Users.

Time Constraints Casual and Competitive users will both have this option.

Champion Points Every player will have access to gaining these champion points.

Login Screen Any user who has an account associated with this product will be able to login.

## 4c Individual Product Scenarios

| Scenario | External Actor |
|---|---|
| Casual Play | Players will be able to queue up for casual play where they don't have to worry about being competitive. |
| Ranked Play | Players will be able to queue up for Ranked play where they gain rank or lose rank depending on the game result. This |
| Time Constraints | Players will have to worry about the time (Time Can be unlimited, 5 minutes, etc) between their moves since the time between choosing a move can vary. |
| Champion Points | Champion Points are the amount of points a player gets after a game finishes. If they win a game they usually get more than if they were to lose a game. There is also a first day win where you receive 400 points just for winning your first game of the day. |
| Login Screen | Players will either login or create an account to get access to the game. |

# 5 Stakeholders

## 5a The Client

Our client is Riot Games, a developer and publisher of a very popular MOBA known as League of Legends that has a total of 115 million players playing the game daily.

## 5b The Customer

We are targeting this game towards customers who enjoy playing Chess or any form of MOBA games. This game is targeted also for customers who enjoy playing strategy games or customers who enjoy playing board games.

## 5c Hands-On Users of the Product

Our hands-on users will include players who are playing the game competitively and they can achieve the rankings of, Novice, Apprentice, Initiate, Adept, Magus, Master, Grandmaster, and Legendary. It will also include casual players, Public Beta Environment testers, and journalists.

## 5d Maintenance Users and Service Technicians

Our maintenance users and service technicians include the balance team, which act as a team that help maintain how the game plays (They make sure nothing is too strong in the game.) Database Administrators who make sure nothing goes wrong with the account information of all of the players. Anti-Cheat which will be installed automatically when the users install the game will help prevent unfair matches and provide fairness to all games. Patch-note pushers will users that help push out new patch notes that are basically in game updates (They essentially push updates.) Public Beta Environment testers are basically hand-on users but they help test the game before patch-note pushers push an update, basically they play the game and test the game on the update that is about to get pushed and help test the game if there are any bugs. Bug fixers help fix bugs and help look over bugs users have reported in game.

## 5e Other Stakeholders

Other stakeholders include marketing experts that can help this game take a good approach on how it should be marketed and if the marketing the game currently has needs improvements.

### 5f User Participation

We can expect users to participate during the development of the product when Alpha is released for this game. When Alpha is released we will release this game on a PBE (Public Beta Environment) launcher that will let users get familiar with the game as well as report any bugs they have encountered.

### 5g Priorities Assigned to Users

Since this game is targeted more for competitive play then casual we will be looking to balance this game out towards the competitive players instead. So if something a casual player thinks is unfair compared to what a competitive player thinks is unfair, we will choose what the competitive player thinks is unfair due to the fact we are marketing this game for a more competitive play-style.

## 6 Mandated Constraints

### 6a Solution Constraints

Cross play will be made available, anti-cheat and anti-bot mechanism will also be made available. Ultimately this game should be made available on every product so that every user has a chance of using this product.

### 6b Implementation Environment of the Current System

This product will be made to be available throughout all platforms including, Nintendo Switch, PC, Xbox, PlayStation, and Mobile. Also due to the fact that this product is made available throughout all these products, this will be made cross-play so players from all platforms can compete with each other.

### 6c Partner or Collaborative Applications

Since this game is free to play we will be looking for partners in cloud based data companies that could help provide a platform to store and backup the users account data. This will include applications like AWS, Google Cloud Platform, IBM, and ClearData.

### 6d Off-the-Shelf Software

ReBlaze is a software that must be included in the final product due to the fact that it is an anti-botting software. Due to this product being advertised as a free to play game across multiple platforms this product will help reduce bots that may clog up the cloud system which can negatively affect the servers.

### 6e Anticipated Workplace Environment

Some human factors regarding the environment in which the game will be used are noisy environments due to the fact that this product will be available throughout each platform so if the user is ever coming home from a train, noise may play a big role into the game, but since the game doesn't have many sounds. Due to it being a type of chess game it shouldn't affect the users play-style as much.

### 6f Schedule Constraints

Most of these things should be done when the Alpha of the product is released due to the fact that the game is made available for viewing and playing purposes. So if these constraints must be made by when Alpha is released so there aren't any server issues or our servers don't get clogged up on release day.

### 6g Budget Constraints

Since this game is being developed in the U.S some of the budgets we will be having during the release making the servers available only in the U.S for a short term. Obviously players from all around the world can play the game but they will have a drastic rise in ping compared to a NA player.

## 7 Naming Conventions and Definitions

### 7a Definitions of Key Terms

Rank will be defined as a player who plays this game competitively and has a specific rank such as Novice, Apprentice, Initiate, Adept, Magus, Master, Grandmaster, and Legendary. Username is the log in for the user when he launches the game that will never be shown to any of the other players and should be kept secret. Profile-Username is the profile name that will be seen by all players and acts as the display name. Champion-Points are points received from playing casual or ranked games that can be used to buy cosmetics in the game. Riot-Points are points that are bought from the Riot store and these will be used to buy better/newer cosmetics.

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

PBE means Public Beta Environment, RP means Riot Points, CP means Champion

Points, Rank in this game is measured from a scale of 1-8, where 1 is the best. So Novice = 8, Apprentice = 7, Initiate = 6, Adept = 5, Magus = 4, Master = 3, Grandmaster = 2, and Legendary = 1. Skins/Cosmetics are defined as something that changes the look of the game. Casual is defined as playing a game without a competitive drive. ESports is defined as the game being played in a competitive manner where the players win a large prize bracket. UML will be updated once the UML diagram is implemented.

