Module Code & Module Title

CS6PO5NT-Final Year Project

Assessment Weightage & Type

Final Year Interim Report (25%)

Year and Semester

2025-26 Autumn

Student Name: Pukar Bohara

London Met ID: 23049326

College ID: np05cp4a230032

Assignment Submission Date: December 30, 2025

Assignment Due Date: December 20, 2025

Internal Supervisor: Mr. Amit Shrestha

External Supervisor: Mr. Bibek Maharjan

Title: Esport Arena (Web Application)

Word Count: 4910

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded

Abstract

The Esport Arena Interim Report shows the first steps and outlines the perspective of a web-based esports tournament administration platform that can address the chaotic and untrustworthy character of competitive gaming tournament organization in Nepal and the South Asian region. The project aims at offering a clear, safe and convenient system whereby the Players, Organizers and admins can control tournaments, finances and communications in one system. At this point, securing access control by user authentication, authorization as well as role-based interface has been laid down firmly.

Esport Arena is developed using the Scrum approach, which guarantees progressive and rolling changes by use of time-bound sprints, the sprints are short but effective, focused on a specific set of features. The main elements addressed in the present stage are the user authentication and authorization, secure registration and log in, assignment of permissions to users, multi-role (Player, Organizer, Admin) Access and base infrastructure of the role-based dashboard. This rigid architecture creates a trustful identity and access layer upon which future tournament, wallet and community modules could comfortably be constructed as well as enabling requirements to be changed depending on supervisor and stakeholder feedback.

The report includes the successful configuration of the authentication process, database designs, user and roles database, and the initial UI/UX of the registration and login process and the planning artifacts like product backlog and Gantt chart. Although more complex features such as automated bracketing, wallet integration with eSewa / Khalti, result of matches, and toxicity filtering will be made in the later sprints, design of a strong user identity and access control will go a long way towards lowering security and scalability risks in the future. It is thus the interim report that captures the basic achievements of the project and draws the roadmap to the development of Esport Arena into a full-fledged esports tournament arena in the next phases of development.

Table of Contents

1 Introduction 1

1.1 Problem Domain 2

1.2 Project as a solution 3

1.3 Project aim and objectives 4

2 Background 4

2.1 Similar Projects 5

2.1.1 Toornament 5

2.1.2 XtremBattle 6

2.1.3 TournaOP 7

2.1.4 Game.tv 8

2.1.5 InGame.gg 9

2.2 How Esport Arena Differentiation. 10

2.3 Review and Comparisons with Similar system 11

2.4 Methodology 12

2.4.1 Considered Methodology 12

2.4.2 Selected Methodology 12

3 Development till date 15

3.1 Requirement Gathering 15

3.1.1 SRS 15

3.1.2 Product Backlog 15

3.1.3 Pre Survey 16

3.2 System Design 19

3.2.1 System Architecture 19

3.2.2 ERD 20

3.2.3 Use Case Diagram 21

3.2.4 Class Diagram 21

3.2.5 Milestone Chart 23
3.2.6 Work Breakdown Structure 24
3.2.7 Style Guide 25
3.3 Sprint 1 (User authentication and authorization) 26
3.3.1 Sprint Planning 26
3.3.2 Design 27
3.3.3 Development 35
3.3.4 Sprint Review 40
3.3.5 Sprint Retrospective 42
3.3.6 Velocity Chart 43
3.3.7 Burndown Chart 44
4 Progress Analysis 45
4.1 Gantt Chart 46
4.2 Progress Table 47
4.3 Analysis 49
5 Further Work 50
6 Conclusion 50
7 References 51
8 Appendix 53
8.1 Background 53
8.1.1 Trends in the Field 53
8.1.2 Benefits of Esport Arena 53
8.1.3 Challenges 53
8.1.4 Future Possibilities 54
8.2 Product Backlog 55
8.3 Pre Survey 58
8.4 Considered methodologies 63

8.4.1 Waterfall Model 63
8.4.2 Prototyping Model 65
8.4.3 Incremental Model 67
8.4.4 Iterative Model 70
8.4.5 Spiral Model 72
8.4.6 Scrum Framework 75
8.5 Work Till Date 76
8.5.1 SRS 76
8.5.2 INTRODUCTION 76
8.5.3 Purpose 77
8.5.4 Intended Audience and Intended Use 77
8.5.5 Definition and acronyms 79
8.5.6 Overview 79
8.5.7 Functional requirements 81
8.5.8 Nonfunctional Requirements 84
8.5.9 Hardware and Software Specifications 84
8.5.10 Other Non-functional Requirements 85
8.5.11 Sprint 1 86
8.6 Progress Analysis 88
8.6.1 Analysis 88
8.7 Further Work 89
8.8 Conclusion 91

Table of Figures

Figure 1 Logo of Esport Arena 1

Figure 2 Toornament 5

Figure 3 XtremBattle 6

Figure 4 TournaOP 7

Figure 5 Game.tv 8

Figure 6 InGame.gg 9

Figure 7 Pre Survey Question 1 16

Figure 8 Pre Survey Question 2 16

Figure 9 Pre Survey Question 3 17

Figure 10 Pre Survey Question 4 17

Figure 11 Pre Survey Question 5 18

Figure 12 System Architecture of Esport Arena 19

Figure 13 ERD for Esport Arena 20

Figure 14 Use Case for Esport Arena 21

Figure 15 Class Diagram of Esport Arena 22

Figure 16 Milestone Chart 23

Figure 17 WBS chart of Esport Arena 24

Figure 18 Collor Palette for Esport Arena 25

Figure 19 Sign Up User Sequence Diagram 27

Figure 20 Sign in User Sequence Diagram 28

Figure 21 Activity Diagram of Sign up User 29

Figure 22 Activity Diagram of Log in User 30

Figure 23 Wireframe of Registration User Page 31

Figure 24 Wireframe of Login User Page 32

Figure 25 Design of Registration Page 33
Figure 26 Design of Login Page 34
Figure 27 Register as a Player Frontend 35
Figure 28 Register as a Organizer Frontend 36
Figure 29 Login as any User Frontend 37
Figure 30 Backend File Path 38
Figure 31 Backend User Model Creation 39
Figure 32 Backend Api Creation using Swagger 39
Figure 33 Velocity Chart of Esport Arena 43
Figure 34 Burndown Chart of Esport Arena Sprint 1 44
Figure 35 Esport Arena Gantt Chart 1 46
Figure 37 Esport Arena Gantt Chart 3 46
Figure 36 Esport Arena Gantt Chart 2 46
Figure 38 Esport Arena Gantt Chart 4 47
Figure 39 Pre Survey Question 6 58
Figure 40 Pre Survey Question 7 58
Figure 41 Pre Survey Question 8 58
Figure 42 Pre Survey Question 9 59
Figure 43 Pre Survey Question 10 59
Figure 44 Pre Survey Question 11 59
Figure 45 Pre Survey Question 12 60
Figure 46 Pre Survey Question 13 60
Figure 47 Pre Survey Question 14 60
Figure 48 Pre Survey Question 15 61

Figure 49 Pre Survey Question 16 61

Figure 50 Pre Survey Question 17 61

Figure 51 Pre Survey Question 18 62

Figure 52 Pre Survey Question 19 62

Figure 53 Scrum framework diagram 75

Table of Tables

Table 1 Review of Similar System 11

Table 2 Justification Table Scrum:1 13

Table 3 Justification Table Scrum:2 13

Table 4 Justification Table Scrum: 3 13

Table 5 Justification Table Scrum: 4 14

Table 6 Justification Table Scrum: 5 14

Table 7 Sprint 1 Backlog 26

Table 8 Planned Items for Sprint 1 41

Table 9 Incomplete Item for Sprint 1 41

Table 10 Sprint 2 Retrospective 42

Table 11 Progress Table of Esport Arena 47

Table 12 Product Backlog for Esport Arena 55

Table 13 Justification Table Waterfall Model: 1 63

Table 14 Justification Table Waterfall Model: 2 64

Table 15 Justification Table Waterfall Model: 3 64

Table 16 Justification Table Waterfall Model: 4 64

Table 17 Justification Table Waterfall Model: 5 65

Table 18 Justification Table Prototyping Model: 1 66

Table 19 Justification Table Prototyping Model: 2 66

Table 20 Justification Table Prototyping Model: 3 66

Table 21 Justification Table Prototyping Model: 4 67

Table 22 Justification Table Prototyping Model: 5 67

Table 23 Justification Table Incremental Model: 1 68

Table 24 Justification Table Incremental Model: 2 68

Table 25 Justification Table Incremental Model: 3 69

Table 26 Justification Table Incremental Model: 4 69

Table 27 Justification Table Incremental Model: 5 69

Table 28 Justification Table Iterative Model: 1 70

Table 29 Justification Table Iterative Model: 2 71

Table 30 Justification Table Iterative Model: 3 71

Table 31 Justification Table Iterative Model: 4 71

Table 32 Justification Table Iterative Model: 5 72

Table 33 Justification Table Spiral Model: 1 73

Table 34 Justification Table Spiral Model: 2 73

Table 35 Justification Table Spiral Model: 3 74

Table 36 Justification Table Spiral Model: 4 74

Table 37 Justification Table Spiral Model: 5 74

Table 38 High Level Description of Sign-up User 86

Table 39 High Level Description of Log in User 86

Table 40 Expanded Use Case Description for Sign Up User 87

Table 41 Expanded Use Case Description for Sign In User 88

1 Introduction

The Esports activity in Nepal and South Asia is quickly rising, events like community and regional events organized to play famous games like PUBG Mobile and Free Fire are rising worldwide and attracting thousands of players. Regardless of this expansion, the majority of tournaments are structured with the help of disjointed tools, including social media conversations, spreadsheets, and manual payments. Some of the common issues that may arise out of this disjointed method are lack of clarity about registration procedures, conflicts in schedules, ineffective communication, biased decision-making, and lack of financial goodwill. Organizers have a hard time controlling brackets, checking the results of matches, and having difficulties with ways to pay, as players suffer because of ambiguous gameplay guidelines, lack of updates, bad communication, and concerns about fairly giving out prizes.

Therefore, Esport Arena is suggested to overcome these difficulties by becoming a single centralized online tournament management solution that introduces all of the main tournament activities to one system, which is structured and well-organized. The system is aimed at matching Players and Organizers; this is controlled by one Admin who oversees control and policies at the system level. Esport Arena will incorporate secure authentication, role-based dashboards and well-defined workflows in the creation, joining and running of tournaments. The service aims to minimize manual work and achieve greater transparency by offering such features as automated tournament brackets, virtual payments made with the coin system combined with local gateways such as eSewa and Khalti, submission of match results with evidence, and social controls such as forums and notifications. With a more systematic, transparent and reliable organization, Esport Arena will adjust the overall experience of participating in the esports tournament and help facilitate the long-term expansion of the competitive gaming environment in the region.

Figure 1 Logo of Esport Arena

1.1 Problem Domain

In Nepal and in emerging markets of esports, most online tournaments continue to be operated using manual tools such as spreadsheets, chats and ad-hoc payment tools, resulting in errors and limiting scalability. Manual collection of the player information, bracket keeping and payout of prizes such as prize money requires organizers to use multiple and unconnected systems, which heightens chances of delay in processing results, errors in processing and overpayment or underpayment of prizes. To players this may be late updates or no updates at all, schedules that are not clear and they just do not know whether their prize money will be delivered or not, this will make them not trust community events as much as they should (Mittal, 2023).

The distribution of prizes is perhaps a weakest area: as the process of distributing the prizes implies gathering sensitive information manually, calculations, and payment in small bits, the organizers have to face security challenges, bottlenecks, and frustrated winners who have to wait several weeks before receiving their payment. According to studies or industry reports, this type of manual workflow is inefficient, whatsoever written by human hands, and hard to audit particularly with increase in participants and prize pools. Simultaneously, abusive communication and unpleasant play in competitive games are also known to be a central concern, and some studies have shown that the existing moderation systems and tools are inefficient and should be improved to be capable of controlling abusive chat and ensuring compliant and pleasant playing conditions. The combination of such problems indicates that an esports tournament platform is specifically needed that can easily automate brackets and money transfers and can also facilitate decent organizational communication, internet monitoring, and management of all participants (Smith, 2023).

1.2 Project as a solution

Esport Arena is one of the modern solutions that can help overcome the management issues related to esports in Nepal and South Asia. Although esports competition is rapidly developing in these areas, various events continue to be organised with default and fragmented software. Esport Arena provides players, organizers, and admins with a well-structured, automated, and easy-to-use platform that connects the whole experience in hosting, joining, and administering competitive games.

With the ever-expanding world of esports, Nepal has begun paying some attention to esports via bodies like the Nepal Esports Association (NESA) and such national competitions as the Nepal Esports Championship. This development notwithstanding, there are still concerns of awareness, infrastructure and organization slowing long-term growth. Esport arena is aimed at transparency, security and fair play to enable development of a stable and sustainable esports ecosystem.

Key Features

Automated Tournament Management: Quickly create tournaments with automated creation and management of tournament registration, match brackets and automated results management.

Secure Payment System: Coin Wallet that is virtual and has local gateways such as eSewa and Khalti to carry out transactions securely.

Toxicity Detection: The AI-driven chat moderation can decrease the amount of abusive language and encourage a high level of respect towards players and the game.

Notifications and Updates: Instant information on the matches and updates to the users by using in-app notifications.

Community Forum: Special areas to communicate, exchange ideas and make announcements between players and the organizers.

1.3 Project aim and objectives

The purpose of Esport Arena is to create a safe and convenient web application that will hold esports tournaments safely, transparently, and without prejudice to the players, organizers, and admins. The Esport Arena platform is primarily aimed at ensuring that the process of conducting esports tournaments become more convenient, transparent, and secure to all participants. It achieves this by providing easy-to-use dashboards to various positions, paying and receiving money easily, artificial intelligence such as identifying toxic chat, real-time updates, discussion features, and high security to ensure that everything is clear and secure.

Objectives:

- Offer safe user authentication and role-based Dashboards to players, organizers and admins.
- Make management and registration of tournaments and brackets simpler with automated registrations and brackets.
- Prepare the ground on the safe financial handling by using virtual wallet system and local gateway such as eSewa and Khalti.
- Enforce effective communication by use of notification and scheduled community section to communicate with the users.
- Protect all data with encryption and secure login.

2 Background

Esports tournaments are becoming more frequent, and it is not an easy task to conduct them without any problems and with fairness and efficiency to both the organizer and players. Esport Arena is being built as the centralized hub that has a single system combining Players, Organizers, and Admins. It allows users to find, participate, and organize tournaments in a well-organized and clear way, which eliminates the use of diffused chats and manual procedures. The goal of the platform is to reduce the amount of confusion, save time, and make competition gaming experience more organized and enjoyable.

Further Description

2.1 Similar Projects

2.1.1 Toornament

Toornament is an online management platform of esports tourism that is popular and used to arrange online and offline tournaments or competitions worldwide. It helps organizers to establish tournaments, handle participants, and post brackets and results in a web-based system. The platform has numerous types of games and forms of competitions, so it can be used in minor events within a community or major competitions (play.toornament, 2025).

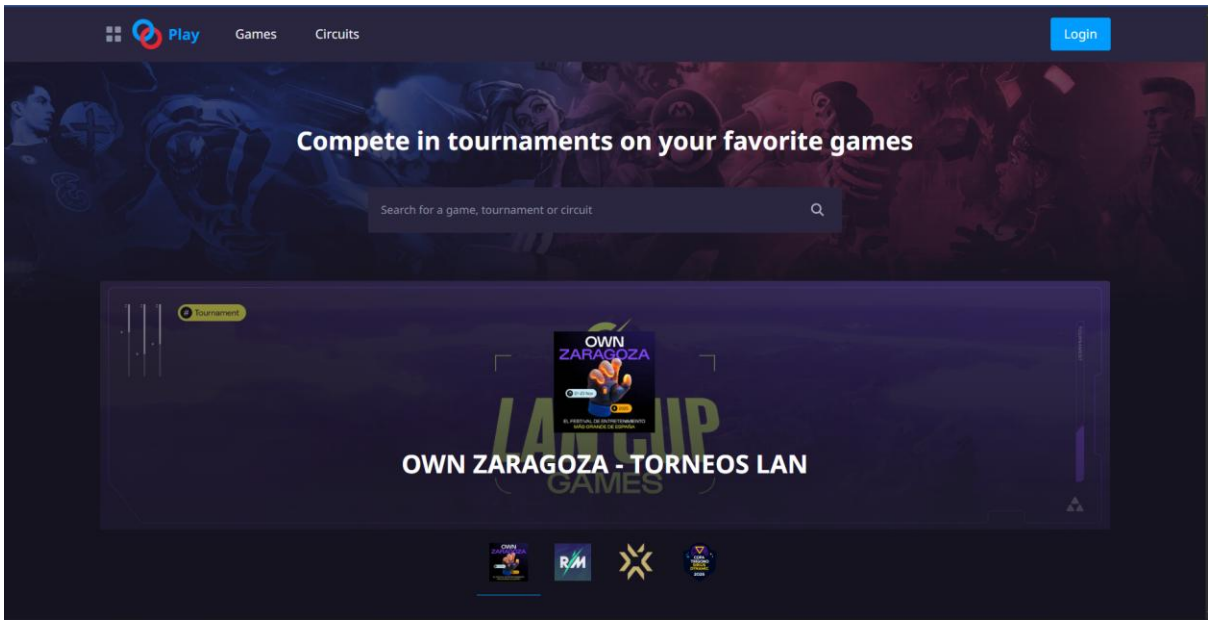


Figure 2 Toornament

Features & Strength:

- Supports the creation of tournaments in multiple formats, stages, and web based.
- Enables online registration, seeding, scheduling and displaying of results.
- Offers small, medium, and large size tournaments, APIs and widgets.

Limitations:

- None of the inbuilt digital wallet of handling entry fee and prize pool.
- The transactions with payments and payouts must be processed using third parties.
- Rudimentary local features and automatic toxicity control.

2.1.2 XtremBattle

XtremBattle is a Nepal based esports site where it primarily functions as an information centre to local gaming events. It publicizes future events, provides tournament information and prizes, and keeps individuals connected with esports events in Nepal. Nevertheless, majority of operations involved in tournament are processed manually out of the platform (xtrembattle, 2022).

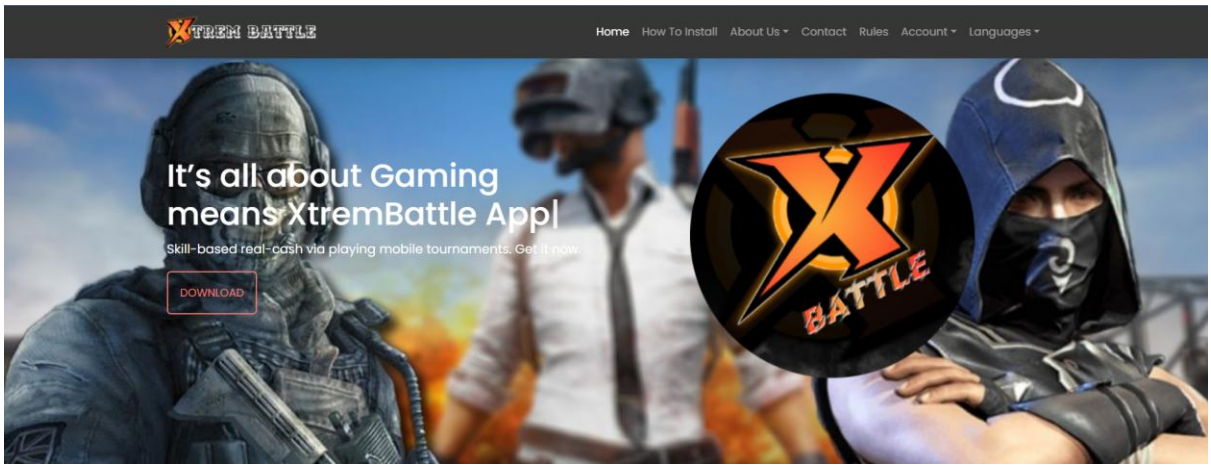


Figure 3 XtremBattle

Features & Strength:

- Shows esports tournaments in the future, including barebones information.
- Stocks regulations, game details, and contact details of organizers.
- Dedicates itself to growing the Nepali esports community by giving news updates.

Limitations:

- None of the system of automatically managing bracket or match results.
- Locks added the feature of payment or digital wallets.
- None of the role-based dashboards of players, organizers, or admins.

2.1.3 TournaOP

TournaOP is a competitive game platform that is aimed primarily at PUBG Mobile and Free Fire users. It enables them to make their own custom rooms, enroll in matches, and receive rewards in a built-in points system. The platform is a blend of a hosting platform and a gaming marketplace that isn't built around a tournament structure but is aimed at quick competitive experiences (TournaOP, 2025) .

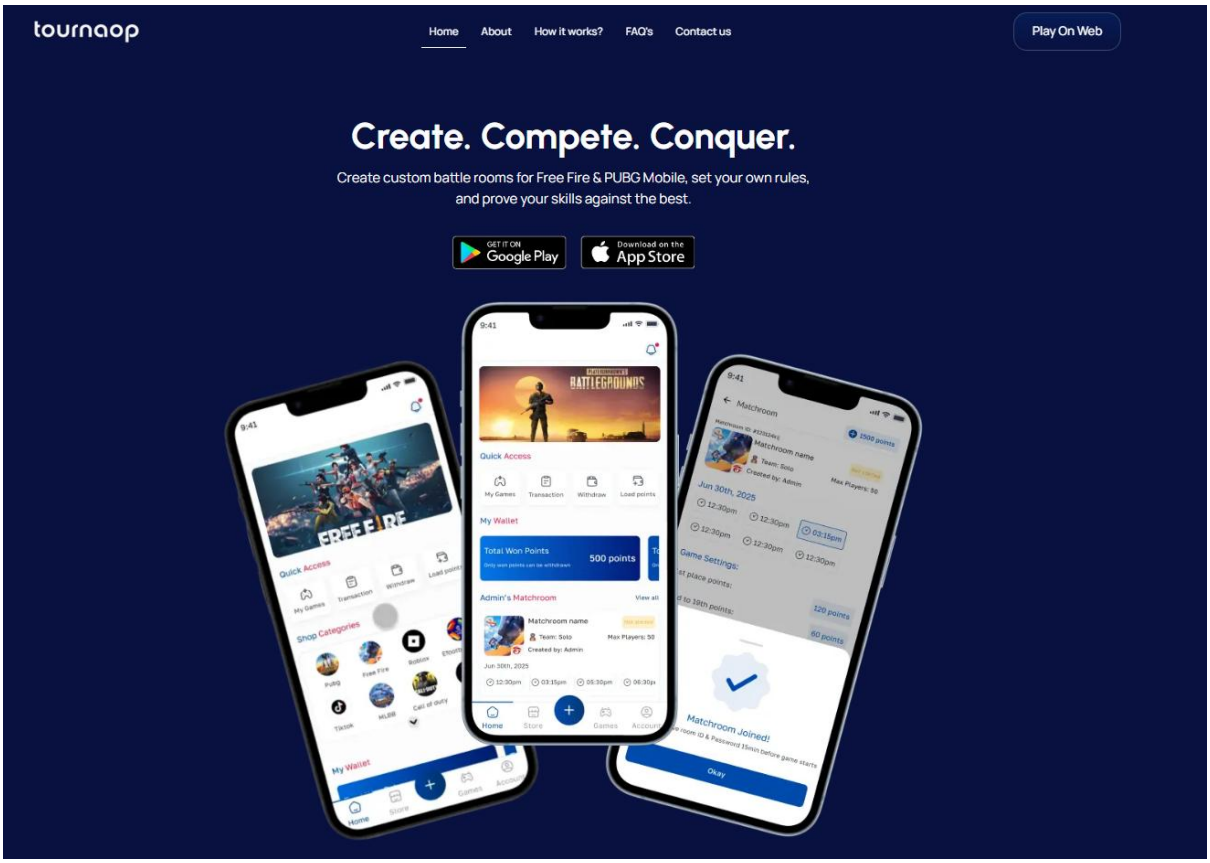


Figure 4 TournaOP

Features & Strength:

- Allows participants to make and customize match rooms with settings.
- Relies on an in-app points system to get into matches and as a reward system.
- Has an inbuilt gaming and UC coin store.

Limitations:

- And does not support multi-round or large tournament brackets.
- Wallet system is not transparent or tournament specific to the organizers.
- Poor dispute management tools and sophisticated moderation.

2.1.4 Game.tv

Game.tv is an international esports network which covers hosting of tournaments in a wide range of popular games. It allows the organizers to operate automated contests and players can comfortably find tournaments, sign up and follow results. The platform is scalable and aimed at automation particularly of communal and influencer-led events (Game.tv, 2024).

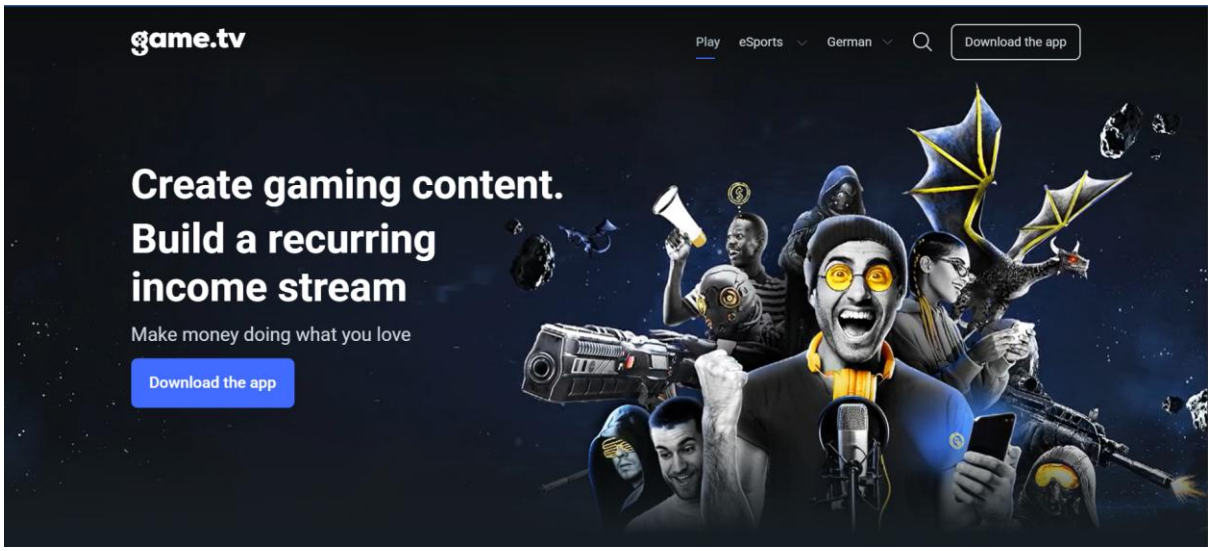


Figure 5 Game.tv

Features & Strength:

- Discovers and registers tournament through web and mobile applications.
- Automates schedules, brackets and live updates of results.
- Enables interactivity with the community by supporting applications such as Discord.

Limitations:

- None of convincing digital wallets to store organizer-controlled finances.
- Restricted personalization in the payment and reporting systems.
- Little emphasis on geographic and local pay.

