ONLINE ESPORTS COMPETITIVE TEAM FINDER WEB APP WITH BUILT-IN TOURNAMENT CREATOR, CUSTOM LOBBY MATCH-UP AND PROFILE RATINGS BASED ON PARTICIPATIONS AND COMPLIMENTS

BY
YEE ZI YANG

A REPORT
SUBMITTED TO
Universiti Tunku Abdul Rahman
in partial fulfilment of the requirements
for the degree of
BACHELOR OF COMPUTER SCIENCE (HONOURS)
Faculty of Information and Communication Technology
(Kampar Campus)

JAN 2022

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: ONLINE ESPORTS COMPETITIVE TEAM FINDER WEB APP WITH BUILT-IN TOURNAMENT CREATOR, CUSTOM LOBBY MATCH-UP AND PROFILE RATINGS BASED ON PARTICIPATIONS AND COMPLIMENTS

Academic Session: JAN 2022

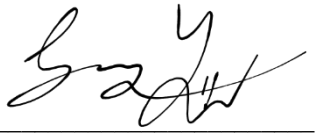
I YEE ZI YANG

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,



(Author's signature)

(Supervisor's signature)

Address:
12, Persiaran Merdeka
Taman Merdeka 30100,
Ipoh Perak

Syed Muhammad Syed Omar
Supervisor's name

Date: 20 April 2022

Date: 21/04/2022

FACULTY/INSTITUTE* OF Information and Communication Technology

UNIVERSITI TUNKU ABDUL RAHMAN

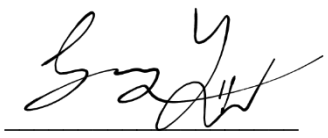
Date: 20 April 2022

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ___ *Yee Zi Yang* ___ (ID No: *18ACB02834*) has completed this final year project/ ~~dissertation/ thesis*~~ entitled "Online Esports Competitive Team Finder Webapp with Built-in tournament creator, custom lobby match-up and profile ratings based on participations and compliments" under the supervision of Encik Syed Muhammad Bin Syed Omar (Supervisor) from the Department of Information Systems, Faculty/Institute* of Information and Communication Technology, and _____ (Co-Supervisor)* from the Department of _____, Faculty/Institute* of _____.

I understand that University will upload softcopy of my final year project ~~/ dissertation/ thesis~~* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

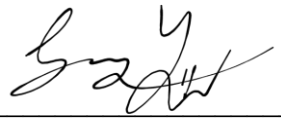
Yours truly,



(*Yee Zi Yang*)

DECLARATION OF ORIGINALITY

I declare that this report entitled "**ONLINE ESPORTS COMPETITIVE TEAM FINDER WEB APP WITH BUILT-IN TOURNAMENT CREATOR, CUSTOM LOBBY MATCH-UP AND PROFILE RATINGS BASED ON PARTICIPATIONS AND COMPLIMENTS**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : 

Name : YEE ZI YANG

Date : 20th Apr 2022

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Encik Syed Muhammad Bin Syed Omar who has given me this bright opportunity to engage in Web Development and UX design project. It is my first step to establish a career in the web design field. A special thanks to you.

ABSTRACT

This is a web application development project for academic purposes. The following proposal project title is to provide students or those who are not familiar with the concept of competitive gaming scene with the history of Esports and its associated competitive options. The proposed project title also provides several solutions to the underlying problems that are faced in the current competitive gaming scene involving around the hassle of finding a team or matchmaking, flexibility of lobby making and tournament hosting and also the problem of player toxicity. The proposed solution and the work done that revolves around the aforementioned problem would be providing a platform for players for the sole purposes of team building across multiple games and regions, other than that, a customizable lobby and tournament would also be viable. To tackle the problem of human nature, hence toxicity, one of the solutions is to introduce a rating system that is calculated based on the players' participation in lobbies or tournaments and also compliments from other players.

TABLE OF CONTENTS

FRONT PAGE **i**
REPORT STATUS DECLARATION FORM **ii**
FYP THESIS SUBMISSION FORM **iii**
DECLARATION OF ORIGINALITY **iv**
ACKNOWLEDGEMENTS **v**
ABSTRACT **vi**
TABLE OF CONTENTS **vii**
LIST OF FIGURES **x**
LIST OF TABLES **xii**
LIST OF ABBREVIATIONS **xiii**

CHAPTER 1 INTRODUCTION **1**
1.1 Problem Statement and Motivations 1
1.2 Objectives 3
1.3 Project Scope 4
1.3.1 Team Finder Module 4
1.3.2 Custom Lobby Module 4
1.3.3 Custom Tournament Module 5
1.3.4 Player Rating Module 5
1.4 Contributions 6
1.5 Report Organization 6

CHAPTER 2 LITERATURE REVIEW **7**
2.1 Base System 8
2.1.1 Base System Analysis 8
2.1.2 Base System Strengths 9
2.1.3 Base System Weaknesses 9
2.1.4 Base System Recommendations 10
2.2 Match.tf System 11
2.2.1 Match.tf Analysis 11
2.2.2 Match.tf Strengths 12
2.2.3 Match.tf Weaknesses 12

2.2.4 Match.tf Recommendations 13
2.3 FACEIT System 14
2.3.1 FACEIT Analysis 15
2.3.2 FACEIT Strengths 15
2.3.3 FACEIT Weaknesses 16
2.3.4 FACEIT Recommendations 17
2.4 Community PUG System 18
2.4.1 Community PUG Analysis 19
2.4.2 Community PUG Strengths 19
2.4.3 Community PUG Weaknesses 20
2.4.4 Community PUG Recommendations 21
2.5 Review Summary 22

CHAPTER 3 SYSTEM METHODOLOGY AND APPROACH 23
3.1 System Design Diagrams 23
3.1.1 System Architectural Diagram 23
3.1.2 Use Case Diagram 24
3.1.3 Activity Diagram 25

CHAPTER 4 SYSTEM DESIGN 28
4.1 System Design Block Diagram 28
4.1.1 Front-End Block 29
4.1.2 Back-End Block 29
4.1.3 Code Repository 29
4.2 System Component Specifications 30
4.2.1 Front-End Component 30
4.2.2 Back-End Component 33
4.2.3 Repository Component 38

CHAPTER 5 SYSTEM IMPLEMENTATION 40
5.1 Development Workflow and Methodologies 40
5.2 System Setup and Configuration 41
5.3 Implementation Issues and Challenges 43
5.4 System Testing 44

5.4.1 Alpha Testing 44
5.4.2 Beta Testing 44
5.4.3 Developmental Logging 45

CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION 46
6.1 Functional and Objectives Evaluation 46
6.2 Non-Functional Evaluation 47
6.3 GUI and UX Evaluation 48

CHAPTER 7 CONCLUSION AND FUTURE WORKS 49
7.1 Overall Conclusion 49
7.2 Future Works 49

REFERENCES 50
WEEKLY LOG 51
POSTER 64
PLAGIARISM CHECK RESULT 65
FYP2 CHECKLIST 689

LIST OF FIGURES

Figure Number	Title	Page
Figure 1.1.1	Overwatch Avoid Teammate Function	2
Figure 2.0.1	Team Fortress 2 Poster	7
Figure 2.0.2	TF2 Insomnia61 Finals, 2017	7
Figure 2.1.1	TF2 System UI via ToonsHud	8
Figure 2.2.1	Match.TF System UI	11
Figure 2.3.1	FACEIT System UI	14
Figure 2.3.2	FACEIT TF2 Lobby Hub	14
Figure 2.4.1	Example of Community PUG using a Discord Server	18
Figure 3.1.1	System Architectural Diagram	23
Figure 3.1.2	Use Case Diagram	24
Figure 3.1.3	Register Activity Diagram	25
Figure 3.1.4	Login Activity Diagram	25
Figure 3.1.5	Players Activity Diagram	26
Figure 3.1.6	Team Activity Diagram	26
Figure 3.1.7	Lobby Activity Diagram	27
Figure 3.1.8	Tournament Activity Diagram	27
Figure 4.1.1	System Design Block Diagram	28
Figure 4.2.1	Exposed Public Folder	30
Figure 4.2.2	Nginx Config	31
Figure 4.2.3	Deploy YAML File	38
Figure 4.2.4	GitHub Action Secrets	39
Figure 5.1.1	Rapid Development Life Cycle	40
Figure 5.2.1	Environment Variables	42
Figure 5.4.1	PM2 Monitoring Function	45
Figure 6.1.1	Osmium Player Profile	46
Figure 6.2.1	Digital Ocean Droplet Metrics	47

Figure 6.3.1	Osmium Home Page	48
Figure 6.3.2	Osmium Lobby Profile	48

LIST OF TABLES

Table Number	Title	Page
Table 2.5.1	System review summary and project challenges	22
Table 4.2.1	Front-End Specifications	32
Table 4.2.2	Back-End Specifications	37
Table 4.2.3	Repository Specifications	39

LIST OF ABBREVIATIONS

PUG	Pick Up Games
LAN	Local Area Network
PVP	Player versus Player
NEET	Not in education, employment, or training
CS:GO	Counter Strike: Global Offensive
TF2	Team Fortress 2
VAC	Valve Anti Cheat
UI	User Interface
UX	User Experience
AC	Anti-Cheat
OS	Operating System
EULA	End-User Licensing Agreement
RAD	Rapid Application Development
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
SASS	Syntactically Awesome Style Sheet
JS	JavaScript
PHP	Hypertext Pre-processor
CI/CD	Continuous Integration/Continuous Deployment
API	Application Programming Interface
QoL	Quality of Life
MS	Microsoft
GB	Giga Byte
MVC	Model View Controller
SSH	Secure Shell
FTP	File Transfer Protocol
DM	Direct Message

Chapter 1

Introduction

In this chapter, the motivation behind the proposal of the project title, the problem statement and intended solution, as well as with the paired project scope and objectives will be discussed alongside the potential impact and significance the project is going to bring to the gaming community and team building options. As a background info, the emergence of Esports title has increased over the years, and yet there still exist difficulties in integrating the community or players into the competitive scene, which leads to the problem identified below.

1.1 Problem Statement and Motivations

The problems that are currently experienced by the previously aforementioned players are mostly spawned from the fundamentally flawed system of some of the Esports titles which require the help of 3rd party software or community to intercept. Other than technical problems, the human cause is also one of the reasons contributing to the problem faced in this field. The problem domain identified are as follows:

I. **Matchmaking and Team-up**

In the early days where matchmaking service did not exist yet, in which the system used to match another player into a game lobby. People often compete with each other within a LAN environment where they would bring their home computer together and link up with each other via a common ethernet cable. Over the years as internet accessibility has become more accessible to the general public, the issue with matchmaking is seemingly solved with a built-in system in the majority of the competitive video game titles. However, while matchmaking service is present in most video game titles in the form of Elo system or Glicko2 system [1], it has yet to solve the problem of creating a custom lobby match-up or simple way to find a team to play within a similar skill group.

II. **Flexibility of System**

Although some titles have gone and provided some form of private lobby, these however are still limited as a private lobby requires players to find each other manually either through a common community or via their friend-list. On the other hand, tournament options within a video game are still absent in seemingly all of the game titles, excluding the competitive game mode.

III. **Human Interaction**

Other than lobbies and tournaments, another problem that persists would be the problem of toxicity of a player. While some games handle toxicity with a plethora of options, such as "Overwatch" where it allows the player to avoid a certain individual that is providing a negative experience to their game [2]. Most of the time, these "toxic" players are often ignored by the official system, and it would damage the reputation of a game title gradually.

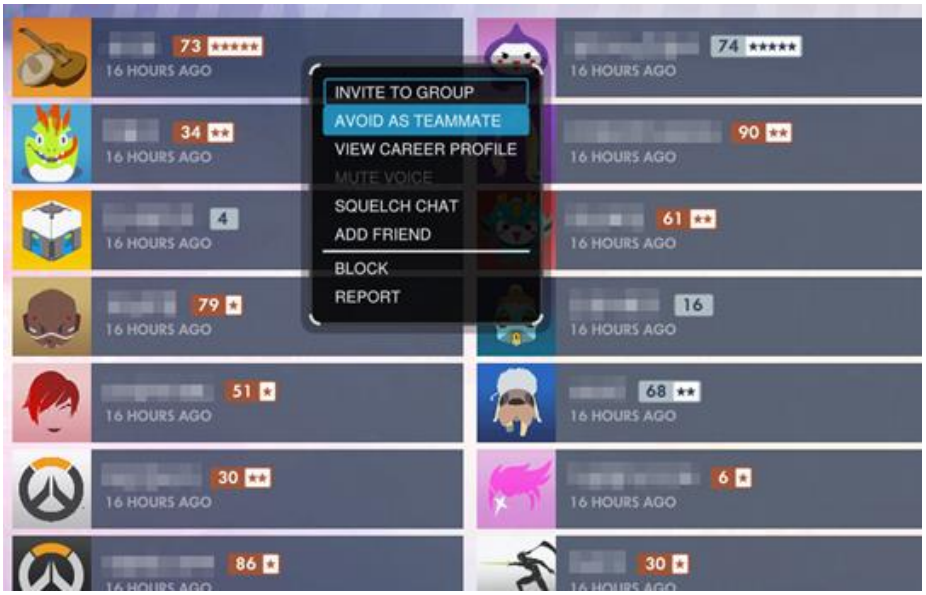


Figure 1.1.1 - Overwatch avoid teammate function

1.2 Objectives

The main objective of the project is to provide an easier way for players in all skill ranges to participate in a competitive scene with the primary intent of learning and having a wonderful experience. This however can be further divided into 3 sub-objectives as follows:

I. To develop a web app that enables players to find a team based on their respective Esports title, regional preference and also the appropriate skill level. This is to ease the process of players into fitting themselves into the competitive scene and provide a user-friendly learning experience for each other.

II. To develop a web app that allows the creation of custom lobbies or tournament events by players with versatile options such as the size of the lobby, Esports titles' game settings, tournament mode, etc. This allows individuals or teams to participate in an all-out competitive setting to compete with each other to test their limits and also enjoy a match based on their setting requirements.

III. To develop a web app that accounts for player base toxicity with a proposed player rating system based on participation and compliments from other players. This would create a user-friendly environment to learn, train and play, which could help develop players with a crucial sportsmanship attitude in the future.

1.3 Project Scope

The proposed project is to develop a web application that acts as a general team finder for a selected variety of Esports compatible titles. Besides that, the web app would also let users create a tournament event or custom lobby intended for pick-up games, PUG or scrimmage between teams, Scrim in short. The dynamic and versatile customization options of lobbies and tournaments provided could ease newcomers and veteran players to find matches that suit their skill level, as well as creating a perfect environment for learning for each other. With the subject of player toxicity, we intend to solve the problem by introducing a player rating profile that rates a player's status quo via their participation rate in matches and compliments gained from other players.

In terms, there would be 4 main modules that will be the main pillar for the proposed web application development project that will complement the established objectives:

1.3.1 Team Finder Module

In this module, users/players can search through a variety of teams in a table-sort of an interface where players are able to view the team primary details such as team names and team capacity, team region, and of course, the team's selection of Esports title that they will be participating in with its recommended skill-level assigned. The players are also able to use filter and search terms in order to find their designated Esports team, invitations from teams are also available from team leaders to find players they seek on their team.

1.3.2 Custom Lobby Module

While each Esports title would provide different lobby settings, we will try to implement templates based on the provided Esports as a starting point for a custom lobby. The custom lobby would also be advertised publicly on a lobby menu, otherwise, the lobby is private, and players would have to invite others via common keyword/room code to enter the lobby. While the server providing module is still in consideration due to limitations, one of the lobby players would be assigned as room master to create a game server within the base game for the lobby match to start.

1.3.3 Custom Tournament Module

In this module, selected individuals by the community leaders or moderators are able to create a custom tournament that is recognized by the player rating module with set rules based around a designated Esports title. The tournament master is assigned and is given options as to how the tournament placements are going to proceed, with examples such as Single Eliminations, Round-Robin, etc. Individuals or teams are able to sign up given the appropriate requirements and would be awarded player or team ratings and other prizes given by the tournament master.