### 7c Data Dictionary for Any Included Models

Data properties that are relevant to this project include the contents of a user account, so this will store all of the users information, such as the rank, time played, money spent, skins/cosmetics they have. Also the data structures of rank are relevant to this project due to

the ranks having a leader board where players who win a game go up and players who lose go down in rank. Each player will have a rank but also have a division rank which will separate plays based on what play style they have.

## 8 Relevant Facts and Assumptions

### 8a Facts

Chess has been around since the 7th century with 360,000 current active tournament players. League of Legends has a current daily player count of 115 million. MOBA is a current big genre of games where prize brackets for competitive play go over $1 million for teams who win tournaments.

### 8b Assumptions

Assumptions are that most users who play the game know basic strategy games, the basics of Chess, have played a MOBA before, understand logical deduction, understand Chess Strategies, understand the movement of the pieces, understand the concept of base and towers.

# II Requirements

## 9 Product Use Cases

### 9a Use Case Diagrams

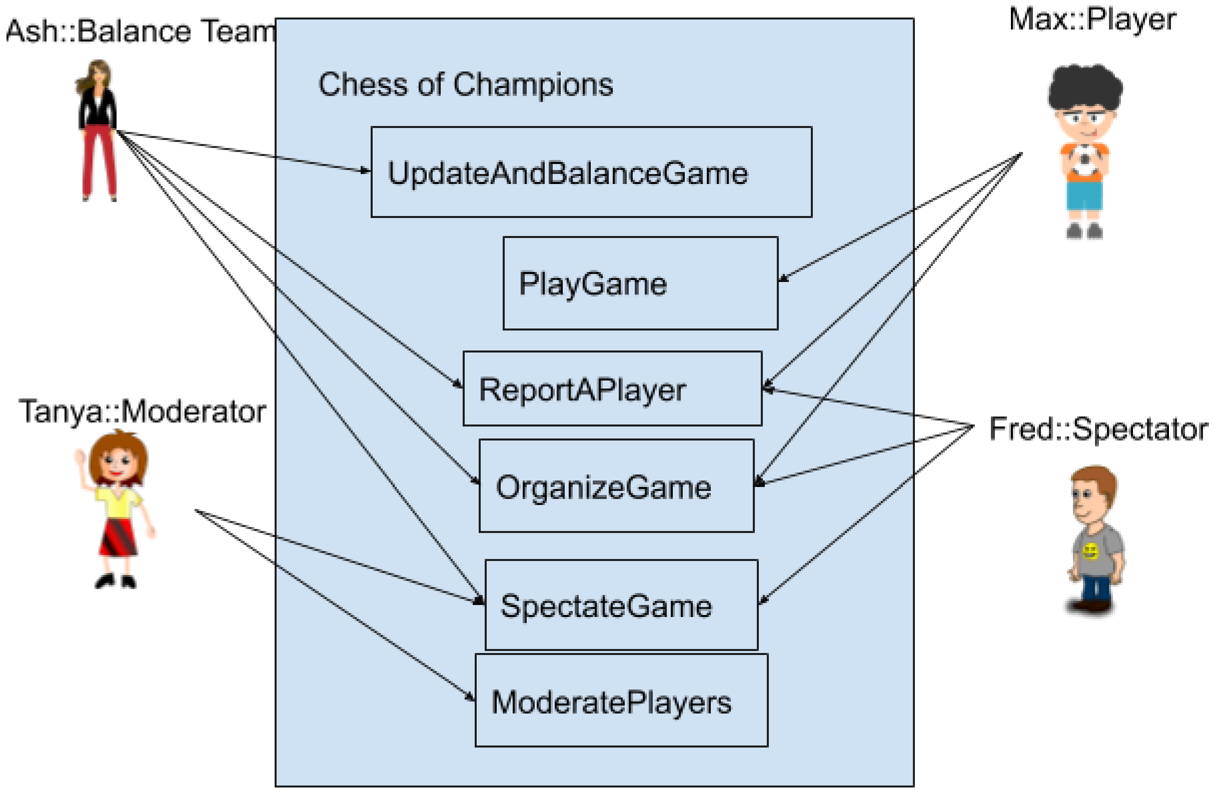


**Figure 1 - Chess of Champions *Play a game* Use Case Diagram**

## 9b Product Use Case List

*Chess of Champions*

| Use Case | Brief Description | User/Actor | Objective |
|---|---|---|---|
| UpdateAndBalanceGame | -Balance Team can balance the game then push these updates<br>-Balance the game based on something they deem *too strong* or *too weak* | Balance Team | Updating the game and ensuring it doesn't get stale. Along with balancing out strategies so the game won't be centered around that one strategy. |
| PlayGame | -Player can queue up for a game and be able to play one<br>-Player can queue up for either a ranked or casual game<br>-Player needs to have an account registered to play a game<br>-Player must not be banned to play a game | Player | Playing the game either casually or competitively. Without being put at a disadvantage by cheaters. Players must be honest players as well, not have any ban history or ban evasion. |
| OrganizeGame | -Choose between Ranked or Casual play<br>-Can select time limit for however long the game lasts for<br>-For Spectators they can watch their favorite players play against other people | Player<br>Spectator<br>Balance Team | Any of the actors are able to choose the preferences they prefer in a game. These include if they want the game to be casual or ranked and at a specific time limit. |