2.1.5 InGame.gg

The InGame.gg is a South Asia-centred esports hub where professional tournaments and leagues are arranged. Through the site, players can subscribe to events, see brackets, and follow results. The platform is dedicated to organizing its own regional events instead of having full customization of independent organizers (InGame.gg, 2024).

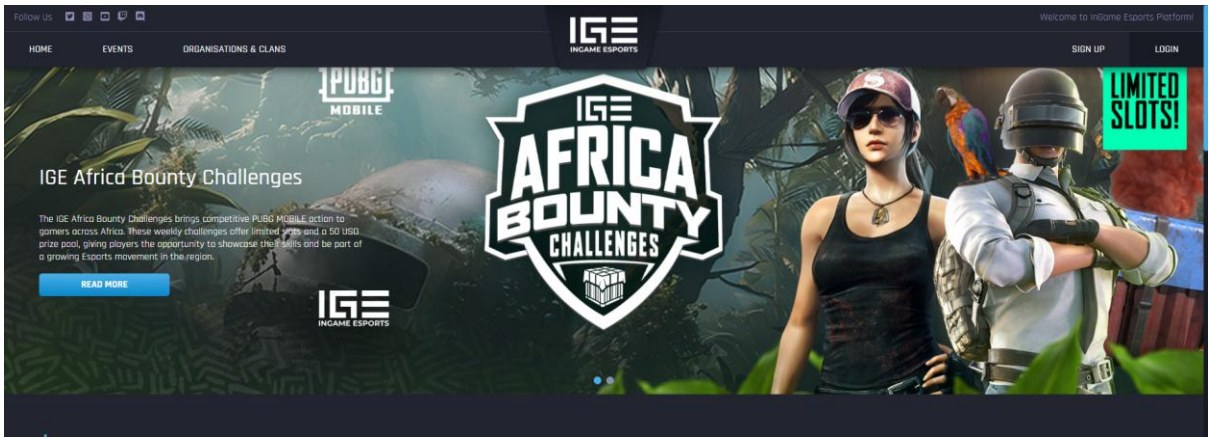


Figure 6 InGame.gg

Features & Strength:

- Registration of South Asian esports tournaments is also possible online.
- Lists fixtures, brackets and findings on the site.
- Organizes numerous regional esports on a single platform.

Limitations:

- No customizable organizational digital wallets.
- The platform regulates entry fees and payouts.
- Poorly developed in-built moderation and financial reporting tools.

2.2 How Esport Arena Differentiation.

As a solution, Esport Arena complements the existing platforms and local tools in managing esports tournaments by closing the gaps existing in current tournaments.

Structured Tournament Management:

- Accommodates full tour features, such as registration, brackets, and resultant monitoring.
- Allows Players, Organizers and Admins to use distinct dashboards in order to be clearer and more controlled.

Secure Wallet System:

- Uses a virtual coin wallet to handle entry fees, and distribution of prize.
- Plans integration with eSewa and Khalti to pay secure and traceable money.

Fair Play and Community Focus:

- Supplies evidence result submission to curb controversy.
- Prepares AI-based chat moderation and notifications, and forums to aid in fair play and structured communication.

2.3 Review and Comparisons with Similar system

Table 1 Review of Similar System

Features	Esport Arena	Toornament	XtremBattle	TournaOP	Game.tv	InGame.gg
User Registration & Profiles	✓	✓	✓	✓	✓	✓
Tournament Creation	✓	✓	✗	✓	✓	✓
Automated Brackets	✓	✓	✗	✗	✓	✓
Role-based Dashboards	✓	✓	✗	✗	✓	✗
Secure Payment / Wallet System	✓	✗	✗	✓	✗	✗
Local Payment Integration	✓	✗	✗	✗	✗	✗
Real-time Notifications & Updates	✓	✓	✗	✓	✓	✓
Community Forum / Interaction	✓	✗	✗	✗	✓	✗
Fair Play & Result Verification	✓	✓	✗	✗	✓	✓
Toxicity Detection / Moderation	✓	✗	✗	✗	✗	✗

2.4 Methodology

2.4.1 Considered Methodology

There were several possible software development methodologies suitable for building Esport Arena. I have considered the following approaches for this project:

- Waterfall Model
- Prototyping Model
- Incremental Model
- Iterative Model
- Spiral Model

Each of these models was reviewed with respect to Esport Arena's needs for evolving requirements, multiple modules, and regular feedback. Even though they all were efficient in certain contexts, they were not finally chosen because they do not provide the same level of flexibility, sprint-based planning, and role clarity that Scrum offers for this dynamic esports' platform.

2.4.2 Selected Methodology

Scrum is especially appropriate to Esport Arena given that the system needs regular upgrades, mods, and fast upgrades which is a vital attribute in a contemporary industry such as esports. Scrum work particularly well in an environment related to games and interactive settings due to its intuition in fast iteration, structured feedback cycles and flexibility to alter technical requirements (intersog, 2020). These same attributes render Scrum a perfect model to develop a platform that would need incorporation of real-time functionality, secure payments, and developing user demands.

Table 2 Justification Table Scrum:1

Case Study Scenario	**Managing high-frequency variations in the rules of tournaments, APIs, or payment gateways of Esport Arena.**
Features	Scrum applies time boxed sprints that have a prioritized product backlog that can be updated during the beginning of a sprint.
Description	This flexibility will enable Esport Arena to modify features very quickly in response to a change in tournament formats or that the project requires payment and notification APIs to be adjusted, without altering the entire project plan.

Table 3 Justification Table Scrum:2

Case Study Scenario	**Problems with authentication, tournament logic and wallet flows should be identified as early as possible.**
Features	Every sprint has a small increment of working software that is testable and well-designed, developed, and tested.
Description	It can be found and corrected at an early stage through the development and testing of Esport arena modules in short sprints, preventing rework, and improving results.

Table 4 Justification Table Scrum: 3

Case Study Scenario	**Observing current work on user stories such as the login, tournaments and wallet features.**
Features	Scrum delivers visibility into the progress and blockers regularly using such general practices as daily stand-ups, sprint backlogs, and sprint reviews.
Description	All these events are used to monitor the progress that Esport arena has achieved to date, to ensure that activities progress according to schedule and fast decision-making should priorities or problems emanate.

Table 5 Justification Table Scrum: 4

Case Study Scenario	**Handling a multi-module esports server under Academic time and resources.**
Features	Scrum breaks down the project into small sprints which are realistic and have deliverables and acceptance criteria.
Description	Division of Esport Arena into sprint-sized milestones (e.g. authentication, tournaments, wallet, forums) helps to make planning realistic, reduces complexity and inclines to steady and measurable movement towards the end system.

Table 6 Justification Table Scrum: 5

Case Study Scenario	**There is more value attached to players, organizers, and admins which should be prioritized.**
Features	Scrum maintains a prioritized order product backlog based upon business value and user influence and reviewed with each sprint.
Description	In the case of Esport Arena, it implies that such fundamental elements as secure login, entry to the tournaments, and wallet flow are provided earlier in the process than unimportant extras.

The complete description of the Scrum Framework is provided in the appendix section. (Scrum Framework)

3 Development till date

3.1 Requirement Gathering

Majority of requirements of Esport Arena were collected via an online survey via Google Form that was distributed to Nepalese gamers and small-scale tournament organizers to learn their actual issues related to community esports. The feedback included such concerns like manual registration, ambiguous brackets, cash-based payments, absence of records of transactions, disconnected communications, and unhealthy chat, and requested secure local-wallet innovations (eSewa/Khalti), automated tournaments, transparent wallets, and real-time notifications. These observations were written down and transformed into elaborate user stories and SRS requirements that inform the design and development of the platform.

3.1.1 SRS

A Software Requirement Specification (SRS) is a well-organized, detailed documentation of a software system which states both the functional and non-functional requirements and forms the basis around which developers, testers, supervisors and future maintainers base their work. An SRS bridges the gap between the conceptual notion and actual development by stating the purpose of the system, its behaviour, limitations, and expectations placed on it by users in a clear way. (Overall SRS)

3.1.2 Product Backlog

The product Backlog of the entire project is provided in the appendix section of the project report. (Product Backlog)