1.3.4 Player Rating Module

Each player/user is given their own personal page after registration on the web application, where they can view their participation history, current player ratings, player compliments, current team and other personal details. The main function of this module is to calculate the rating based on the participation rate and player compliments in a human-readable index such as an arbitrary number for their ratings. Otherwise, the alternative rating can be solely based on the other player's compliment with valid reasoning. Of course, downvoting a player is also an option but is closely monitored by community moderators in the case of abusing the aforementioned function. If a player rating falls to some threshold (in some extreme cases), the account would be barred from recognized tournaments and other lobbies with set rating requirements and will need to file a report to the community assembly (made up of multiple volunteers recognized by the community) in order to reverse their barred status.

1.4 Contributions

While 3rd party software that enhances the user experience is not a new topic in this field, but the general approach with the proposed objectives and scope is something of a refreshing idea that aims to aid both the competitive and casual player base at the same time while acknowledging the problem faced by both party with different goals and intentions, as previously, majority of the application focuses mainly on the competitive side of the player base without some concern of the toxicity problem.

1.5 Report Organization

The report is organized into 7 major chapters: Chapter 1 – Introduction, Chapter 2 – Literature Review, Chapter 3 – System methodologies and approach, Chapter 4 – System Design, Chapter 5 – System Implementation, Chapter 6 – System Evaluation and Discussion and Chapter 7 – Conclusion and Future Works. The first chapter is the introduction to this project which includes problem statement and motivations, defining project scope and objectives as well as contributions. Second chapter is a literature review conducted on various similar systems that involves team building, match making and else. The third chapter discusses on the methodologies of the system and approach, conducted via using various diagram such as system architectural diagram, use case diagram and activity diagram. Chapter 4 discusses about the overall system design and the major component of the system alongside their specifications. The fifth chapter explains the implementation of the system and also the developmental workflow, system setup, issues, testing methods. Chapter 6 mainly focuses on the evaluation of the system from both developer and tester's feedback. Finally, in chapter 7, where conclusion would wrap up the whole project.

Chapter 2

Literature Review

In this chapter, multiple established systems are reviewed and analysed for their respective strengths in tackling the problem and the weaknesses the system unfortunately suffers. Thoughts on how the weaknesses can be overcome are discussed thoroughly in detail.

For a unified view, the Esports title, “Team Fortress 2” or TF2 for short, are chosen as the base for all of the reviews in the hope to negate differences and reduce the complexity that arises due to completely different Esports title; though, in general, all Esports title would result in the same conclusion.

Figure 2.0.1 - Team Fortress 2 Poster

Figure 2.0.2 - TF2 Insomnia61 Finals, 2017

2.1 Base System

Published in 2007, TF2 was innovative in both its design and gameplay, while the official competitive scene would not come until late 2016, many of the 3rd party community had already established a general setting for the competitive scene [3].

2.1.1 Base System Analysis

While the proposed project sought to be a 3rd party software, it would not hurt to analyse and study the base system and its intention of the original developer in the making of the system.

During the launch of the competitive model in 2016, doubted the “Meet your Match” update, the system received changes to its graphical user interface, GUI, which includes simple details and history of the competitive match being played.

In addition to the competitive mode being launched, the update also reinforced system matchmaking in casual mode, whereas previously you just browse through a list of the server to join. Though community servers are still unaffected by the matchmaking and still be played using the local server browser.

Figure 2.1.1 - TF2 System UI via ToonsHud

2.1.2 Base System Strengths

Based on the Elo-based matchmaking system, the overall matchmaking experience is tied to the actual skill level of the player during the match-up. As such the queue time for competitive mode is slightly longer than of the casual mode due to the importance of ensuring a fair and square playing field between 2 similarly skilled players.

The system also utilizes a non-kernel anti-cheat system known as Valve Anti Cheat, VAC for short to stop cheaters and bots from interfering with the match. The VAC process is an automated process, as such, the cheater would have a hard time bypassing the barrier since VAC runs on a machine learning algorithm that analyses potential cheats or any other malicious app being hooked to the main system [4].

2.1.3 Base System Weaknesses

While the system reinforces the matchmaking with Elo-based algorithm, the system restriction however is disapproved by the majority of the competitive player base as it forces mandatory and unnecessary system changes to the game to ensure fairness in a match-up.

The custom server lobby and tournament lobby are practically non-existent to the system, and it would require the player to resort to 3rd party community or software to create a custom lobby they wish to play. While community servers still exist in the system, the flexibility or dynamic lobby creation are unfortunately still lacking from the system.

On the topic of player toxicity, as TF2 runs on the game launcher, Steam, which is also made by the same company, its player profile is reflected in the external launcher. Valve, the company that made both the products, had tried to solve the problem with a reporting system and suppression of vague language during a match, it has yet to prove any success in tackling the subject.

While VAC was able to block the majority of the cheaters, malicious individuals were still able to crack through the system with ease due to the fact that the anti-cheat was made in 2002 and is considered outdated by today's standard [5].

2.1.4 Base System Recommendations

While the system has done thorough work in making sure the gameplay environment is fair, the mandatory system changes are just unnecessary, to begin with, especially the mangling of the player's graphical settings. And thus, a recommendation that the system should just drop the system changes on the software level all while still preserving the Elo-based matchmaking concept.

Creating a playable server on an outdated engine requires tremendous work and resources to maintain, as such the system should adapt to new engine changes or rework their networking protocols to support implementation on a much easier custom lobby creator, and even tournament creator if necessary.

The problem of cheater and anti-cheater are way beyond the league of the current proposed project as it is a matter of subject on machine learning and algorithms. Though said, a suggested solution to the reporting system by introducing player ratings and a new report system to counteract cheater and toxic player alike. As easier as said on paper, the work needed to be done involves the changes to the entire player profile database, which is very unlikely and unprofitable for the company.

2.2 Match.tf System

Match.TF is a 3rd party and community made web application that provides players with the ability to host and run their tournament under the supervision of the moderators and finding their dream team to play on and experience the competitive scene [6].

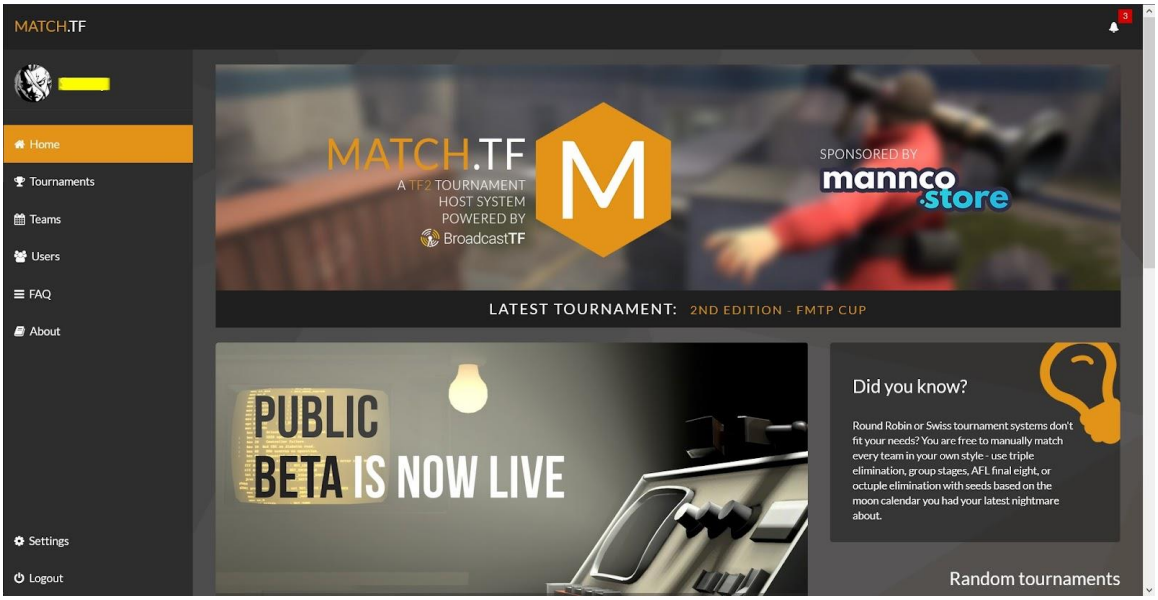


Figure 2.2.1 - Match.TF System UI

2.2.1 Match.tf Analysis

The sole objective of the system is to provide the ability for players to host their tournaments in different settings, all while accompanying different competitive leagues. Many of the community competitive league utilize this website to host their seasonal ranking ladder as it provides everything necessary for a tournament or event to be hosted.

On the other hand, the establishment of a team finder makes the system much more welcoming for beginners and veterans alike in the competitive scene, allowing the player to browse through potential team to participate and join into the action immediately with the transfer option during a tournament.

2.2.2 Match.tf Strengths

The primary strength of the system would be its tournament hosting service. The hosting service provides everything a tournament would need, including a customizable tournament page, a powerful match system such as pick-and-ban, etc.

Another strength that the system provides is the team finder module, it enables the player to create a team and invite other players to participate on behalf of their team in any of the hosted tournaments.

The system also serves as an extension to the base system discussed in the previous review, where it covers the limitations of the base system which maintaining originality from it, making it much more user-friendly to those familiar with the base system and overall league rules.

2.2.3 Match.tf Weaknesses

While the system is able to cover the creation of tournament that is absent from the base system, it is still unable to cover the problem of creating a flexible lobby that suits the player requirement. The tournament creation process requires the attention of the site moderator to be approved before it can be run, it is great to some extent, but the approval time might hinder the time schedule of some community tournament.

The team finder module also lacks the necessary information required for players to consider such as the team skill-level requirement, tournament, and match history, etc. The way they provide the team information only extends to a single blog-like page, which is not very intuitive.

As mentioned before, the system serves as an extension to the base system, and such, the problem of cheaters and player toxicity still persists via the base system approaches on how it handles the problem.

2.2.4 Match.tf Recommendations

The solution to the problem of creating a custom lobby might just be the simple act of launching a new subcategory just like what the system did with the tournament creator. On the other hand, the approval of a tournament might only be necessary on large scale events while minor tournament such as charity tournament can be created without the approval of the site moderator, which speeds up the overall deployment and initiation of the event.

The team finder module would require a few new additions in order to keep up with the user requirement, for such, an update can be done to the module in order to show team region, team skill-level (based on leagues) and also some other contact information in their team page. History of the team participation is also an important feature to be considered in this context as it allows the player to generalize if the team commitment to a tournament is absolute or not.

On the subject of tackling the player interaction problem, a suggestion on decoupling the player profile from the base system so that the system itself would have a player rating system separate from the base system, with this in mind, creating a user-friendly environment might become much simpler and easier. As such, the site moderator should also actively monitor the report system to reinforce the environment that the player base seeks.

2.3 FACEIT System

FACEIT is a 3rd party community web application, which also comes with its own mobile and desktop application, where the system often collaborates with another base system to enhance the overall user experience. The FACEIT system is made to be an incredibly competitive platform which rewards a player based on their participation and ranking up [7].

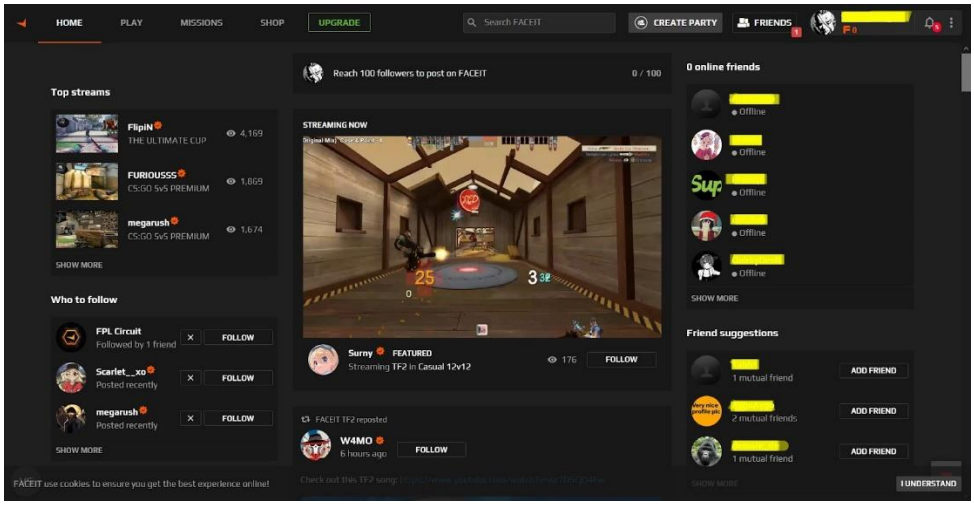


Figure 2.3.1 - FACEIT System UI

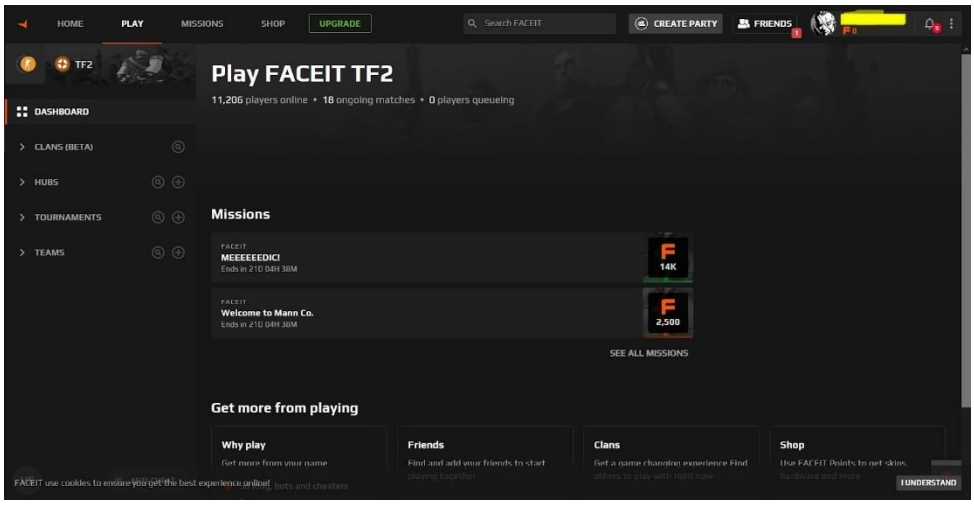


Figure 2.3.2 - FACEIT TF2 Lobby Hub

2.3.1 FACEIT Analysis

Made by an actual backed company instead of a community funded project, this system provides all the requirements and functionality that the base system simply lacks. Their first approach was on the Esports title, CS:GO, which is a major success in terms of popularity and flexibility for competitive players. TF2 support, on the other hand, was implemented instead in March 2021 due to the success of the previous support of the Esports title.

The system also provides its own unique matchmaking system that is separate from the base system where FACEIT provides its own ranking system where the player can climb from its initial placement match. Though the common 6 versus 6 styles of game mode, which is the main staple of the TF2 competitive still absent, instead 12 versus 12 is the main game mode the system currently provides which prompt to promote as a new and overhauled system compared to the base system.

2.3.2 FACEIT Strengths

The system provides the creation of tournament via official collaborations and representative from the competitive league, the system also provides a weekly tournament ladder for players to compete with that reset on a weekly basis to encourage the player base to engage in the scene while awarding them with the appropriate awards such as point to be spent in exchange for base system items.

While team finder in this system exists as sort of a hub where players can be invited together and play accordingly with each other. The system also introduces the player rating system in the form of Behavioural Rating after the end of each match which serves to show how a player behaves and plays during a match.

The system introduces a totally new anti-cheat system, doubted, FACEIT AC, which is a superior anti-cheater system compared to the base system due to its kernel-level system scanning. With this in mind, the system's AC is promoted to the extent that it encourages players from the base system to play on this platform instead as it provides a much better user-friendly experience.

2.3.3 FACEIT Weaknesses

The creation of the tournament, though robust, by the entire process, are monitored by the FACEIT administrator and must be approved before initiating, which consumes time and resources for the organizer. Furthermore, the tournament is only beheld by official and recognized collaborators with the system itself, meaning that normal players cannot simply create their own tournament without a reliable credibility backing them up.