| | | | |
|---|---|---|---|
| | -For the Balance Team they can push for new game limits, such as a meta time limit or ranked players being matched with the same ranked divisions as each other. Or pair casual players with other casual players in the same skill bracket. | | |
| SpectateGame | -Can be able to view any current game being played<br>-Can be able to view current players of the current game<br>-Option to view the current players ranked of the current game<br>-For Moderator they can spectate games of players they deem suspicious<br>-For the Balance Team they can analyze these games to see if the game needs balancing (Basically checking if some strategy is too overpowered compared to others.)<br>-Games are able to be saved within 30 days of being played (Any games that have been played and pass the 30 days expiration are deleted) in the database for moderating or analyzing.<br>-Games can be viewed live or at any time as long as they don't go past the 30 day mark (So if a game is played at 09/03/25, it can't be viewed on 10/04/25, but | Spectator<br>Moderator<br>Balance Team | Any of these actors should be able to spectate any of the games that are currently being played without little to no delay. |

| | | | |
|---|---|---|---|
| | can be viewed on 10/02/25.) | | |
| ModeratePlayers | -A Moderator can be able to view a players account history (Checking if they were banned/muted before.)<br>-A moderator can ban a player based on evidence that the player was cheating<br>-A moderator can mute a player they deem as toxic/offensive during the current game.<br>-View everyone who is currently spectating the current game<br>-Get notified when a suspicious player queues up their next game<br>-Get notified when a player gets reported | Moderator | A moderator should be able to ban or warn players based on suspicious players. They should also be able to respond to the reports made by other actors and react to those reports. |
| ReportAPlayer | -Able to send a report of the player you deem as suspicious. With a brief description on what they are doing that is suspicious | Player<br>Spectator<br>Balance Team | Any user that feels as if they are at an unfair advantage due to a 3rd party program should be able to report a player they deem as suspicious. |

## 9c Individual Product Use Cases

**Use case ID:** Moderate **Name:** Ban/Mute/Warn Players

**Pre-conditions:** Suspicious player has been reported by a player, spectator, or a balance team member.

**Post-conditions:** Suspicious players account history is provided, showing past offenses, warnings, or temporary bans.

**Initiated by:** Moderator

**Triggering Event:** Suspicious player is seen as suspicious due to either cheating, abusing a bug, or being toxic.

**Additional Actors:** Spectator, Player, Balance Team

**Sequence of Events:**

1. A player is reported by a spectator, another player, or a balance team member due to suspicious behavior as a result the Moderator is alerted.

2. Moderator retrieves the reported players past account data, showing all previous offensives.

3. Moderator spectates the current suspicious player game or the past games they have played.

4. Moderator uses anti-cheat software and it's deployed, checking to see if it was bypassed or if it was set off by the reported player during that current game or past game. Or checks the text logs if the reported player was saying anything offensive.

5. Moderator checks if the reported player is flagged for trying to bypass the anti-cheat software or if the player is flagged for offensive language.

6. Players are either banned for cheating, warned/muted for offensive language depending how offensive it was. If a player had previous offenses moderator bans them due to being warned/muted before.

**Alternatives:** Reported player is banned / warned / muted depending on the offense.

**Exceptions:** Reported player was false reported meaning the report wasn’t true.

## 10 Functional Requirements

**F1 - Patching/Update Requirement**

**Description:** The system must provide a way to send out game updates to every client

**Rationale:** The game will be maintained and balanced frequently so it is important that players are able to receive updates quickly and continue playing.

**Fit Criterion:** Players must be able to receive updates as soon as they are pushed out and the game client should be able to receive and install updates quickly.

**Acceptance Tests:** Test1- Patches/Updates Sent

**F2 - Account Management Requirement**

**Description:** The system must provide a way for players to create accounts and log in to accounts that will hold that player’s information.

**Rationale:** A User’s account will hold important information such as past game records, account status, and friends

**Fit Criterion:** Players must be able to create an account and log in to that account which the game client should be able to hold the accounts past game records, account status, and friends.

**Acceptance Tests:** Test2 - Account Information Stored

**F3 - User Interface Game Search Requirement**

**Description:** The system must provide the player with an interface to choose the specifications of the game they want to play

**Rationale:** There are different game modes and specifications so the player must be able to easily choose which specific mode they want to play

**Fit Criterion:** The player should be able to select a game mode to look for a match

**Acceptance Tests:** List ID# and/or names here . . .

### F4 Multiple Connection Requirement

**Description:** The system must be able to provide a stable connection between all players and spectators in a game.

**Rationale:** Each game will have multiple players connected at once and in order for a good experience, players should be able to have a stable connection to each other
**Fit Criterion:** Players should have no connection errors during the course of a game

**Acceptance Tests:** List ID# and/or names here . . .

### F5 Post Game User Interface Requirements

**Description:** The system must provide a way to see results of a game and interact with other players after a game.

**Rationale:** Players should have some way of receiving feedback of the game they just played and be able to interact with the other players in some way. This interface will also allow players to report suspicious activity.