3.1.3 Pre Survey

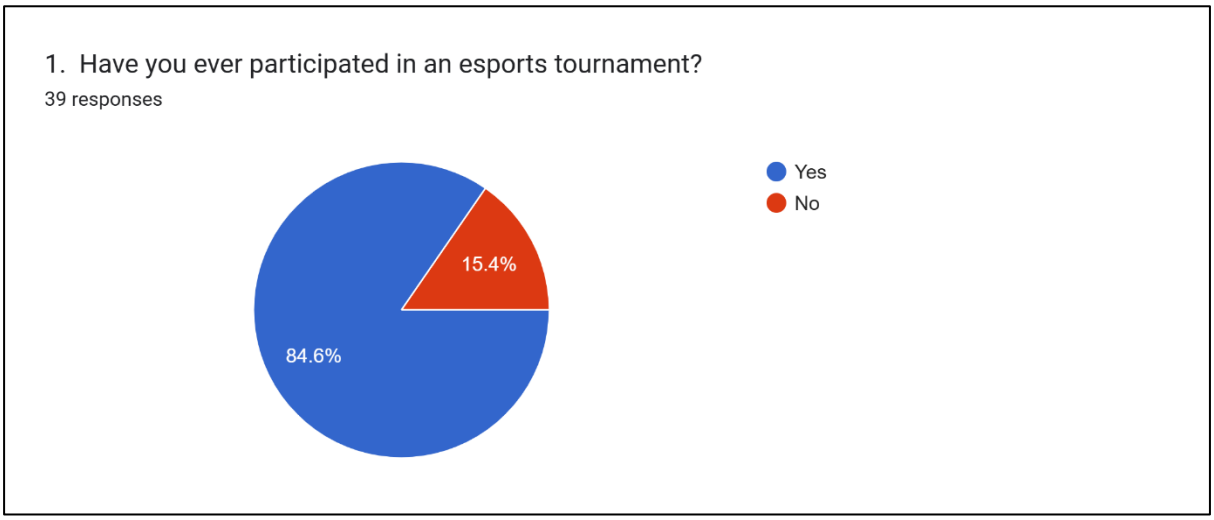


Figure 7 Pre Survey Question 1

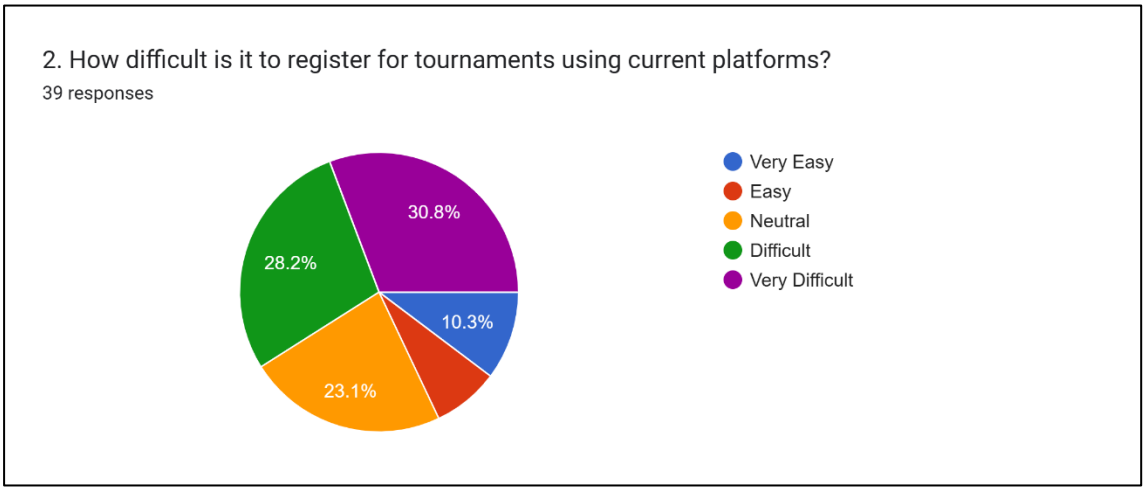


Figure 8 Pre Survey Question 2

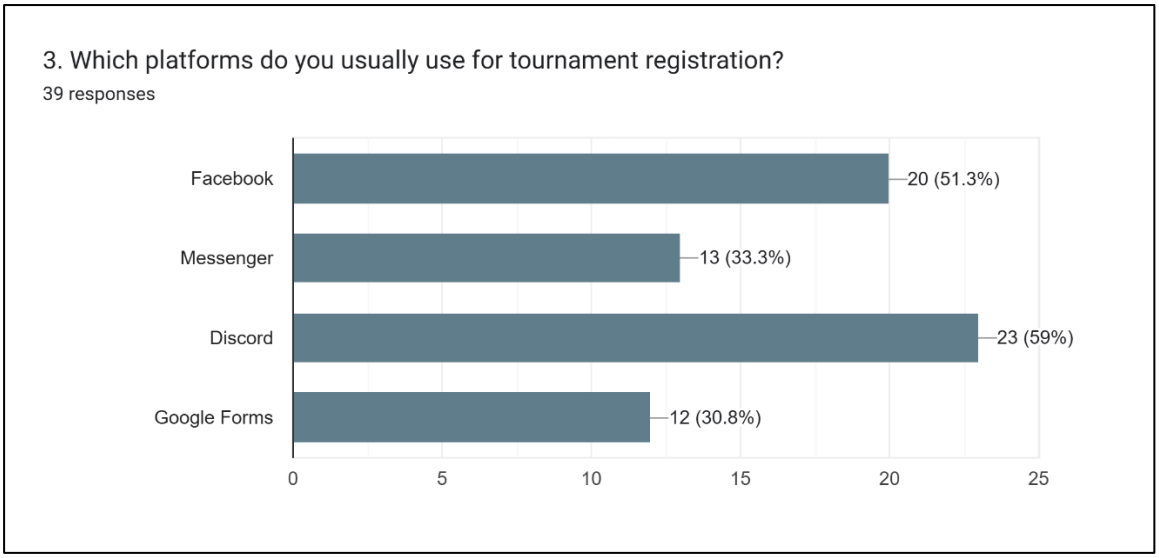


Figure 9 Pre Survey Question 3

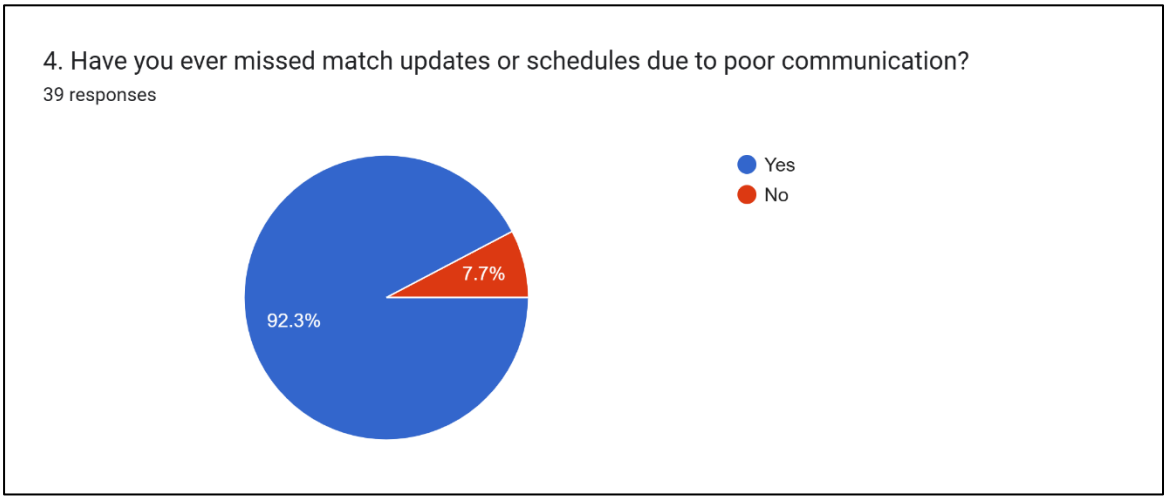


Figure 10 Pre Survey Question 4

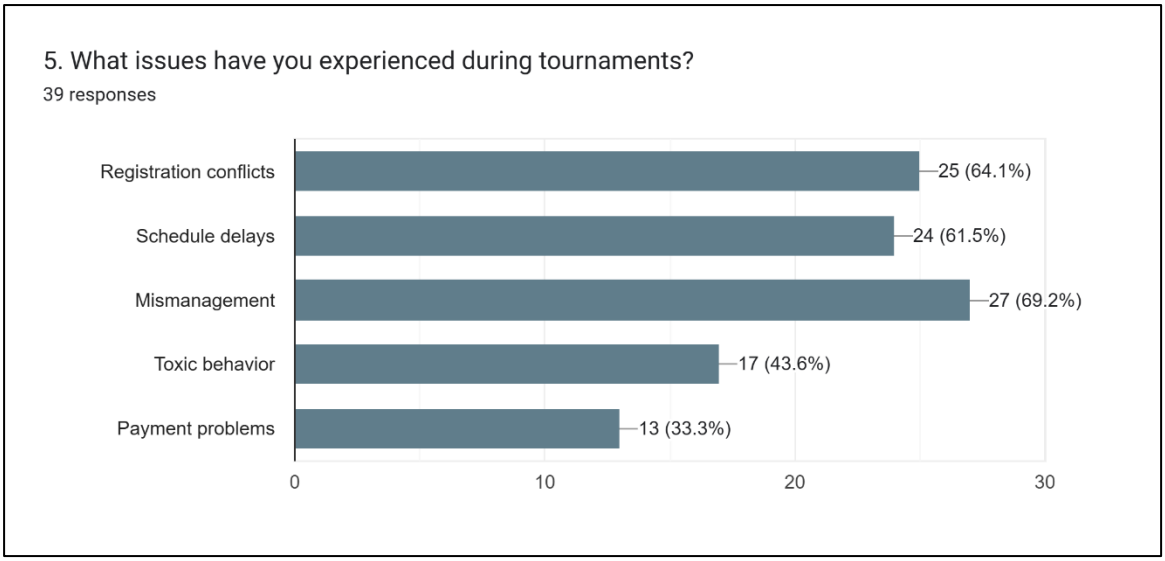


Figure 11 Pre Survey Question 5

View More Response

3.2 System Design

3.2.1 System Architecture

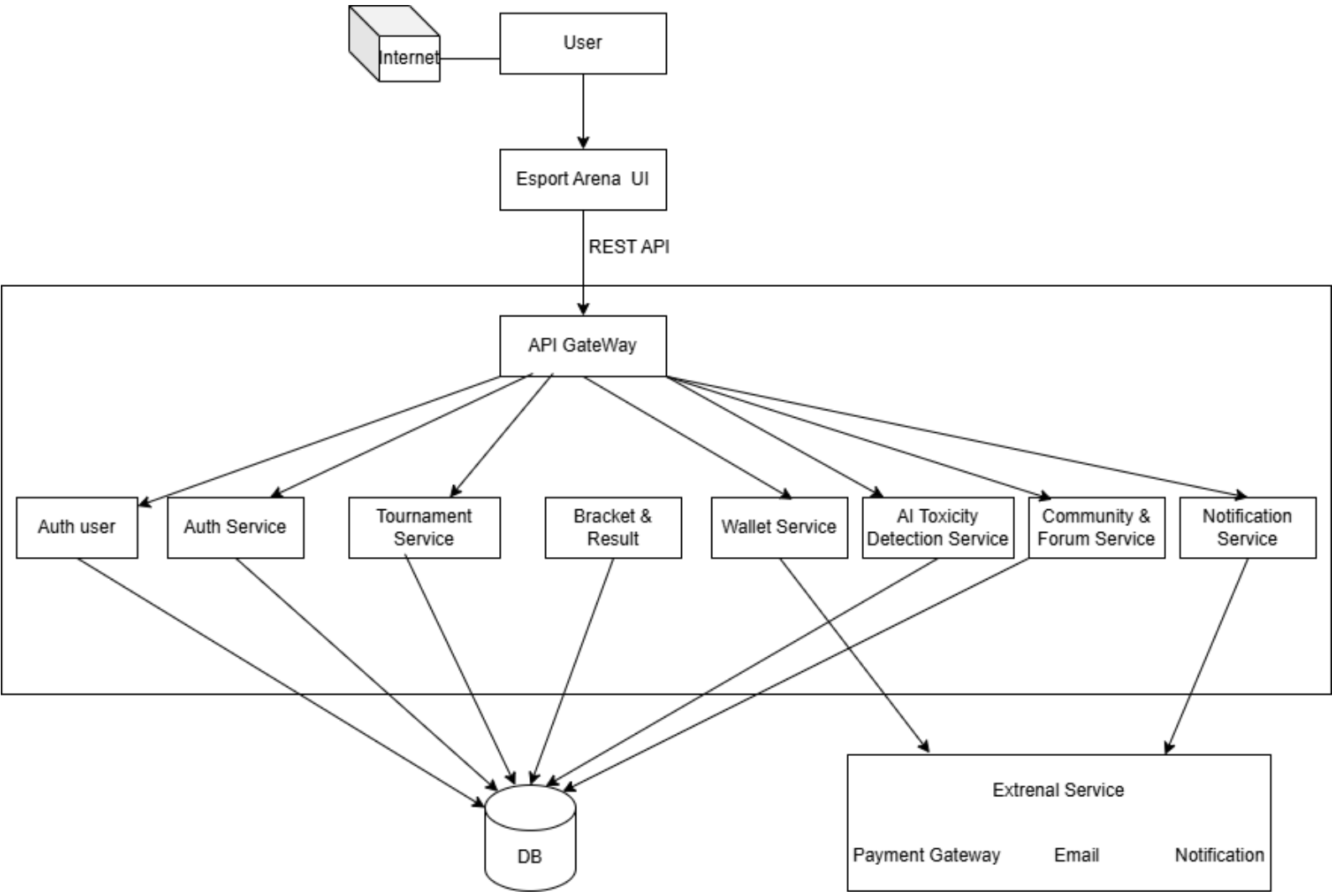


Figure 12 System Architecture of Esport Arena

3.2.2 ERD

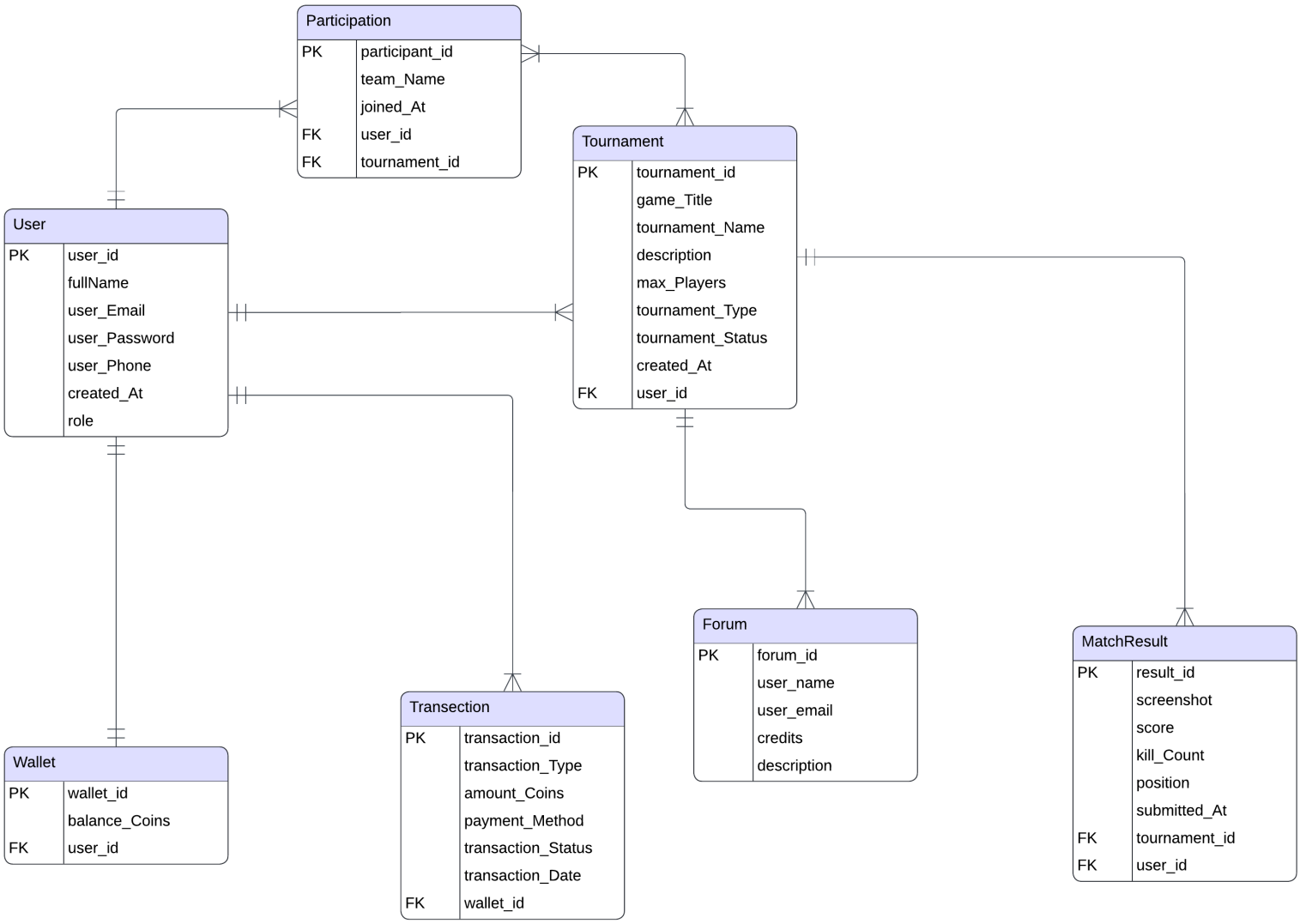


Figure 13 ERD for Esport Arena

3.2.3 Use Case Diagram

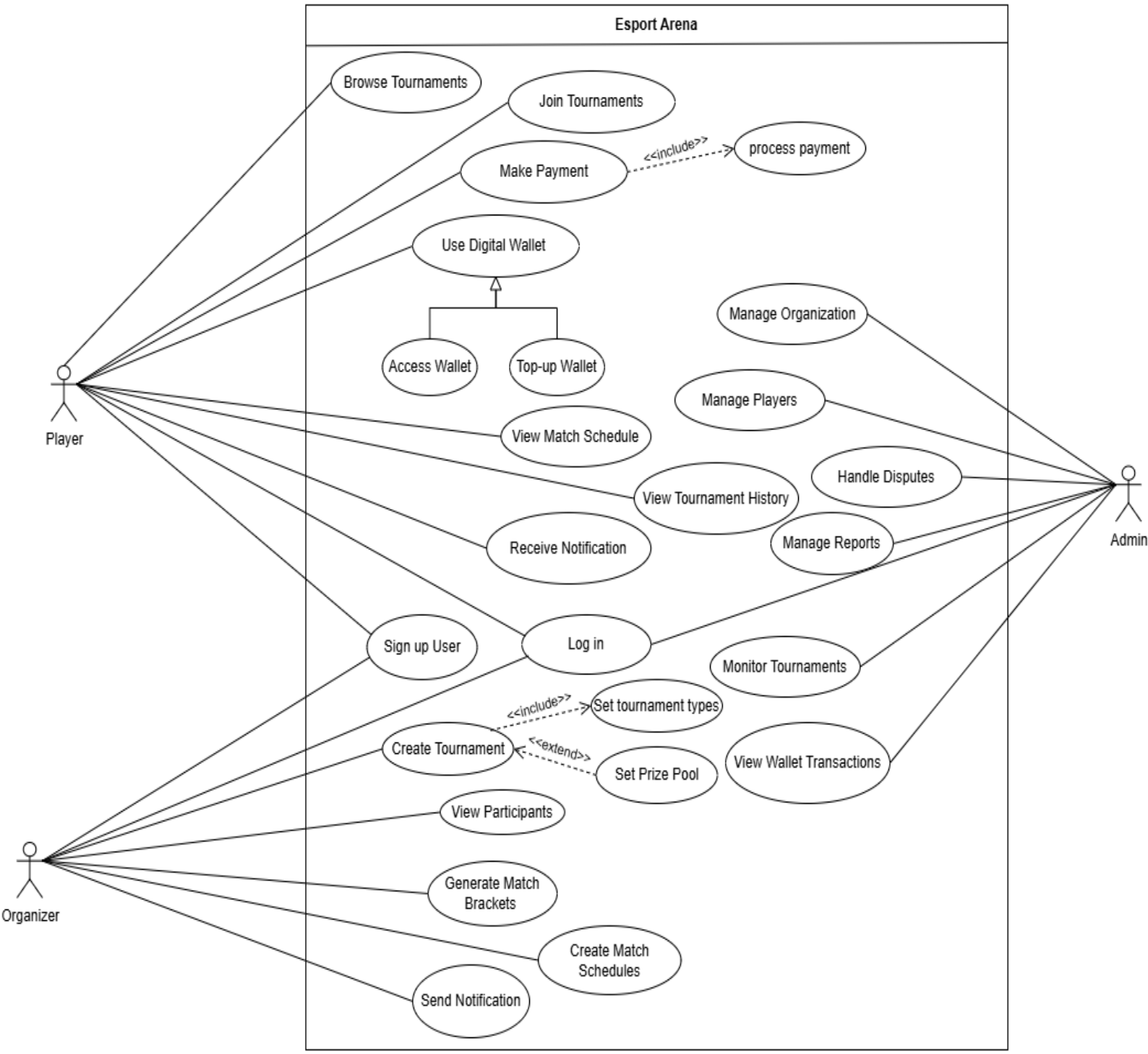


Figure 14 Use Case for Esport Arena

3.2.4 Class Diagram

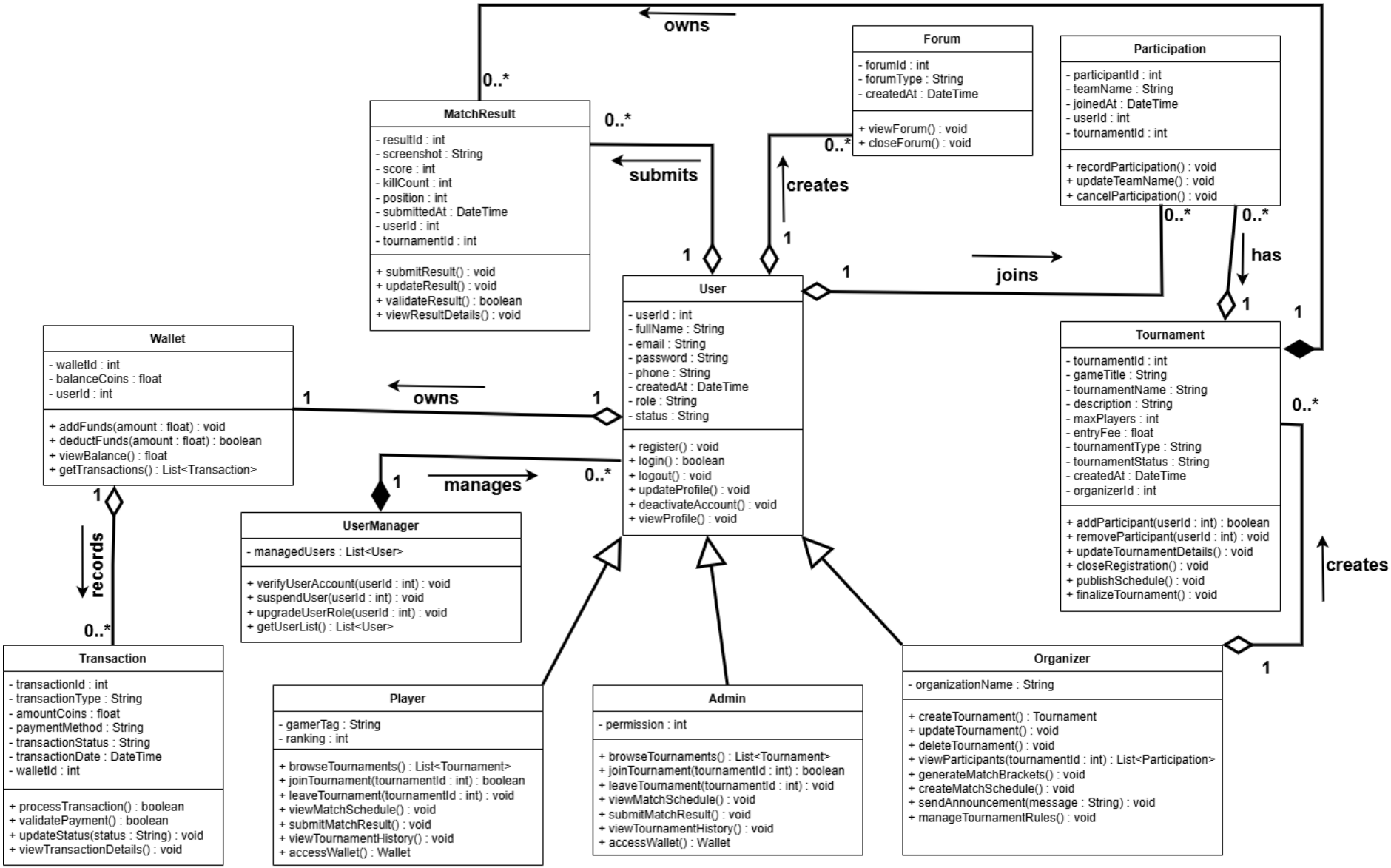


Figure 15 Class Diagram of Esport Arena

3.2.5 Milestone Chart

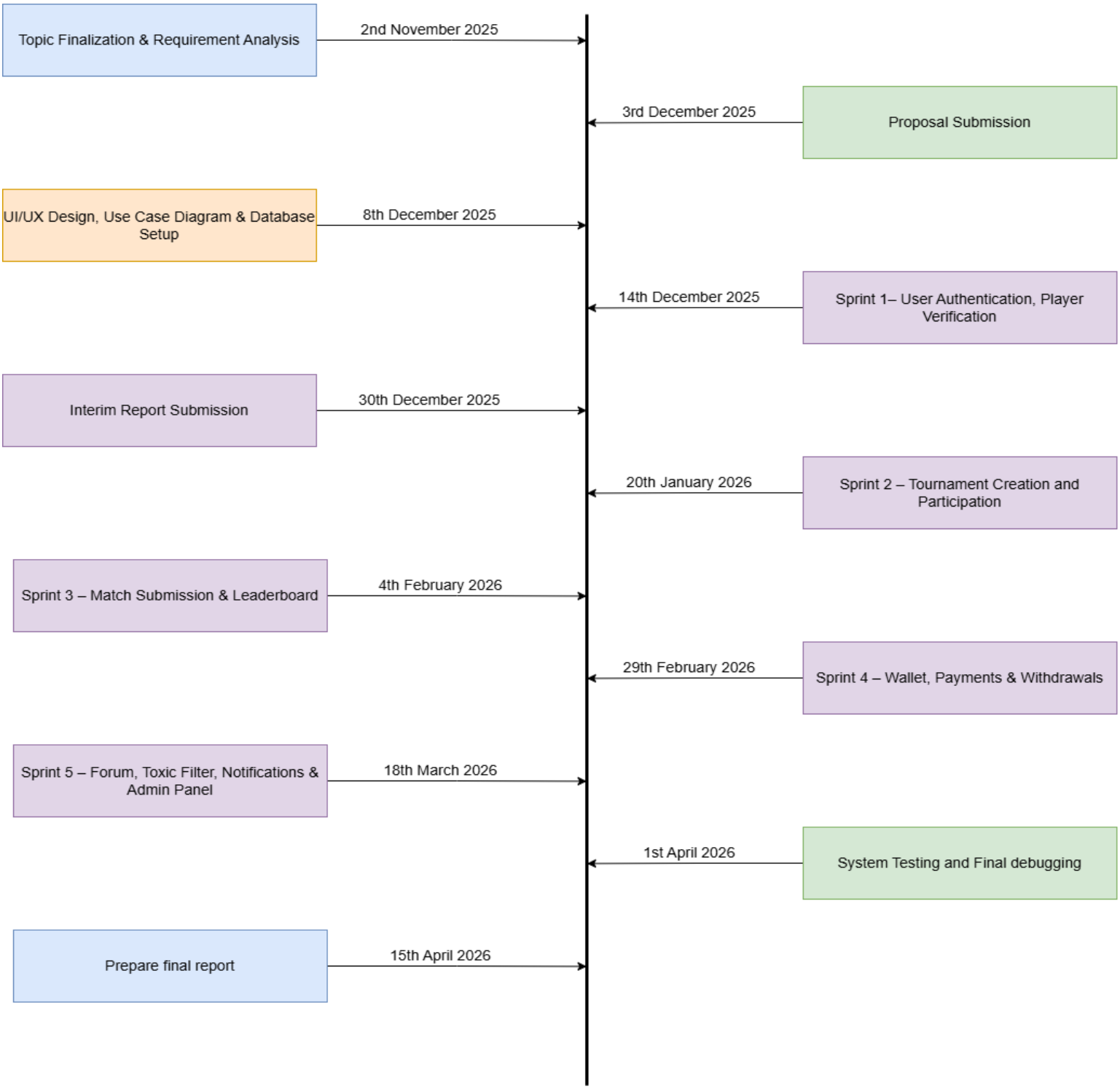


Figure 16 Milestone Chart

3.2.6 Work Breakdown Structure

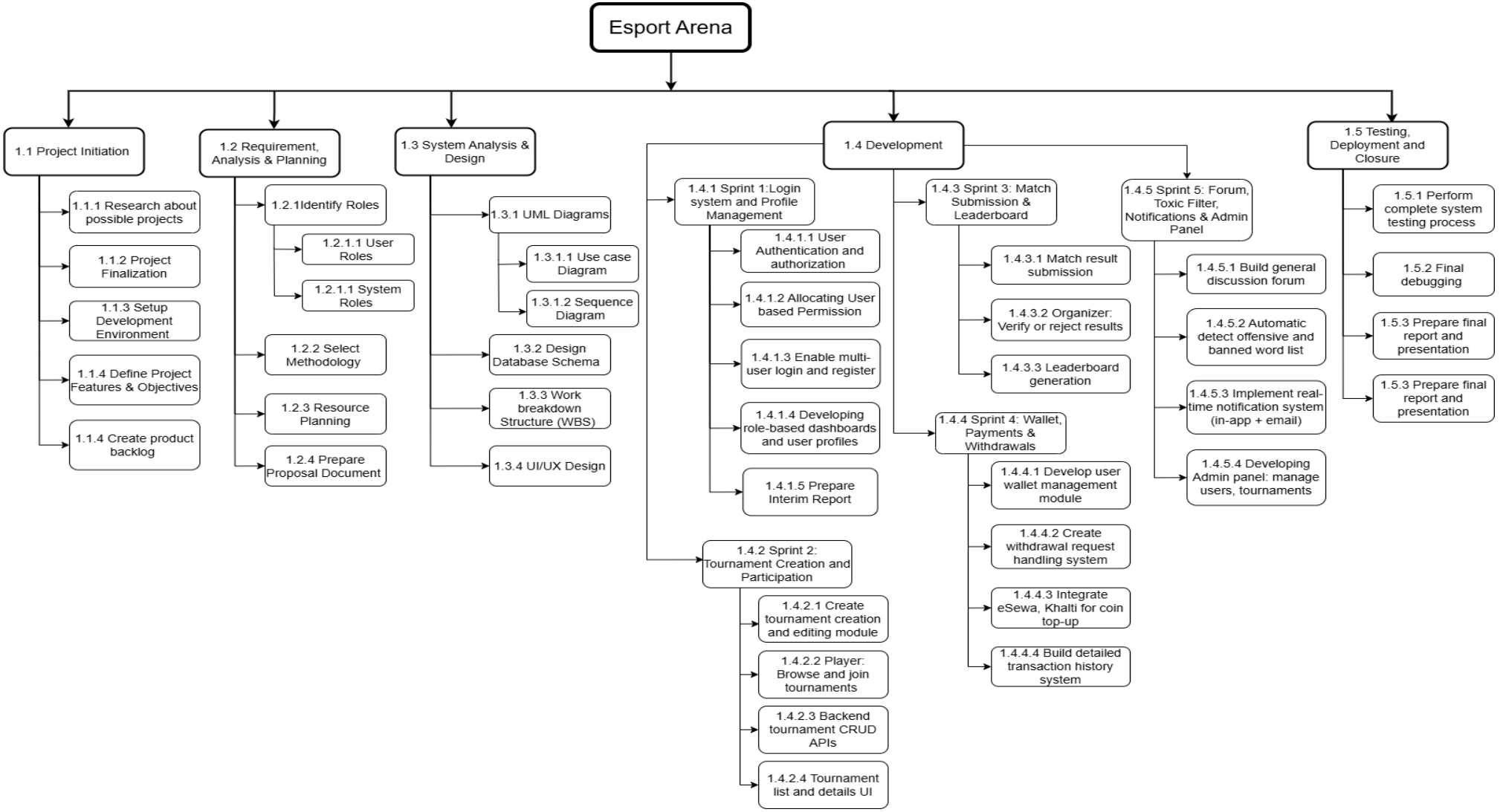


Figure 17 WBS chart of Esport Arena

3.2.7 Style Guide

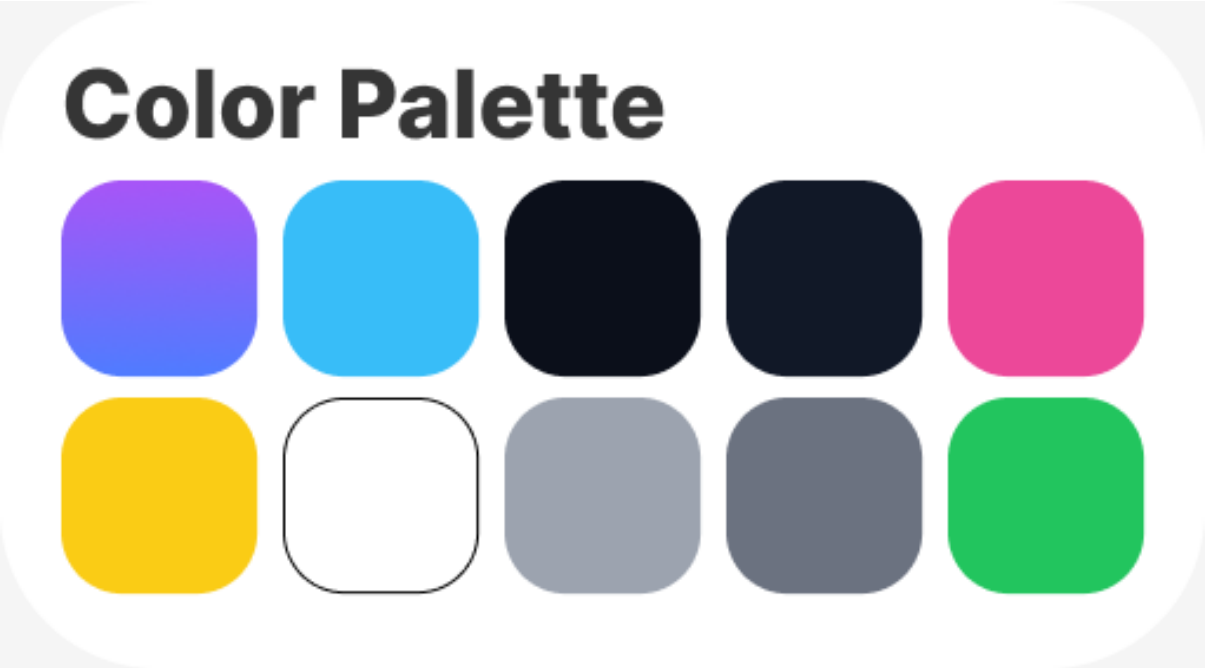


Figure 18 Collor Palette for Esport Arena

Primary Colour: Gradient Blue Purple (##A855F7 + #4F7CFF)

Secondary Colour: #38BDF8

Font: "Inter", sans-serif

Heading: 48 px

Body Content / Paragraphs: 16px

3.3 Sprint 1 (User authentication and authorization)

3.3.1 Sprint Planning

Sprint No: 1

Start Date: 07-12-2025

End Date: 30-12-2025

Sprint Goal: Introduce the basic authentication and role configuration of Esport Arena in such a way that players and hosts can create an account and sign in safely and have their first dashboards as a starting point of future tournament and wallet functionalities.

Sprint Backlog

User Story ID	User Story	Story Points	Priority
US001	As a new player, I would like to achieve registration with my personal information and verify OTP, so that I will be able to open a safe account on Esport Arena.	5	High
US002	As a player, I would like to sign in using my email and password, to be able to reach my dashboard and features.	3	High
US013	As a host, I would like to register my details and verification OTP. so that I will be able to open a safe account on Esport Arena.	5	Medium
US014	As a host, I desire to log in to my host account, so I can create and manage tournaments.	3	Medium

Table 7 Sprint 1 Backlog

3.3.2 Design

3.3.2.1 High Level Description

The high-level description is provided on the section of the appendix. (High Level Description)

3.3.2.2 Expanded Use Case Description

The expanded-level description is provided on the section of the appendix. (Expanded Level Description)