Though team exist on this platform, unfortunately, does not provide the necessary details that a team finder module should have. Making it only useful for known players to invite each other and play for a game. Custom lobby making too are non-existent in this system.

While the system itself promotes a fair competitive scene, however, the system focuses too much on the competitive aspect of the game that any casual player wishing to play are forced to be disappointed by the lack of casual play that the base system provides.

While the FACEIT AC does solve the problem of cheaters, however, the kernel-level intrusion on the OS raises some privacy and security concerns for many players.

2.3.4 FACEIT Recommendations

Security on AC aside, the overall system does seem promising and is able to cover the majority of the functionalities that the base system lacks. a recommendation for the system to implement a much more beginner-friendly environment such as casual queue matchmaking so that they can join the competitive scene much easier.

The creation of custom tournament and lobby should be made much easier for normal users so that they can enjoy the game in their own way, a suggestion for the system to separate official tournaments from the community one so that it is more flexible and could satisfy a part of the community that wishes to play a match with a set rule in mind.

2.4 Community PUG System

Over the years, many people resort to making their own system with just a communication medium as their staging ground such as Discord or TeamSpeak, where people would organize some sort of event and play together; originally from real sports, the term pick-up game has integrated into the Esports scene over time [8].

The general idea of the community system is that people would use much simpler and primitive methods to solve the problem that the base system have by using a manual matchmaking system, where players are hand-picked by a leader, doubted the “Captain” or by an automated sorting system to play a match where skill-level does not matter, a pick-up game, or PUG for short.

Other than PUG, the system can also be utilized for small tournament events or organized team competition scrimmage, Scrim for short, which is made much easier with how this system is able to adapt to the requirement of the players.

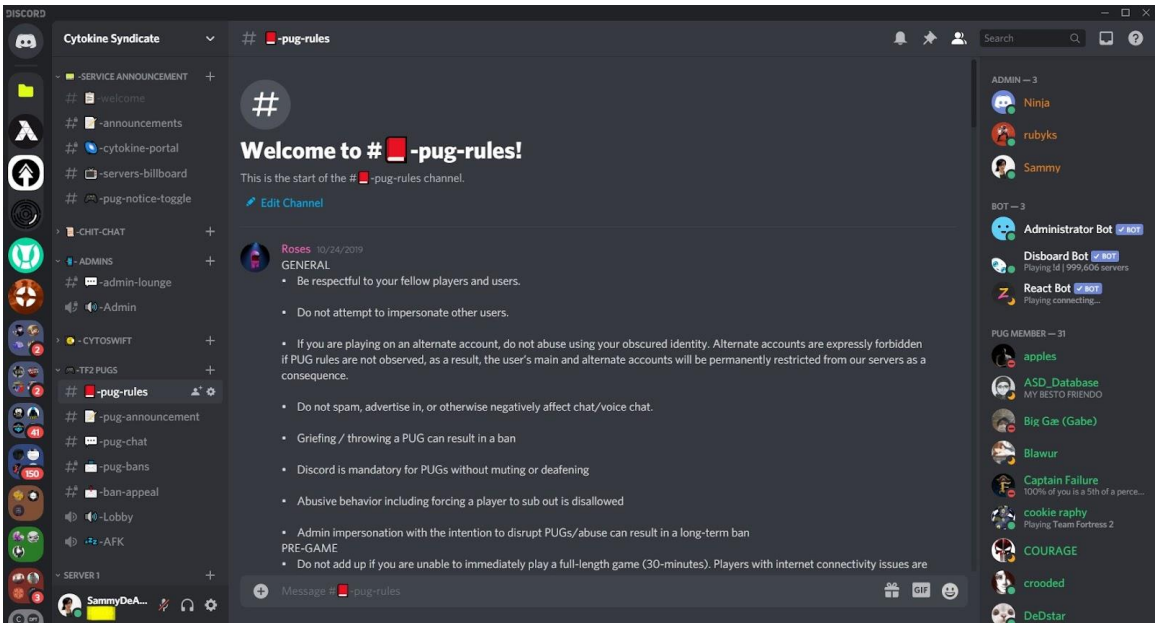


Figure 2.4.1 - Example of Community PUG using a Discord Server

2.4.1 Community PUG Analysis

As mentioned before, the community PUG system is organized by actual players or volunteered moderators themselves. Many players from the different community are able to organize a PUG with ease using the tools already provided, in this case, the communication stage like the Discord, where users can simply create a new server or new room within the server to accommodate their needs in the competitive scene.

Every community may come with a diverse set of rules depending on the player elected moderator itself, as such, there are a diverse group of players that have a totally different playstyle that suits them while playing in a match.

2.4.2 Community PUG Strengths

As the name implies, this system is entirely governed by the players and thus organizing tournaments or lobbies are very efficient in terms of time and participation rate. With popular communication group such as Discord, creating a custom tournament or lobby are mere clicks away from their keyboard, hence the flexibility on this system is absolute.

Players throughout the same community server are able to communicate with each other to when to organize a PUG, and also to team up together for a game, including try-out, where potential players are testing in a scrimmage match for their capabilities in a tournament or ranked season for a competitive league.

2.4.3 Community PUG Weaknesses

Although the creation of tournament and lobby is quite easy, it is to the point of anarchy. The rules surrounding the community PUG server may change frequently without notices and can be unreasonable at times. As such, synchronization between community servers is always impossible and unpredictable. The staging of a PUG would also be very time-consuming because it is a manual matchmaking, which means, a PUG can only start depending on how many players that will be participating in the game for that particular community server.

The usage of external software for a staging ground for the system are dangerous and volatile as if the software ceases to function, then the complete system would not function at all. As such, community PUG is very reliant on the medium and may become troublesome in the event that the medium was to cease to operate.

Another major limitation of this system is that player toxicity has a huge presence in this community PUG servers. Although player-elected moderators may govern their players properly on some servers, many of them still left unchecked as some server's PUG are self-serviced. Out of all the systems mentioned, player toxicity here posed the largest threat to the overall competitive scene due to the overall anarchic nature of how the system is operated.

Aside from player interactions, this system is not very user-friendly because as mentioned, every community PUG server would have a different ruleset and playstyle, and thus players would have a challenging time at finding the right community PUG to play on.

2.4.4 Community PUG Recommendations

The system generally trades quality over quantity and time, the whole point of the system is their unique flexibility and dynamic changes coupled with customizable matchmaking makes this system almost unique as its own. That said, a suggestion would be that the community PUG should decouple itself from the need for a medium for staging ground and made itself self-sustainable such as making a web application instead of hosting the PUG.

The problem of player toxicity with this system is ridiculously hard to tackle due to the overall nature of the system, though a suggestion that the system could follow a common guideline provided by the EULA of the communication medium so that it can maintain a user-friendly environment for all players.

In terms, the community PUG system's problem would be the hardest to resolve due to how it runs as an anarchic system, in the end, it all depends on the player on how they would make it out of the system.

2.5 Review Summary

In terms, the overall review can be summarized as follow, including the proposed title with challenges that need to be overcome during the development phase:

Table 2.5.1 - System review summary and project challenges

Comparing Systems	Strengths	Weaknesses/Challenges
Base System	• Elo-based matchmaking for fair play • Non-intrusive anti-cheat protection	• Unnecessary system changes during queue • Non-existent custom lobby or tournament • Out-dated anti-cheat • Player profile is tied to the game launcher
Match.TF System	• Tournament hosting service • Team finder module • Familiarity to the base system	• Approval of tournament takes time • Team description lacks details • Player profile is tied to the game launcher
FACEIT System	• Custom hub and tournament creation • New behavioural player rating system • Separate matchmaking system • Powerful anti-cheat	• Only collaborators can host tournaments • Non-existent custom lobby or team • Non-newcomer friendly • Intrusive kernel-level anti-cheat
Community PUG System	• Flexible lobby and tournament creation • Easy and quick customization • Fast matchmaking	• Player dependent matchmaking • Is tied to a communication staging ground • Anarchy system, no authority governing
*Proposed Title	• Dynamic and flexible lobby and tournament creation • Detailed team finder module • Unique player rating system	• Hosting service still dependant on the player side • Lack of automated matchmaking for players ease of function • Communication methods are reliant on the base system

* Refer to Proposed Project Title on the front page

Chapter 3

System Methodologies and Approach

In this chapter, the system methodologies, and approaches to the development of the web application are presented and justified to suit the aforementioned objectives established in the previous chapter.

3.1 System Design Diagrams

3.1.1 System Architectural Diagram

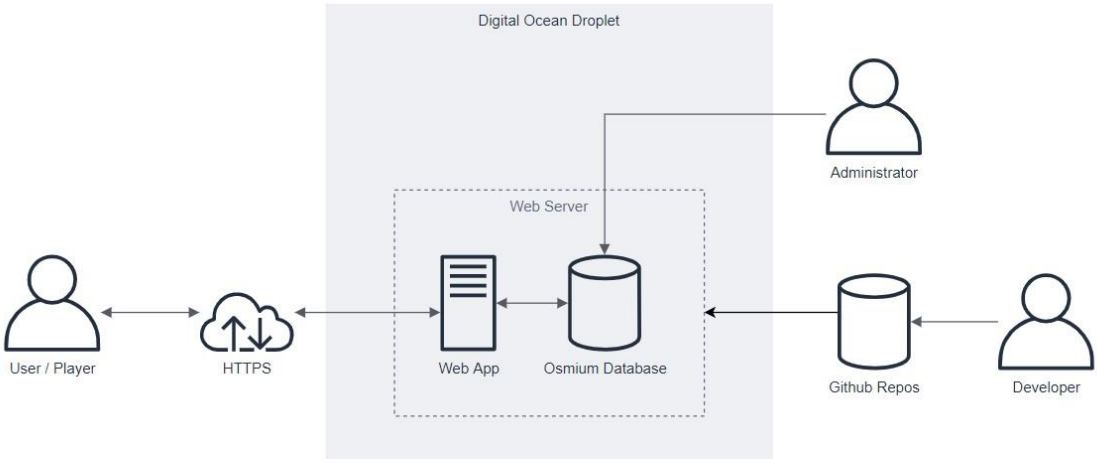


Figure 3.1.1 - System Architectural Diagram

The architecture of the overall web application is pretty simple and straight-forward. The web application is hosted in a web server which is provided in a virtual cloud environment called droplet in the Digital Ocean Droplet Service. The 3 primary actors in the system would be the user/player, who interacts mainly with the web app's service; administrator, which are able to administrate over the web application database via SSH connection; and also, the developer where development changes in the web application are carried out in a private Git Hub repository, where changes are automatically applied to the web server via CI/CD process.

3.1.2 Use Case Diagram

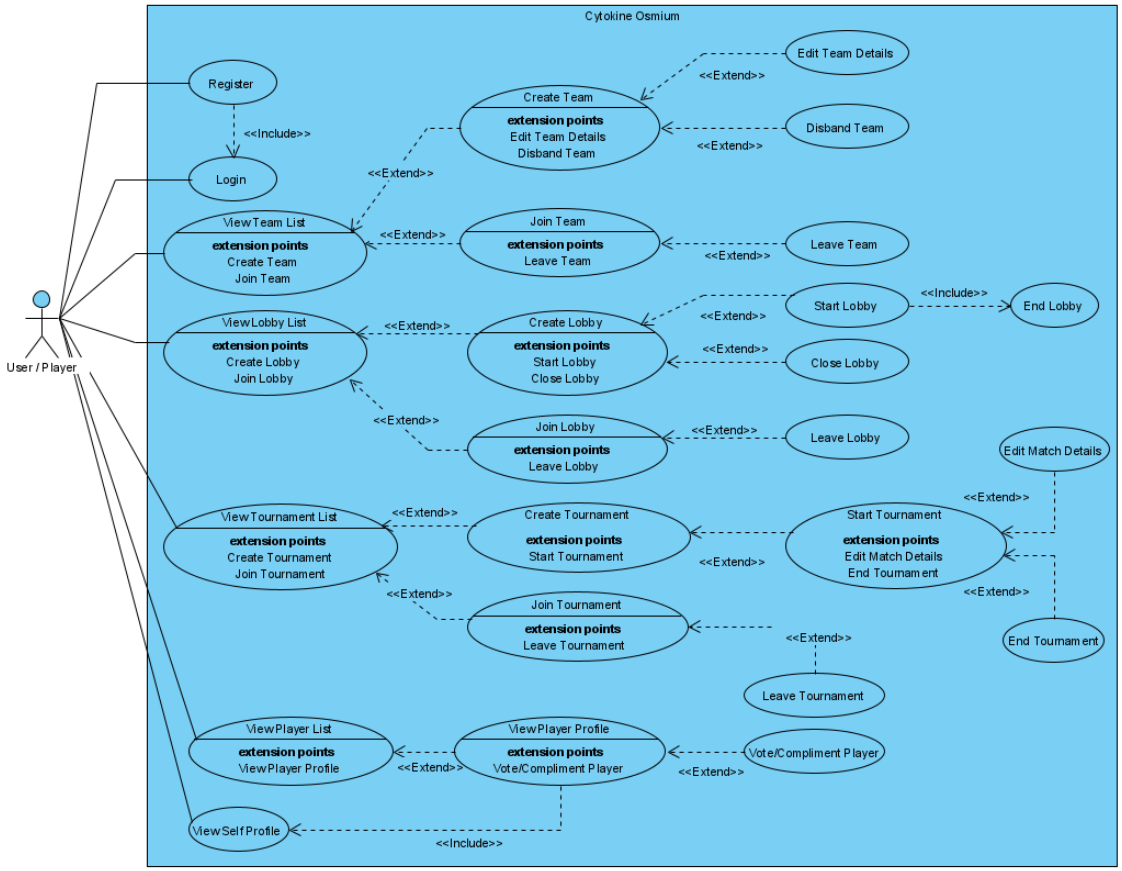


Figure 3.1.2 - Use Case Diagram

And add-on note for the use case here in which, the use-case: Login are essential for unlocking the user/players to other functionality deeper in the extension. The following use-case are presented as such that the user is logged into the system with a verified status. Else, the user would only be able to view teams, lobbies, tournaments, and players; other functionalities such as creation, deletion, joining and your profile would not be accessible.

3.1.3 Activity Diagram

The following activity diagrams would follow the core functionalities of the system, in which will be Register, Login, Players, Teams, Lobbies, Tournaments activities.