**Fit Criterion:** There should be an organized result screen with statistics of the match information and a way for players to interact with each other by friending, chatting, or reporting

**Acceptance Tests:** List ID# and/or names here . . .

### F6 Moderation Requirement

**Description:** The system must provide a way to have users with higher permissions to be able to moderate other players and also have an automated moderation system

**Rationale:** There will be players that will try to cheat or may say harmful things to other players so there should be a group to ensure this doesn't happen
**Fit Criterion:** Player accounts must be able to have different statuses with different permissions. An accounts status should be easily recognizable. Obvious offences should be detected and dealt with automatically

**Acceptance Tests:** List ID# and/or names here . . .

### F7 Spectator Requirement

**Description:** The system must provide a way for players to view ongoing games in real time with an easy to use spectator client

**Rationale:** Players may enjoy watching their friends play and if there is a large event such as a tournament, there must be a way for games to be watched

**Fit Criterion:** Any ongoing game must be able to be spectated by anyone with the correct permissions

**Acceptance Tests:** List ID# and/or names here . . .

**F8 User Interface Profile Information**

**Description:** The system must be able to provide the user with a way to view their profile and relevant information regarding it such as past game stats and saved game recordings

**Rationale:** Being able to see past records is useful for players to improve

**Fit Criterion:** Players should be able to view information about the rank, past game stats, and past recordings of games

**Acceptance Tests:** List ID# and/or names here . . .

## 11 Data Requirements

### D1 - Moderator Account Information

**Description:** The system must be able to provide the moderator the players account information which displays the players past offensives.

**Rationale:** The system must be able to hold the player account information which holds important information such as the number of times they have been reported, warned, muted, and banned.

**Fit Criterion:** The moderator should be able to view the offensives players have made by accessing the account information at any given time.

**Acceptance Tests:** List ID# and/or names here . . .

### D2 - Player Account Information

**Description:** The system must be able to provide the players the players account information which displays the players email, username, and their name.

**Rationale:** The system must be able to collect the player account information that the player is able to access holds important information such as the email they registered with, their name, and their current account user name.

**Fit Criterion:** The moderator should be able to view their email, username, and their name by accessing the account information and this information must be updated after each game the players has played.

**Acceptance Tests:** List ID# and/or names here . . .

## D3 - Player Report Information

**Description:** The system must be able to provide the Moderator of the reasoning the player reports information when a player is reported.

**Rationale:** The System must hold the player reports information which holds information such as the reason the player has been reported, either being cheating, offensive language, offensive username, or bug exploiting.

**Fit Criterion:** The application should be able send the reasoning behind the report of the current player to the moderator everytime the player has been reported.

**Acceptance Tests:** List ID# and/or names here . . .

## D4 - Game Patch Information

**Description:** The system must be able to provide the Players the current Patch/Update Information on what the current launcher/game is running on.

**Rationale:** The system must have the current Patch/Update Information displays info such as the current patch, and the updates that occurred in the game.The players should be able to know what current patch they are in the current game and what updates occurred in the current patch such as bug fixes, balance to the game, etc.

**Fit Criterion:** The application must be able to take multiple accesses to this information from any given player and should be able to display this information at any given time.

**Acceptance Tests:** List ID# and/or names here . . .

# 12 Performance Requirements

## 12a Speed and Latency Requirements

### PR1 - Connection Between Two Players

**Description:** The system must be able to provide a stable connection towards all players who are currently playing a game.

**Rationale:** The system must be able to connect players who may be in different locations/time zones.

**Fit Criterion:** Each player should be able to play the game with little to no lag no matter what their location/time zone is.

**Acceptance Tests:** List ID# and/or names here . . .

### PR2 - Connection a Game and Spectators

**Description:** The system must be able to provide a delay for spectators as they are viewing a live game.

**Rationale:** The system must delay the people who are spectating the game, so this ensures there is no cheating coming from the spectators (No backseat gaming / Ghosting).

**Fit Criterion:** Each game that is viewed by a spectator must be provided a 3 minute delay to ensure that there is no cheating coming from the spectators.

**Acceptance Tests:** List ID# and/or names here . . .

# 13 Dependability Requirements

## 13a Reliability Requirements

#### DR1 - System Fails

**Description:** The System must never be allowed to fail when it is not being updated/patched and provide users access to the system at all times.
**Rationale:** The System must provide servers that are not allowed to fail and servers that are continuously allowing users access to the system.

**Fit Criterion:** The application must not ever fail no matter how many players are on the server at any given time.

**Acceptance Tests:** List ID# and/or names here . . .

#### DR2 - Patch/Updates

**Description:** The System must be able to be shut down each time a patch/update is about to be pushed, so a new update for the game can be pushed to all players.

**Rationale:** The System must provide the Balance Team a way of shutting down the system to provide a push for the update of the game.

**Fit Criterion:** The application must be able to shutdown then reboot in 10 minutes to provide players with a new patch/update.

**Acceptance Tests:** List ID# and/or names here . . .

### 13b Availability Requirements

#### AV1 - Servers Runtime

**Description:** The System must be able to run continuously which allows for players to constantly play games and access their account.

**Rationale:** The System must provide servers that are able to be played / accessed / log into/ create an account /for all players, the system will be unable to provide players to play the game if the servers are not running continuously.

**Fit Criterion:** This application must be able to maintain 24/7 servers for players to access which the players can use to play the game or log into or create an account..

**Acceptance Tests:** List ID# and/or names here . . .

## 14 Maintainability and Supportability Requirements

## 14a Maintenance Requirements

### MSR1 - Balance Team Updates/Patches

**Description:** The System must be able to retrieve updates and maintenance from the balance team.

**Rationale:** The System must provide a way for the balance team to push for updates/patches, if they are unable the system will be unable to maintain a bug free / fixed game.

**Fit Criterion:** This application must be able to allow the balance team to push updates/patches at any given time which the balance team can give the players a ET on when the update will occur.

**Acceptance Tests:** List ID# and/or names here . . .

## 14b Supportability Requirements

### SR1- Ticket Support

**Description:** The System must be able to provide support for the players through which the player sends support tickets to the help desk.

**Rationale:** The System must be able to provide a way for players to get in contact with the support team in order to ask about their account status / account information which then the support team can provide the help the player needs.