3.3.2.3 Sequence Diagram

Sign Up User

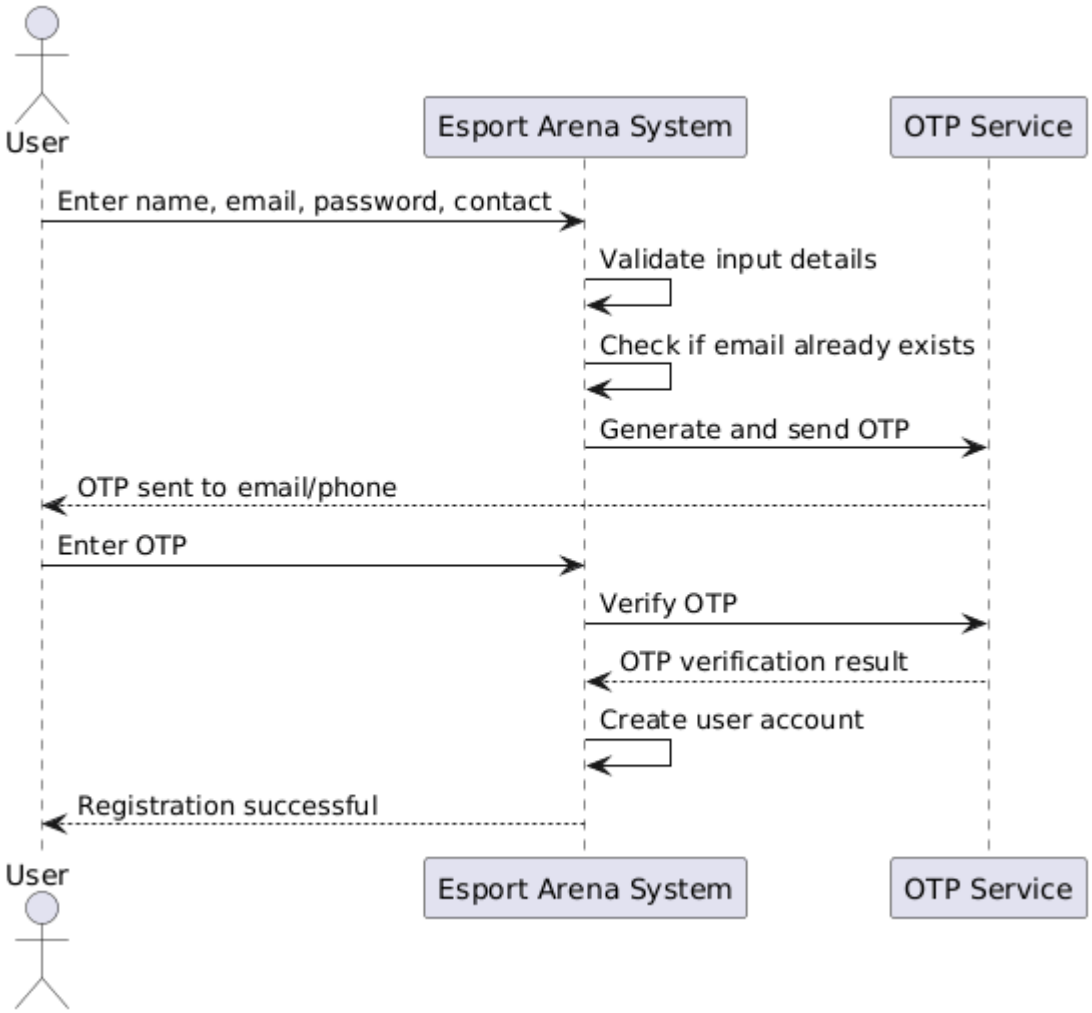


Figure 19 Sign Up User Sequence Diagram

Sign In User

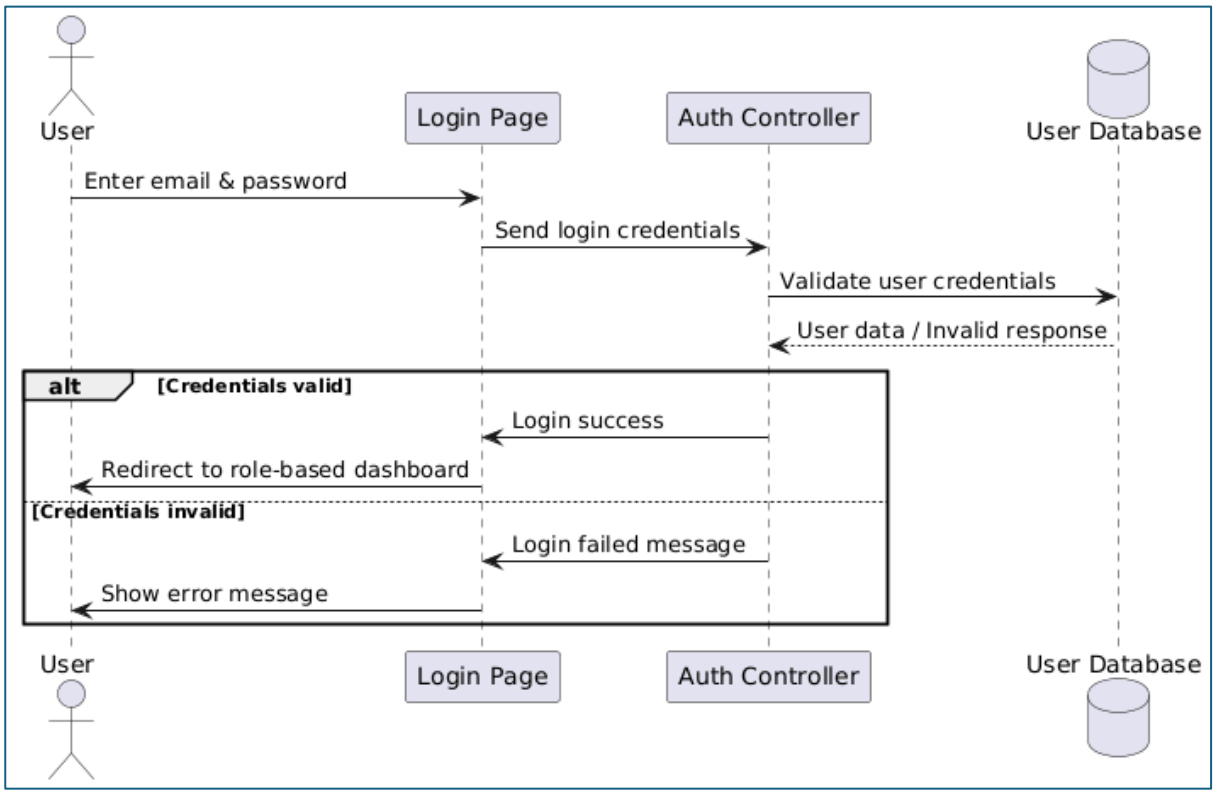


Figure 20 Sign in User Sequence Diagram

3.3.2.4 Activity Diagram

Registration Page

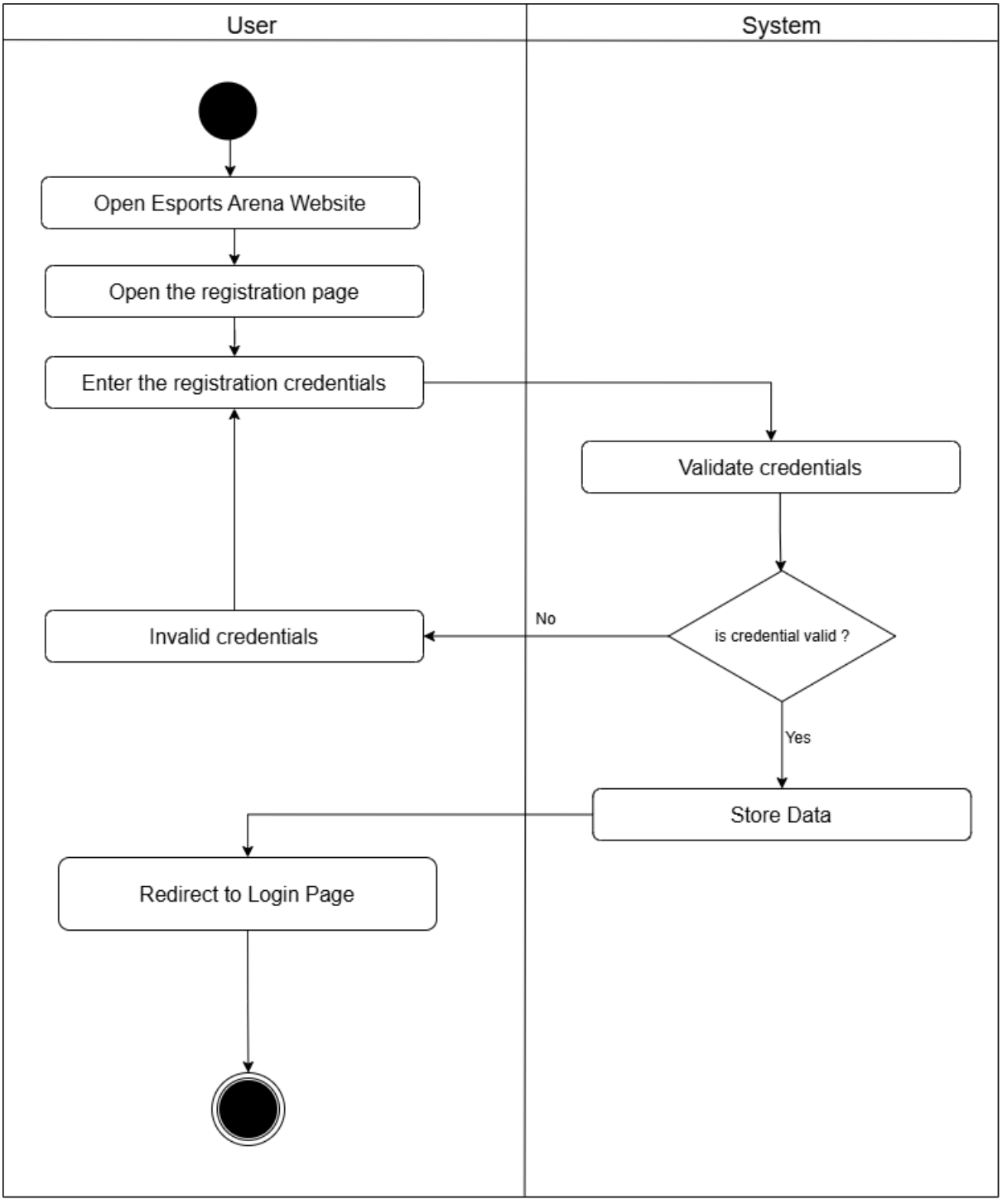


Figure 21 Activity Diagram of Sign up User

Login Page

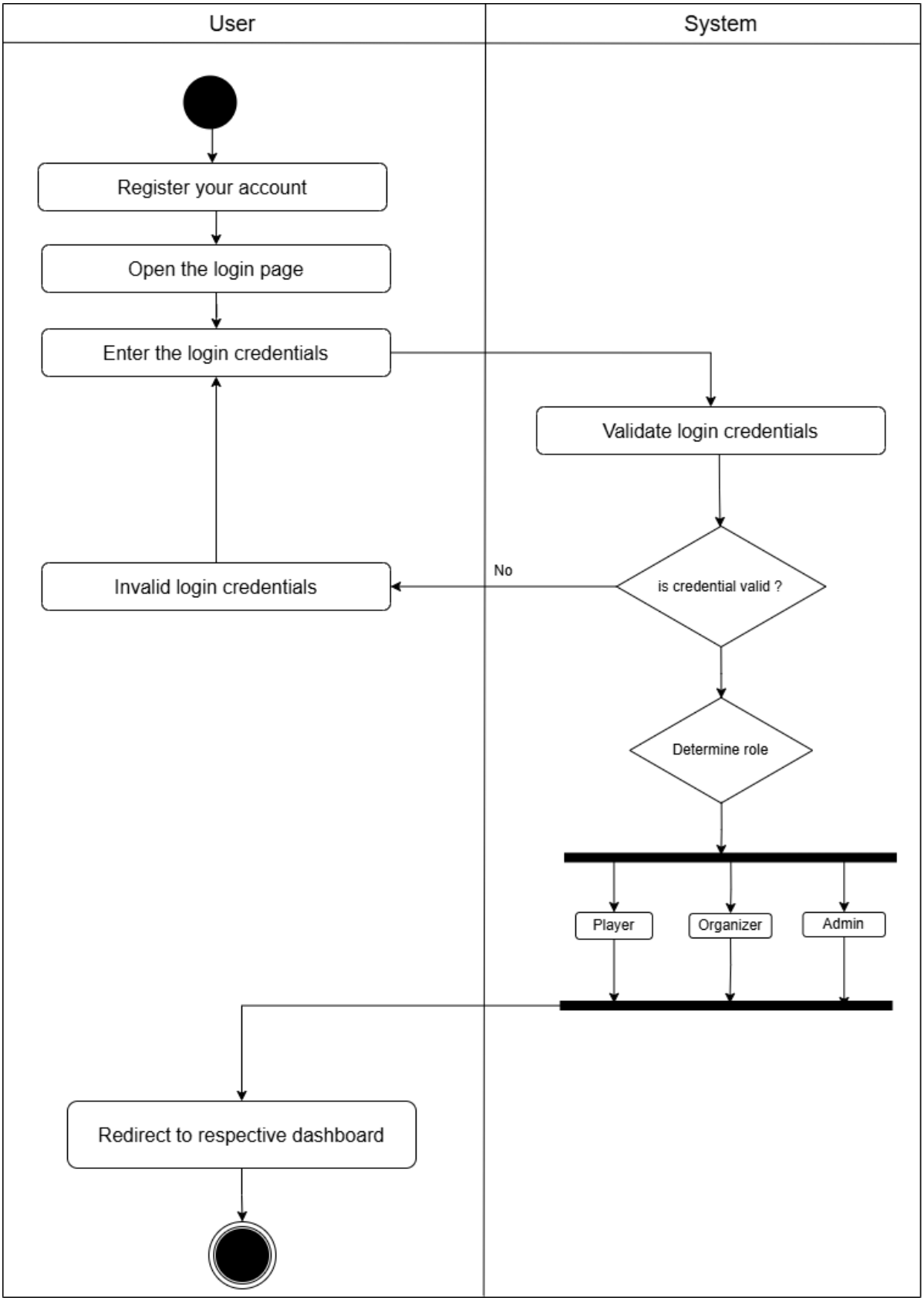


Figure 22 Activity Diagram of Log in User

3.3.2.5 Wireframes

Registration User Page

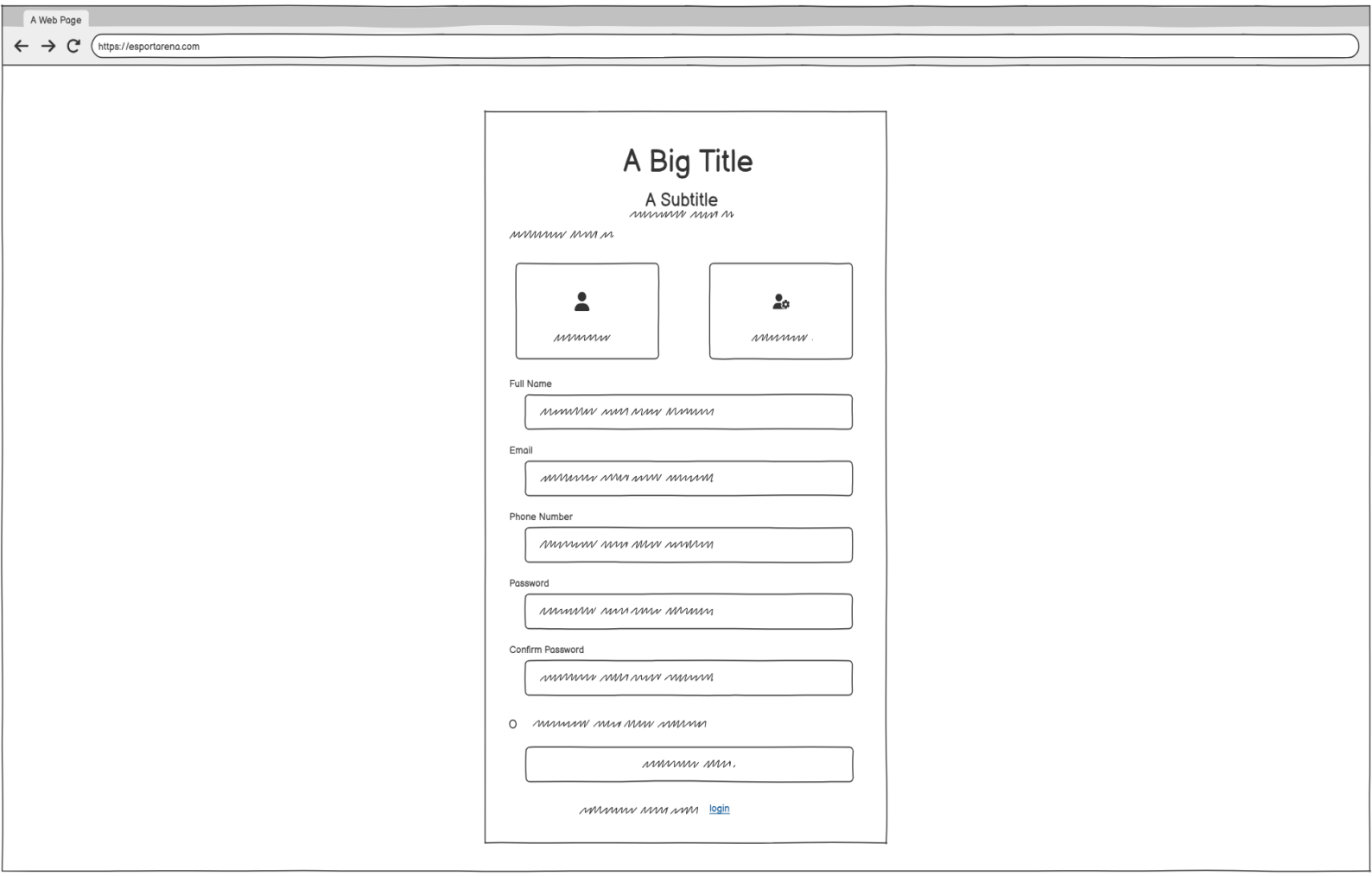


Figure 23 Wireframe of Registration User Page

Login User Page

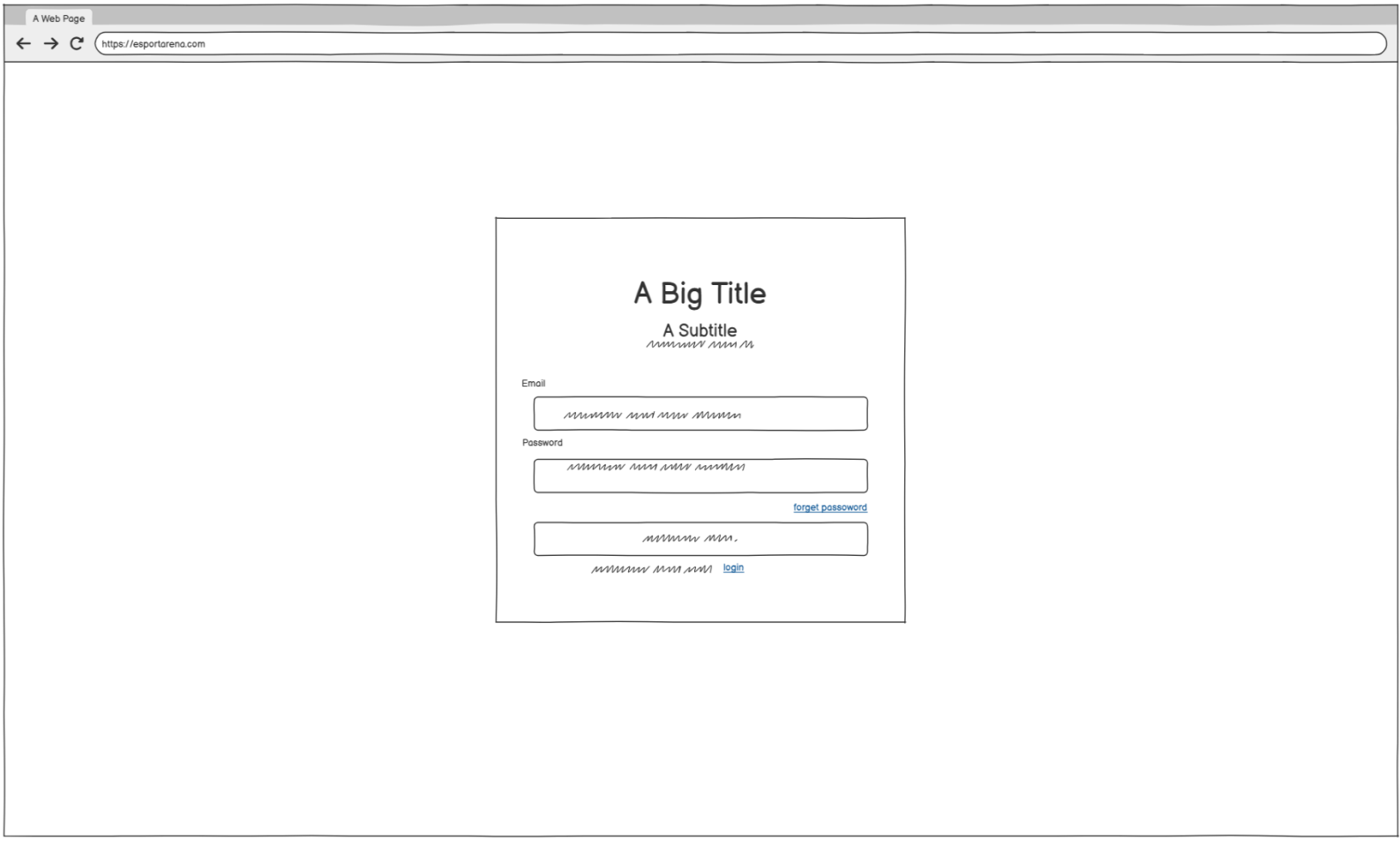


Figure 24 Wireframe of Login User Page

3.3.2.6 Mock Up

Registration Page

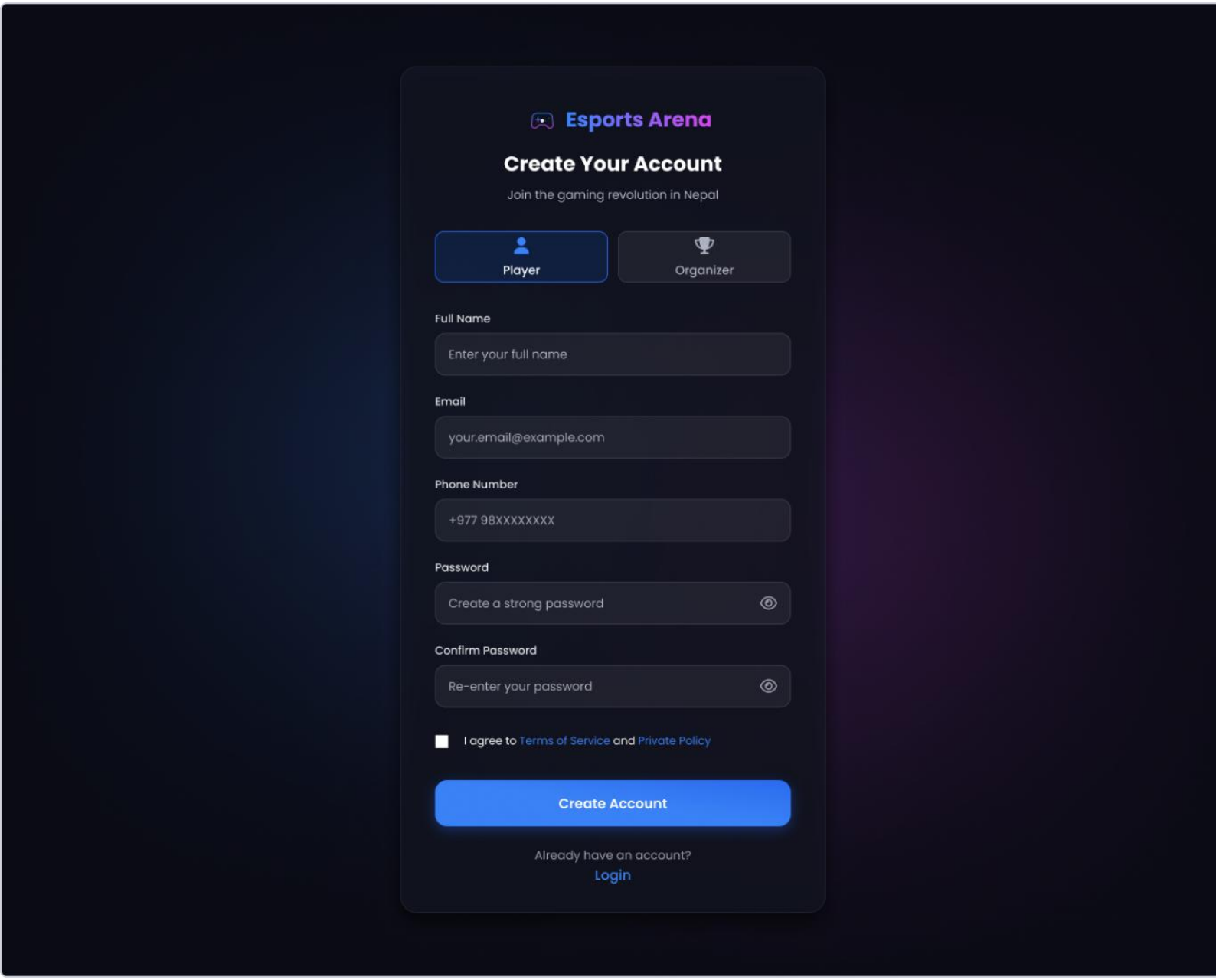


Figure 25 Design of Registration Page

Login Page

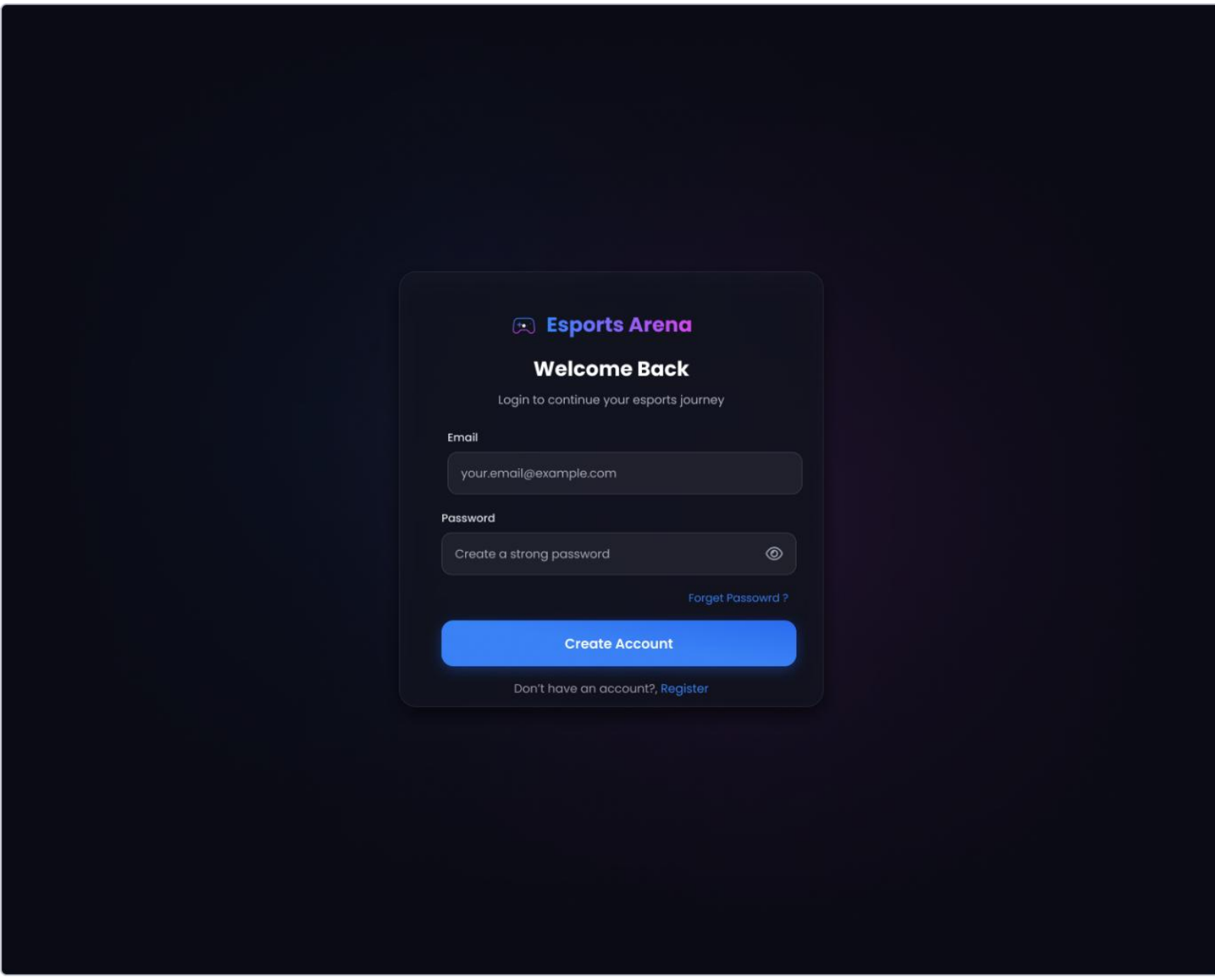


Figure 26 Design of Login Page

3.3.3 Development

3.3.3.1 Frontend

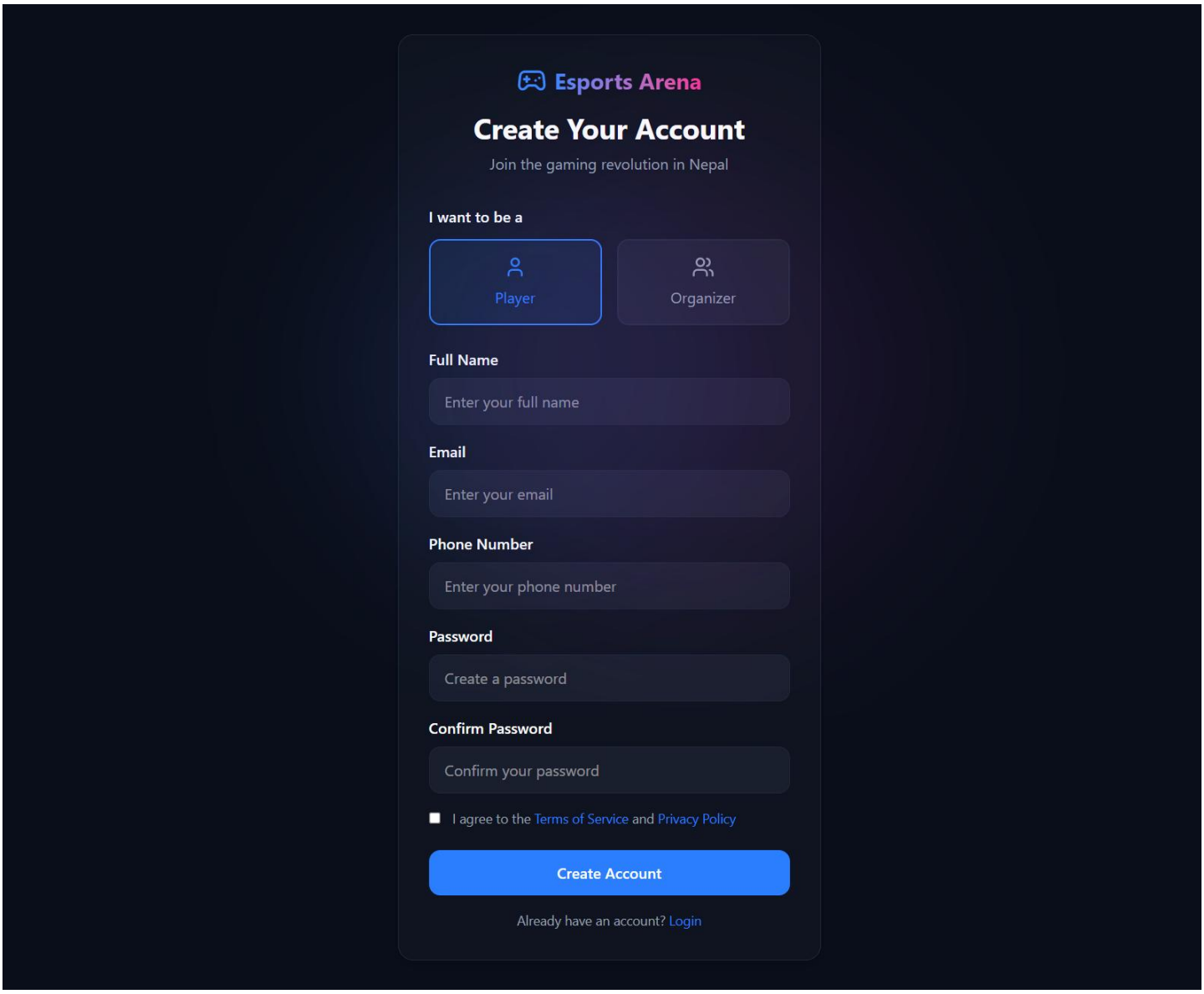


Figure 27 Register as a Player Frontend

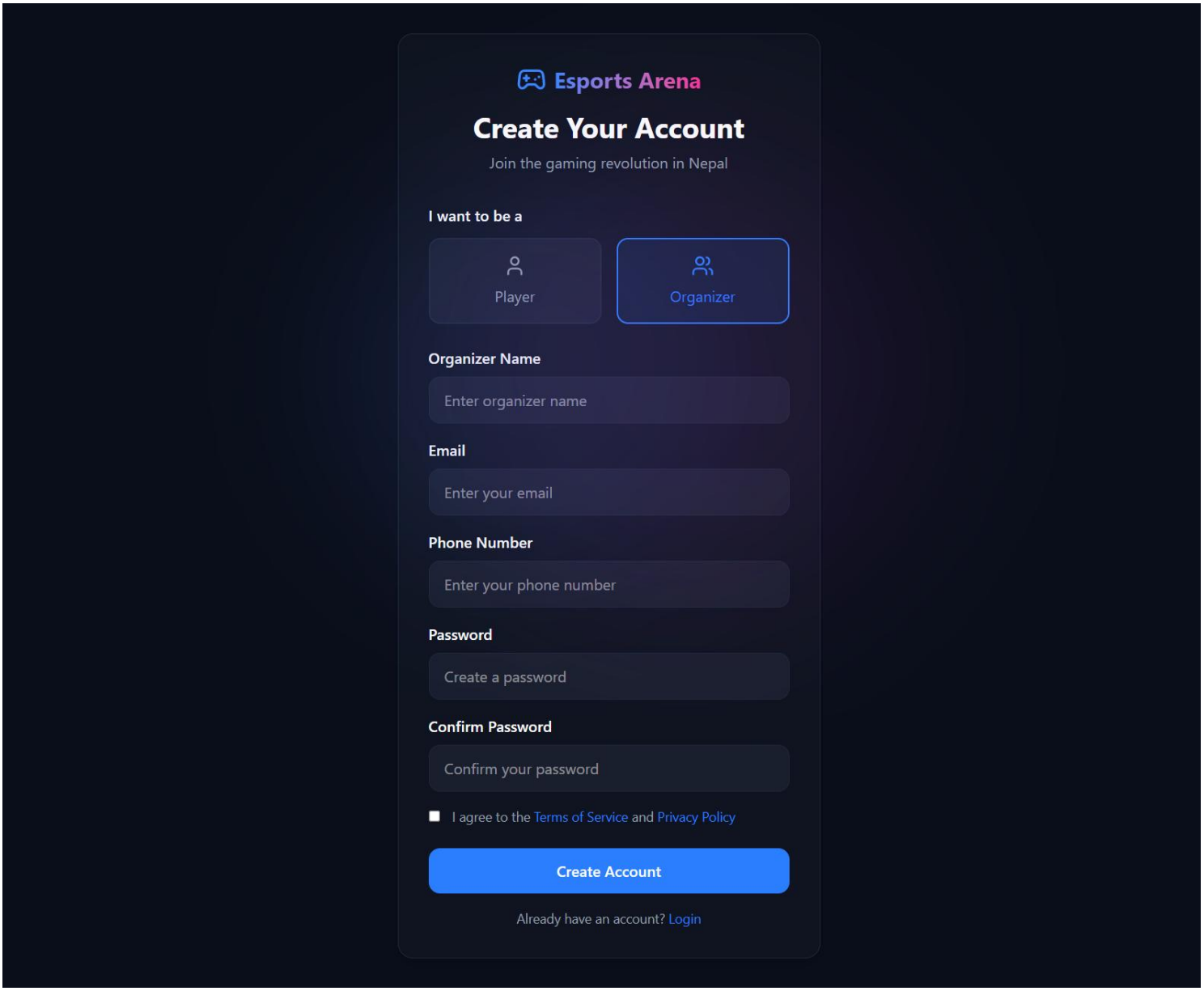


Figure 28 Register as a Organizer Frontend

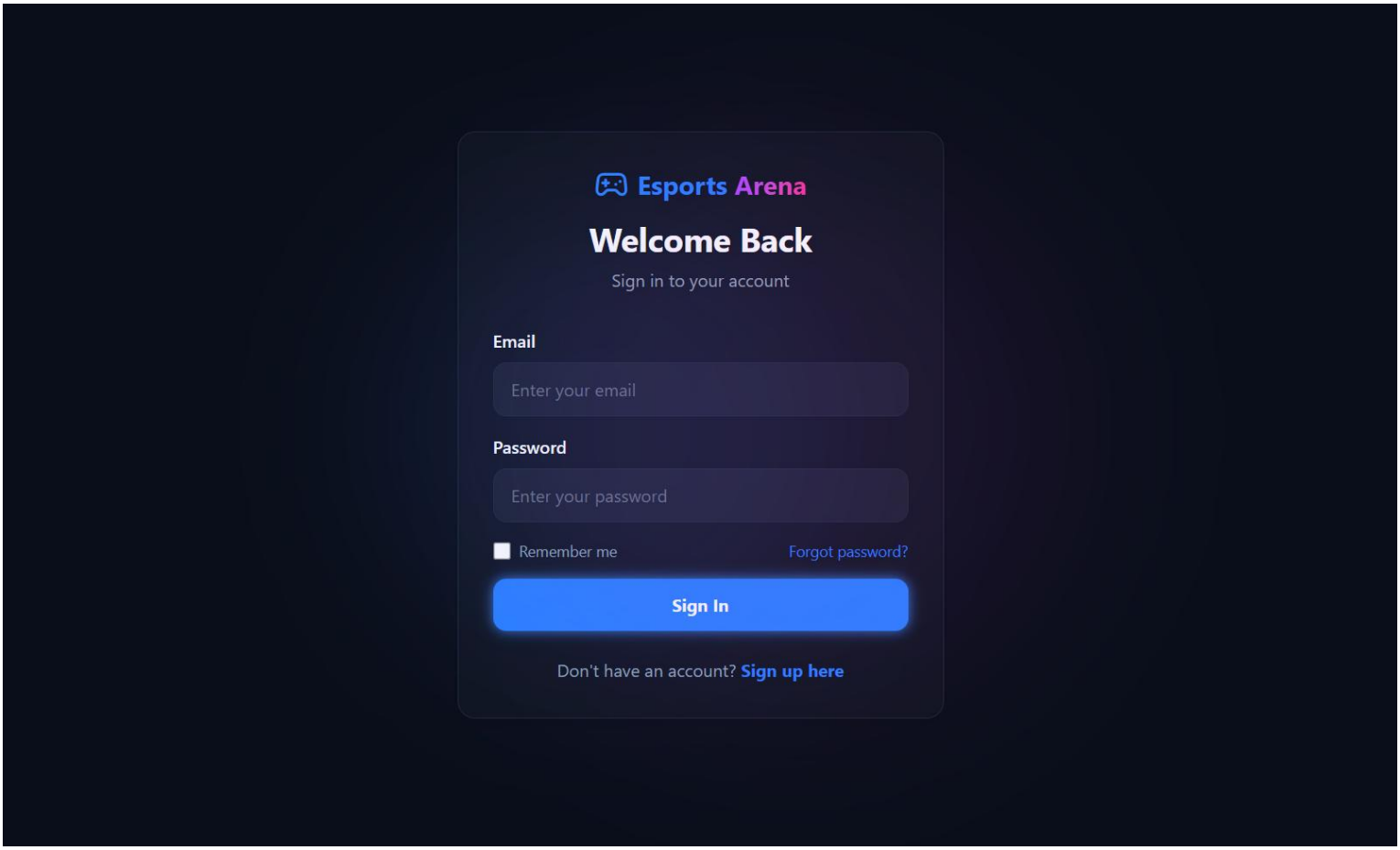


Figure 29 Login as any User Frontend

3.3.3.2 Backend

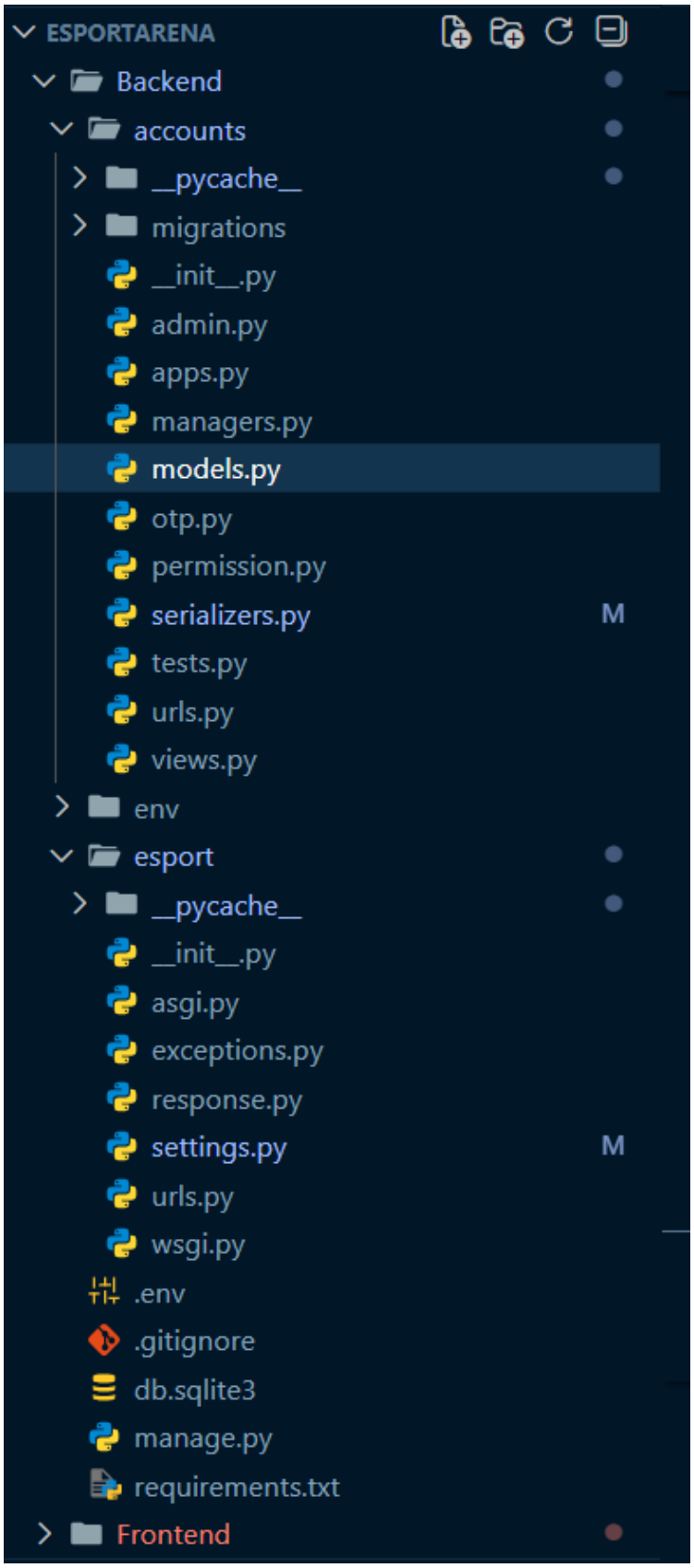


Figure 30 Backend File Path

```python
# Custom User Model
class User(AbstractUser):
    class UserRoles(models.TextChoices):
        ORGANIZER = "Organizer", _("Organizer")
        PLAYER = "Player", _("Player")
        SUPERADMIN = "SuperAdmin", _("SuperAdmin")

    username = None
    first_name = None
    last_name = None
    email = models.EmailField(_('email address'), unique=True)

    name = models.CharField(max_length=255, blank=True, null=True)
    phone_number = models.CharField(max_length=15, blank=True, null=True)
    is_verified = models.BooleanField(default=False)

    is_organizer = models.BooleanField(default=False)
    date_joined = models.DateTimeField(auto_now_add=True)
    role = models.CharField(
        choices=UserRoles.choices,
        max_length=20,
        default=UserRoles.PLAYER
    )
```

Figure 31 Backend User Model Creation

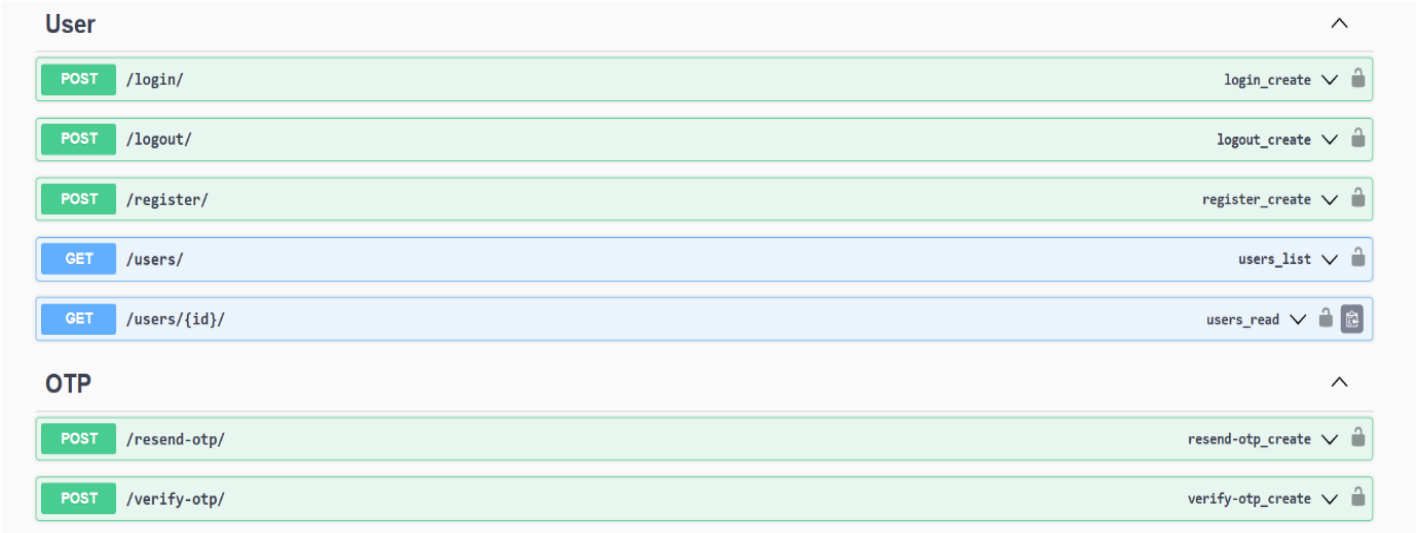


Figure 32 Backend Api Creation using Swagger

3.3.4 Sprint Review

Sprint No: 1

Start Date: 07-12-2025

End Date: 30-12-2025

Sprint Goal: Introduce the basic authentication and role configuration of Esport Arena in such a way that players and hosts can create an account and sign in safely and have their first dashboards as a starting point of future tournament and wallet functionalities.

Sprint status

1. **Status of project:** OTP based player registration (US001) and basic host verification flow (US013) are done and users can now create the accounts and authenticate using the login form.
2. **Project on track:** The sprint went as planned; though, using redirected logged-in players to a complete dashboard is on the backburner since the dashboard interface and data will be implemented in subsequent sprints.
3. **Sprints left for completion:** 4
4. **Status of project completion:** Approximately 15 18% percent of core functionality is completed, which includes authentication base and role configuration.
5. **Stories initiated:**

- US001 As a new player, I would like to achieve registration with my personal information and verify OTP, so that I will be able to open a safe account on Esport Arena.
- US002 As a player, I would like to sign in using my email and password, to be able to reach my dashboard and features.
- US013 As a host, I would like to do it with verification information, so that I can handle professional tournaments within the platform.

Planned Items

User Story ID	User Story	Story Points	Priority
US001	As a new player, I would like to achieve registration with my personal information and verify OTP, so that I will be able to open a safe account on Esport Arena.	5	High
US002	As a player, I would like to sign in using my email and password, to be able to reach my dashboard and features.	3	High
