# American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science &Technology (FST)

Spring 23 24

Section: A

Software Quality Assurance and Testing

## E-Sports Tournament Management System

A Report submitted

By


| SN | Student Name | Student ID |
|---|---|---|
| 1 | Jahidul Islam Shikdar | 21-44634-1 |
| 2 | Devdoot Parial | 21-45061-2 |
| 3 | Md. Aminul Islam | 20-43854-2 |


**Checked By Industry Personnel**

Name:

Designation:

Company:

Sign:

Date:

# Software Test Plan
for
# < E-Sports Tournament Management System >

Version 1.0 approved
Prepared by <Devdoot Parial, Jahidul Islam Shikdar, Md.Aminul Islam>
< American International University-Bangladesh>
<May 14, 2024>

# Table of Contents



# Revision History

| Revision | Date | Updated by | Update Comments |
|---|---|---|---|
| 0.1 | 2024.05.12 | Jahidul Islam | First Draft |
| 0.2 | 2024.05.12 | Devdoot | Second Draft |
| 0.3 | 2024.05.12 | Aminul | Third Draft |
| 0.4 | 2024.05.12 | Jahidul Islam | Fourth Draft |
| | | | |

# 1. TEST PLAN IDENTIFIER:RS-MTP01.3

## 2. REFERENCES

- https://www.scribd.com/document/605352030/srs-gaming
- https://www.slideshare.net/slideshow/the-complete-srs-documentation-of-our-developed-game/65337752

## 3. INTRODUCTION

**Background to the Problem**

- Esports has rapidly grown into a multi-billion-dollar industry with a worldwide audience, attracting both professional players and amateur gamers. However, regions like Bangladesh struggle to build a robust esports infrastructure due to the lack of a centralized management system. This absence leads to fragmented tournament registration, inconsistent qualification processes, and high costs as players resort to international platforms, which stifles the growth of the local Esports scene. A comprehensive, localized esports management system could address these issues, providing Bangladeshi players with more opportunities and driving the growth of the esports industry in the region.

- The root cause of Bangladesh's esports problems is the absence of a centralized esports management system. This leads to fragmented tournament registration, inconsistent qualification standards, high costs due to reliance on international platforms, and limited opportunities for local players. As a result, the growth of the esports industry in Bangladesh is stunted. A comprehensive local platform could solve these issues by streamlining registration, reducing costs, ensuring consistent tournament rules, and creating more opportunities for Bangladeshi players, thereby fostering a thriving esports community and industry in the region.

- Addressing these issues through a localized esports management system is crucial. It will streamline tournament organization, reduce costs for players, ensure consistency in rules and standards, and ultimately foster a vibrant esports community in Bangladesh. This will open doors for local talent and contribute to the growth and sustainability of the esports industry in the region.

**Solution to the Problem**

- The proposed solution to address fragmented tournament management in Bangladesh's esports scene is to create a comprehensive Esports Tournament Management System

(ETMS). This platform centralizes all aspects of tournament management, streamlining processes like registration, scheduling, and team management, thereby reducing errors and ensuring consistency. It also lowers costs by providing a local alternative to international platforms, enhancing the user experience with a user-friendly interface, real-time scoring, and leaderboards. The ETMS is scalable to accommodate various tournament sizes, making it feasible and aligned with business objectives, thereby supporting the growth of Bangladesh's esports industry.

- The Esports Tournament Management System (ETMS) is a comprehensive platform designed to manage and streamline all aspects of esports tournaments. It provides a unified interface for players, teams, and organizers, making it easier to register, create teams, and participate in tournaments. Key features include user registration, tournament scheduling, real-time scoring, leaderboards, and tournament management, ensuring smooth operations and consistent results. Additionally, the ETMS offers robust security to protect user data and financial transactions. Overall, the system aims to enhance the tournament experience, promote the growth of the esports community, and support the development of a vibrant esports ecosystem in Bangladesh.