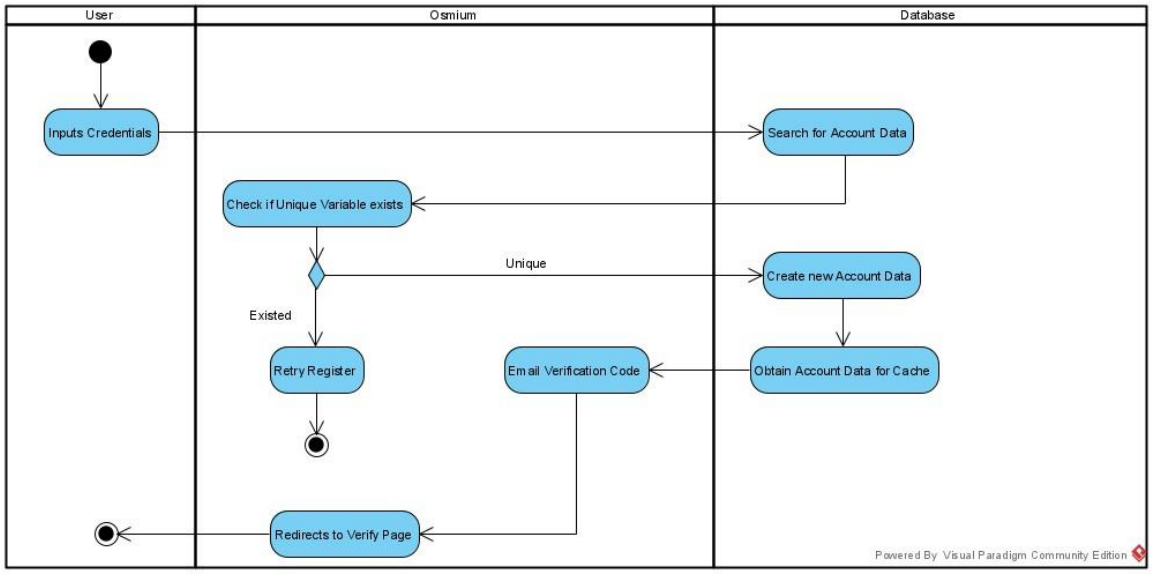


Figure 3.1.3 – Register Activity Diagram

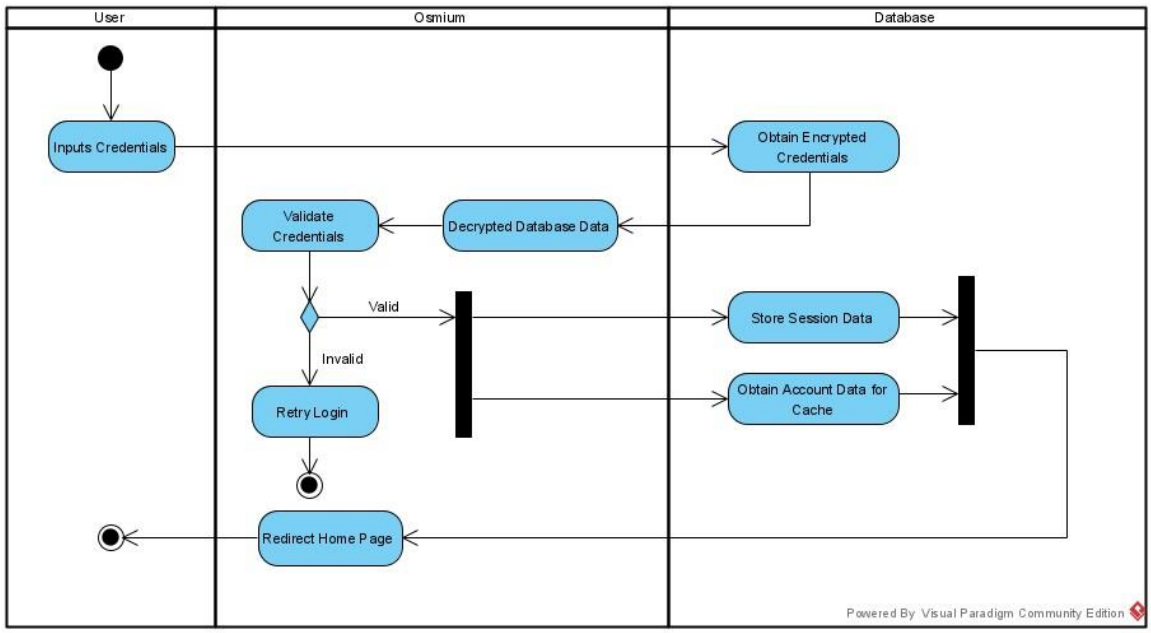


Figure 3.1.4 - Login Activity Diagram

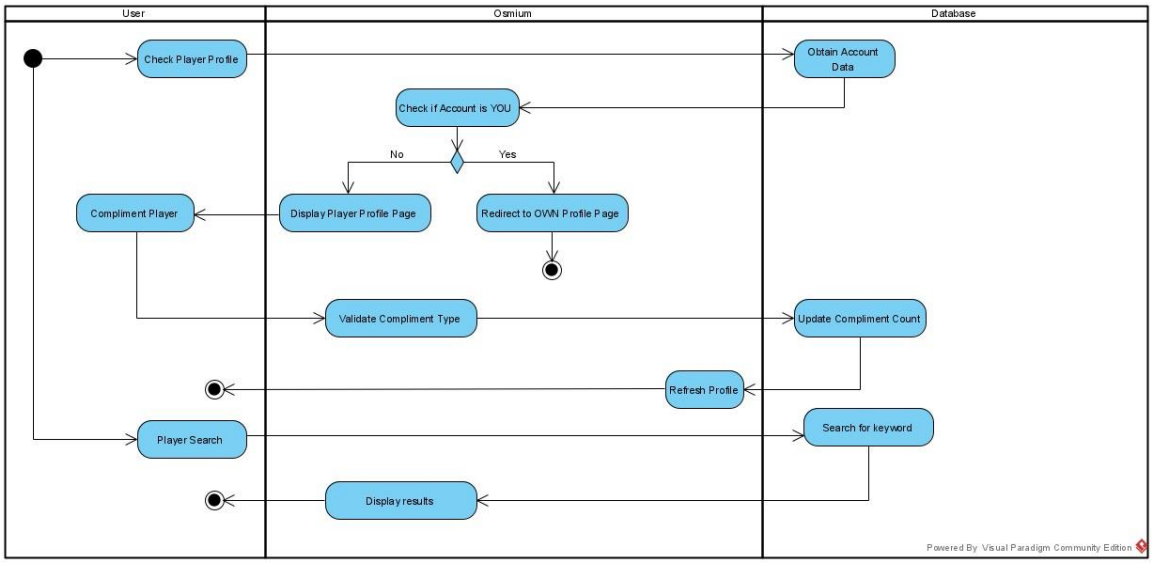


Figure 3.1.5 – Players Activity Diagram

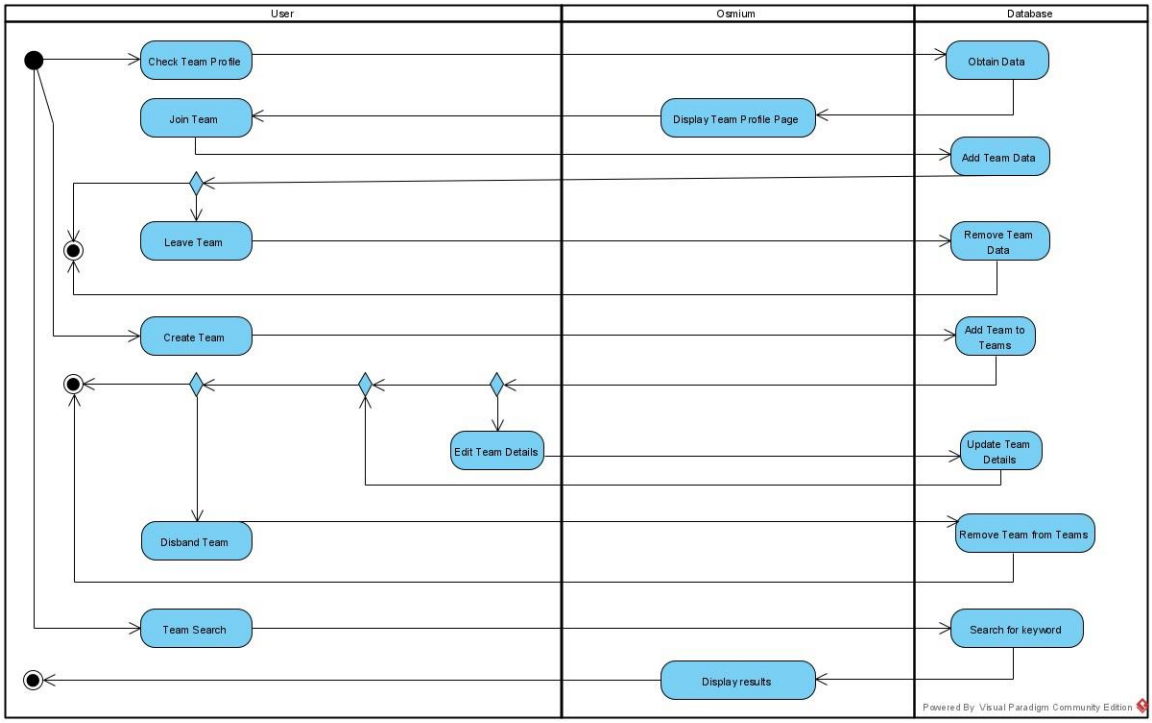


Figure 3.1.6 – Teams Activity Diagram

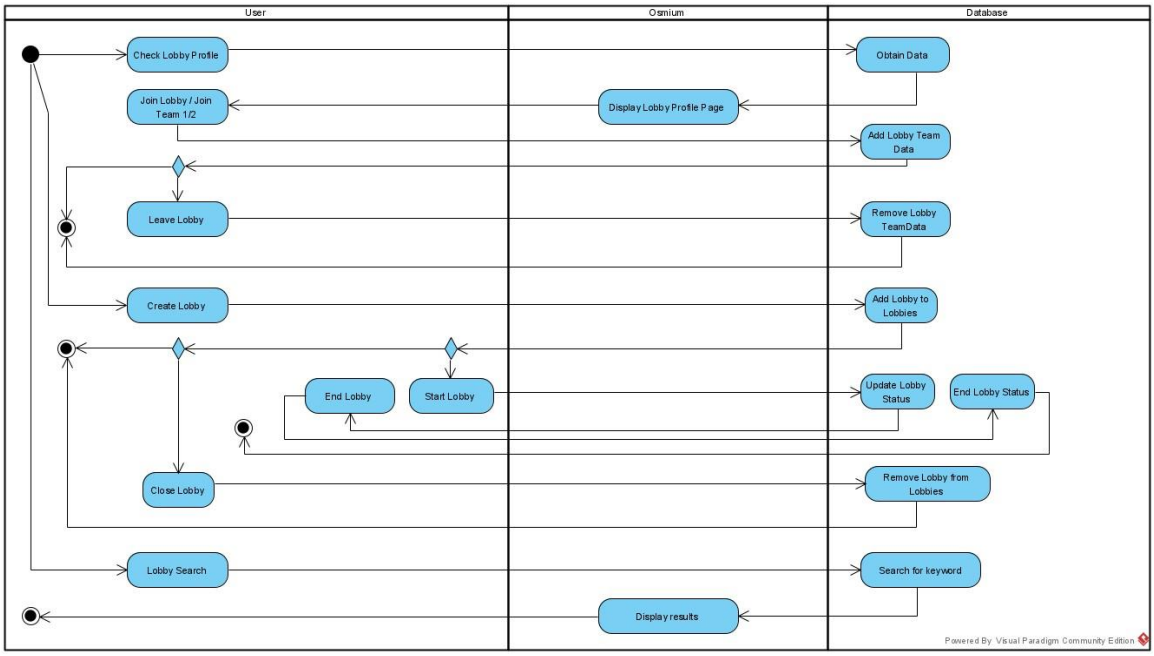


Figure 3.1.7 – Lobbies Activity Diagram

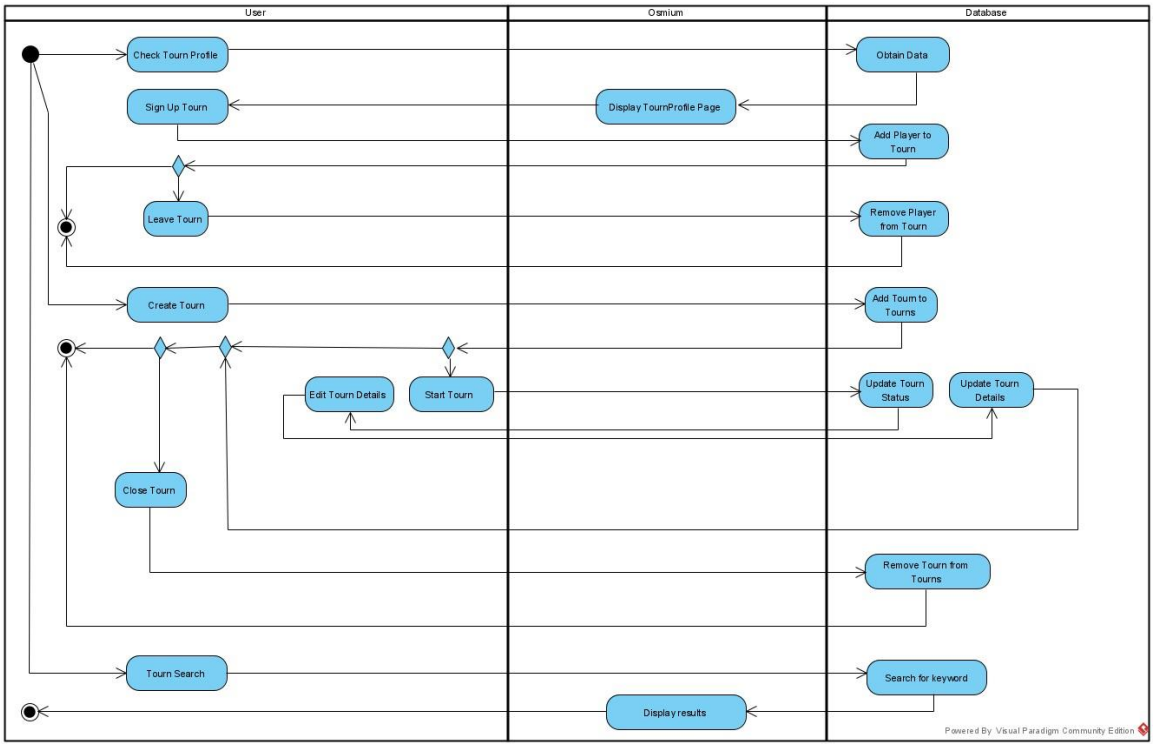


Figure 3.1.8 – Tournaments Activity Diagram

Chapter 4

System Design

In this chapter, the system design, and the development of the project in details will be discussed. The connection between the components and interactions of the modules would also be concluded here. All information in this chapter would be displayed in a way that would allow individuals to reconstruct and setup the system for future uses.

4.1 System Design Block Diagram

The overall web application is built on the platform of remote hosting service with the concept of MVC model, in which the primary web server would be Nginx. The following illustrates the overall web server system flow at the current development stage.

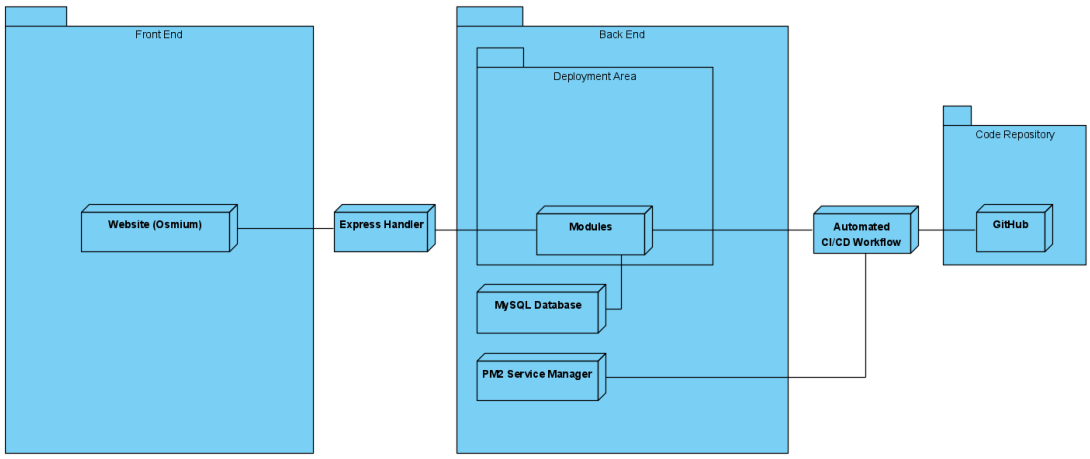


Figure 4.1.1 - System Design Block Diagram

The overall system is categorized into three main packages, mainly the front-end, back-end and the repository of the source code for the web application.

4.1.1 Front-End Block

The front-end of the system mainly consisted of serving the website to the users on the internet. Any interactions here are handled by the Express framework where it will be handling all the server responses such as GET and POST. Information is usually carried through via the usage of sessions instead of cache or cookies for security reasons. The session's information is generally handled by express built-in modules which integrate with the local-hosted database, which will be discussed in the next section.

4.1.2 Back-End Block

The back end of the system is built on the foundations of NodeJS + Express framework. Here, the functional modules communicate with the front-end, such as relaying information or rendering pages, via the Express handler. All the crucial modules are held within a deployment area in the hosting server, which will be discussed in the next section. Some of the application’s modules would also communicate with the local database via secured port and API built into the framework, for instance, the sessions management of the web application is handled as such where the session information is stored in the local database instead of the web browser for seamlessness between channels. Other than that, the server hosting service also comes with PM2, which is a daemon process manager that handles the life cycle of the web application. The service manager would be controlled by the automated workflow in the code repository upon any recent updates or commits to the repository.