US013	As a host, I would like to register my details and verification OTP. so that I will be able to open a safe account on Esport Arena.	5	Medium
US014	As a host, I desire to log in to my host account, so I can create and manage tournaments.	3	Medium

Table 8 Planned Items for Sprint 1

Incomplete Item

User Story ID	User Story	Story Points	Priority
US002	As a player, I would like to sign in using my email and password, to be able to reach my dashboard and features.	3	High

Table 9 Incomplete Item for Sprint 1

Additional Items

No additions in this sprint were made; emphasis was on authentication and host verification.

Total Story Points planned: 16

Points achieved on the total number of stories: 13.

3.3.5 Sprint Retrospective

Sprint No: 1

Start Date: 07-12-2025

End Date: 30-12-2025

Table 10 Sprint 2 Retrospective

What Went Well	What to Improve	Action items
Introduced OTP player registration as well as basic host verification.	The redirection of log-in to full dashboard player was not completed by the end of the sprint.	Consider development of player dashboard and routing a priority in Sprint 2 to make the login flow completely closed.
Install Trello board and sprint backlog to visualize the Product Backlog, In-Progress and Done cards.	The estimation that was done on US002 was a bit optimistic; dashboard integration requires extra time.	Also set aside more hours on each authentication story which has higher points and estimate will be reviewed later.
Acquired the practical experience on the structure of the React/Django project and authentication flow.	Scarcity of time to write the unit tests of registration and login APIs.	Otherwise, add basic test cases of registration and login as different tasks during the following sprint.

3.3.6 Velocity Chart

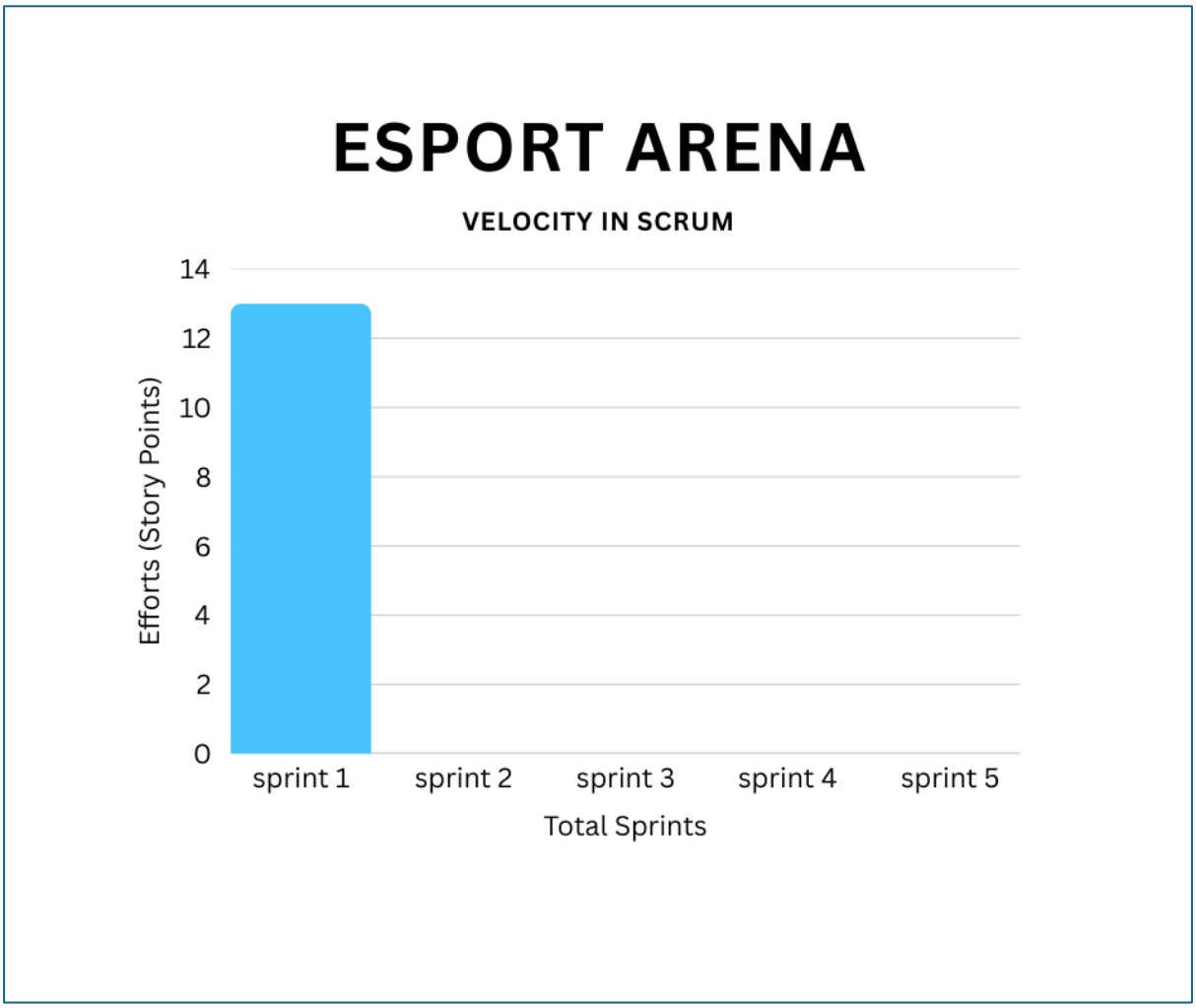


Figure 33 Velocity Chart of Esport Arena

3.3.7 Burndown Chart

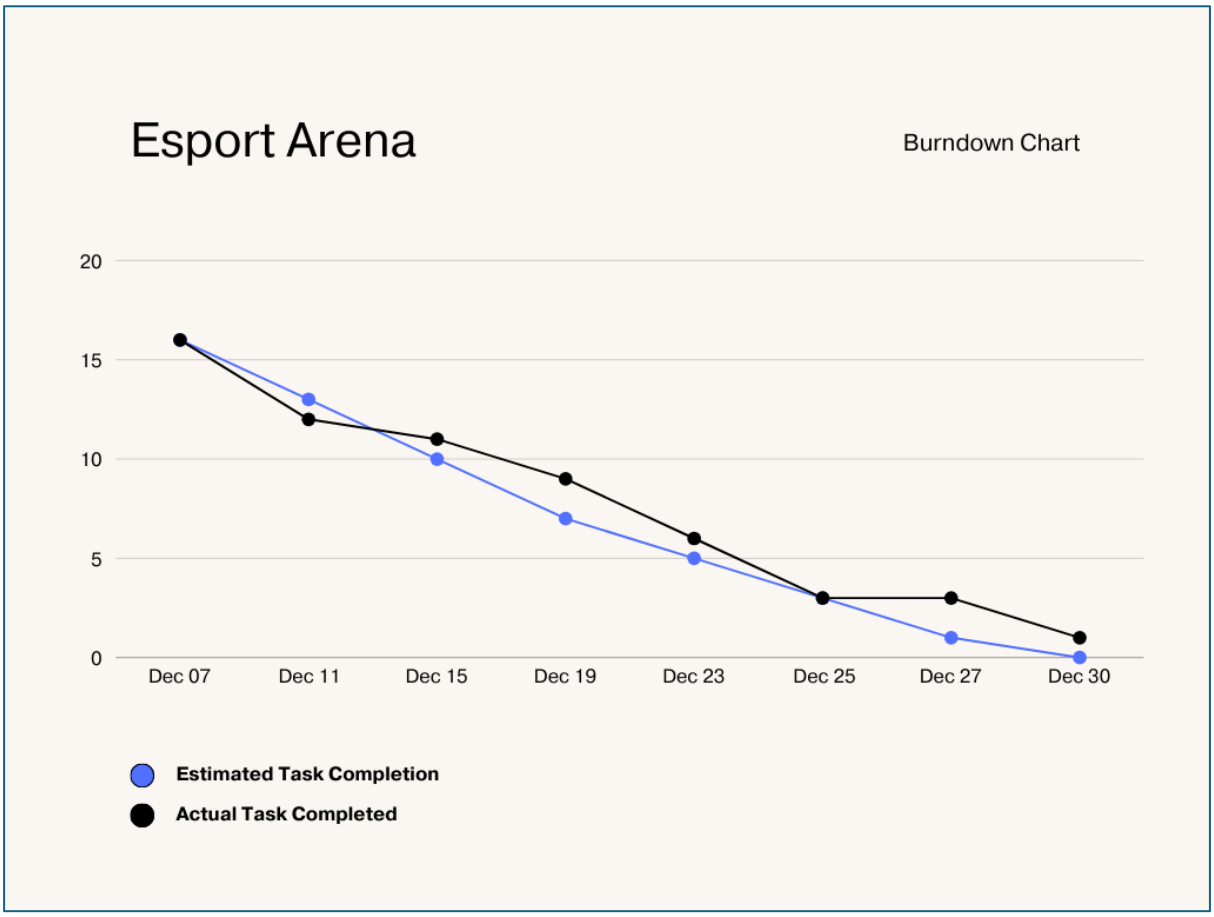


Figure 34 Burndown Chart of Esport Arena Sprint 1

4 Progress Analysis

The Esport Arena began developing by creating a clear understanding of the project scope and objectives to address problems associated with the current situation of the first pilot in managing tournaments manually, unfair results, poor communication, and unstable cash transfers in Nepali esports. These were collected via an online survey and informal interviews with the local players and small tournament hosts and then bundled into a detailed Software Requirements Specification along with a product backlog in the form of user stories complete with Fibonacci story points. Scrum as the guiding methodology with a Gantt chart and Trello board visualisation was used to plan the entire schedule in sprints and track the backlog, tasks in a sprint, and the completion status.

The following artefacts of the system design were generated; use case diagrams, class diagrams, sequence diagrams and collaboration diagrams which illustrated interactions among Players, Hosts, Admins, tournaments, wallet and notification services. The schema of the database was described as an ERD, which contained the core entities (User, Tournament, Participation, Wallet, Transaction, and Forum) whereas the registration, the login screen, dashboards, and the tournament listing were schematised with wireframes and high-fidelity mock-ups.

Sprint 1 focused on establishing the bottom layer: registration of players with an OTP verification, registration and verification of hosts and redirection logic with roles that will be ready in prospective dashboards. Simple frontend and backend project structure was implemented to allow subsequent sprints to add tournament management, wallet top-up and withdrawals, communication via forums and monitor the administration without significant refactoring. The constant testing of registration and login flow ensured that they were in sync with the SRS and the identified defect were reduced at an early stage, leaving the project in a good position to continue with other sprints on tournaments and financial features.