- Several software solutions for esports tournament management, like Challenge, Tournament, and Battle, offer various tools for organizing and managing tournaments. However, these platforms are often designed for a global audience and may not meet the unique needs of a localized region like Bangladesh. They can lead to increased costs and complexities due to international navigation requirements. The proposed Esports Tournament Management System (ETMS) aims to address these issues by providing a localized solution specifically tailored for Bangladesh. It integrates best practices from existing platforms while focusing on the specific needs of Bangladeshi esports organizers and players, promoting the growth of the local esports scene.

# 4. REQUEIREMNT SPECIFICATION

## 4.1 System Features

### 4.1.1 User Registration and Authentication

#### 4.1.1.1 Description

Allows players, team managers, and tournament organizers to register accounts securely and authenticate their access to the system.

**Priority Level**: High

**Precondition:** user have valid user id and password

#### 4.1.1.2 Functional Requirements

**FR1:** Provide a registration form for users to create accounts with necessary information (username, email, password).

**FR2:** Implement secure authentication mechanisms (verification, password) for login.

**FR3:** Allow password reset functionality through a verified email process.

#### 4.1.1.3 Non-Functional Requirements

**NFR1:** The system must comply with industry-standard encryption protocols to ensure secure transmission of sensitive data like passwords and personal information.

**NFR2:** The registration and authentication process should be intuitive and user-friendly, with clear instructions and error messages.

**NFR3**: The system should be modular and well-documented to facilitate easy maintenance and future enhancements by development teams.

### 4.1.2 Tournament Creation and Management

#### 4.1.2.1 Description

This feature empowers tournament organizers to create, manage, and schedule esports tournaments efficiently. It includes setting up tournament details, managing participants, and defining rules.

**Priority Level:** High

**Precondition:** User must have access to the tournament creation and management interface

#### 4.1.2.2 Functional Requirements

**FR4:** Provide an intuitive interface for organizers to create new tournaments, defining game titles, rules, schedules, and registration criteria.

**FR5:** Allow tournament organizers to set up brackets, manage team registrations, and update match schedules seamlessly.

#### 4.1.2.3 Non-Functional Requirements

**NFR4**: The system should have low latency when loading and updating tournament information, even when handling large datasets and concurrent user interactions.

**NFR5**: The system should be designed to scale horizontally to accommodate an increasing number of tournaments and users over time.

**NFR6**: Ensure high availability of the system, with a minimum uptime of 99.9%, to prevent disruptions to tournament scheduling and management.

### 4.1.3 Team Creation and Management

#### 4.1.3.1 Description

Allows players to form teams, manage team details, invite members, and handle team-related activities within the system.

**Priority Level**: Medium

**Precondition:** Users must have registered accounts on the system.

#### 4.1.3.2 Functional Requirements

**FR7:** Enable players to create and manage team profiles, including team name, logo, description, and roster management.

**FR8:** Implement team invitation functionality for team managers to invite players to join their teams or accept invitations.

**FR9:** Allow team managers to manage team members, handle substitutions, and update team information.

#### 4.1.3.3 Non-Functional Requirements

**NFR7**: The system should provide quick response times when loading team profiles and managing team-related activities, even with a large number of teams and members.

**NFR8**: The system should be capable of scaling horizontally to accommodate an increasing number of teams and users over time without sacrificing performance.

**NFR9**: The system should be modular and well-documented to facilitate easy maintenance and future enhancements by development teams.

### 4.1.4 Match Result Reporting and Recording

#### 4.1.4.1 Description

This feature enables participants to report match results accurately, ensuring fair play and transparent tournament progress tracking.

**Priority Level:** High

**Precondition:** Participants must have completed matches within the tournament.

#### 4.1.4.2 Functional Requirements

**FR10:** Enable participants to report match results within the system, including scores, match details, and any additional relevant information.

**FR11:** Implement validation mechanisms to verify match result submissions and resolve any discrepancies through a dispute resolution process.