4.1.3 Code Repository

The deployment area of the hosting server signifies the working area in which the code base would be deployed from the repository, in this case, GitHub. The GitHub repository for this application is handled automatically via GitHub’s own CI/CD workflow. The workflow will update the composition of files and modules within the deployment area and restarts the PM2 daemon process in the event of the server being overloaded or timed out during deployment phase.

4.2 System Component Specifications

4.2.1 Front-End Component

Majority of the front-end sites are served by the Express middleware with the exceptions of error pages and maintenance pages. Since this is where user interacts, a static folder must be exposes to the public as all other directory that is not exposed by the Express framework are automatically denied access.

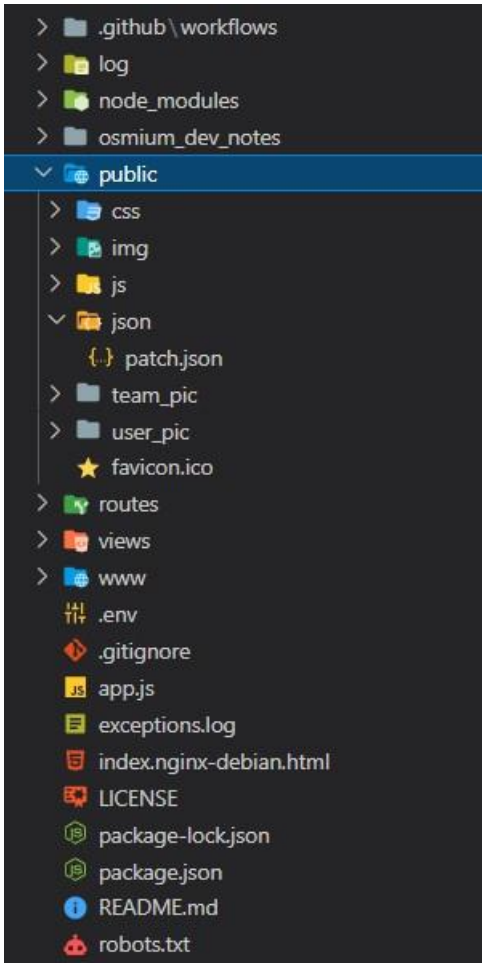


Figure 4.2.1 - Exposed Public Folder

Other than that, the configurations on the web server, in this case, the server running is Nginx on Linux OS. A config file must be set in order for the server to be available to the public website (alongside the domain name that has been set in the Digital Ocean Domain Section) and also enabling SSL HTTPS secure mode.

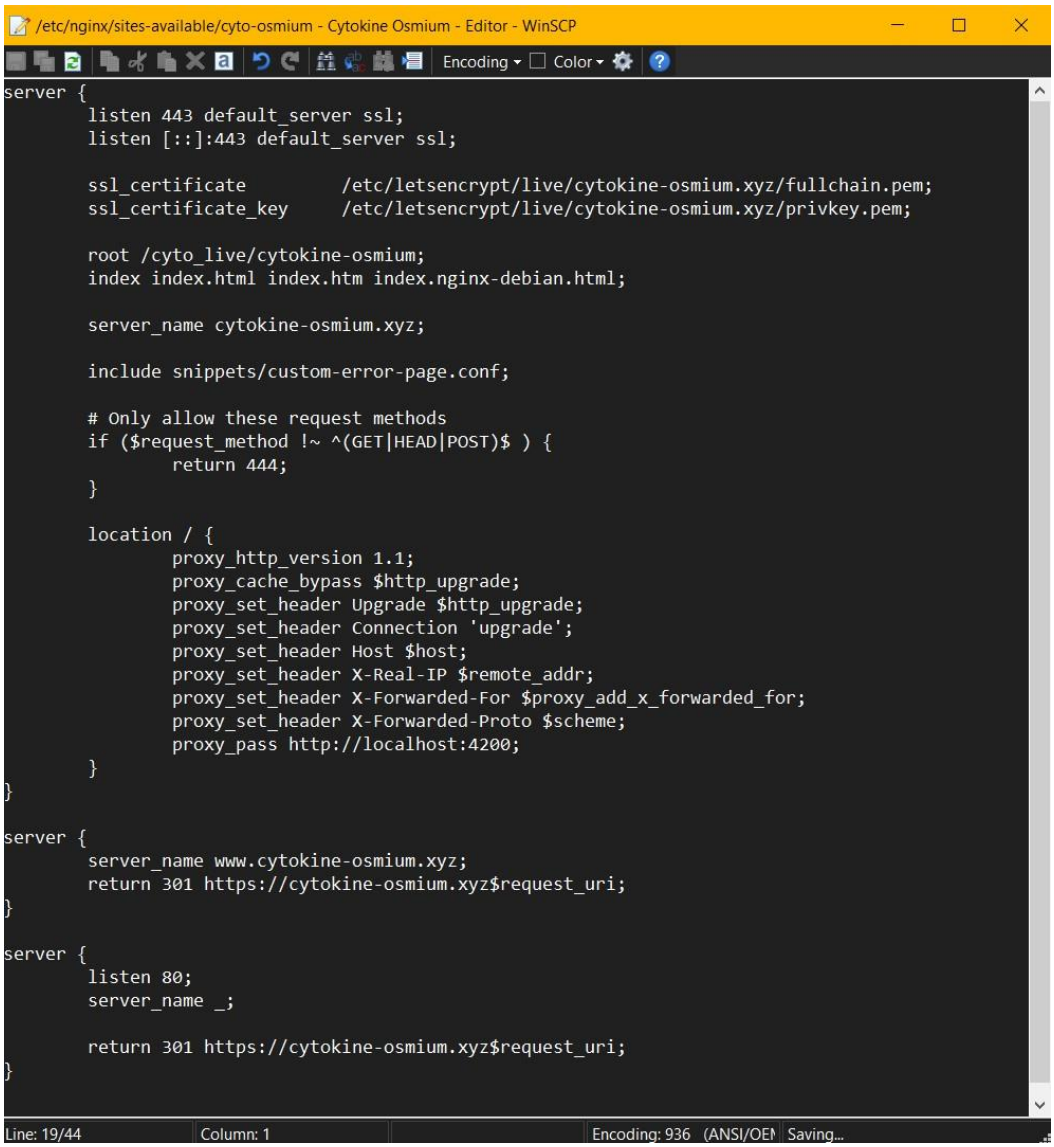


Figure 4.2.2 - Nginx Config

Overall, the front-end components can be summarized as follows:

Table 4.2.1 – Front-End Specifications

Module	Module Version	Functionality
Digital Ocean Cloud Service	-	Popular cloud provider for compute service that comes with various modules.
Digital Ocean Droplet	Basic	Virtual machine that comes with 1vCPU, 1GB RAM and 25GB SSD at a reasonable price point.
Nginx Web Server	1.18.1	Used to host a web server within the droplet, also enables SSL HTTPS secure mode for encryptions as an added security feature for the web application.
Express JS	4.17.1	JavaScript framework for enabling web hosting on specified port, and also includes the middleware which enables communication between front-end and back-end.
Embedded JS (EJS)	3.0.2	Server-side scripting and markup viewing engine used to display dynamic web pages, just like the counterpart: php or jsp.
Bootstrap CDN	4.5.3	Front-End framework which simplifies HTML designs and overall developmental time.

4.2.2 Back-End Component

The back-end side of the web application comprise of the connection between functional modules and database. Because sensitive information is being parsed throughout the application, it is wise to store these variables in an environment variable file that is not exposed to the public, the file is commonly named as ".env" (dotenv).

Aside from the sensitive information, the previously mentioned Digital Ocean Droplet also comes with NodeJS and NPM pre-installed, which allows package and modules installation and updates during maintenance or initialization.

The droplet also comes with pre-installed Daemon scheduler and manager called PM2, this is a crucial module as it is the main module that actually serves our web application. During maintenance and other events, PM2 would manages the application within and would automatically "restart" the app upon crashes or errors.

The Database part of the component is the single holding pillar that allows the web application to function, as such, it would need to be initialized before the web application.

As of current development, there would be 13 tables that needed to be initialized as follows:

```
CREATE TABLE IF NOT EXISTS accounts (
    id              VARCHAR(255)    NOT NULL,
    username        VARCHAR(255)    NOT NULL,
    sitename        VARCHAR(255)    NOT NULL,
    pimage          VARCHAR(255)    NOT NULL,
    email           VARCHAR(255)    NOT NULL,
    password        VARCHAR(255)    NOT NULL,
    register_stamp  DATE            NOT NULL,
    PRIMARY KEY (id),
    CONSTRAINT un_uname UNIQUE (username),
    CONSTRAINT un_email UNIQUE (email)
);
```

```
CREATE TABLE IF NOT EXISTS account_verify (
  id               VARCHAR(255)    NOT NULL,
  verify_status    VARCHAR(255)    NOT NULL,
  verify_code      VARCHAR(255),
  FOREIGN KEY (id)             REFERENCES accounts (id)       ON DELETE
CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS account_status (
    id             VARCHAR(255)    NOT NULL,
    status_text    TEXT            NOT NULL,
    FOREIGN KEY (id)           REFERENCES accounts (id)       ON DELETE
CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS account_special (
    id             VARCHAR(255)    NOT NULL,
    site_privilege VARCHAR(50)     NOT NULL,
    FOREIGN KEY (id)           REFERENCES accounts (id)       ON DELETE
CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS teams (
    team_code      VARCHAR(8)      NOT NULL,
    team_name      VARCHAR(255)    NOT NULL,
    team_desc      TEXT            NOT NULL,
    team_pimage    VARCHAR(255)    NOT NULL,
    team_leader    VARCHAR(255)    NOT NULL,
    team_cTime     DATE            NOT NULL,
    team_invCode   VARCHAR(6)      NOT NULL,
    PRIMARY KEY (team_code),
    FOREIGN KEY (team_leader)  REFERENCES accounts (id)       ON DELETE
CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS team_comp (
    team_code      VARCHAR(8)      NOT NULL,
    team_member    VARCHAR(255)    NOT NULL,
    FOREIGN KEY (team_code)    REFERENCES teams (team_code)   ON DELETE
CASCADE,
    FOREIGN KEY (team_member)  REFERENCES accounts (id)       ON DELETE
CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS account_compliments (
    id              VARCHAR(255)    NOT NULL,
    compliment      INT             NOT NULL,
    disapprove      INT             NOT NULL,
    lobby_complete  INT             NOT NULL,
    tourn_complete  INT             NOT NULL,
    tourn_finalist  INT             NOT NULL,
    special_rating  INT             NOT NULL,
    FOREIGN KEY (id)            REFERENCES accounts (id)        ON DELETE
CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS account_voted (
    id              VARCHAR(255)    NOT NULL,
    voted_id        VARCHAR(255)    NOT NULL,
    vote_desc       VARCHAR(255)    NOT NULL,
    FOREIGN KEY (id)            REFERENCES accounts (id)        ON DELETE
CASCADE,
    FOREIGN KEY (voted_id)      REFERENCES accounts (id)        ON DELETE
CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS lobbies (
    lobby_code      VARCHAR(8)      NOT NULL,
    lobby_name      VARCHAR(255)    NOT NULL,
    lobby_desc      VARCHAR(255)    NOT NULL,
    lobby_leader    VARCHAR(255)    NOT NULL,
    lobby_cTime     DATE            NOT NULL,
    lobby_open      INT             NOT NULL,
    lobby_region    VARCHAR(255)    NOT NULL,
    lobby_pass      VARCHAR(6)      ,
    lobby_status    VARCHAR(255)    NOT NULL,
    lobby_game      VARCHAR(255)    NOT NULL,
    lobby_type      VARCHAR(255)    NOT NULL,
    lobby_skill     VARCHAR(255)    NOT NULL,
    PRIMARY KEY (lobby_code),
    FOREIGN KEY (lobby_leader)   REFERENCES accounts (id)        ON DELETE
CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS lobbies_team (
    lobby_code      VARCHAR(8)      NOT NULL,
    id              VARCHAR(255)    NOT NULL,
    team            INT             NOT NULL,
    FOREIGN KEY (lobby_code)     REFERENCES lobbies (lobby_code) ON DELETE
CASCADE,
    FOREIGN KEY (id)            REFERENCES accounts (id)        ON DELETE
CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS tournaments (
    tournament_code   VARCHAR(8)      NOT NULL,
    tournament_name   VARCHAR(255)    NOT NULL,
    tournament_desc   VARCHAR(255)    NOT NULL,
    tournament_cTime  DATE            NOT NULL,
    tournament_status VARCHAR(255)    NOT NULL,
    tournament_game   VARCHAR(255)    NOT NULL,
    tournament_type   VARCHAR(255)    NOT NULL,
    tournament_skill  VARCHAR(255)    NOT NULL,
    tournament_region VARCHAR(255)    NOT NULL,
    tournament_open   INT             NOT NULL,
    tournament_pass   VARCHAR(6)      ,
    tournament_leader VARCHAR(255)    NOT NULL,
    PRIMARY KEY (tournament_code),
    FOREIGN KEY (tournament_leader) REFERENCES accounts (id)        ON DELETE
CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS tournamens_tree (
    tournament_code   VARCHAR(8)      NOT NULL,
    id                VARCHAR(255)    NOT NULL,
    climb             INT             ,
    bracket           INT             ,
    FOREIGN KEY (tournament_code) REFERENCES tournaments (tournament_code) ON
DELETE CASCADE,
    FOREIGN KEY (id)            REFERENCES accounts (id)        ON DELETE
CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS tournament_match (
    tournament_code   VARCHAR(8)      NOT NULL,
    climb             INT             NOT NULL,
    bracket           INT             NOT NULL,
    id_1              VARCHAR(255)    NOT NULL,
    id_2              VARCHAR(255)    NOT NULL,
    point_1           INT             NOT NULL,
    point_2           INT             NOT NULL,
    match_status      VARCHAR(255)    NOT NULL,
    FOREIGN KEY (tournament_code)  REFERENCES tournaments (tournament_code) ON
DELETE CASCADE,
    FOREIGN KEY (id_1)             REFERENCES accounts (id)        ON DELETE
CASCADE,
    FOREIGN KEY (id_2)             REFERENCES accounts (id)        ON DELETE
CASCADE
);
```

The Database SQL is designed in such a way where if an account is deleted, all associate data within from other tables will be purged alongside. This principle is the same goes to teams, lobbies, and tournaments.

Overall, specification of the back-end components:

Table 4.2.2 – Back-End Specifications

Module	Module Version	Functionality
Ubuntu	20.04	Linux OS that is used during hosting of droplet.
NodeJS	16.8.0	Server-side run-time environment.
NPM	7.21.0	Node Package Manager where it is responsible for updating and maintaining packages within the web application.
PM2	5.2.0	Daemon manager where the web application lifetime would be managed.
MySQL	8.0.28	Database system used for the web application and managing crucial informations.
dotenv	10.0.0	Node package with allows read of sensitive information stored in environment variable file.
bcrypt	5.0.1	Node package that functions as salt hasher for encryption for the password of registered user.
Node-cron	3.0.0	Node package for scheduling repeating task such as deletion of completed lobby or tournaments.
nodemailer	6.7.3	Node package for mailing verification code for registered user.

4.2.3 Repository Component

This section of the component is setup and designed for developers and administrator. It is generally a good practice to create a private GitHub repository for the purpose of the web application, remember to not include sensitive information to be included into the repository.