**Fit Criterion:** The application must be able to receive support tickets and collect the account information of who sent the ticket at least every time a ticket is sent.

**Acceptance Tests:** List ID# and/or names here . . .

## 14c Adaptability Requirements

### AdR1 - System Requirements

**Description:** The System must provide support for all platforms such as Windows, Consoles, MacOS, and Linux and be able to run on any of these platforms.

**Rationale:** The System must be able to run properly throughout all of these platforms; otherwise, the system will be unable to provide an accessible platform for players using these platforms.

**Fit Criterion:** The application should be able to maintain a stable connection with the players without crashing regardless of what system they are using.

**Acceptance Tests:** List ID# and/or names here . . .

# 15 Security Requirements

## 15a Access Requirements

### AR 1 - Account Statuses

**Description:** The system must have distinct account statuses that determine what permissions they have and what they can access

**Rationale:** Normal players do not need access to some functionality but the moderation team should have more access to help moderate players.

**Fit Criterion:** Accounts with moderator status should be able to see chat logs, other players game history, report history and will also have permission to mute or ban other players

**Acceptance Tests:** List ID# and/or names here . . .

## 15c Privacy Requirements

### PrivR1 - Secure Player Account Information

**Description:** The System must provide a means of security for the players account information which other players can't access.

**Rationale:** The System must be able to maintain and keep account information such as emails, names, passwords, and usernames from the view of other players otherwise, the system will be unable to protect the login information for the players allowing unauthorized access to the account.

**Fit Criterion:** The application should be able to maintain a secure database of the players account information 24/7.

**Acceptance Tests:** List ID# and/or names here . . .

## 15e Immunity Requirements

### IR1 - DDOS attacks

**Description:** This System server should have the ability to resist DDOS attacks from affecting the connections of the users.

**Rationale:** The System should be able to provide a stable connection to the servers for all players without the option of facing a drop in connection due to a DDOS attack on the server.

**Fit Criterion:** The application must be able to withstand/resist DDOS attacks at any given attack.

**Acceptance Tests:** List ID# and/or names here . . .

### IR2 - 3rd Party Cheats

**Description:** This System should have the ability to prevent any 3rd party cheats from impacting any player game.

**Rationale:** The System should be able to provide a fair advantage for all players playing the game by providing anti-cheat to prevent 3rd party cheats impacting any game played. Otherwise, if it is unable to provide this it will have players cheating through each game they play providing an unfair advantage.

**Fit Criterion:** The application must be able to block out any 3rd Party Programs that try to interfere with the game whenever the 3rd party program is launched.

**Acceptance Tests:** List ID# and/or names here . . .

# 16 Usability and Humanity Requirements

## 16a Ease of Use Requirements

### EoU 1 - UI Usability

**Description:** The system should be easy to navigate and allow players to move around menus easily without any tutorials

**Rationale:** The interface should be the simplest part of the game. If players are confused by the UI, they may be turned away from trying the game

**Fit Criterion:** Users should be able to navigate to any given menu without any trouble

**Acceptance Tests:** List ID# and/or names here . . .

## 16c Learning Requirements

#### LR1 - Tutorial

**Description:** The System must provide a tutorial on how the game is played to provide an understanding for new players of the game.

**Rationale:** The System should be able to provide new players and current players a tutorial section which will provide information on how the game is played. Otherwise, new players will be unable to understand the basics and concept of the game.

**Fit Criterion:** The application should be able to provide new users an option to play the tutorial when they first log into the game launcher.

**Acceptance Tests:** List ID# and/or names here . . .

#### LR2 - Ranked Explanation

**Description:** The System must provide a ranked explanation to provide players an explanation of the divisions in competitive play.

**Rationale:** The System should be able to provide all players with ranked information on how ranked divisions in the game work and how the player ranks up. Otherwise, the player will not be able to understand the difference between other players' ranks.

**Fit Criterion:** The application should be able to provide all players with ranked information whenever the player accesses the ranked information interface.

**Acceptance Tests:** List ID# and/or names here . . .

## 17 Look and Feel Requirements

### 17a Appearance Requirements

#### ApR1# - Look Requirements

**Description:** The system must provide a colorfullish, large logo, in a cartoonish like appearance to reflect the branding of the product.

**Rationale:** The system should use a colorful, large logo, in a cartoonish like appearance to help appeal towards the targeted audience, otherwise, without the use of these appearances, the product won't match the type of branding this product is supposed to have.

**Fit Criterion:** This application should use the vast majority of color in its appearance and a cartoonish appearance as well to it to feel like the users are playing a kid friendly game, so for every child/teen who browses the product can get appealed towards it.

**Acceptance Tests:** List ID# and/or names here . . .

### 17b Style Requirements

**StR1 - Target Audience Style**

**Description:** The System must provide a friendly/cool packaging product style to provide an appeal towards the teenage/children demographic.

**Rationale:** The system should use a friendly/cool packaging product style to reach/appeal to the target audience, otherwise, the target audience wouldn't be reached.

**Fit Criterion:** The application should be able to appeal towards the target audience of children and teens and maintain this target audience on a daily basis for account creations and logins.

**Acceptance Tests:** List ID# and/or names here . . .

## 18 Operational and Environmental Requirement

### 18c Productization Requirements

**PrR1 - Installation**

**Description:** The System must be available to be downloaded from Steam, Epic Games Launcher, Xbox, Playstation, Switch, stores.

**Rationale:** The system must be able to be downloaded through any of the following online stores that range from any platform, otherwise, if they are unable to be downloaded then this will lower the amount of players for the game.

**Fit Criterion:** This application should be available to download for most platforms without any crashes or downloading errors.