**FR12:** Automatically update tournament brackets and standings based on reported match results to reflect accurate progress and rankings.

#### 4.1.4.3 Non-Functional Requirements

**NFR10:** The system should provide prompt response times when participants report match results, ensuring real-time updates to tournament progress.

**NFR11:** Implement encryption for match result data transmission to protect sensitive information from unauthorized access or tampering.

**NFR12:** Implement load balancing mechanisms to distribute processing load evenly across servers and maintain system performance under heavy usage.

### 4.1.5 Payment and Registration Processing

#### 4.1.5.1 Description

Facilitates secure payment processing for tournament registration fees, ensuring seamless user registration and fee collection.

**Priority Level:** High

**Precondition:**

#### 4.1.5.2 Functional Requirements

**FR13:** Enable participants to register for tournaments by securely paying registration fees through integrated payment gateways.

**FR14:** Provide a seamless registration confirmation process upon successful payment, granting access to the tournament and validating participation.

#### 4.1.5.3 Non-Functional Requirements

**NFR13:** The system must comply with Payment Card Industry Data Security Standard (PCI DSS) requirements to ensure the secure handling of payment card information.

**NFR14:** Implement end-to-end encryption for payment transactions to protect sensitive data from unauthorized access during transmission.

### 4.1.6 Dashboard and Analytics

#### 4.1.6.1 Description

Offers tournament organizers and administrators a dashboard view and analytical insights into tournament statistics and user engagement.

**Priority Level:** Low

**Precondition:** Tournament data must be up-to-date and accurate**.**

#### 4.1.6.2 Functional Requirements

**FR15:** Develop an intuitive dashboard interface displaying tournament statistics, registration metrics, match progress, and user participation.

**FR16:** Generate comprehensive analytics reports showcasing user engagement metrics, tournament success rates, team performances, and other relevant data for informed decision-making.

#### 4.1.6.3 Non-Functional Requirements

**NFR15:** The dashboard should load quickly, providing instant access to tournament statistics and analytics data to minimize user wait times.

**NFR16:** The dashboard and analytics system should be scalable to handle increasing amounts of data as the number of tournaments and users grows over time.

### 4.1.7 Communication and Notifications

#### 4.1.7.1 Description

Facilitates communication between participants, team managers, and organizers through in-app messaging and automated notifications.

**Priority Level:** Low

**Precondition:** Participants must be enrolled or registered in tournaments.

#### 4.1.7.2 Functional Requirements

**FR17:** Implement an in-app messaging and voice calling system allowing users to communicate within the system, facilitating discussions, and coordination.

**FR18:** Configure automated notifications for match schedules, tournament updates, registration confirmations, and essential announcements to keep participants informed.

#### 4.1.7.3 Non-Functional Requirements

**NFR17:** The communication system should be scalable to support a growing user base and increasing message traffic as the popularity of tournaments increases.

**NFR18:** Ensure high availability of the communication system, with minimal downtime to prevent disruptions to participant communication and coordination.

## 4.2 System Quality Attributes

### 4.2.1 Usability

**QA-1:** The user interface shall be intuitive, allowing users to perform common tasks without extensive training.

**QA-2:** The system shall support multiple languages for a diverse user base.

**QA-3:** A user shall be able to login and search for what they want in under one minute or in two and a half minutes maximum.

### 4.2.2 Performance

**QA-5:** The system shall load tournament information and user dashboards within 3 seconds.

**QA-6:** During peak times, the system should handle concurrent user actions without significant latency.

### 4.2.3 Security

**QA-7:** User authentication shall be mandatory for accessing sensitive data or performing critical actions.

**QA-8:** All communication channels including voice calls must be encrypted to prevent unauthorized access.

### 4.2.4 Efficiency

**QA-9:** the platform should use 100 to 300 Mb RAM per tab and 0.1% to 1% of CPU usage apart from the browser’s usage depending on the user usage.