4.1 Gantt Chart

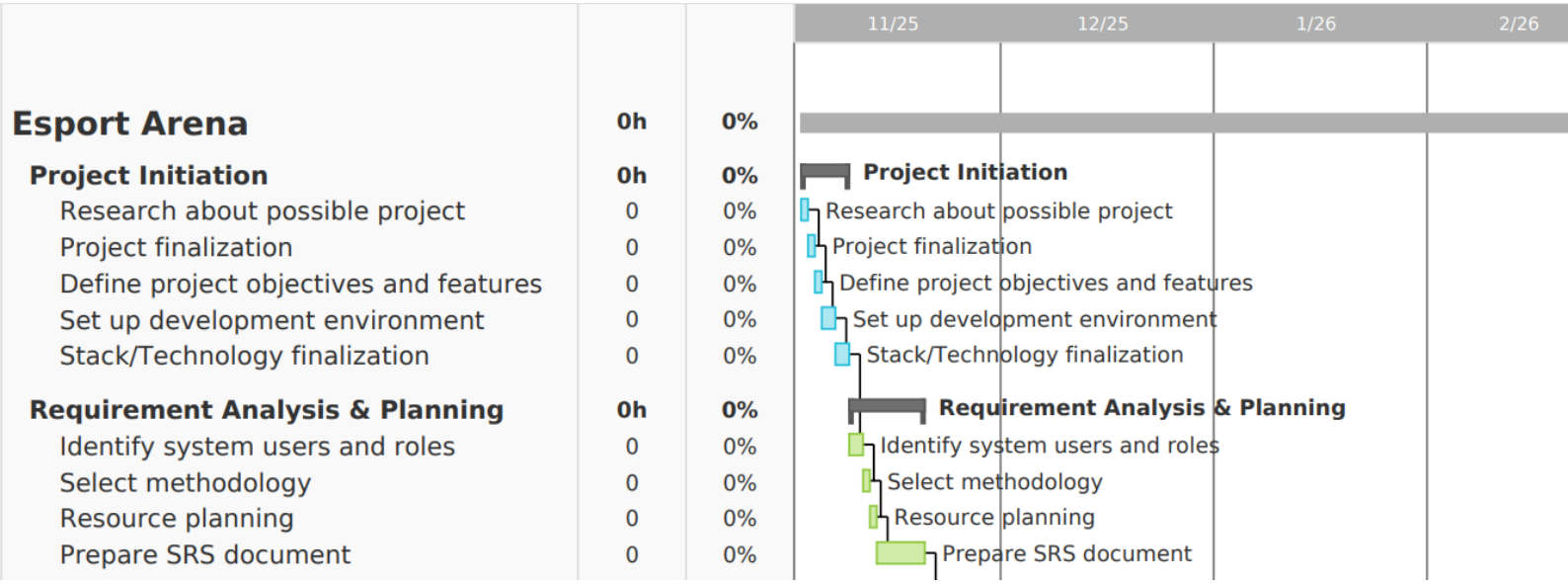


Figure 35 Esport Arena Gantt Chart 1

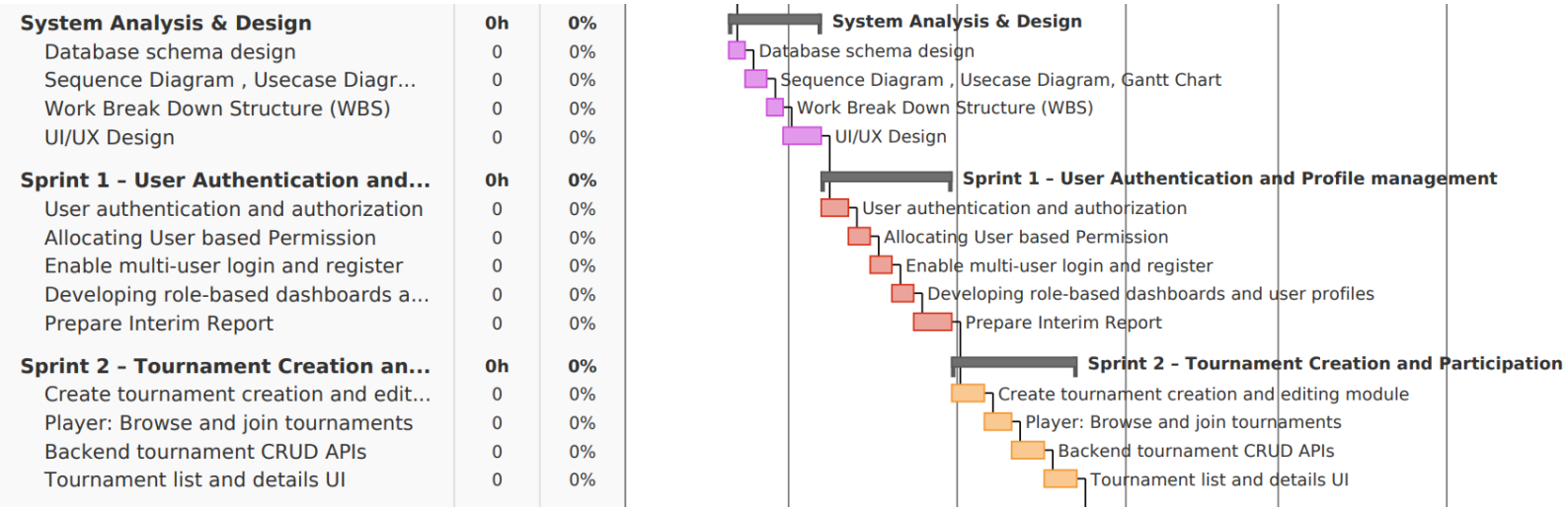


Figure 37 Esport Arena Gantt Chart 2

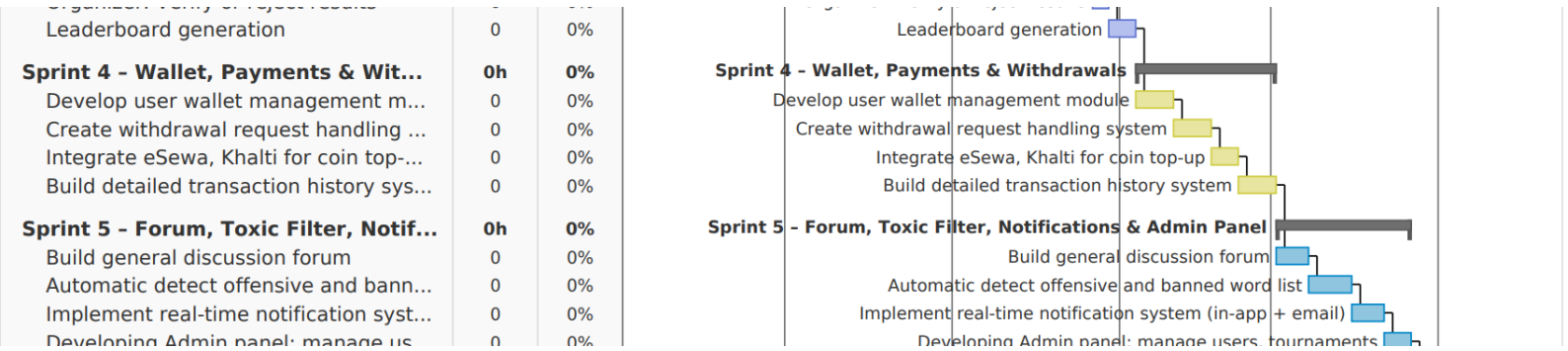


Figure 36 Esport Arena Gantt Chart 3

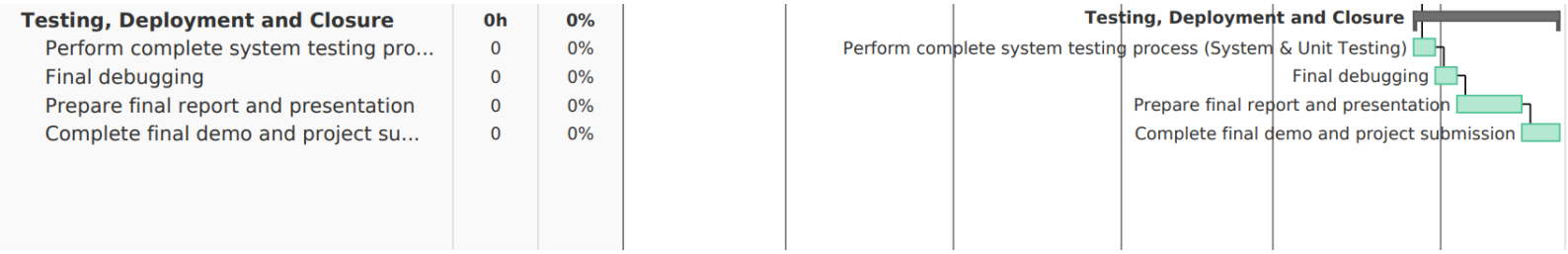


Figure 38 Esport Arena Gantt Chart 4

4.2 Progress Table

Table 11 Progress Table of Esport Arena

S.N.	Task	Status	Progress
1	Project Initiation	Completed	100%
1.1	Research about possible project	Completed	100%
1.2	Project finalization (topic and title)	Completed	100%
1.3	Define project objectives and features	Completed	100%
1.4	Set up development environment	Completed	100%
1.5	Stack/technology finalization	Completed	100%
2	Requirement Analysis and Planning	Completed	100%
2.1	Identify system users and roles (Player, Organizer, Admin)	Completed	100%
2.2	Select methodology (Scrum)	Completed	100%
2.3	Resource planning (time, tools, hardware/software)	Completed	100%
2.4	Prepare SRS document	Completed	100%
3	System Analysis and Design	Completed	100%
3.1	Database schema design	Completed	100%
3.2	Sequence diagram, use-case diagram, Gantt chart	Completed	100%
3.3	Work Breakdown Structure (WBS)	Completed	100%
3.4	UI/UX design and wireframes	Completed	100%

4	Sprint 1 – User authentication and profile management	Partially Completed	60%
4.1	User authentication and authorization (register/login with roles)	Completed	100%
4.2	Allocating user-based permission	Completed	100%
4.3	Enable multi-user login and register	Completed	100%
4.4	Developing role-based dashboards and basic user profiles	Partially Completed	40%
4.5	Prepare interim report	Completed	100%
5	Sprint 2 – Tournament creation and participation	Not Started	0%
5.1	Create tournament creation and editing module	Not Started	0%
5.2	Player browse and join tournaments	Not Started	0%
5.3	Backend tournament CRUD APIs	Not Started	0%
5.4	Tournament list and details UI	Not Started	0%
6	Sprint 3 – Match submission and leaderboard	Not Started	0%
6.1	Match result submission	Not Started	0%
6.2	Organizer verify or reject results	Not Started	0%
6.3	Leaderboard generation	Not Started	0%
7	Sprint 4 – Wallet, payments and withdrawals	Not Started	0%
7.1	Develop user wallet management module	Not Started	0%
7.2	Create withdrawal request handling system	Not Started	0%
7.3	Integrate eSewa, Khalti for coin top-up	Not Started	0%
7.4	Build detailed transaction history system	Not Started	0%
8	Sprint 5 – Forum, toxic filter, notifications and admin panel	Not Started	0%

8.1	Build general discussion forum and tournament-specific forums	Not Started	0%
8.2	Automatic detection of offensive and banned words (toxicity filter)	Not Started	0%
8.3	Implement real-time notification system	Not Started	0%
8.4	Develop admin panel to manage users, tournaments, and withdrawals	Not Started	0%
9	Testing, deployment and closure	Not Started	0%
9.1	Perform complete system testing process	Not Started	0%
9.2	Final debugging and performance/security tuning	Not Started	0%
9.3	Prepare final report and presentation	Not Started	0%
9.4	Complete final demo and project submission	Not Started	0%

4.3 Analysis

The Esport Arena project development is fairly in compliance with the phases outlined in the Gantt chart and roadmap (Scrum-driven).

The preliminarily stages are already accomplished. The project starting activities such as selection of the topic, scope definition, feasibility study, resource scheduling, selection of the technology stack are completed. The requirement analysis and planning are also finished with the help of a detailed SRS, the clear understanding of the user roles, and reasonable selection of the methodology of Scrum. System analysis and design have been finalized i.e. database schema design, ERD, use-case and sequence diagrams, WBS, milestone chart, revised Gantt chart, and UI/UX wireframes. Conceptual clarity and Architectural planning are good at this point and chances of significant requirements changes are minimal.

View Rest Analysis

5 Further Work

Authentication and Role-Based Dashboards Completion: The ongoing activity will entail finishing up with Sprint 1 in such a way that authentication and role control lays a strong foundation in subsequent modules. The current registration and login processes will be improved by means of a more potent validation, error management, and safe password management. Role-based access control will then be completed in such a way that the Players, the Organizers and the Admins will automatically go to their respective dashboards once they are fully authenticated. The dashboards will be customized to have only features and navigation options shown of the user and their role, thus making them clear and secure and have a similar user experience across the system.

Other Further Work

6 Conclusion

The Esport Arena project has gone through the process of developing a rough idea to a concrete and partially executed web application with the aim of consolidating fragmented, manual and usually unreliable management of community esports tournaments in Nepal. The basic lifecycle activities like project initiation, requirement analysis, feasibility study and the selection of methodology are complete. Comprehensive SRS, ERD, UML diagrams, WBS, updated Gantt chart, and UI wireframes are given, which are a clear blueprint of the functional and non-functional expectations of the system so that future development undertakings will be informed of a stable and consistent design.

Overall Conclusion

7 References

Game.tv. (2024) *Game.tv* [Online]. Available from: https://www.game.tv/de [Accessed 19 December 2025].

InGame.gg. (2024) *InGame.gg* [Online]. Available from: https://ingame.gg/ [Accessed 19 December 2025].

intersog. (2020) *Why Scrum Is Important for Game Development* [Online]. Available from: https://intersog.com/blog/development/adopting-scrum-team-best-practices-for-video-game-development/ [Accessed 01 December 2025].

Kirvan, P. (2024) *What is a Waterfall model? Definition and guide* [Online]. Available from: https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model [Accessed 10 December 2025].

Kumar, S. (2023) *SDLC - Iterative Model* [Online]. Available from: https://www.scaler.com/topics/software-engineering/iterative-model-in-software-engineering/ [Accessed 10 December 2025].

Mittal, K. (2023) *How to Distribute eSports Prize Payments* [Online]. Available from: https://usedots.com/blog/how-to-distribute-esports-prize-payments/ [Accessed 18 December 2025].

play.toornament. (2025) [Online]. Available from: https://play.toornament.com/en_US/ [Accessed 19 December 2025].

Rajkumar. (2025) *Software Prototyping Model in Software Engineering* [Online]. Available from: https://www.softwaretestingmaterial.com/software-prototyping-model/ [Accessed 10 December 2025].

Sachan, D. (2025) *Incremental Model in Software Engineering* [Online]. Available from: https://www.scaler.com/topics/incremental-model-in-software-engineering/ [Accessed 10 December 2025].

scrum.org. (2025) *What is Scrum?* [Online]. Available from: http://scrum.org/learning-series/what-is-scrum/ [Accessed 28 November 2025].

Smith, R. (2023) *Five common challenges when organizing online e-sports tournaments* [Online]. Available from: https://www.reedsmith.com/articles/five-common-challenges-when-organizing-online-esports-tournaments/ [Accessed 18 December 2025].

Talreja, A. (2025) *Spiral Model: Definition, Phases, Advantages & Disadvantages* [Online]. Available from: https://teachingagile.com/sdlc/models/spiral [Accessed 10 December 2025].

TournaOP. (2025) *TournaOP* [Online]. Available from: https://tournaop.com/ [Accessed 19 December 2025].

xtrembattle. (2022) *XtremBattle is an Ultimate Solution to all your eSports Games.* [Online]. Available from: https://xtrembattle.com.np/page/index/about-us [Accessed 19 December 2025].

8 Appendix

8.1 Background

8.1.1 Trends in the Field

- **Increased esports activity:** More players and teams are entering online tournaments, and there is a high demand to have reliable digital systems that would handle the event at large scale.
- **Smart tournament:** Web-based tournament management systems are also taking over manual tracking systems, because it offers more control over the registration process, brackets, and match outcomes.
- **Live capabilities:** Players and organizers would also want schedules, match results, and announcements delivered in real-time so that they can coordinate well in case of any event.
- **Playing fair and safety:** It is being increasingly necessary to have secure systems and simple rules, as well as active moderation to ensure trust and professionalism in competitive gaming.

8.1.2 Benefits of Esport Arena

- **Ease of use:** The site will enable all the tournament-related activities to be handled in just one convenient interface without the use of many applications.
- **Transparency:** Result processing and automated brackets ensure minimized human error in all tournaments and make them transparent and trustworthy by everyone involved.
- **Financial design:** This is a scheduled virtual coin wallet integrated with the local gateways like eSewa and Khalti to allow safe and trackable transaction of entry fees and prizes paid out.
- **More responsible environment:** Toxicity filtering and role-based access control AI will be used to promote thoughtful communication and regulated interactions.

8.1.3 Challenges

Prompt coordination: To make sure all users report the schedule modifications and on essential announcements promptly, the notification system should be functioning well.

Competitive integrity: It is essential to provide fair competition with the help of powerful rules, regular verification of results, and avoiding cheating or abuse.

User adoption: There are users who might be used to non-structured tools, and this is where guidance must be provided to them to switch to a structured platform.

Technical complexity: Payment, moderation and real time functions need to be securely integrated and be planned to ensure a well-developed system.

8.1.4 Future Possibilities

Enhanced tournament services: More games, modes and tournament formats can be supported to draw in more esports talent.

High-tech AI applications: The better moderation, matching, and performance analytics can be to improve the quality of tournaments and engagement with the users.

Greater security: This will boost authentication, encryption and fighting fraud to improve confidence in the platform.

Improved community: Forums, user profiles, and social utility can be useful in growing esports communities in the long run.

Esport Arena is created to simplify the process of participation in and organization of esports tournaments and make it more transparent and secure. It has scheduled functions such as automated bracket, structured payments, and other toxicity control and real time communication, which is effective in terms of providing a practical solution to players, organizers and admins. Overcoming such issues as fairness, trust, and coordination, introducing advanced features step by step, Esport Arena will be able to become a dominating force of competitive gaming in its target area.

Go Back

8.2 Product Backlog

Go Back

Table 12 Product Backlog for Esport Arena

User Story ID	User Story	Sprint	Story Points	Priority
US001	As a new player, I would like to achieve registration with my personal information and verify OTP, so that I will be able to open a safe account on Esport Arena.	1	5	High
US002	As a player, I would like to sign in using my email and password, to be able to reach my dashboard and features.	1	3	High
US003	As a player, I would like to observe all the tournaments that are available with details, that I may choose the one with which to associate myself.	2	5	Medium
US004	As a player, I would like to filter tournaments by type of game to play, entrance fee or date, in order that I might have tournaments that are to my liking.	3	3	Medium
US005	As a player, I would like to participate in tournaments with my coins, so that I will be able to play competitive matches.	2	8	High
US006	As a player, I would like to refill virtual coins with the help of eSewa or Khalti, to be able to participate in tournaments with ease.	2	8	High
US007	As a player,	3	5	Medium

	I would prefer to post screen shots of matches, to ensure that my performance is checked by hosts.			
US008	As a player, I would like to get announcements regarding the time of matches and updates, to such an extent that I keep abreast of the tournament.	4	5	Medium
US009	As a player, I should speak in the tournament lobby, so that I will be able to interact with teammates and the host.	3	5	Medium
US010	As a player, I would like the toxic messages to be blocked automatically, with a view of making the chat environment conducive and amiable.	4	8	High
US011	As a player, I would like to view the entire history of my transactions, so that I can see top-ups, fees, and winnings.	3	5	Medium
US012	As a player, I would like to demand remittances of my coins earned, that I may go safe redeem them into money.	4	8	High
US013	As a host, I would like to do it with verification information. so that I can handle professional tournaments within the platform.	1	5	High
US014	As a host, I desire to log in to my host account, so I can create and manage tournaments.	1	3	High
US015	As a host,	2	8	High

	I would like to make tournaments with rules, fees and schedule as well as prizes, that I may be able to plan events successfully.			
US016	As a host, I would like to change the tournament settings, to be able to make changes in schedules or correct errors.	2	5	Medium
US017	As a host, I would wish that the system was to automatically create brackets, that tournaments might be successful and just.	3	8	High
US018	As a host, I would like to check the screenshots uploaded by the players, to the end that I can check the results of matches.	3	5	High
US019	As a host, I would like to update developments and ties the bracket, to have the tournament proceed in a proper way.	3	5	High
US020	As a host, I wish to cause announcements and significant updates; Players can get the important information instantly.	4	3	Medium
US021	As a host, I would like to erase bad or rachitic messages, to keep the tournament chat professional.	4	5	Medium
US022	As an admin, I desire to be able to control and observe all the actions on Esport Arena.	1	5	High
US023	As an admin, I would like to see, shape, or deactivate user accounts, so that I will be able to secure platforms.	2	5	Medium