**Acceptance Tests:** List ID# and/or names here . . .

### 18d Release Requirements

**RR1 - Alpha**

**Description:** The system must be able to provide the players a Alpha version of the product to help test the stability of the game.

**Rationale:** The system must be able to provide a Alpha release version of the game to help provide a way for the developers to game breaking bugs and to help test the servers, otherwise, the developers won't be able to test how the product will hold up on the actual release of the product.

**Fit Criterion:** This application should be able to maintain the servers up for a limited amount of Alpha testers, 400 of these alpha testers should be able to report bugs and not be disconnected from the server.

**Acceptance Tests:** List ID# and/or names here . . .

**RR2 - Beta**

**Description:** The system must be able to provide the players a Beta version of the product to help test the stability of the game.

**Rationale:** The system must be able to provide a Beta release version of the game to help provide a way for the developers to game breaking bugs and to help test the servers and on top of that make sure all of the bugs from the Alpha version are fixed, otherwise, the developers won't be able to test how the product will hold up on the actual release of the product.

**Fit Criterion:** This application should be able to maintain the servers up for a limited amount of Beta testers, 1,400 of these beta testers should be able to report bugs and not be disconnected from the server or not run into any of the previous bugs they encountered in Alpha release.

**Acceptance Tests:** List ID# and/or names here . . .

**RR3 - Final Release**

**Description:** The system must be able to provide the players a final version of the product that doesn't provide any of the bugs from the Alpha or Beta versions.

**Rationale:** The system must be able to provide a final version of the game in order for players to fully play the game without any of the bugs in Alpha or Beta release, otherwise, the players won't be able to play a bug free game that they were testing alpha and beta versions for to fix those bugs.

**Fit Criterion:** This application should be able to maintain the servers up for any amount of players, these players should be able to report bugs and not be disconnected from the server or not run into any of the previous bugs they encountered in Alpha or Beta release.

**Acceptance Tests:** List ID# and/or names here . . .

## 21 Requirements Acceptance Tests

### 21a Requirements – Test Correspondence Summary

| Test 1 | Test 2 | Test 3 | Test 4 | Test 5 | Test 6 | Test 7 | Test 8 | Test 9 | Test 10 | Test 11 | Test 12 | Test 13 | Test 14 | Test 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F1<br>F4<br>DR2<br>MSR2 | F2<br>D1<br>D2 | F3 | F3<br>EoU1 | F5 | F4 | F6<br>IR2 | F7<br>PR2 | F6<br>AR1<br>D2<br>D3 | F8<br>D2 | D3 | D4 | MSR1<br>RR1<br>RR2<br>RR3 | F1<br>MSR1 | DR1<br>AV1 |

| Test 16 | Test 17 | Test 18 | Test 19 | Test 20 | Test 21 | Test 22 | Test 23 | Test 24 | Test 25 |
|---|---|---|---|---|---|---|---|---|---|
| SR1 | ADR1 | AR1 | PrivR1 | IR1 | LR1 | LR2 | ApR1<br>StR1 | PrR1 | RR1<br>RR2<br>RR3 |

### 21b Acceptance Test Descriptions

#### Test1 - Updates/Patches Sent

**Description:** Two players who are in different time zones are able to receive the same patch/update as soon as it is released.

#### Test2 - Account Information Stored

**Description:** A player is able to create an account and play at least one game on the game, adding the player they played against. Then log out and log back into the game which they can see their past match history along with the player they added onto their friends list.

### Test3 - Game Selection

**Description:** A player will login to the game and then queue up for a ranked game. When the game ends, the user can check the match history of the game and if it shows that the player has played a ranked game instead of a normal game the test is passed.

### Test4 - Interface Usability

**Description:** A user will be told to navigate and search for a specific game mode selection with no instructions. There should be no confusion on how to navigate the menus.

### Test5 - Add Friend

**Description:** A player will login to the game and then search for another existing player, which then the player can add the friend. The player that is searched for will then accept the friend request which then both players should see each other's names on both of their lists.

### Test6 - Game Connection

**Description:** A player will connect into a game with another player. The two users should be able to play through an entire game without any connection issues.

### Test7 - Report Feature

**Description:** Two players will queue up into a game, where one player types profanity and uses a 3rd party program while the other player reports the player. If the reported player is banned then the test is passed.

### Test8 - Spectate

**Description:** Two players will add each other as friends, one player will queue up into a game, while the other spectates the game. Both players will record both of the games. If the videos match each other then the spectate test passes.

### Test9 - Mod Account Info

**Description:** A moderator will login the game and select a player off their friends list, the player being a player who has been reported 5 times, warned 1, and muted 2. The moderator will select the option to view the account information of that player, if the account information matches the number of times the player has been reported, warned, and muted then the test passes.

### Test10 - Player Account Info

**Description:** A player will create a new account with the following information, email: l@gmail.com, username: l, name: l, the player will then request their account information. If the following information displayed by the product matches what the user typed in then the following test passes.

### Test11 - Report Reason

**Description:** A player will queue up into a game then cheat with a 3rd party program, while the other player reports this player for cheating. When the moderator receives this report call it should match with the reason the player was reported for if it doesn't then the test doesn't pass.

### Test12 - Patch Version

**Description:** A player should be able to login the game and check the current version of the client. If the player isn't able to check the current version of the game then the test fails.

### Test13 - Update Game

**Description:** A player will login to the game then check that the following version is Ver: 1.2, then the balance team should push an update for Ver: 1.2.1, when the player logs back into the game and their games version says 1.2.2 then the test is passed.