### 4.2.5 Maintainability

**QA-10:** The system's codebase shall follow standard coding practices and be well-documented.

**QA-11:** System updates and patches shall be easily deployable without disrupting ongoing tournaments.

### 4.2.6 Scalability

**QA-12:** The system should accommodate a minimum of 10,000 concurrent users during peak tournament times.

### 4.2.7 Compatibility

**QA-13:** The system shall be compatible with major web browsers (Chrome, Firefox, Safari, Edge) and mobile devices (iOS, Android).

## 4.3 System Interface

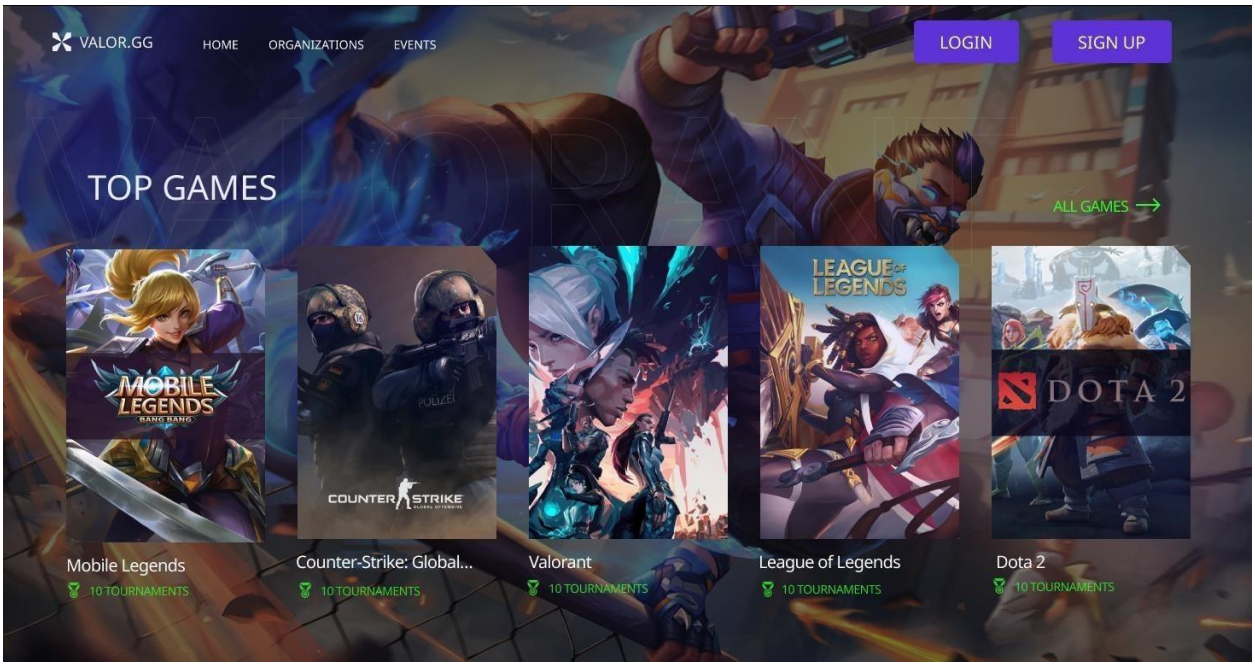



FIGURE:TOP GAME SECTION

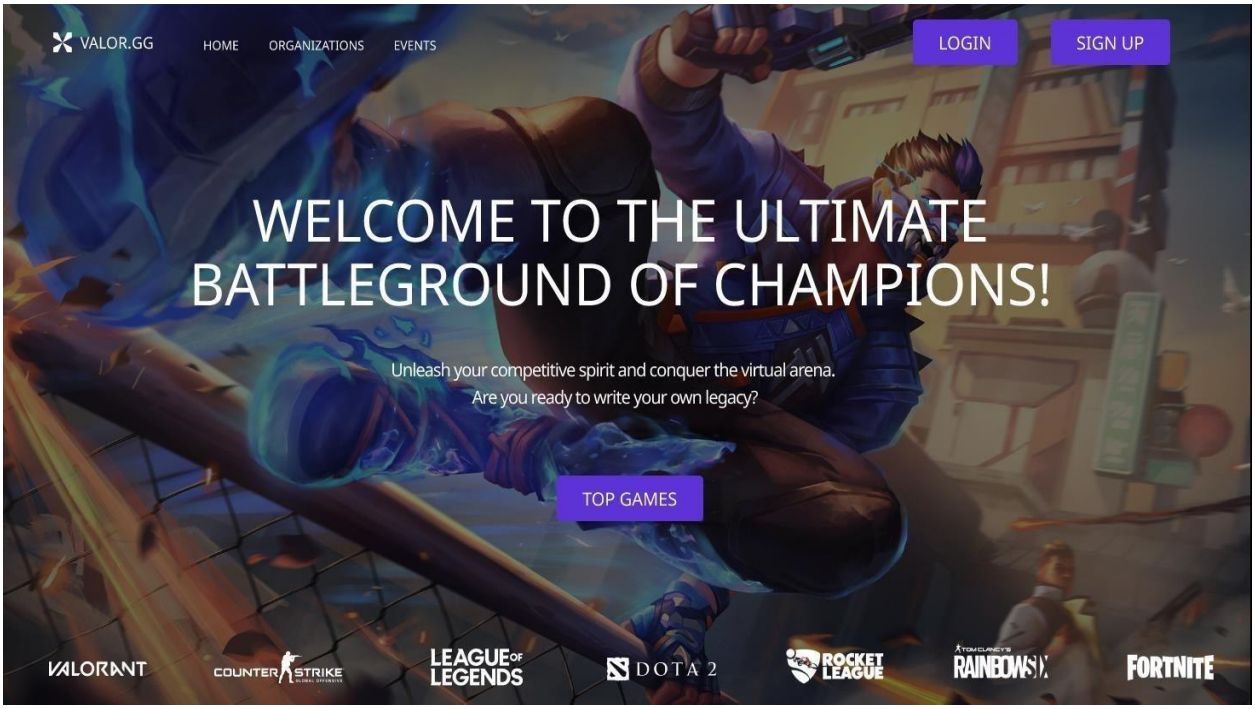