8.3 Pre Survey

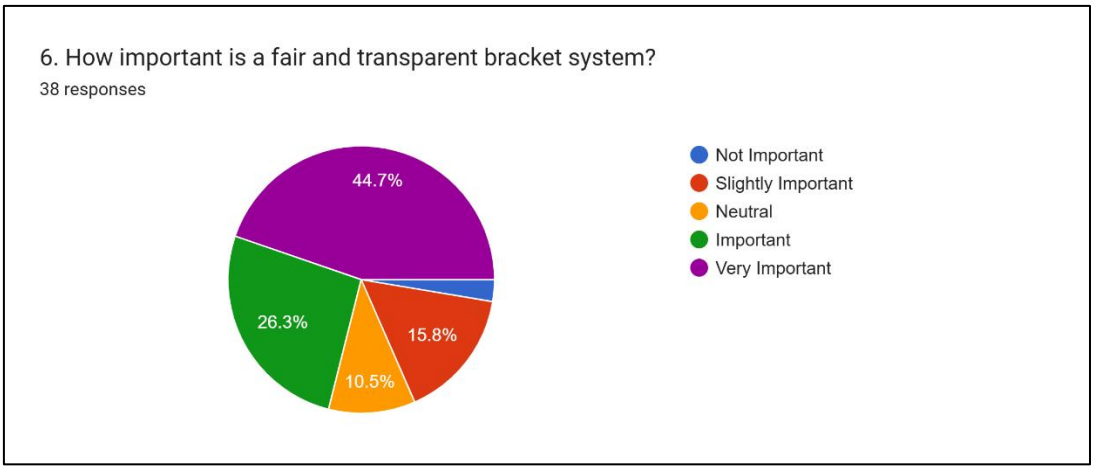


Figure 39 Pre Survey Question 6

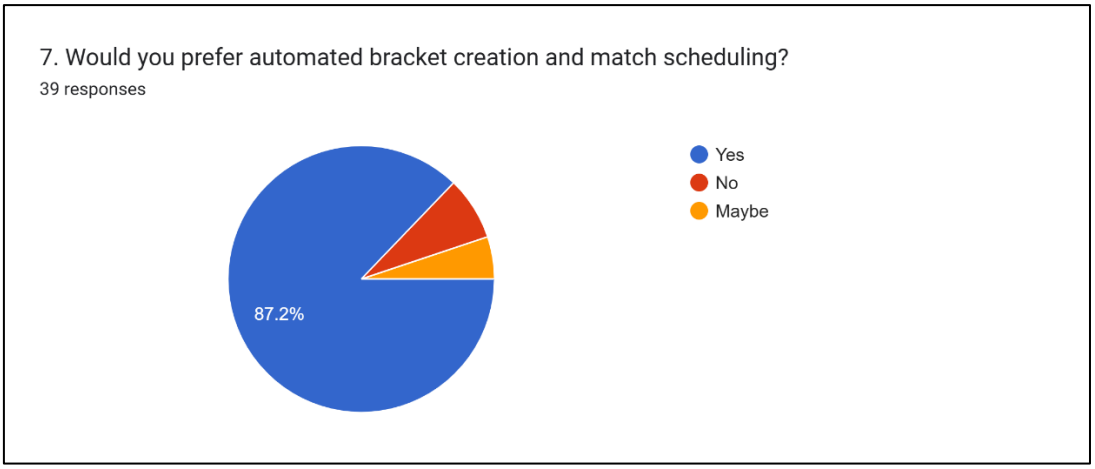


Figure 40 Pre Survey Question 7

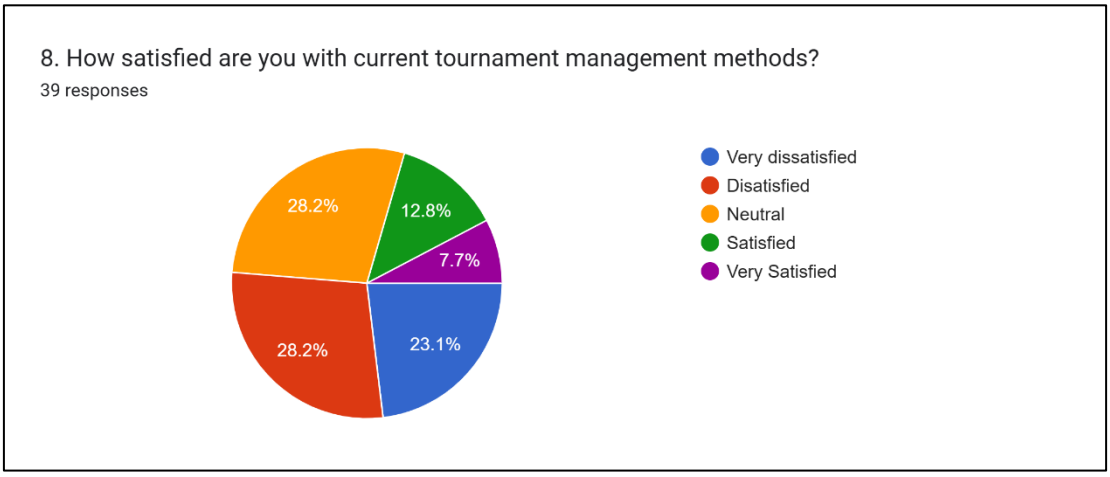


Figure 41 Pre Survey Question 8

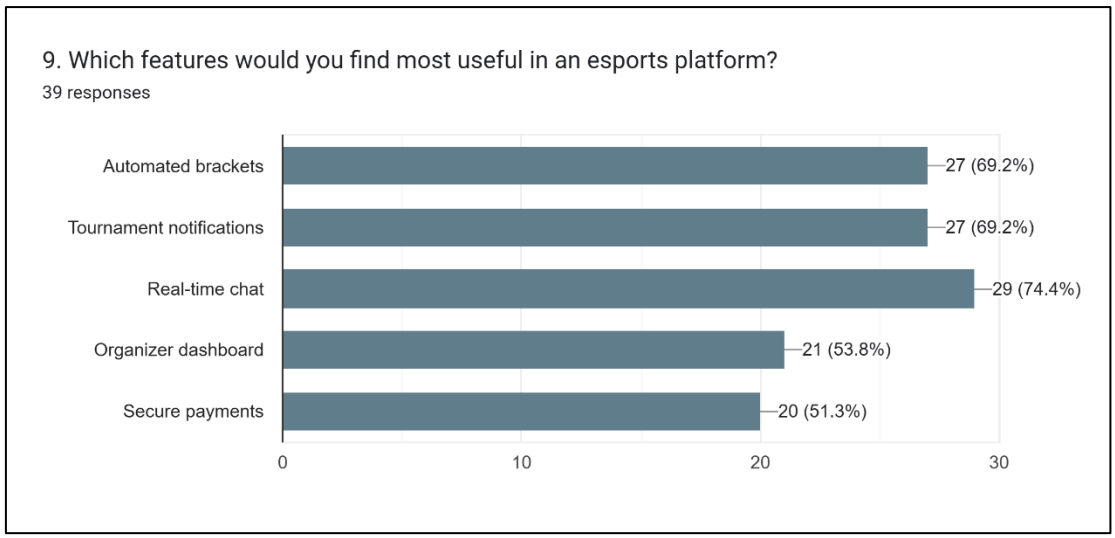


Figure 42 Pre Survey Question 9

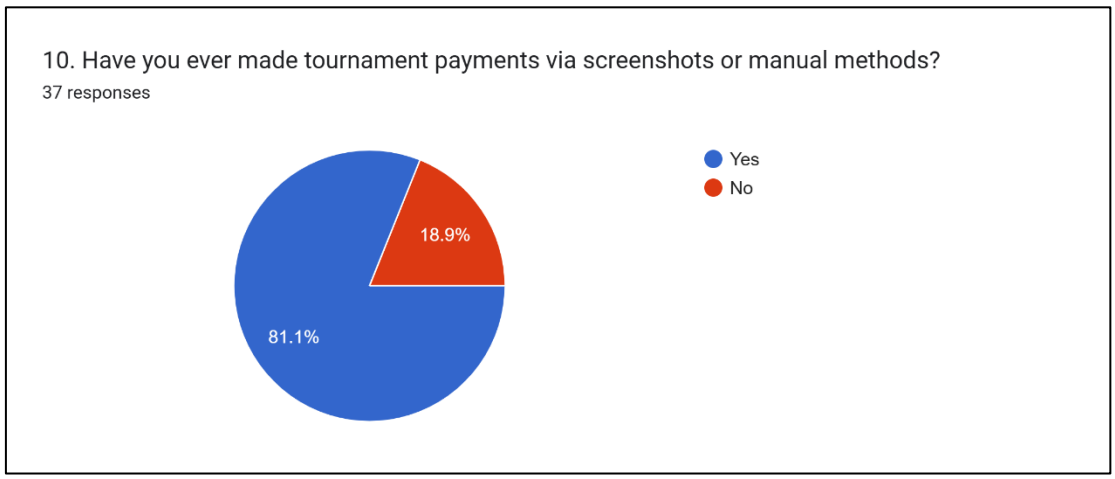


Figure 43 Pre Survey Question 10

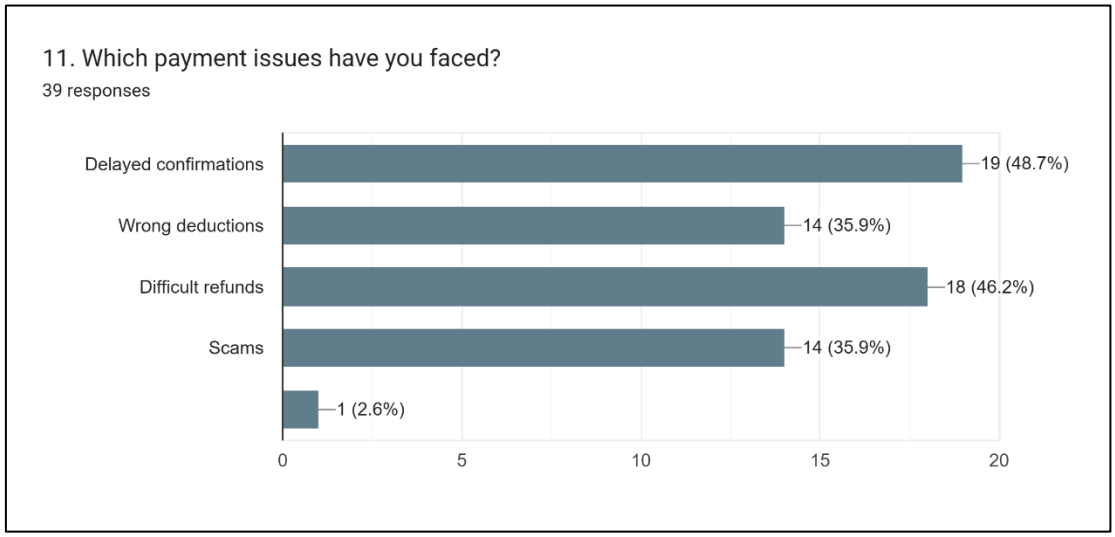


Figure 44 Pre Survey Question 11

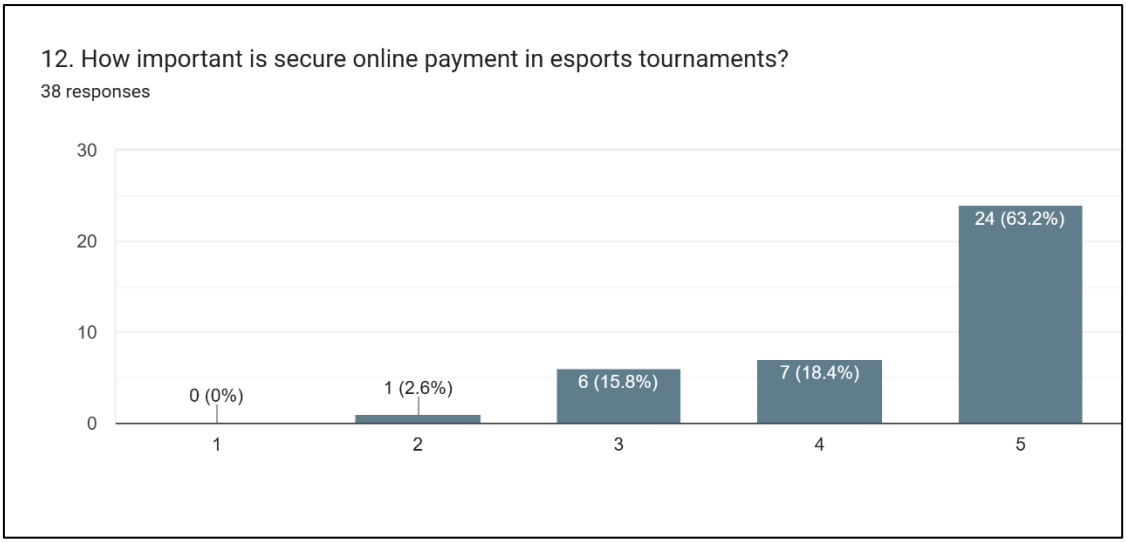


Figure 45 Pre Survey Question 12

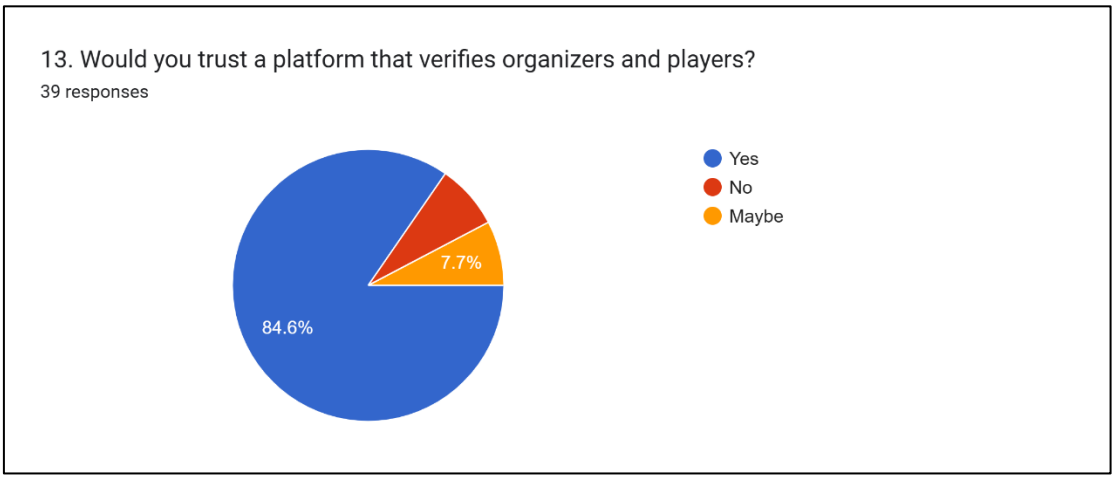


Figure 46 Pre Survey Question 13

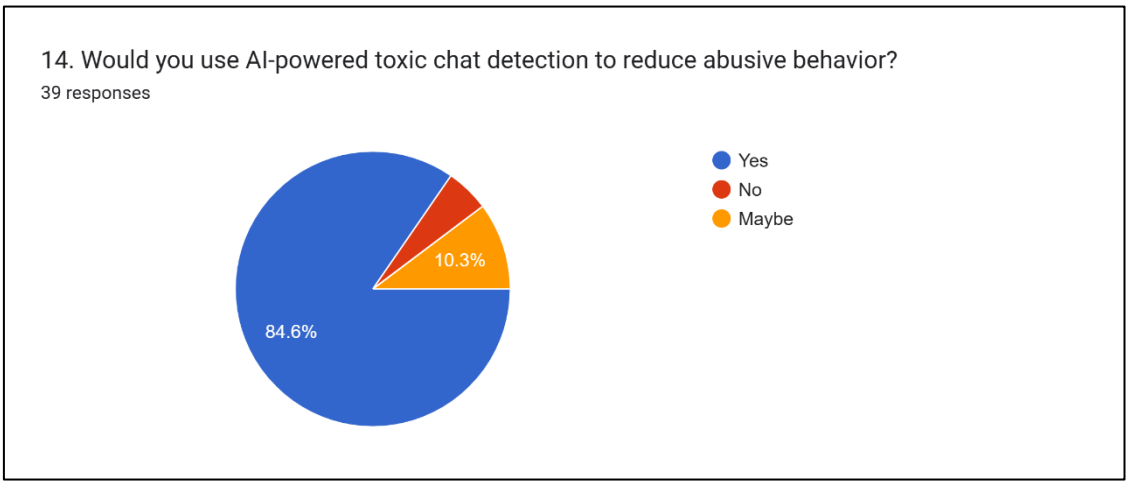


Figure 47 Pre Survey Question 14

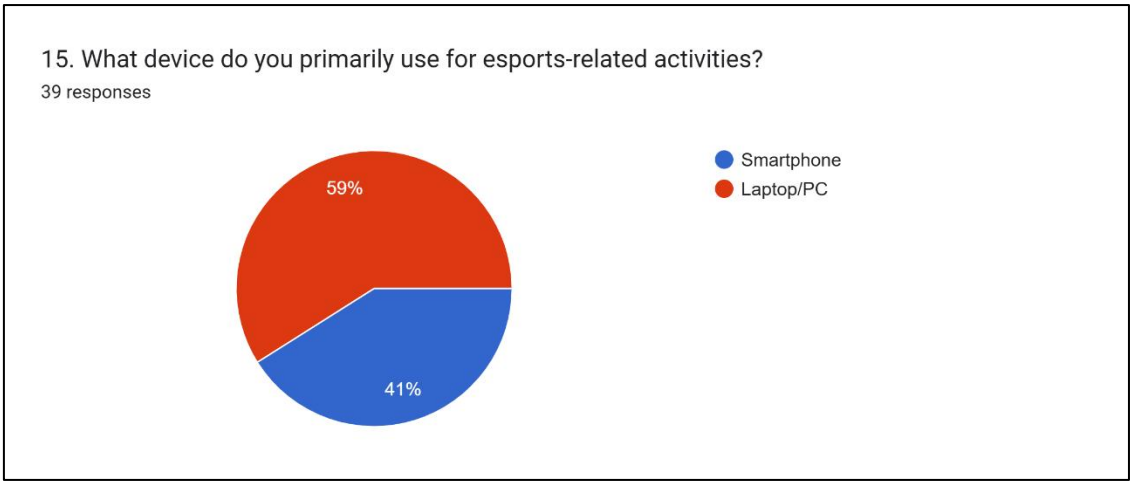


Figure 48 Pre Survey Question 15

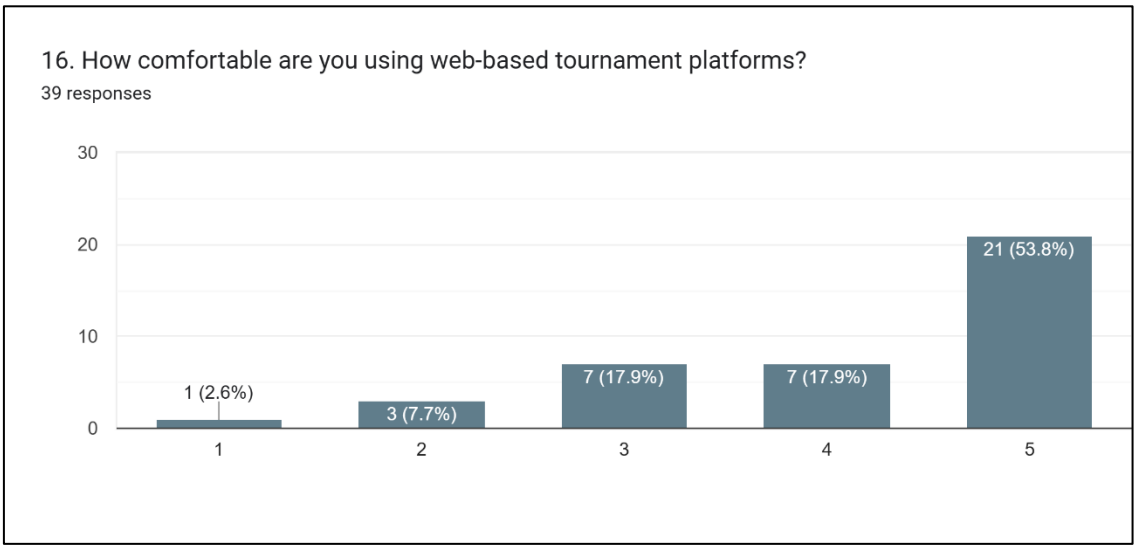


Figure 49 Pre Survey Question 16

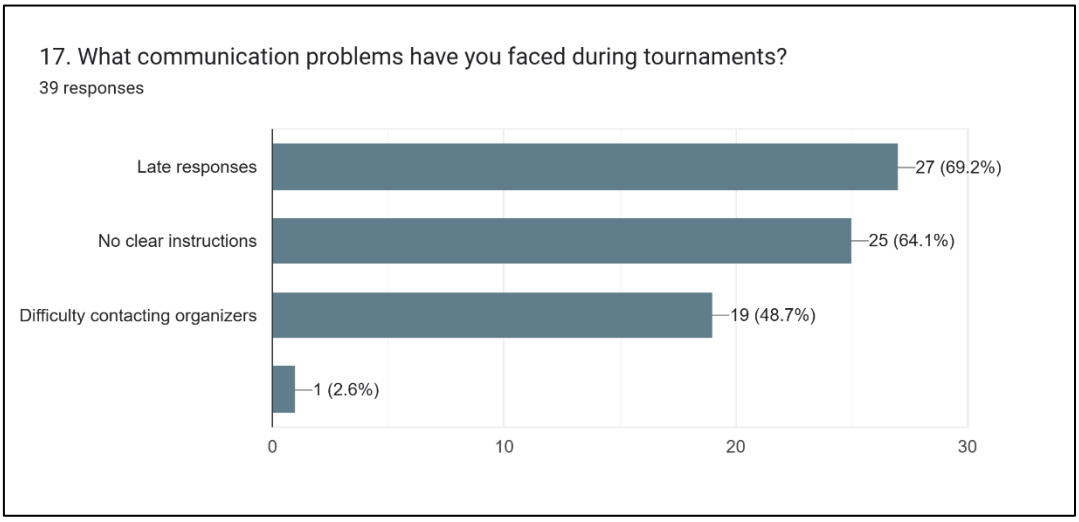


Figure 50 Pre Survey Question 17

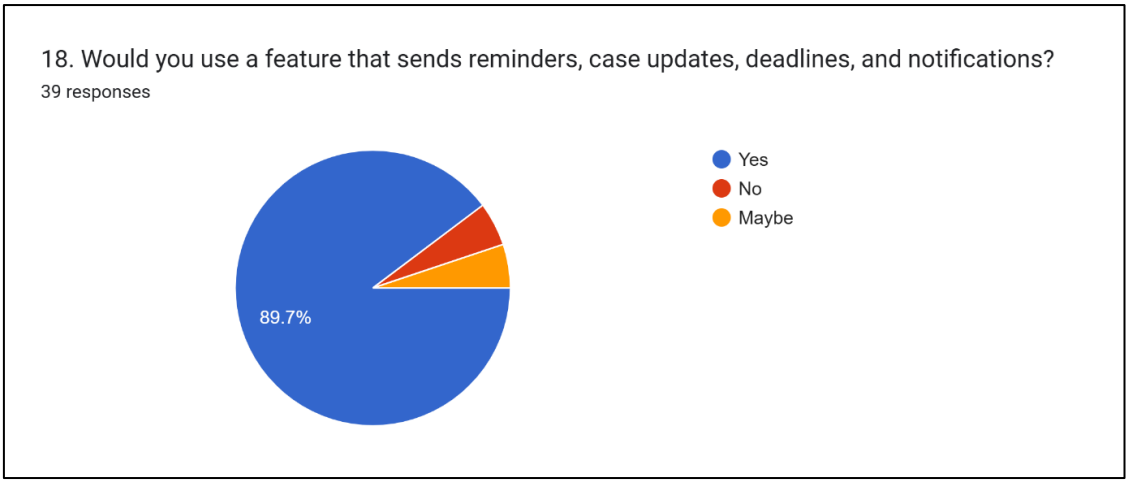


Figure 51 Pre Survey Question 18

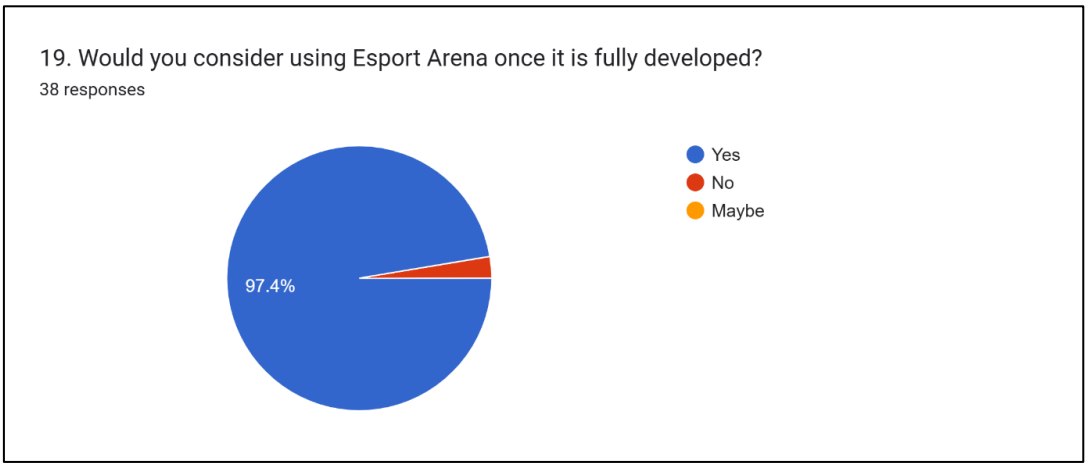


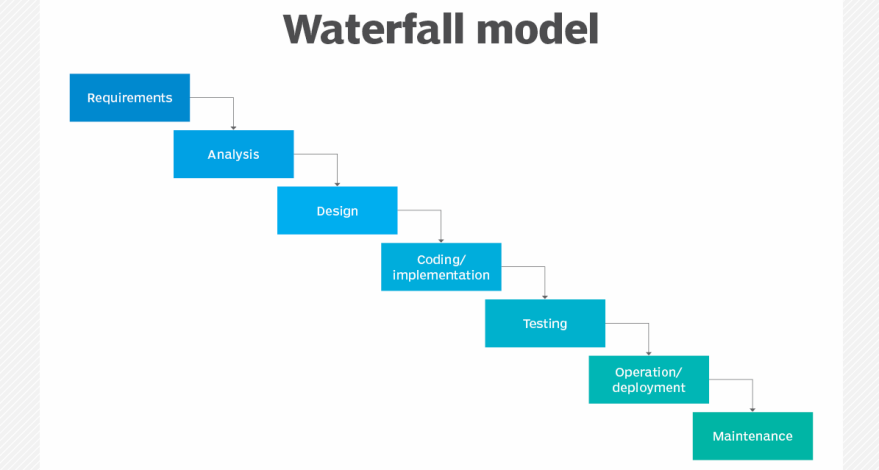
Figure 52 Pre Survey Question 19

Go Back

8.4 Considered methodologies

8.4.1 Waterfall Model

The Waterfall Model refers to sequential type with all the requirements being defined first and each stage must be done prior to the start of the next. It is very organized and is suitable with predictable and stable projects. It is straightforward software systems and can be useful due to its clarity and documentation. It is however inefficient when there are frequently changing requirements, or when a response to early feedback is required. Its inflexible nature makes it expensive to do changes as soon as the development is underway. (Kirvan, 2024)



Reason for not choosing waterfall model:

Table 13 Justification Table Waterfall Model: 1

Case Study Scenario	**Creating an esports platform where the rules of the tournaments, user interface (UI) flows, and payment functions are regularly altered.**
Features	Sequential and hard phase-wise development in which the next phase must be fulfilled before another can be initiated.
Justification	Payments flows, tournament formats, and user experience should be updated on a regular basis at Esport Arena depending on the feedback. Waterfall is not dynamic as it costs a lot and takes a long time to implement changes once the requirement or design stage is completed.

Table 14 Justification Table Waterfall Model: 2

Case Study Scenario	**The requirement to have a register and log-in and simple dashboards and real users.**
Features	Such software is only provided at the last stage of implementation.
Justification	Waterfall does not allow usable output until toward the end of the process. In the case of Esport arena, this would delay the core flow testing such as authentication and tournament views, taking the risk of significant issues in the usability being found far too late.

Table 15 Justification Table Waterfall Model: 3

Case Study Scenario	**The incorporation of eSewa/Khalti and notification services that might have to be implemented multiple times.**
Features	Rigid linear flow with limited design, development and testing.
Justification	Often, payment gateway and notification integrations have to be retested and adjusted. In Waterfall, it is not very easy to reverse testing back to redesign integration logic thus leading to delays and increased rework cost.

Table 16 Justification Table Waterfall Model: 4

Case Study Scenario	**Better performance, scaling and security as tournaments and number of users increase.**
Features	The non-functional requirements are frozen early and communicated significant validation around the end.
Justification	Security and performance need to be adjusted to in a progressive manner by Esport Arena under higher load. The late kind of

	testing used by waterfall makes it difficult to refreeze these qualities as it goes.

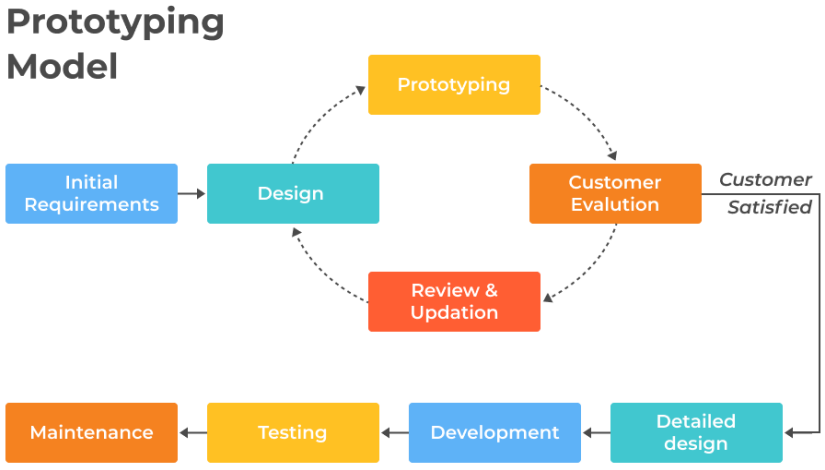
Table 17 Justification Table Waterfall Model: 5

Case Study Scenario	**Developing interactive web platform and real-time updates and changing UI behaviour.**
Features	Documentation-enzymatic, plan-oriented process most appropriate to stable, well-defined systems.
Justification	Esport Arena requires regular changes in UX in terms of interactive elements (live lists, future forums, notifications). The inflexibility and initial documentation of waterfall render it less appropriate compared to the Agile style.

Go Back

8.4.2 Prototyping Model

The Prototyping Model emphasizes on creating a rough working system that enables the users to visualize the features and further explain the requirements. It stimulates a high level of usage and is also good in systems that require transparent interfaces. Nevertheless, prototypes can make unreasonable expectations and the more prototypes produced, the more effort and money are spent. Another result of the model is the frequent change of scope depending on the feedback of users. (Rajkumar, 2025)



Reason for not choosing prototyping model:

Table 18 Justification Table Prototyping Model: 1

Case Study Scenario	**Multiple closely integrated modules will be developed including tournaments, wallet, withdrawals, chats and admin panels.**
Features	Focus on developing fast prototypes to iron out the requirements prior to complete development.
Justification	Esport Arena would require a number of prototypes (wallet, tournaments, payments, forums), which would time and effort up. The administration of numerous prototypes of interrelated modules becomes ineffective.

Table 19 Justification Table Prototyping Model: 2

Case Study Scenario	**The presentation of wallet or payment pages to the users at their initial stages.**
Features	Prototypes are realistic even incomplete logic and security.
Justification	Players and organizers may believe that the prototype accepts actual payments or stable tournaments. This may create misunderstanding and an illusion during the project.

Table 20 Justification Table Prototyping Model: 3

Case Study Scenario	**Introduction of sound verifying of payment, bracket generating and transaction history.**
Features	The emphasis is more on the front-end behaviour rather than serious, deep, production-level back-end logic.

Justification	Esport Arena needs (wallet rules, ranking logic) are difficult to implement in throwaway prototypes and require an additional effort to recreate everything with the correct architecture.

Table 21 Justification Table Prototyping Model: 4

Case Study Scenario	**Gathering user ideas post-screen UX mock-ups.**
Features	Endorses frequent feedback which can add continuously a new feature request.
Justification	In the Esport arena case, every prototype round had the possibility of adding new types of tournaments or wallet features that would leave the scope left unchecked and applicability under time pressure.

Table 22 Justification Table Prototyping Model: 5

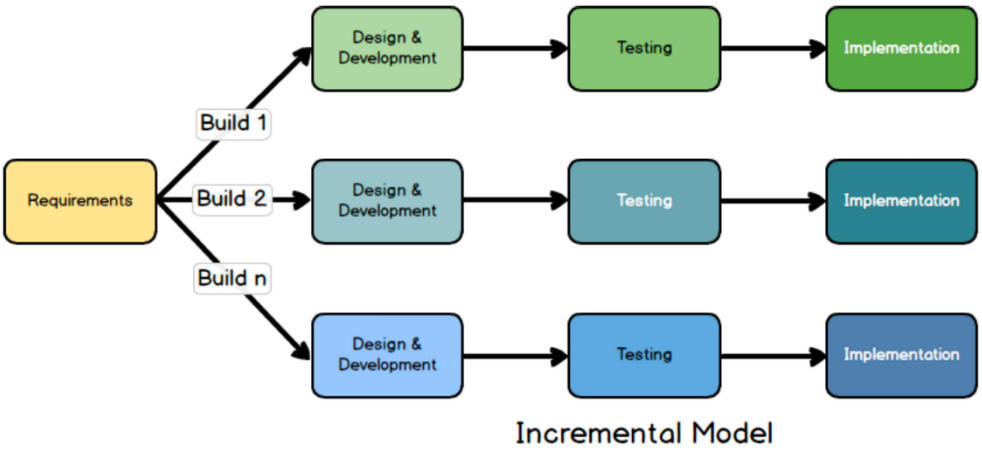
Case Study Scenario	**Real-time coordination of tournaments, payments, notifications, and moderation of a chat.**
Features	Best on small, interface-oriented systems, not integrated platforms.
Justification	When a set of Esport Arena modules has to be managed with several prototypes, it turns out to be difficult and inaccurate.

Go Back

8.4.3 Incremental Model

The Incremental Model develops the system in other small functional modules delivered as time moves by. It enables launching of partial forms of the product either early or adding those more aspects progressively. The model is efficient when there is knowledge about basic requirements, whereas dependency modules may make the

development problematic. There must be solid planning and architecture to make sure the increments integrate. (Sachan, 2025)



Incremental Model

Reason for not choosing Incremental model:

Table 23 Justification Table Incremental Model: 1

Case Study Scenario	**Splitting tournaments, payments, participation, and notifications into increments.**
Features	System is provided in autonomous bursts that provide a range of features.
Justification	The attendance of the tournaments is highly based on wallet, schedule and validation. It is possible to construct them in different stages making them not complete enough and where a certain section cannot operate effectively without the other.

Table 24 Justification Table Incremental Model: 2

Case Study Scenario	**Introducing a combination of multiple independently developed increments into a single working platform.**
Features	Both increments are developed and designed individually and then brought together.

Justification	In the case of Esport Arena, the large number of increments (tournament, wallet, forums, administration, etc.) will prove incompatible assumptions and additional integration issues.

Table 25 Justification Table Incremental Model: 3

Case Study Scenario	**Locking overall architecture prior to understanding all information about tournaments and wallet features.**
Features	Needs a constant world architecture to enable incremental features to plug in.
Justification	The rules and financial streams related to competition in Esport Arena can change. Premature fix of the architecture may make subsequent changes expensive or may cause work-around.

Table 26 Justification Table Incremental Model: 4

Case Study Scenario	**Desire frequent sprint reviews and backlog reprioritization.**
Features	The features are bigger and less time-boxed compared to Scrum sprint.
Justification	Esport Arena has a benefit of shorter cycles, which have definite Scrum rituals (planning, review, retrospective). These roles and meetings are not characterized by incremental only.

Table 27 Justification Table Incremental Model: 5

Case Study Scenario	**Dealing with constant adjustments of tournament rules, UI, and payment logic as time goes by.**

Features	The features will work well when there are significant requirements known and when increments are bound together with changes.
Justification	Esport Arena is a multi-module dynamic platform; one may find it more difficult to prioritize and restructure continuously than in Scrum based on Incremental Model only.

Go Back

8.4.4 Iterative Model

The Iterative Model plans the system in the cyclic process, step by step resulting in a more sophisticated model. It is best when it is not clear what one needs at the outset. The model gives an opportunity to continuously improve, though it does not have structured roles and sprint cycles as in Agile. Too much iteration will also add greater cost and development time. (Kumar, 2023)

Reason for not choosing Iterative model:

Table 28 Justification Table Iterative Model: 1

Case Study Scenario	**Authentication and payments, and time-bound work with clear-cut roles and time work.**
Features	Cycles of design and develop without strict sprint backlog.

Justification	Esport Arena is in need of established roles (Product Owner, Scrum Master) and predictable sprints, pure Iterative Model lacks any of these requirements.

Table 29 Justification Table Iterative Model: 2

Case Study Scenario	**Always improving tournament features and dashboards.**
Features	Promotes continual repetitions which can continue to add or rework functionality.
Justification	Esport Arena may not have an effective backlog coupled with a robust sprint planning framework which will lead to rampant scope creep in tournament and schedule features.