### Test14 - Update Notification

**Description:** A player should be able to login the game which then the balance team should push an update for the game at any given time limit. If the player receives that the game is being updated in that same time limit then the test passes.

### Test15 - Server Runtime

**Description:** A player should be able to login to the game every 5 minutes then log back out every 5 minutes for every day in a week. If the player disconnects due to the servers and not the player's connection then the test fails.

### Test16 - Ticket Support

**Description:** A player should be able to login the game where the player's username is “Test” and send a ticket to the support with the following message, “I don’t know what email this account is on.” If the support team receives that exact message with the exact username then the test passes.

### Test17 - System Requirements

**Description:** A player should be able to download and login to the same account with the Username: Test, throughout all of the following platforms, Windows, Consoles, MacOS, and Linux. If the player isn’t able to connect or download the game from any of these devices the test fails.

### Test18 - Account Status

**Description:** A moderator should be able to log into the game then check the status of their account which then they can compare it to a non-moderator account, if both status are different the test passes.

### Test19 - Flagging Access

**Description:** A player should be flagged if they are logging into an account from a different IPaddress blocking their account. If the player isn’t flagged then the test fails, but if it does it passes.

### Test20 - Blocking DDOS

**Description:** A DDOS attack should be initiated into the servers, if the servers contain to stay up then the test passes.

### Test21 - Tutorial

**Description:** A player should be able to create a new account and once they load into the game they should be prompted to take the tutorial, they should be able to decline or accept. If they aren't given this message after creating a new account then the test fails.

### Test22 - Ranked

**Description:** A player should be able to log on to the game which they can click the ranked explanation GUI, which should explain the importance of rank and the different divisions. If there is no screen for clicking the ranked explanation then the test fails.

### Test23 - Look/Feel

**Description:** A player should be able to log on to the game and if they are greeted with colorful colors, large logos, and cartoonish feel to the game, then the test passes.

### Test24 - Download from online Store

**Description:** A player should be able to find the game on either Steam, Epic Games Launcher, Xbox, Playstation, Switch, stores. If it is found on all of these stores then the test passes.

### Test25 - Report Bugs

**Description:** A player should be able to log into the game which then they can click "report a bug", the player will type Reported Bug: "Test", if the Balance team sees the bug message as "Test", then the test passes.

# III Design

## 22 Design Goals

*SV: Identify the important design goals that are to be optimized in the proposed design.* Your text goes here . . .

## 23 Current System Design

*SV: **IF** the proposed new system is to replace an existing system, then the current system should be described here. Otherwise insert a brief statement that there is no pre-existing system.*

Your text goes here . . .

## 24 Proposed System Design

*This section will make heavy use of class diagrams, and also sequence and deployment diagrams where noted. However don't overlook finite state, activity, communication, or other diagram types as needed for effective communication.*

### 24a Initial System Analysis and Class Identification

*SV: Perform grammatical and similar analyses to identify the most import and obviously needed classes, and to organize them into an initial class structure. An initial class diagram is appropriate, containing few if any internal details.*

Your text goes here . . .

### 24b Dynamic Modelling of Use-Cases

*SV: Insert sequence diagrams of ( at least the most important ) use-cases, as a means of identifying other needed classes.*

Your text goes here . . .

### 24c Proposed System Architecture

*SV: Identify the Software Architecture to be applied to this project, such as Client Server, Repository, MVC, etc., along with justification for the choice.*

Your text goes here . . .

### 24d Initial Subsystem Decomposition

*SV: A slightly more detailed class diagram, showing the classes identified in sections 24a, 24b, and 0 above, partitioned into subsystems. For each subsystem provide a brief description of the subsystem, including its key responsibilities. There should still be few if any internal details.*

Your text goes here . . .

## 25 Additional Design Considerations

*SV: The sections listed here do not need to be presented in the order given, and may not all be relevant for any particular project. Those that are relevant can help identify additional classes that are needed as a result.*

### 25a Hardware / Software Mapping

*SV: This is particularly important for distributed systems, such as those employing a client-server architecture. Use a deployment diagram to indicate which subsystems are mapped onto which piece(s) of hardware, and what communication subsystems need to be added to the system as a result.*

Your text goes here . . .

### 25b Persistent Data Management

*SV: Document the classes and perhaps subsystems necessary to store persistent data when the system shuts down, and to restore that data when the system starts back up again.*

*Reiterate key data structures and information as necessary for the understanding of this design phase. Refer the reader back to the data dictionary in section* ***Error! Reference source not found.*** *to avoid undue repetition, while reviewing only the most relevant items here.*

Your text goes here . . .

### 25c Access Control and Security

*SV: Identify the access control and security concerns for this system, and the new classes and/or subsystems that must be added to handle those concerns.*

Your text goes here . . .

### 25d Global Software Control

*SV: Identify the global software control concerns for this system, and the new classes and/or subsystems that must be added to handle those concerns.*

Your text goes here . . .

### 25e Boundary Conditions

*SV: Identify the boundary condition concerns for this system, and the new classes and/or subsystems that must be added to handle those concerns. In particular consider startup, shutdown ( normal or abnormal ), and the creation and/or maintenance of any configuration files, databases, or similar supporting data files.*

Your text goes here . . .

### 25f User Interface

*SV: Include a preliminary user interface design here, possibly as a rough sketch or other mockup, in order to identify additional classes needed to implement the interface.*

Your text goes here . . .

### 25g Application of Design Patterns

*SV: Any design patterns applied as a result of previous sections should have been addressed there, and identified as such at the time. Use this section to document only the additional design patterns that were not previously covered elsewhere. ( If any. )*

Your text goes here . . .