As for GitHub CI/CD, it can be done by creating a workflows file named “deploy.yaml” inside the .github folder of the repository. Inside this deployment file is where the executable commands for updating and maintaining the web application is written.

```yaml
name: Build & Deploy
on:
  push:
    branches: [master]

jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - name: Deploy NodeJS app
        uses: appleboy/ssh-action@v0.1.2
        with:
          host: ${{secrets.SSH_HOST}}
          key: ${{secrets.SSH_KEY}}
          username: ${{secrets.SSH_USERNAME}}
          command_timeout: 30m

          script: |
            pm2 stop app
            cd cyto_live
            git clone git@github.com:SammyDeAgent/cytokine-osmium.git
            cd cytokine-osmium
            git add -A
            git commit -a -m 'Commiting linux dependencies'
            git pull origin master
            rm -rf node_modules/
            npm install
            npm audit fix
            pm2 start app
            pm2 save
            echo "Deploy to Digital Ocean Droplet successful."
```

Figure 4.2.3 - Deploy YAML file

Other than that, an important piece of variable within the CI/CD process would be the SSH secrets. These variables are configured in the repository secrets section and can be obtained by generating RSA key pairs in the host machine. In this case, generating the keys in the Digital Ocean Droplet's virtual machine (Linux Distributed in this case).

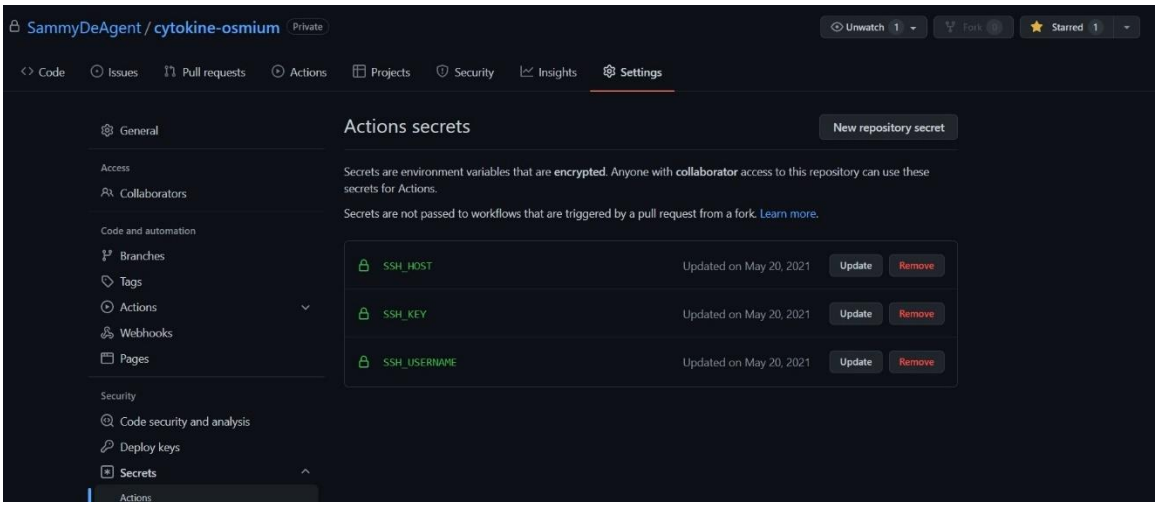


Figure 4.2.4 – GitHub Action Secrets

Overall, specification of the repository components:

Table 4.2.2 – Repository Specifications:

Module	Module Version	Functionality
GitHub	-	Version control and CI/CD middleware for the web application.
VS Code	-	A popular IDE that comes with various useful extension.

Chapter 5

System Implementation

In this chapter, system implementation and configuration of both hardware and software of the system are discussed in detail and shown. Alongside with the development workflow, methods, and also developmental testing methods.

5.1 Development Workflow and Methodologies

The developmental process of the project follows an Agile approach as transition between the phase are much faster and more preferred in the event where design of the project would need to be revisited.

And thus, to accompany prototyping, testing and refined in a rapid session, the agile method of rapid application development is chosen to be the workflow of the project.

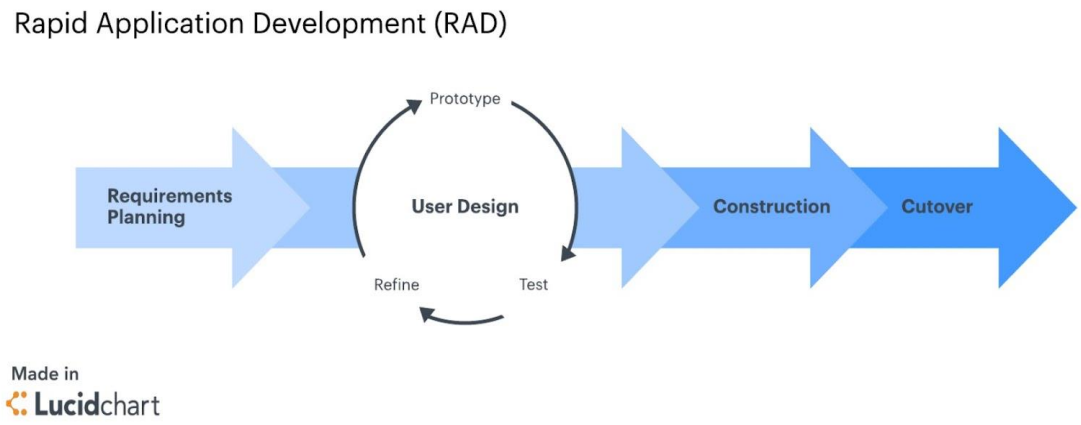


Figure 5.1.1 – Rapid Development Life Cycle

5.2 System Setup and Configuration

The implementation of the system starts with initiating a new compute deployment from any cloud or web service, in this case, Digital Ocean Droplets. After that, copy the virtual machine IP in order to SSH into the droplet using FTP and SSH programs such as PuTTY or WinSCP. The password of both SSH and FTP protocol into the virtual machine are the same password when initiating the compute device.

Upon successful SSH into the compute device, ensure that a web server service is present, if not install one, in this case: Nginx web server. Configure the Nginx web server alongside the config file to enable SSL HTTPS (ensure that a domain for the website is available). After that, install pre-requisite modules such as NodeJS, NPM, PM2, MySQL and Git into the machine if not yet installed.

Setup MySQL by following the built-in instructions and create a schema named “Osmium”, here generate the aforementioned 13 essential tables. The complete table specification is found in Chapter 4.

In the root directory, create a new folder called “live”, inside this folder, you can now git clone the repository via the following command:

```
git clone git@github.com:SammyDeAgent/cytokine-osmium.git
```

The repository is private at the moment of development, and thus for usage please contact the main GitHub account **SammyDeAgent** for access.

After cloning the repository, setup the environment variable file at the root of the project folder as such:

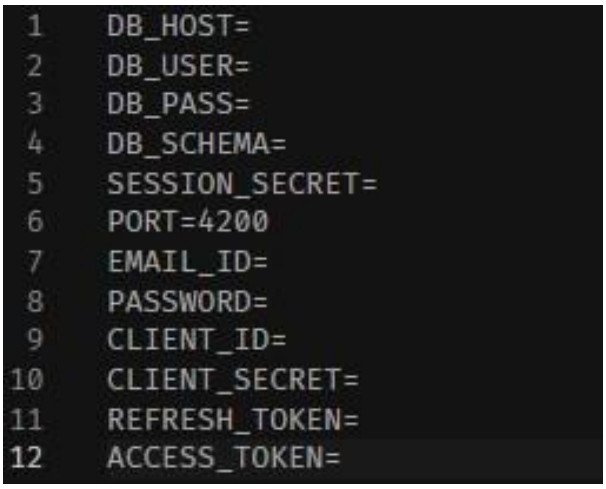


Figure 5.2.1 – Environment Variables

These variables are used throughout the project, as such it is crucial to set the variable beforehand. The “DB” section of the variable are the credentials as the database you have set previously. Session secret are used to store session information, it can be your own secrets. The port of the web application is opened at port 4200 as a proxy port, all request would still need to come through port 80 or port 443. The port in the environment variable file can be changed, just remember to change the port of the proxy in the Nginx config too.

For registration verification, you would require creating a Gmail account for the nodemailer alongside the Oauth 2.0 credentials that can be generated via Google Workspace and APIs Oauth 2.0 Client Generation.

After finishing configuration, the web application is now ready to be started. Simply initiate a PM2 command:

```
pm2 start app
```

The web application would now be hosted in the domain provided in the cloud service or the Ipv4 address of the virtual machine at port 4200.

5.3 Implementation Issues and Challenges

On the side of implementation, there have been an issue where the droplet would vanish without explanation, and thus it is wise to create a snapshot or backup of the droplet in case of emergency.

The usage of CPU and RAM would fluctuate randomly between normal and 100% (overload status). After throughout investigations, this has been believed to be caused by insufficient memory and resource(as the droplet has only 1GB and 1vCPU). On the other hand, requests from bots and crawlers have also been a contributing factor to the abnormal resource usage of the virtual machine.

A main programmatic challenge would be the tournament module, as storing the entire tournament tree requires in-depth processing, reconstructing and data storing. Not to mention, calculation between opponents and ascending the “climb” of a player would also be requiring a lot of design and planning. Thought the module is functional, it could be riddle with exploits or bugs that would require for testing by numbers.

5.4 System Testing

In the project, developmental testing is done in a sense that the process is quick, and result is clear. For instance, by usage of alpha and beta testing, debugging and system testing are done at a quick speed and allow allocation of more time on development.

5.4.1 Alpha Testing

After a certain version of the project development is done, alpha test can immediately take place. The test is usually done by internal developers and trusted tester before the service could go live.

The environment in which the test is conducted usually is in the developers or tester's local environment. The testing process are done by following component test (sole function), then proceeding to integration test (connection with another module), usually by checking the functionalities or expected output for a certain action.

The overall results and feedback from the tests are presented through a meeting after the test is complete or via direct messaging to the developer.

5.4.2 Beta Testing

After alpha testing is conducted, the updated version would then go live by deploying it into the live server by updating the web application.

During these, user or players are able to access the web application and act as testers together for the version test.

The overall results and feedback of the test are recorded via 2 ways, either from user inputs through support communication channel such as developer's DM or Email. Or by viewing the developmental logging system.

5.4.3 Developmental Logging

Developmental logging is one of the ways to receive results and feedback from the previous beta testing. The logging system is integrated into the code base itself, as such any action of the user would be logged in this way by timestamp and actions. One of PM2 monitoring function allows for this result viewing.

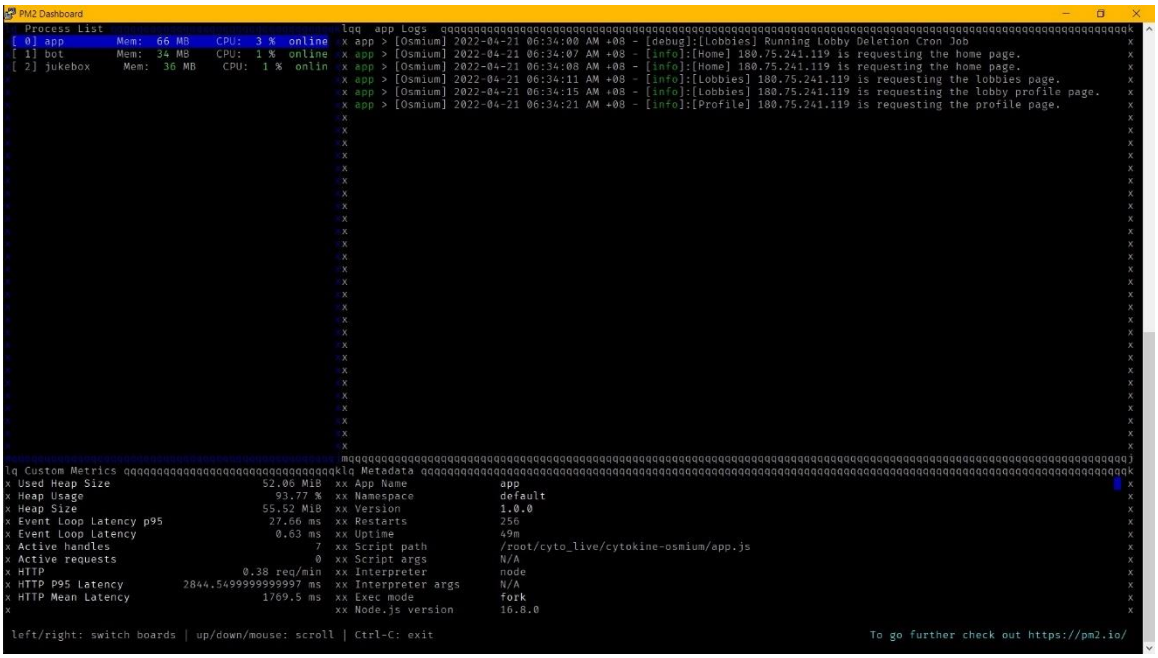


Figure 5.4.1 – PM2 Monitoring Function

Aside that, the logging would also be saved in a log file in the “logs/” directory of the project under the name “combined.txt”. In the event of abnormal response, the logs can be viewed as reference to what is going on without constant monitoring the system.

Chapter 6

System Evaluation and Discussion

The following evaluation are formed after many thoughts from both developer itself and testers feedback on the current implemented system.

6.1 Functional and Objectives Evaluation

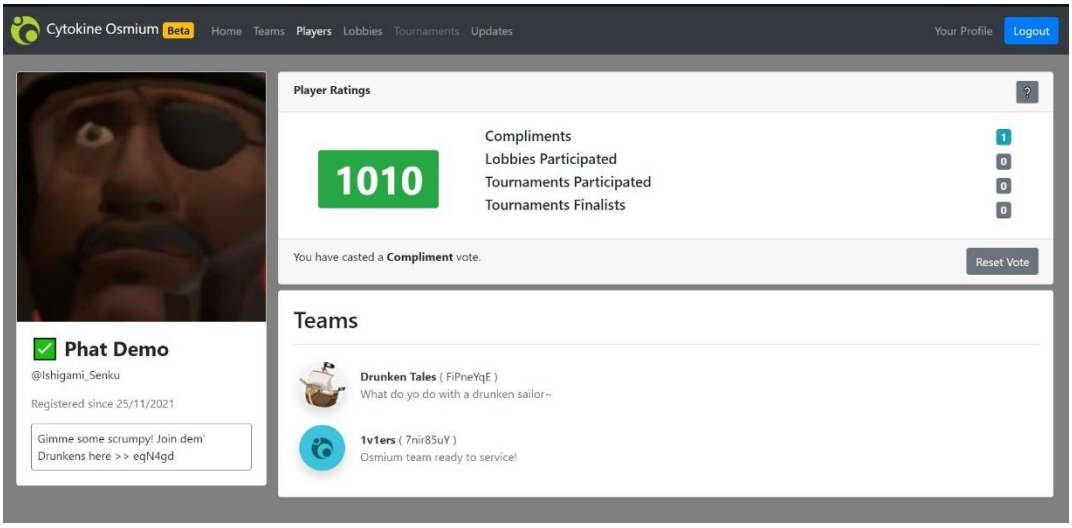


Figure 6.1.1 – Osmium Player Profile

Overall, all objectives proposed in the beginning are able to meet satisfactory standards. The project is able to meet the conditions of objective I and III where team building, and rating system modules are based on. Though minor adjusted in the rating system is still undergoing experiments as user could potentially abuse the rating system.

On the topic of Objectives II, lobby, and tournament creation functionalities are able to be done but the resulting work lacks some QoL functions such as selection of tournament modes or kicking uninvited player.

While the project also suffers some minor bugs and exploits, the whole project is still being able to meet the requirements set beforehand successfully.