Table 30 Justification Table Iterative Model: 3

Case Study Scenario	**Replacing payment flows or bracket logic multiple times.**
Features	It can have iterations over the same components.
Justification	In the case of Esport Arena, the redesign of payment and match workflow can be counterintuitive and expensive to conduct without small, prioritized sprints.

Table 31 Justification Table Iterative Model: 4

Case Study Scenario	**Prioritizing features with several modules and keeping all people on track.**
Features	Gives cycles of improvement but leaves out roles and ceremonies.

Justification	Tournaments, wallet and forums require systematic coordination of Esport arena. Absence of the role definitions may complicate coordination and make decisions more difficult.

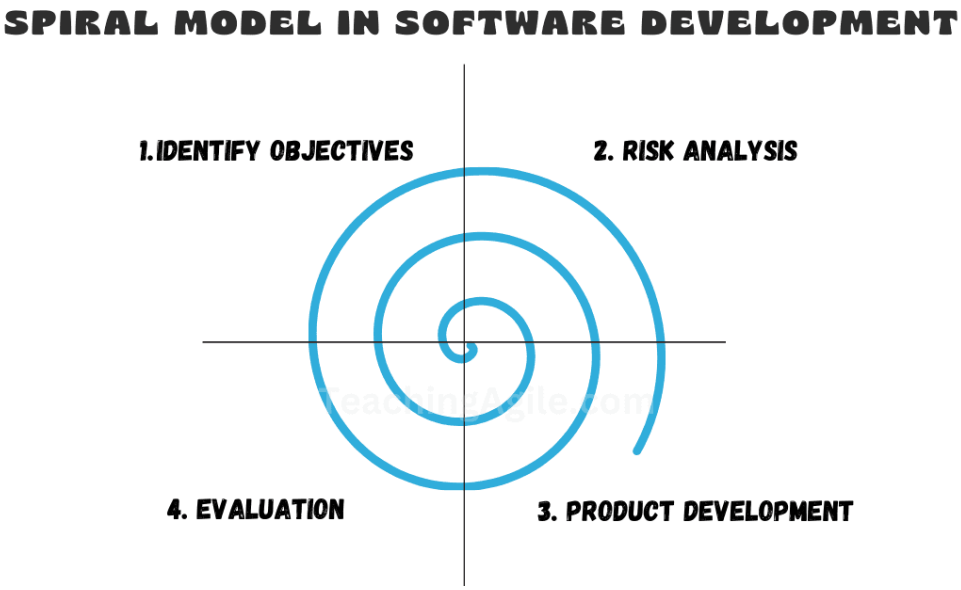
Table 32 Justification Table Iterative Model: 5

Case Study Scenario	**Creating an esports multi-module platform in real-time.**
Features	Ideal in incremental development but less effective in managing large and complex projects.
Justification	Catering with live tournaments, payment, and moderation Esport arena is more structured and better positioned to control risks than the basic Iterative Model is.

Go Back

8.4.5 Spiral Model

The development of the Spiral Model is a combination of the risk's evaluation with the iterative development in each of the phases. It is particularly appropriate to large and risky systems that need to be strictly validated. It is flexible and rich in risk management, which makes it powerful and expensive. It requires professional analysis and much documentation, posing too big a task to do by medium-sized or scholarly student projects. (Talreja, 2025)



Reason for not choosing Spiral model:

Table 33 Justification Table Spiral Model: 1

Case Study Scenario	**Limited budget and time student project.**
Features	Integrates the development by boulders with thorough risk study on a cycle basis.
Justification	Spiral Model engages in massive risk evaluation and planning, thereby raising cost and complications. In the case of Esport arena being a student FYP, this amount of overhead is not essential.

Table 34 Justification Table Spiral Model: 2

Case Study Scenario	**Documentation management of a medium-sized academic project.**
Features	Each phase and risk evaluation step must have detailed documents.
Justification	Esport Arena already generating proposal, SRS, interim and final reports. The project scope would not allow Spiral to add the additional documentation cycles.

Table 35 Justification Table Spiral Model: 3

Case Study Scenario	**Developing a working system in an academic year.**
Features	Every spiral cycle has planning, risk analysis, engineering as well as evaluation.
Justification	These recurrent loops stretch the development period. In the case of Esport Arena, this may postpone the introduction of the fundamental features such as tournaments and wallet to an inappropriate time frame.

Table 36 Justification Table Spiral Model: 4

Case Study Scenario	**Small project team with no specialized risk-management positions.**
Features	It needs practiced risk analysts and management.
Justification	Esport arena is produced by one individual/ small team hence using all the Spiral risk techniques it is unrealistic and excusable.

Table 37 Justification Table Spiral Model: 5

Case Study Scenario	**Developing an esports management system instead of a system of critical safety.**
Features	Intended to operate in high-risk, such as defence and aviation or banking.
Justification	Whereas reliability is significant, Esport arena is not the project that needs the rigor that Spiral demands. The risk level of the project would not be equal to the overhead.

Go Back

8.4.6 Scrum Framework

Scrum is an agile methodology that Scrum.org created to help to handle complicated projects with the help of iterative and incremental development. It breaks down work into short fixed-length cycles known as sprints. A sprint consists of a plan, development, testing, review and reflection, whereby the teams can monitor progress made, early identification on problems and continued improvement. Scrum is focused on collaboration, transparency, and flexibility and has very specific roles, including Product Owner, Scrum Master, and Development Team. (scrum.org, 2025)

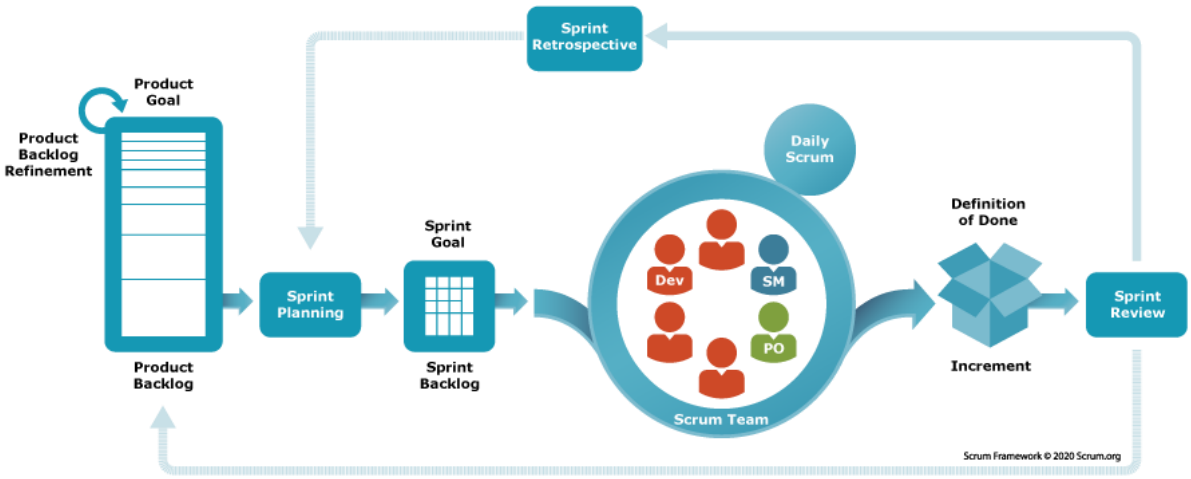


Figure 53 Scrum framework diagram

Scrum framework functions based on a cycle which involves the following:

- Developing and ranking the Product Backlog.
- Sprint Planning to choose tasks. Daily Scrum - coordination.
- Incremental development and testing.
- Sprint Review for feedback.
- Sprint Retrospective for improvement.

Go Back

8.5 Work Till Date

8.5.1 SRS

The SRS document describes the complete functional and non-functional requirements of the Esports Arena platform. It serves as a guide to the developers, stakeholders and users and gives them an idea of the features of the system, performance expectations and scope of the entire project. In the case of Esports Arena, the modules defined in the SRS include user registration, tournament administration, wallet and payment integration, match result, and communications forums. It also provides the limitations of security, scalability and the reliability of the system.

This ensures that the stakeholders will be on the same page of what the system is aiming to provide.

Project Title: Esports Arena

Category: Web Application (React + Django REST)

8.5.2 INTRODUCTION

A Software Requirement Specification (SRS) is a well-organized, detailed documentation of a software system which states both the functional and non-functional requirements and forms the basis around which developers, testers, supervisors and future maintainers base their work. An SRS bridges the gap between the conceptual notion and actual development by stating the purpose of the system, its behaviour, limitations, and expectations placed on it by users in a clear way.

The Esports Arena SRS describes all primary system functionalities related to online tournament hosting and participation. This system provides users with different functionality, such as authentication, tournament creation, automated bracket generation, match result verification, wallet management, payment integration, and community forums. It also covers critical non-functional requirements such as performance expectations, security protocols, and system availability of the system.

This system allows esports/casual Players participate in tournaments, Organizers can offer events and Admins can oversee activities in the platform. It combines virtual coin systems, payment APIs and semi-automatic withdrawal processing to facilitate a smooth competitive gaming experience.

8.5.3 Purpose

The purpose of Esports Arena is to provide a structured, automated, and user-friendly platform for hosting online gaming tournaments. The platform links Organizers to Players/Teams and allows teams to compete on a competent digital process. Esports Arena aims at easing the process of tournament management and providing secure wallet management, communication, and equitable result-checking system.

Key Objectives of Esports Arena:

- Organized and automated esports tournament hosting
- Fair match result processing
- Safe and secure wallet payment integration
- Seamless withdrawal handling
- Engaging community interaction using forums
- Scalable and reliable competitive platform

8.5.4 Intended Audience and Intended Use

Players

Players are individuals/teams who participate in tournaments hosted on the platform. They are the ones who use the system to join matches, submit results, track wallet balances, and interact with each other in forums.

Players may include:

- **Casual Gamers**: Players who join to have fun and entertain and compete in smaller-scale tournaments.
- **Competitive Esports Players**: Individuals who frequently join esports events and aim to improve rankings and win prize rewards.
- **Students and young adults:** Young individuals who are actively engaged in online gaming competitions who do not have access to organized offline events.
- **Community Gamers:** Members of gaming groups or clans looking for a structured and fair tournament environment.

Organizers

Organizers are those who are responsible for creating and managing tournaments, which can be free or paid. They configure match rules, verify results, and coordinate the flow of the tournament.

Organizers may include:

- **Gaming communities:** Groups or clubs that frequently conduct tournaments for their members.
- **Esports Event Hosts:** Individuals or teams who specialize in conducting online or offline esports competitions.
- **Campus or Local Organizers:** Colleges, gaming cafés, or small organizations hosting community-level tournaments.
- **Professional Hosts:** Individuals who run tournaments for revenue, sponsorship, or brand visibility.

Admin

The admin is the one who controls and supervises the entire Esports Arena platform. The admin ensures that all platform operations run securely and efficiently by following the system guidelines and policies.

Admin responsibilities include:

- **Monitoring activities:** Overseeing Players and Organizers for making sure of fair usage while preventing fraud, exploitation, or policy violations.
- **Handling Disputes:** Resolving escapades between Players and Organizers like wrongful findings, suspicious activities or mismanagement of tournaments.
- **Platform Security:** Protecting the user data, wallet transactions and authentication process using correct security standards and compliance practices.
- **Content Moderation:** Monitoring posts, forum posts, profiles and announcements made by tournament about observing the rules of the community.
- **System Maintenance:** Tracking performance of platforms, fixing technical failures, updating system components, and high availability when tournaments are under way.

8.5.5 Definition and acronyms

Player: A gamer who enters and takes part in tournaments.

Organizer: This is the user who organizes and runs tournaments.

Admin: A system controller that oversees the general activity.

Wallet: A balance of virtual coins to use in entry and withdrawal in a tournament.

API: Application Programming Interface which is utilized in payment and authentication.

CRUD: CRUD operations.

UI: User Interface whereby users communicate with the platform.

Tournament Bracket: computer generated framework to control the flow of matches.

8.5.6 Overview

User Needs

Easy-to-Use Interface: It needs to have a simple intuitive and user-friendly interface that enables the Players, Organizers and Admins navigate the platform easily without the need to be experienced in matters of technology.

Secure Login: The customer must be able to log in safely using a safe authentication system to secure personal information and wallet details.

Tournament Participation & Management: The players require an efficient system that allows them to browse, enter, and monitor tournaments whereas the Organizers need a structured interface that allows them to create and manage events effectively.

Digital Wallet & Payment Integration: The system must have easy deposit and withdrawal capabilities by integrating with local digital payment gateways to process tournament entry fees and the payment of prizes.

Notifications & Alerts: The users are supposed to get real-time notifications on the match schedules, verification of results, wallet, and tournament announcements.

Feedback & Reporting: To ensure the quality and fairness of the tournaments, the players must be able to report the problems, submit the match proofs and give feedback.

Assumptions

- Users possess basic knowledge of using mobile or web applications.
- All users (Players and Organizers) have access to a stable internet connection during tournaments and wallet transactions.
- The platform will run on modern devices, updated browsers, and reliable network environments.
- Organizers will follow standardized tournament rules and maintain fair play guidelines.

Dependencies

Payment Gateway Integration: The payment gateway is integrated into the system for making seamless digital wallet operations possible regarding deposits, tournament entry fee payments, and withdrawals.

Notification System Integration: Email or in-app notification services will be integrated for sending updates related to matches, payments, and results of tournaments.

Cloud Database & Hosting Services: This is necessary for storing the user profiles, tournaments, transactions, and match data securely.

Media Upload Service: For submitting match screenshots or video proof during disputes or result verification.

8.5.7 Functional requirements

8.5.7.1 User Management

Requirement Id	Function	Requirement Description	Priority
FR-7.5.7.1.1	Player Registration	The system shall allow Players to register using email, phone number, and password.	High
FR-7.5.7.1.2	Organizer Registration	The system shall allow Organizers to register and host tournaments.	High
FR-7.5.7.1.3	Login	The system shall authenticate all users (Player, Organizer) using secure login.	High
FR-7.5.7.1.4	Role-Based Access	The system shall provide role-based access to separate dashboards.	High
FR-7.5.7.1.5	Profile Management	Users shall be able to update their profile details.	Medium

8.5.7.2 Tournament Management

Requirement Id	Function	Requirement Description	Priority
FR-7.5.7.2.1	Create Tournament	The system shall allow Organizers to create tournaments with details like title, game, entry fee, prize, and limits.	High
FR-7.5.7.2.2	View Tournaments	The system shall display all tournaments categorized as Upcoming, Ongoing, and Completed.	High
FR-7.5.7.2.3	Join Tournament	The system shall allow Players to join free and paid tournaments.	High
FR-7.5.7.2.4	Auto Grouping	The system shall automatically generate teams or brackets after the join deadline.	High
FR-7.5.7.2.5	Edit Tournament	Organizers shall be able to edit tournament details before it begins.	Medium

FR-7.5.7.2.6	Status Update	The system shall update tournament status automatically during progression.	High

8.5.7.3 Participation & Match Result Management

Requirement Id	Function	Requirement Description	Priority
FR-7.5.7.3.1	Join Participation	Players shall be able to join tournaments they are eligible for.	High
FR-7.5.7.3.2	Submit Results	Players shall be able to submit match results with screenshot proof.	High
FR-7.5.7.3.3	Verify Results	Organizers shall verify submitted results by approving or rejecting them.	High
FR-7.5.7.3.4	Update Leaderboard	The system shall update the leaderboard automatically after verification.	High

8.5.7.4 Wallet & Virtual Coin System

Requirement Id	Function	Requirement Description	Priority
FR-7.5.7.4.1	Wallet Creation	Every Player shall have a wallet to store virtual coins.	High
FR-7.5.7.4.2	Coin Top-Up	The system shall allow Players to top up wallet coins via eSewa or Khalti.	High
FR-7.5.7.4.3	Entry Fee Payment	The system shall deduct coins automatically when joining a paid tournament.	High
FR-7.5.7.4.4	Transaction Log	The system shall maintain a complete transaction history for each wallet.	Medium

8.5.7.5 Withdrawal Management

Requirement Id	Function	Requirement Description	Priority
FR-7.5.7.5.1	Request Withdrawal	The system shall allow Players to request withdrawals by entering method and mobile number.	High
FR-7.5.7.5.2	Process Withdrawal	The system shall process withdrawals using automated payout APIs.	High
FR-7.5.7.5.3	Track Withdrawal	The system shall allow Players to track withdrawal status.	Medium

8.5.7.6 Forum & Communication

Requirement Id	Function	Requirement Description	Priority
FR-7.5.7.6.1	General Forum	The system shall provide a general discussion forum for all users.	High
FR-7.5.7.6.2	Tournament Forum	The system shall provide tournament-specific forums for communication and updates.	High
FR-7.5.7.6.3	Toxic Filter	The system should filter the toxic words used by a player/organizer.	Medium

8.5.7.7 Admin Management

Requirement Id	Function	Requirement Description	Priority
FR-7.5.7.7.1	View Users	The system should allow Admin to watch all players and organizers inside the system.	High
FR-7.5.7.7.2	Monitor Transactions	The system should allow the admin to watch all wallet	High

		transections and withdrawals by users.	
FR-7.5.7.7.3	User Feedback	The system should allow the admin to get feedback by the users.	Medium

8.5.8 Nonfunctional Requirements

Requirement Id	Category	Requirement Description
NFR-7.5.8.1	Security	All user data is encrypted; JWT is used for authentication.
NFR-7.5.8.2	Performance	The system support hundreds of active tournament participants with smooth response times.
NFR-7.5.8.3	Scalability	The system is designed to handle increasing numbers of tournaments and users.
NFR-7.5.8.4	Usability	Clean, modern UI with easy navigation for gamers.
NFR-7.5.8.5	Maintainability	Codebase structured to allow future extensions.
NFR-7.5.8.6	Compatibility	This system supports all modern browsers and devices.

8.5.9 Hardware and Software Specifications

Software Specification

- Compatible browsers: Chrome, Firefox, Safari, Edge
- Operating Systems: Windows 10+, macOS
- HTTPS for secure data transfer
- PostgreSQL database

Hardware Specification

- Minimum 4GB RAM

- Display resolution 1024×720 or higher
- Dual-core processor (i5 or equivalent recommended)
- Stable internet connection (10 Mbps minimum)

External Interface Requirements

- Payment API Integration: The system support eSewa and Khalti for coin top-ups and withdrawals.
- Email Notifications: The system will send OTP, withdrawal updates, and match updates.
- Cloud Storage: It stores screenshot proofs for match results.

8.5.10 Other Non-functional Requirements

Non-functional requirements are defined for the qualities and operational constraints under which the Esports Arena platform has to operate to perform optimally, securely, scalable, and to the satisfaction of all stakeholders. They play an important role in maintaining reliability, efficiency, fairness, and overall usability throughout the competitive gaming activities. They help ensure that the platform will always handle traffic effectively, conduct financial transactions in a secure manner, and provide seamless experiences to Players, Organizers, and Admins.

- Performance
- Scalability
- Security
- Availability and Reliability
- Usability
- Maintainability
- Compatibility
- Extensibility
- Environmental Impact

These non-functional requirements make sure that Esport Arena be in a high-performing, secure and user-friendly platform which will be capable of adapting to the future demands while maintaining equality, fairness, trust and reliability with its users.

Go Back

8.5.11 Sprint 1

8.5.11.1 High Level Description

Use Case: Sign up User

Table 38 High Level Description of Sign-up User

Use Case	Sign up User
Actor	User
Description	The Esport Arena platform has a new user sign-in, where one is required to type his or her name, email and password and other details like contacts. The input is validated in the system, an existing account using the same email is checked and the information of the new user is stored safely. After registration is successfully completed, the system will notify the user about the successful registration. Once activated, the user will be allowed to access the general features which include logging in, navigating through tournaments as well as services of the available platform.

Use Case: Log in User

Table 39 High Level Description of Log in User

Use Case	Log in User
Actor	User
Description	The Esport Arena platform is accessed by a registered user by entering their email and password. The system not only checks the credentials, but it also checks whether the user is present in the database. Once validation is conducted, the user gets forwarded into the respective dashboard depending on his role (Player or Organizer). Wrong logs lead to an error message making access secure. On successful log in, individually different users are then able to do role-based activities in the system.

Go Back

8.5.11.2 Expanded Level Description

Go Back

Use Case: Sign up User

Actor: User

Description: The Esport Arena platform has a new user sign-in, where one is required to type his or her name, email and password and other details like contacts. The input is validated in the system, an existing account using the same email is checked and the information of the new user is stored safely. After registration is successfully completed, the system will notify the user about the successful registration. Once activated, the user will be allowed to access the general features which include logging in, navigating through tournaments as well as services of the available platform.

Typical Course of Events

Table 40 Expanded Use Case Description for Sign Up User

Actor Action	System Response
1. User opens the "Sign up" page.	2. System presents the registration form.
3. User feeds in details and e-mails the form.	4. System checks fields and determines whether the email is already in existence.
	5. In case valid and unique, the user data are stored in system as securely as possible and default role is established.
6. User sees success message.	7. System affirms registration and provides connection to log in.

Use Case: Log in

Actor: User

Description: The Esport Arena platform is accessed by a registered user by entering their email and password. The system not only checks the credentials, but it also checks whether the user is present in the database. Once validation is conducted, the user gets forwarded into the respective dashboard depending on his role (Player or Organizer). Wrong logs lead to an error message making access secure. On successful log in, individually different users are then able to do role-based activities in the system.

Typical Course of Events

Table 41 Expanded Use Case Description for Sign In User

Actor Action	System Response
1. User opens the "Log in" page.	2. System shows login form
3. Clicking person types in email and password and clicks submit.	4. System receives input and search by email.
	5. Checking in the system password; on the correctness, role identification.
6. User sees success message.	7. If the credentials match, the system retrieves the user's role (Player or Organizer).

8.6 Progress Analysis

8.6.1 Analysis

The project has been developed to Sprint 1: User Authentication and Profile Management. Basic registration and log in processes have already been in place with role based authentication and authorization partially completed. Players, Organizers, Admins access is already being set to multi-user access, role-specific dashboards and profiles are being developed. This sprint establishes the security and identity layer all other modules

such as tournaments, wallet, communication and the control of the administration will be based on.

Nonetheless, Sprints 2-5 are going to be non-existent. It has not begun to develop tournament creating and participating, match result entry and standings, wallet and payment services, forum and toxicity blocking, notification facilities, or the administration console. On the same note, the Testing, Deployment and Closure phase has not been initiated yet. This causes the work done on the project up to that point to be biased, with more effort going to analysis and design and the first implementation sprint, and with most work being on functional development and quality assurance.

Overall, the project is in interim phase: planning and design artefacts are ready and aligned, and the initial critical sprint is partially applied. It is now working on finishing Sprint 1 to create a stable framework of authentication and dashboard and then tournament, financial and communication modules will have to be implemented and systematic testing and implementation will have to be done to meet the final submission deadline of 22nd April.

Go Back

8.7 Further Work

Organizing Tournaments and playing: Once the identity layer is stabilized, the next area will be developed as Sprint 2, encompassing the creation of tournaments and their participation. Organizer dashboards will have expanded features of the full tournament management including fields to manage game title, format, entry fee, prize pool, maximum participants and schedule. These forms will communicate using backend CRUD APIs to create, update and cancel tournaments as well as amplifying business requirements like registration dates and participant quotas. On the player side, traveling to and entering tournaments will be introduced with filters based on the type of game, date, and the price of joining, and all the join requests will be checked against the balance in the wallet and the number of vacancies in the tournament. This project will enable the transformation of the site to a handy tournament centre beyond a mere system of logging into.

Match Result Submission and Leaderboard: The recording and validating match outcomes mechanisms will be provided by Sprint 3. Interfaces will be provided to players of the game to enter match result and screenshot evidence, which will be kept safely and

linked to the appropriate tournament and participant records. Reviewing submissions, verifying evidence, and correcting scores where needed as well as settling disputes will be provided to the organizers with dedicated views. Approved results will be apportuned to brackets by backend services, and result in dynamic leaderboards, where players or teams are ranked according to the number of points, number of kills, and position. This set of features will make competitive results open, auditing, and automatic in terms of reflection in the tournament standings.

Payments, wallet and withdrawals: Sprint 4 will be the next significant piece of work and will complete the financial layer of Esport Arena. The wallet module will be expanded in the way that each user has a coin balance with proper transaction history. This will be integrated with eSewa and Khalti to enable top-ups of coins to be performed safely, in addition to payment initiation and on-failed conditions. Any paid tournament entry fee will automatically be deducted off the wallet and prizes and winnings given according to the results of the tournament. A withdrawal system will be built, which will give the possibility to players to cash-out coins and admins to check in, give out, or deny the request. This stage will ensure that financial activities are secure, equitable and in line with local digital payment.

Forum, Toxicity Filtering and Notifications: Sprint 5 will focus on the features of communication and community management. Ancillary forums such as general and tournament specific ones shall be adopted to facilitate communication between the players and the organizers. The functionality of posting messages will be combined with a toxicity filter element which recognizes prohibited or offensive words and blocks or provides warnings to such materials, which aids in promoting a respectful atmosphere. Simultaneously, a notification service will be created to provide in-app real-time notifications regarding new events like the creation of a new tournament, the match schedule, and the approval of the result as well as updates on wallets or withdrawals. Any notification will be saved in a manner that the users can revisit some important messages in future, thereby minimizing the likelihood of missing some information.

Administration, Monitoring and Governance: In addition to communication functionalities, an administration panel will be developed that will give centralised control over the platform. Admin dashboard will provide a summary of the number of users, running tournaments, financial activity, and reported problems. Admins can use

this interface to confirm new organizer accounts, suspend or reactivate users, track the activities of the tournament, and manage the requests to withdraw money. Audit trails and logs will also be adopted to document important administrative activities to facilitate accountability and further analysis. This assignment will guarantee that Esport Arena will be operated safely and openly as the activities increase.

Testing, Optimisation and Final Reporting: The last stage of work will be associated with the quality assurance and documentation in accordance with the Testing, Deployment and Closure aspect of the Gantt chart. Authentication, tournaments, wallet, results, forums, notifications and admin operations will be provided with comprehensive unit tests and integration tests. Acceptance testing and system testing will then be performed using real tournament conditions to test on performance, security, usability and erroneous handling. On these results, specific optimisation and debugging will be implemented, then a stable build is deployed to the hosting environment. It will be concluded by preparing the final report, presentation, which will record the entire development lifecycle, design decisions, testing results, and how Esport arena counteracts the problems found in the problem domain in the first place.

Go Back

8.8 Conclusion

The initial sprint on the implementation side has provided a secure authentication layer with player and organizer registration and login flows and role-based access differentiating between players, organizers and admins. This identity and access infrastructure is the foundation of all further modules, being less dangerous in terms of security and enabling every group of users to interact with the platform using customized dashboards. Though there are still no significant functional areas, including tournament creation, result handling, wallet and payments, forums, notification, and the admin panel, they are already broken down into detailed user stories and sprint tasks, which enables the way of working and facilitates incremental delivery.

In the future, the remaining sprints will gradually turn Esport Arena into a full-fledged tournament management system by realizing all of the modules intended, incorporating local payment gateways, toxicity control, and administrative control. These will then be developed and fully tested, optimised regarding performance and security and deployed to a live environment and the final report and presentation finalised before the 22nd April

submission. As a project, Esport Arena will not only comply with its academic ambitions, but would also provide a viable, transparent, and scalable infrastructure that will improve the organisation, equity, and experience of esports events in both Nepali and South Asian context.

Go Back