## 26 Final System Design

*SV: Include here the final version of the overall system design, incorporating all the subsystems and classes added as a result of additional design considerations. Multiple diagrams may be needed, possibly starting with an overall package diagram showing* *all the different subsystems and the ( important ) classes contained within each one. Still not a lot of internal details.*

Your text goes here . . .

## 27 Object Design

*This section documents the internal details of each class, to the extent that they can be designed at this time. Included should be the class interfaces ( public method signatures and responsibilities ) and constraints. It is probably best to break this section up into subsections corresponding to subsystems as documented above, and/or by ( Java ) packages if those are designed. It may also be appropriate to address additional design pattern considerations here, but not to the point of being redundant of previous documentation.*

*Certain methods, such as simple getters, setters, and constructors are not always documented, unless there is something special about them such as in the Singleton or Factory Method design patterns.*

### 27a Packages

*SV: If the design involves assigning classes to packages ( .e.g Java packages ), then the packages to be created should be documented here.*

Your text goes here . . .

### 27b Subsystem I

Your text goes here . . .

### 27c Subsystem II

Your text goes here . . .

### 27d etc.

Your text goes here . . .

# IV Project Issues

## 28 Open Issues

*SV: Issues that have been raised and do not yet have a conclusion.*

Your text goes here . . .

## 29 Off-the-Shelf Solutions

*SV: Discussion of products or components currently available that could either be incorporated into the new solution or simply used instead of developing ( parts of ) the new solution. The distinction between sections 35 a, b, and c is subtle, and not very important.*

Your text goes here . . .

### 29a Ready-Made Products

*SV: Products available for purchase that could be used either as part of a solution or instead of ( a part of ) a solution.*

Your text goes here . . .

### 29b Reusable Components

*SV: Similar to 35a, but for components such as libraries or toolkits instead of fully blown products.*

Your text goes here . . .

### 29c Products That Can Be Copied

*SV: Products that could legally be copied would typically be past projects developed by the same development group, provided there were no restrictions that would prevent their reuse.*

Your text goes here . . .

## 30 New Problems

*SV: The proposed new system certainly has its benefits, but it could also raise new problems. It is a good idea to identify any such potential problems early on, rather than being surprised by them later.*

### 30a Effects on the Current Environment

*SV: Could the new system have any adverse effects on the working environment, e.g. the way people do their jobs?*

Your text goes here . . .

### 30b Effects on the Installed Systems

*SV: Could the new system have any adverse effects on other hardware or software systems?*

Your text goes here . . .

### 30c Potential User Problems

*SV: Could the new system have any adverse effects on the users of the software? Could users possibly have a negative response to the new system?*

Your text goes here . . .

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

*SV: Are there any ( physical ) limitations in the expected environment that could inhibit the proposed product? ( e.g. weather, electrical interference, radiation, lack of reliable power, etc. )*

Your text goes here . . .

### 30e Follow-Up Problems

*SV: Basically any other possible problems that could occur.*

Your text goes here . . .

## 31 Migration to the New Product

*SV: This section only applies when there is an existing system that is being replaced by a new system, particularly when data must be preserved and possibly translated / reformatted. Otherwise just write "Not Applicable" under section 38 and remove sections 38a and 38b.*

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

*SV: These are a list of requirements relevant to the migration procedures. For example a requirement that the two systems be run in parallel for a time until the client is satisfied with the new system and the users know how to use it.*

Your text goes here . . .

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

*SV: This section specifically addresses **data** that must be preserved and/or translated / reformatted during the migration process.*

Your text goes here . . .

## 32 Risks

*SV: Consideration of the potential risks that could cause the project to fail / underperform.*

Your text goes here . . .

## 33 Costs

*SV: An estimate of what it will cost to complete this project. Think not only in terms of dollars, but also time, resources, lost opportunities, etc.*

Your text goes here . . .

## 34 Waiting Room

*SV: This is a place to record ideas or wishes that will not be included in the current release of the product, but which might be worth reconsidering at a later date.*

Your text goes here . . .

## 35 Ideas for Solutions

*SV: When developing requirements only, it is not the role of the business analyst to dictate the implementation of the solution. However they can pass along any ideas they have here as suggestions to the developers. For CS 440 this report includes system and object design, so this section would make suggestions for implementation and* *testing that would come after design, such as the use of a particular language, IDE, library, or other tools.*

Your text goes here . . .

### 36 Project Retrospective

*SV: At the conclusion of the ( CS 440 ) project, reflect back on what worked well and what didn't, and how the process could be improved in the future.*

Your text goes here . . .

## V Glossary

*SV: The glossary is a more complete and inclusive dictionary of defined terms than that found in section I.7.a, the latter of which only covered the most important key terms needed to understand the report.*

Your text goes here . . .

## VI References / Bibliography

*This section describes the documents and other sources from which information was gathered. This sample bibliography was generated using the "Insert Citation" and "Bibliography" buttons in the "Citations & Bibliography" section under the "References" tab of MS Word. Creating new citations will not update this list unless you click on it and select "Update Field". You may need to reset the style for this paragraph to "normal" after updating.*

[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.

## VII Index

*This section provides an index to the report. The sample below was generated using the "Mark Entry" and "Insert Index" items from the "Index" section on the "References" tab, and can be automatically updated by right clicking on the table below and selecting "Update Field". To*

*remove marked entries from the document, toggle the display of hidden paragraph marks ( the paragraph button on the "Home" tab ), and remove the tags shown with XE in { curly braces. }*

63 Requirements................................ 35, 51,

58

Test..................................................... 64, 65

Design ................................................ 61,