6.2 Non-Functional Evaluation

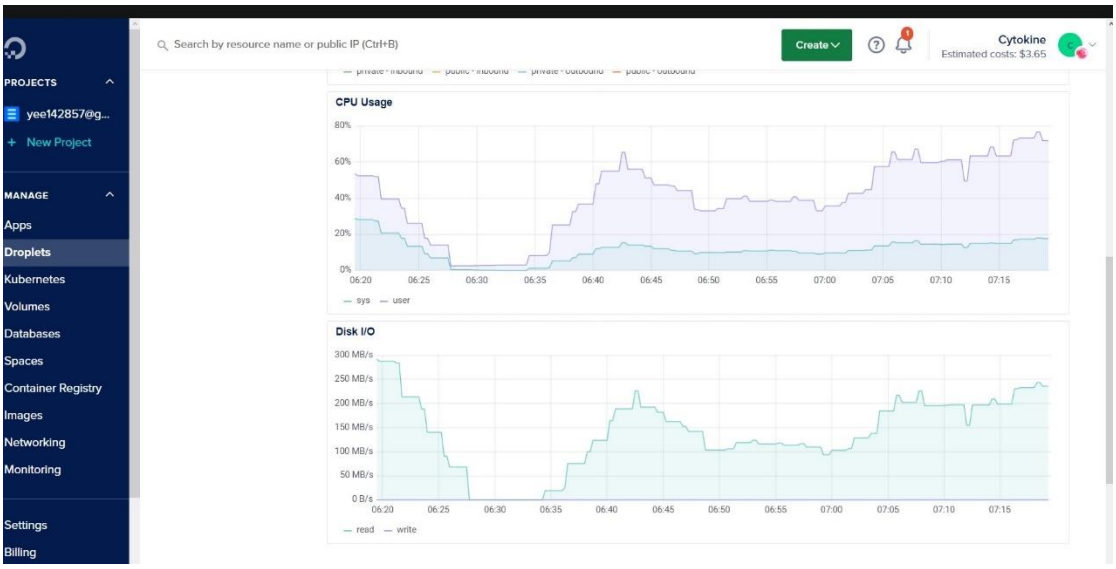


Figure 6.2.1 – Digital Ocean Droplet Metrics

Performance wise, the web application does suffer from major downtime due to random and unexpected overloading CPU and RAM usage. Based on earlier testing, some request to the server would be considered as DDoS attack or brute-force probing attack, which would somewhat explain the dip in performances.

On the subject of security, with the usage of Oauth 2.0 in the mail service and bcrypt encryption using salt and hashing, the security of the web application is acceptable in most conventional cases. In addition, the introduction of robot's protocol and SSL HTTPS essentially made the web application very secure.

Hacking attempts such as SQL injection are deemed impossible due to the setup of the code, any inputs from the user request are automatically encoded as string, which nullifies the incoming attack on the database.

However, improvements are still be made such as introducing blacklist on known bot address or scheduled cron job to restart the server to prevent connection leak from the MySQL service.

6.3 GUI and UX Evaluation

Generally, from the feedback of the testers, both the GUI and UX factor of the web application are deemed satisfactory by standard, however the testers might have been able to point out some inconsistency in design and lacking some QoL functions such as kicking player or auto-refresh on page, while some were pondering on confusing buttons. But user experience wise, the testers are able to navigate through easily and without needing prior context, which meets the condition as the webapp is made for experienced players in mind.

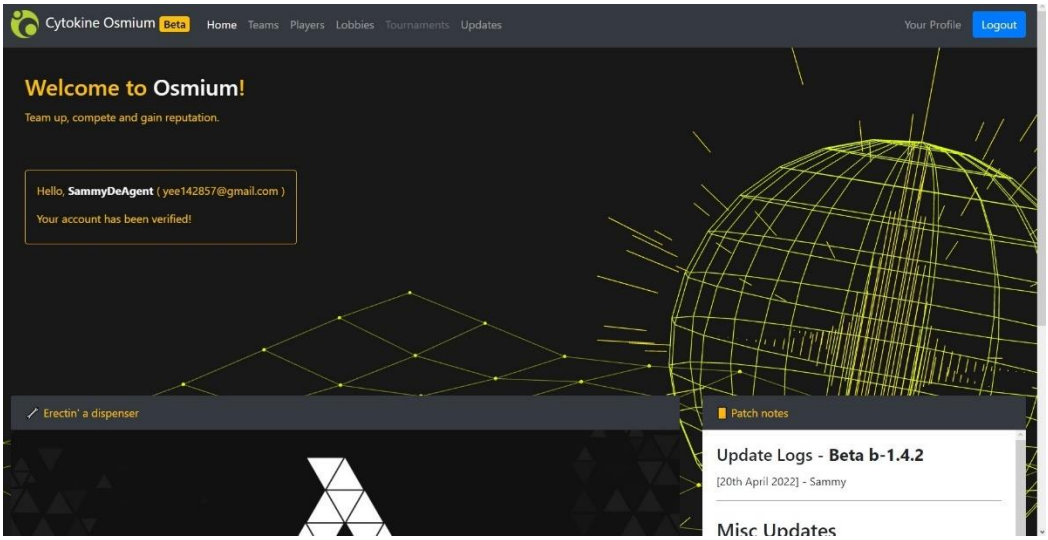


Figure 6.3.1 – Osmium Home Page

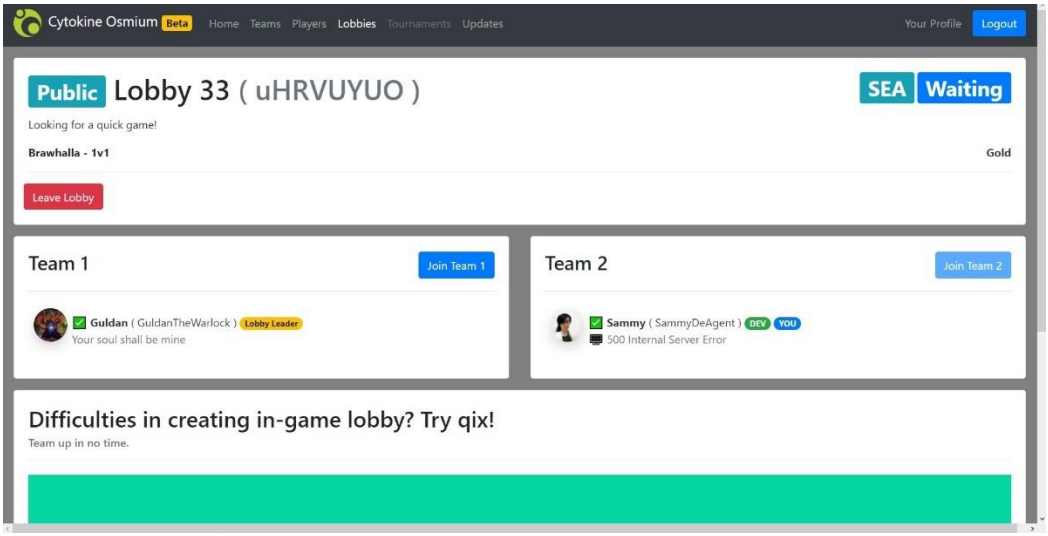


Figure 6.3.2 – Osmium Lobby Profile

Chapter 7

Conclusion and Future Works

7.1 Overall Conclusion

Concisely, the overall project is able to meet over ~90% of the objectives requirement that are set beforehand. Over the developmental period, problems that surface would get fixed and refined on haste due to efficient system testing methods. Evaluation of the project also shows that users (testers included) are able to navigate through the web app with ease and are capable to use the functionalities provided without external help. Though some minor bugs may remain unnoticed, the user experience of the webapp are still able to satisfy majority of the testers. The system's performance might be on the lacking side, but the high security of the webapp is able to keep up with the non-functional requirements. However, whether or not the webapp would get picked up by a large audience if still up for debate, only time could tell.

7.2 Future Works

The development of the full-stack web application is not an easy task at hand, as one would need to tackle both front-end and back-end at the same time. Further works are still needed to be done such as minor QoL functions and improvement to the design and look of the web site. Would recommend collaborations with other developers if ever one would be doing a similar project.

REFERENCES

[1] M. Moncav, “What is Elo? An explanation for competitive gaming’s hidden rating system,” Dots Esport, 2020. [Online]. Available: https://dotesports.com/general/news/elo-ratings-explained-20565. [Accessed 3 April 2021].

[2] Overwatch Blizzard, “Overwatch Patch Notes,” [Online]. Available: https://playoverwatch.com/en-us/news/patch-notes/ptr/2018/03. [Accessed 3 April 2021].

[3] Valve Corporations, “Team Fortress 2,” Team Fortress 2, 2007. [Online]. Available: https://www.teamfortress.com/. [Accessed 18 April 2022].

[4] Valve Corporations, “Valve Anti-Cheat - Valve Developer Community,” Valve, [Online]. Available: developer.valvesoftware.com. [Accessed 5 April 2021].

[5] T. Wilde, “Team Fortress 2 is under attack by bots, and not the fun kind (Updated).,” PC Gamer, 2020. [Online]. Available: https://www.pcgamer.com/team-fortress-2-is-under-attack-by-bots-and-not-the-fun-kind/. [Accessed 5 April 2021].

[6] Match.TF, “Home - match.tf,” Match.TF, [Online]. Available: https://match.tf/. [Accessed 5 April 2021].

[7] FACEIT Esports, “Home - FACEIT.com,” FACEIT Esports, [Online]. Available: https://www.faceit.com/en/home. [Accessed 18 April 2022].

[8] Info Bloom, “What is a Pick up Game? (with pictures),” Info Bloom, [Online]. Available: https://www.infobloom.com/what-is-a-pick-up-game.htm. [Accessed 5 April 2021].

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 1**
Student Name & ID: Yee Zi Yang	
Supervisor: 18ACB02834	
Project Title: ONLINE ESPORTS COMPETITIVE TEAM FINDER WEB APP WITH BUILT-IN TOURNAMENT CREATOR, CUSTOM LOBBY MATCH-UP AND PROFILE RATINGS BASED ON PARTICIPATIONS AND COMPLIMENTS	

1. WORK DONE

Overall clean up and restructuring of code base by using new efficient coding method, such as front-end components/sections and reformatting of indentation.

2. WORK TO BE DONE

Minor quality of life and bug fixes from user direct reports from the alpha/beta testing.

3. PROBLEMS ENCOUNTERED

RAM and CPU usage still occasionally maxes out for no reason, need to review the problem much more in-depth.

4. SELF EVALUATION OF THE PROGRESS

Distributed Computer System introduces new learning platform (AWS), might need to allocate time for that and development.

SyMhmd

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 2**
Student Name & ID: Yee Zi Yang	
Supervisor: 18ACB02834	
Project Title: ONLINE ESPORTS COMPETITIVE TEAM FINDER WEB APP WITH BUILT-IN TOURNAMENT CREATOR, CUSTOM LOBBY MATCH-UP AND PROFILE RATINGS BASED ON PARTICIPATIONS AND COMPLIMENTS	

1. WORK DONE

Introduced some fixes and overall design overhaul, included patch notes section for future updates.

2. WORK TO BE DONE

Investigate high CPU and RAM usage, creation of development logging system to monitor traffic and activities.

3. PROBLEMS ENCOUNTERED

Comminating new CI/CD cycle are starting to fail the GitHub actions, might need to increase the timeout of the deployment.

4. SELF EVALUATION OF THE PROGRESS

Spent too much time on some module alone, development time might be affected. Need to think of a way to simplify testing.

SuMhmd

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 3**
Student Name & ID: Yee Zi Yang	
Supervisor: 18ACB02834	
Project Title: ONLINE ESPORTS COMPETITIVE TEAM FINDER WEB APP WITH BUILT-IN TOURNAMENT CREATOR, CUSTOM LOBBY MATCH-UP AND PROFILE RATINGS BASED ON PARTICIPATIONS AND COMPLIMENTS	

1. WORK DONE

Integrated new developmental logging system, which improves debugging time and enable investigation on CPU and RAM issue.

2. WORK TO BE DONE

Player rating modules next on the list.

3. PROBLEMS ENCOUNTERED

Random IP, which speculates to be bots or web crawler are requesting security breaching URL, fortunately sensitive info is not exposed to the public site. However, CPU and RAM usage are still a concern.

4. SELF EVALUATION OF THE PROGRESS

Internet is a scary place; these random IP came from every region. If left a loophole unchecked, whole project might get decimated, thus need to be careful on the security side.

SuMhmd

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 4**
Student Name & ID: Yee Zi Yang	
Supervisor: 18ACB02834	
Project Title: ONLINE ESPORTS COMPETITIVE TEAM FINDER WEB APP WITH BUILT-IN TOURNAMENT CREATOR, CUSTOM LOBBY MATCH-UP AND PROFILE RATINGS BASED ON PARTICIPATIONS AND COMPLIMENTS	

1. WORK DONE

Preliminary player rating/compliment system, where ratings are calculated based on lobby, tournament participation, tournament finalist and player voting.

2. WORK TO BE DONE

Rework deployment workflow and solve CPU and RAM issues.

3. PROBLEMS ENCOUNTERED

Might need to adjust the value during the calculation of the rating system to avoid abuse. Tournament system need some more in-depth design due to it being a unique data structure to handle.

4. SELF EVALUATION OF THE PROGRESS

Need to catch up on other assignment soon, development time might get affected. Progress is looking grim on the tournament system.

SyMhmd

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 5**
Student Name & ID: Yee Zi Yang	
Supervisor: 18ACB02834	
Project Title: ONLINE ESPORTS COMPETITIVE TEAM FINDER WEB APP WITH BUILT-IN TOURNAMENT CREATOR, CUSTOM LOBBY MATCH-UP AND PROFILE RATINGS BASED ON PARTICIPATIONS AND COMPLIMENTS	

1. WORK DONE

Adjusted deployment workflow to ease server resource usage during CI/CD updates.

2. WORK TO BE DONE

Optimize database query pathway to avoid connection leak and integrate robots' protocol for web crawlers to reduce impact on server.

3. PROBLEMS ENCOUNTERED

CPU and RAM issue might stem from database query connection leak and also crawlers' activity.

4. SELF EVALUATION OF THE PROGRESS

Need to fasten the progress as there is still lobby and tournament system yet to be implemented. Emphasize on the tournament module.

SuMhmd

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 6**
Student Name & ID: Yee Zi Yang	
Supervisor: 18ACB02834	
Project Title: ONLINE ESPORTS COMPETITIVE TEAM FINDER WEB APP WITH BUILT-IN TOURNAMENT CREATOR, CUSTOM LOBBY MATCH-UP AND PROFILE RATINGS BASED ON PARTICIPATIONS AND COMPLIMENTS	

1. WORK DONE

Optimization on the database query and cron-scheduler to prevent unexpected connection leak. Also added new robots' protocol for web crawlers.

2. WORK TO BE DONE

Lobby system implementation.

3. PROBLEMS ENCOUNTERED

Registration module not working due to new Google's Oauth term and conditions.

4. SELF EVALUATION OF THE PROGRESS

Sudden Oauth update affected development time on other modules, taking this as a reminder that unexpected event might occur and be ready to face it.

SuMhmd

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 7**
Student Name & ID: Yee Zi Yang	
Supervisor: 18ACB02834	
Project Title: ONLINE ESPORTS COMPETITIVE TEAM FINDER WEB APP WITH BUILT-IN TOURNAMENT CREATOR, CUSTOM LOBBY MATCH-UP AND PROFILE RATINGS BASED ON PARTICIPATIONS AND COMPLIMENTS	

1. WORK DONE

Fixed registration module via Google's Oath2 method for mailing verification codes and opened template page on lobbies site.

2. WORK TO BE DONE

Addition work on lobby system.

3. PROBLEMS ENCOUNTERED

No problem encountered so far.

4. SELF EVALUATION OF THE PROGRESS

Upcoming other subject's workload might affect development, need to relocate time much more efficient during coding session.

SyMhmd

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 8**
Student Name & ID: Yee Zi Yang	
Supervisor: 18ACB02834	
Project Title: ONLINE ESPORTS COMPETITIVE TEAM FINDER WEB APP WITH BUILT-IN TOURNAMENT CREATOR, CUSTOM LOBBY MATCH-UP AND PROFILE RATINGS BASED ON PARTICIPATIONS AND COMPLIMENTS	

1. WORK DONE

Halted due to technical issue.

2. WORK TO BE DONE

Fix the current technical issue and start lobby system implementation.

3. PROBLEMS ENCOUNTERED

Digital Ocean Droplet where I deployed my project vanished, contacting support team, and hopefully can be recovered.

4. SELF EVALUATION OF THE PROGRESS

Remember to back up the database and any info as to unexpected event such as this.

SyMhmd

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 9**
Student Name & ID: Yee Zi Yang	
Supervisor: 18ACB02834	
Project Title: ONLINE ESPORTS COMPETITIVE TEAM FINDER WEB APP WITH BUILT-IN TOURNAMENT CREATOR, CUSTOM LOBBY MATCH-UP AND PROFILE RATINGS BASED ON PARTICIPATIONS AND COMPLIMENTS	

1. WORK DONE

Digital Ocean Droplet Recovered. Preliminary lobby system where user is able to create new lobby with specification such as game title, game mode, game region, skill level and etc.

2. WORK TO BE DONE

FYP2 Report writing and tournament system final drafting and implementation.

3. PROBLEMS ENCOUNTERED

There is an exploit regarding the lobby system where user other than lobby leader itself can start or end the lobby, needs to be fixed asap.

4. SELF EVALUATION OF THE PROGRESS

This week development might be on the smaller side due to AWS learning, hope this would not affect the dev time any further. Also, last week's technical issue reminds me that backup is extremely important.

SuMhmd

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 10**
Student Name & ID: Yee Zi Yang	
Supervisor: 18ACB02834	
Project Title: ONLINE ESPORTS COMPETITIVE TEAM FINDER WEB APP WITH BUILT-IN TOURNAMENT CREATOR, CUSTOM LOBBY MATCH-UP AND PROFILE RATINGS BASED ON PARTICIPATIONS AND COMPLIMENTS	

1. WORK DONE

Fixed certain bug and exploit in lobby module. Testing tournament module's associating database in offline mode for tree structure storing.

2. WORK TO BE DONE

Tournament module implementation.

3. PROBLEMS ENCOUNTERED

Design is one thing, implementation is another. Tournament module might be a roadblock currently.

4. SELF EVALUATION OF THE PROGRESS

Due to the time constraints, might have to do tournament module for individual user only. Team tournament really requires much more time, never underestimate the saying: good theory, bad executing.

SyMhmd

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 11**
Student Name & ID: Yee Zi Yang	
Supervisor: 18ACB02834	
Project Title: ONLINE ESPORTS COMPETITIVE TEAM FINDER WEB APP WITH BUILT-IN TOURNAMENT CREATOR, CUSTOM LOBBY MATCH-UP AND PROFILE RATINGS BASED ON PARTICIPATIONS AND COMPLIMENTS	

1. WORK DONE

Halted due to another subject's workload.

2. WORK TO BE DONE

Tournament module implementation.

3. PROBLEMS ENCOUNTERED

Triple workload from different subjects, development is postponed to complete other assignments and follow-up on AWS learning platform.

4. SELF EVALUATION OF THE PROGRESS

Unfortunately, I cannot tackle all 3 subjects at the same time. Thus, this week development would have to be delayed. Work smarter, not harder; I hope this quote won't bite me back in the end.

SeMhmd

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 12**
Student Name & ID: Yee Zi Yang	
Supervisor: 18ACB02834	
Project Title: ONLINE ESPORTS COMPETITIVE TEAM FINDER WEB APP WITH BUILT-IN TOURNAMENT CREATOR, CUSTOM LOBBY MATCH-UP AND PROFILE RATINGS BASED ON PARTICIPATIONS AND COMPLIMENTS	

1. WORK DONE

Preliminary tournament module creation, currently only support individual participation, not teams. And the structuring of code needs to be refined as well.

2. WORK TO BE DONE

Polishing the tournament module further, if can, feature a tournament style binary tree to the public site.

3. PROBLEMS ENCOUNTERED

Tournament module which encompasses the tree data structure on the website is hard to execute, as storing the tree would require more in-depth tinkering.

4. SELF EVALUATION OF THE PROGRESS

Finally, the usage of unique data structure. Need to review on past data structure subject to fully cover the tournament module implementation.

SuMhmd

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 13**
Student Name & ID: Yee Zi Yang	
Supervisor: 18ACB02834	
Project Title: ONLINE ESPORTS COMPETITIVE TEAM FINDER WEB APP WITH BUILT-IN TOURNAMENT CREATOR, CUSTOM LOBBY MATCH-UP AND PROFILE RATINGS BASED ON PARTICIPATIONS AND COMPLIMENTS	

1. WORK DONE

Polishing of tournament module system, hopefully it works as intended.

2. WORK TO BE DONE

Further code, GUI, and functionality polishing; might require collaborators in the future.

3. PROBLEMS ENCOUNTERED

Minor bugs and errors in the webapp generally.

4. SELF EVALUATION OF THE PROGRESS

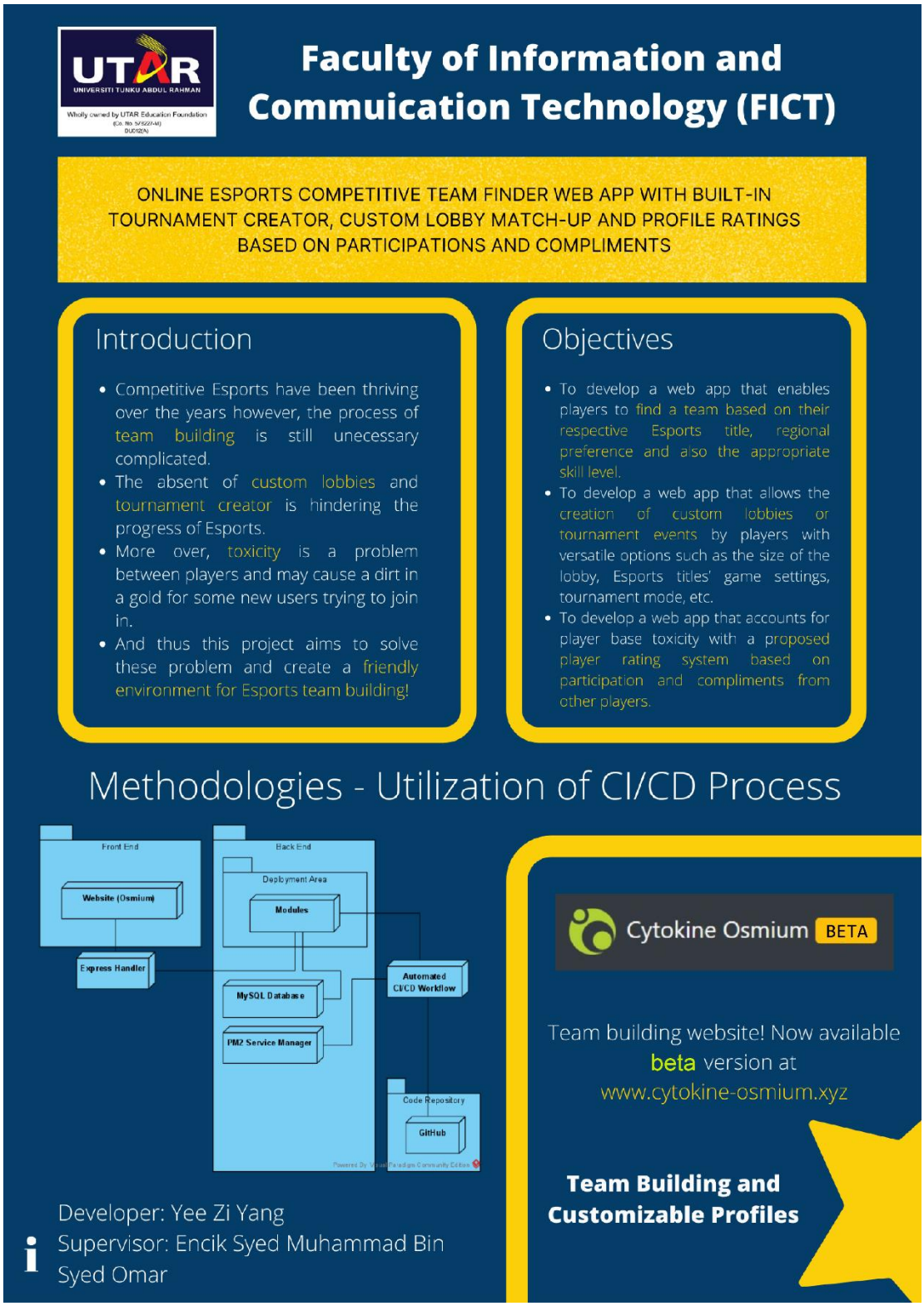
Including FYP1 and IIPSPW, this would conclude the whole developmental life cycle of the webapp. Honestly, project at this scale is underestimated by me, who thought full-stack project can be easily achieve via one-man-army. I can see why full-stack project in the public requires help in all sections and rarely one person handles both front and back end simultaneously. May this be an experience for me in the work field.

SyMhmd

Supervisor's signature

Student's signature

POSTER


UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
Wholly owned by UTAR Education Foundation
Faculty of Information and Commuication Technology (FICT)
ONLINE ESPORTS COMPETITIVE TEAM FINDER WEB APP WITH BUILT-IN TOURNAMENT CREATOR, CUSTOM LOBBY MATCH-UP AND PROFILE RATINGS BASED ON PARTICIPATIONS AND COMPLIMENTS
Introduction
Competitive Esports have been thriving over the years however, the process of team building is still unecessary complicated.
The absent of custom lobbies and tournament creator is hindering the progress of Esports.
More over, toxicity is a problem between players and may cause a dirt in a gold for some new users trying to join in.
And thus this project aims to solve these problem and create a friendly environment for Esports team building!
Objectives
To develop a web app that enables players to find a team based on their respective Esports title, regional preference and also the appropriate skill level.
To develop a web app that allows the creation of custom lobbies or tournament events by players with versatile options such as the size of the lobby, Esports titles' game settings, tournament mode, etc.
To develop a web app that accounts for player base toxicity with a proposed player rating system based on participation and compliments from other players.
Methodologies - Utilization of CI/CD Process
Front End
Website (Osmium)
Express Handler
Back End
Deployment Area
Modules
MySQL Database
PM2 Service Manager
Automated CI/CD Workflow
Code Repository
GitHub
Cytokine Osmium BETA
Team building website! Now available beta version at www.cytokine-osmium.xyz
Team Building and Customizable Profiles
Developer: Yee Zi Yang
Supervisor: Encik Syed Muhammad Bin Syed Omar

PLAGIARISM CHECK RESULT

ONLINE ESPORTS COMPETITIVE TEAM FINDER WEB APP WITH BUILT-IN TOURNAMENT CREATOR, CUSTOM LOBBY MATCH-UP AND PROFILE RATINGS BASED ON PARTICIPATIONS AND COMPLIMENTS

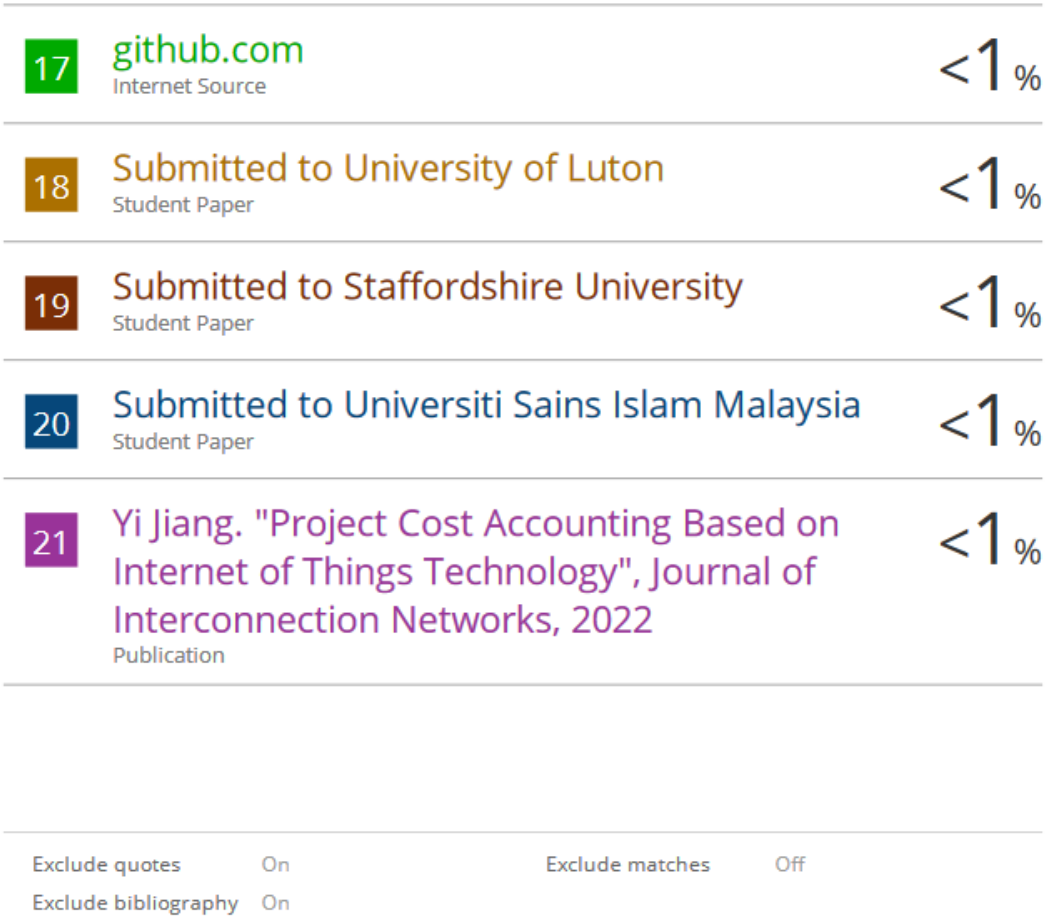
ORIGINALITY REPORT

6%	6%	4%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universiti Tunku Abdul Rahman Student Paper	1%
2	Submitted to University of Stirling Student Paper	1%
3	Submitted to University of Northampton Student Paper	1%
4	Alberto Hernández Chillón, Diego Sevilla Ruiz, Jesús García Molina. "Chapter 4 Athena: A Database-Independent Schema Definition Language", Springer Science and Business Media LLC, 2021 Publication	<1%
5	Submitted to University of Greenwich Student Paper	<1%
6	Submitted to Queensland University of Technology Student Paper	<1%

7	Submitted to Manchester Metropolitan University Student Paper	<1 %
8	eprints.utar.edu.my Internet Source	<1 %
9	blog.richigarza.me Internet Source	<1 %
10	Submitted to University of Warwick Student Paper	<1 %
11	Submitted to University of Surrey Student Paper	<1 %
12	Submitted to unibuc Student Paper	<1 %
13	E Sutanta, E K Nurnawati. "The Design of Relational Database for Multipurpose WebGIS Applications", Journal of Physics: Conference Series, 2019 Publication	<1 %
14	fict.utar.edu.my Internet Source	<1 %
15	Submitted to Royal Holloway and Bedford New College Student Paper	<1 %
16	Submitted to University of Newcastle upon Tyne Student Paper	<1 %


17
github.com
Internet Source
<1 %
18
Submitted to University of Luton
Student Paper
<1 %
19
Submitted to Staffordshire University
Student Paper
<1 %
20
Submitted to Universiti Sains Islam Malaysia
Student Paper
<1 %
21
Yi Jiang. "Project Cost Accounting Based on Internet of Things Technology", Journal of Interconnection Networks, 2022
Publication
<1 %
Exclude quotes On
Exclude matches Off
Exclude bibliography On

Plagiarism Check Result

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Yee Zi Yang
ID Number(s)	18ACB02834
Programme / Course	Bachelor of Computer Science (Honours)
Title of Final Year Project	Online Esports Competitive Team Finder Webapp with Built-in tournament creator, custom lobby match-up and profile ratings based on participations and compliments

Similarity	**Supervisor's Comments** **(Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 6 % **Similarity by source** Internet Sources: 6 % Publications: 4 % Student Papers: 5 %	False positive - The similarity index of Turnitin hits the SQL codes and reports organization.
Number of individual sources listed of more than 3% similarity: 0	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

SyMhmd

Signature of Supervisor

Name: Syed Muhammad Syed Omar

Date: 21/04/2022

Signature of Co-Supervisor

Name: _____

Date: _____

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

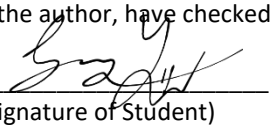
CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	18ACB02834
Student Name	Yee Zi Yang
Supervisor Name	Encik Syed Muhammad Bin Syed Omar

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
-	Front Plastic Cover (for hardcopy)
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
-	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
-	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.



(Signature of Student)
Date: 20 April 2022