FIGURE: LANDING PAGE

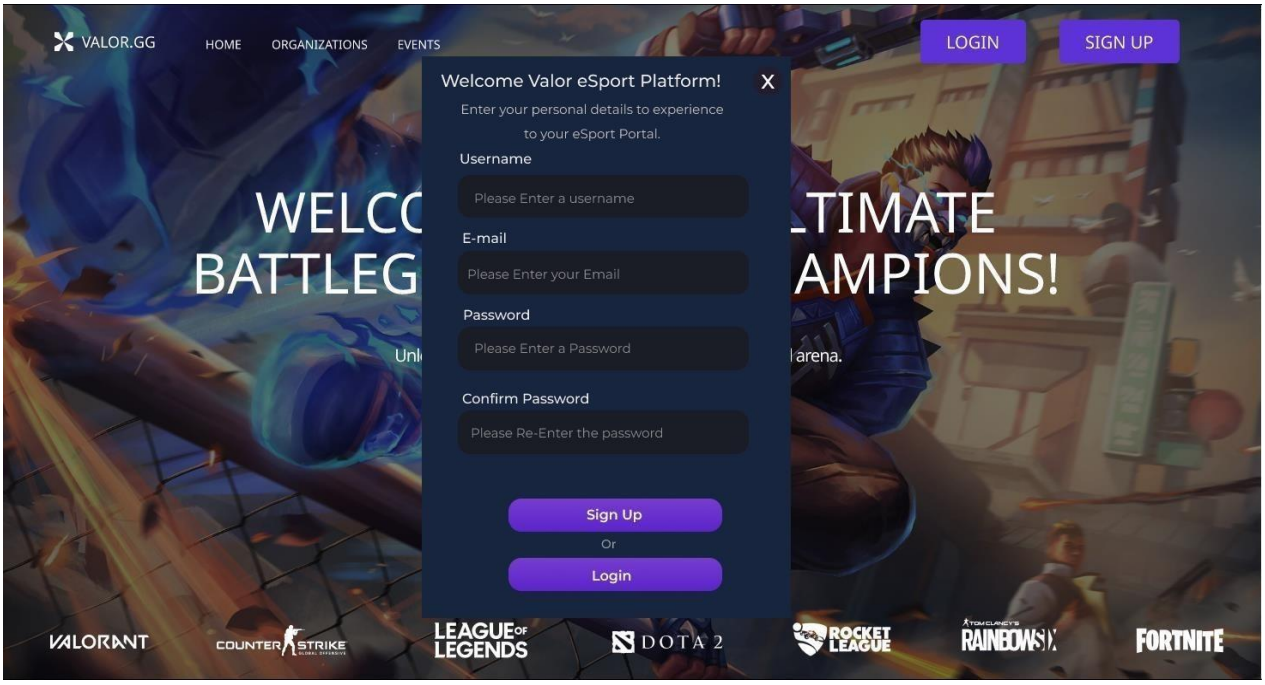



FIGURE: SIGNUP PAGE

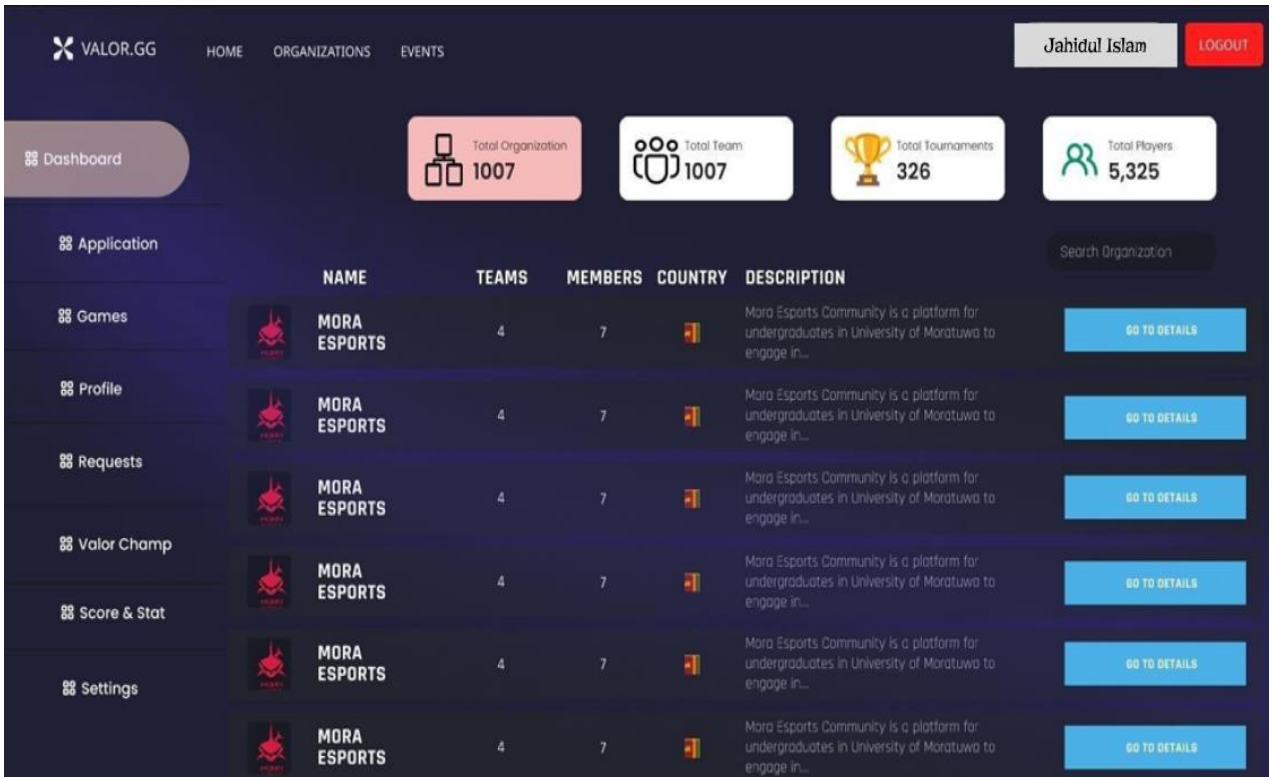



FIGURE: ADMIN DASHBOARD

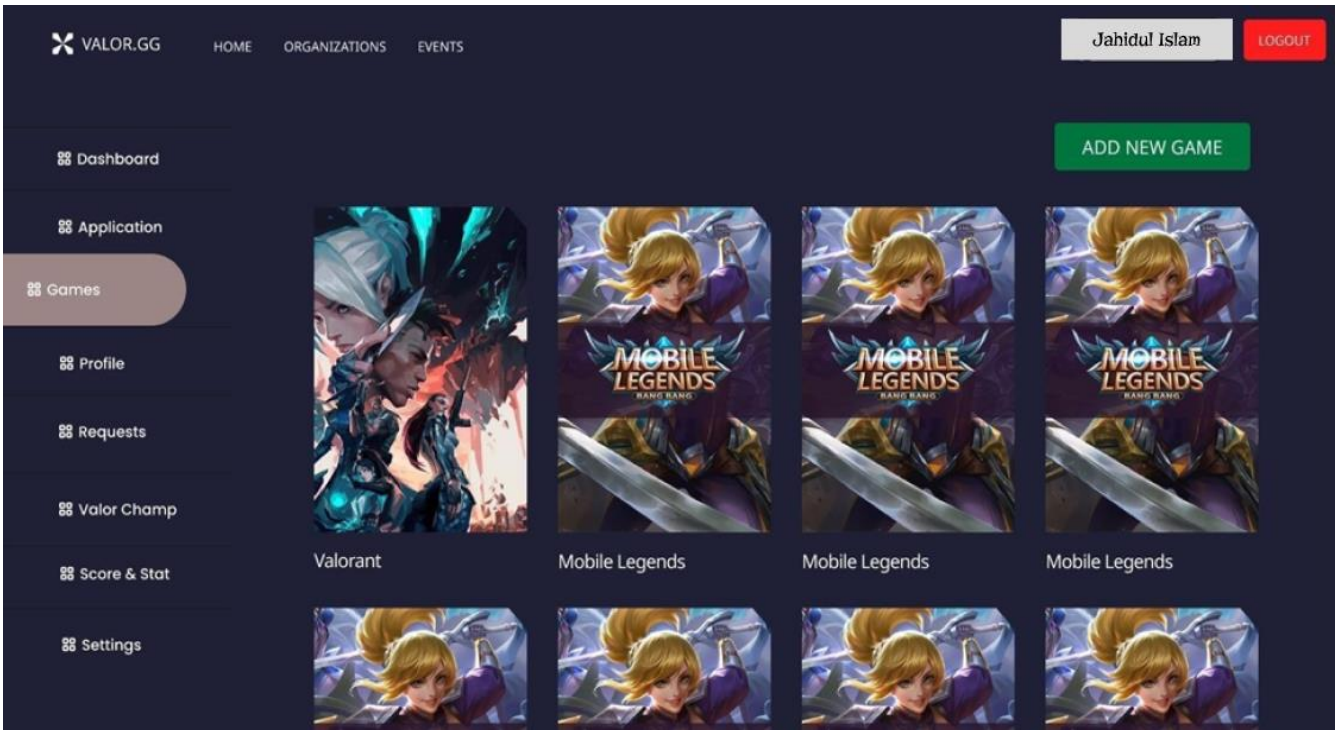



FIGURE: GAME MANAGING PAGE

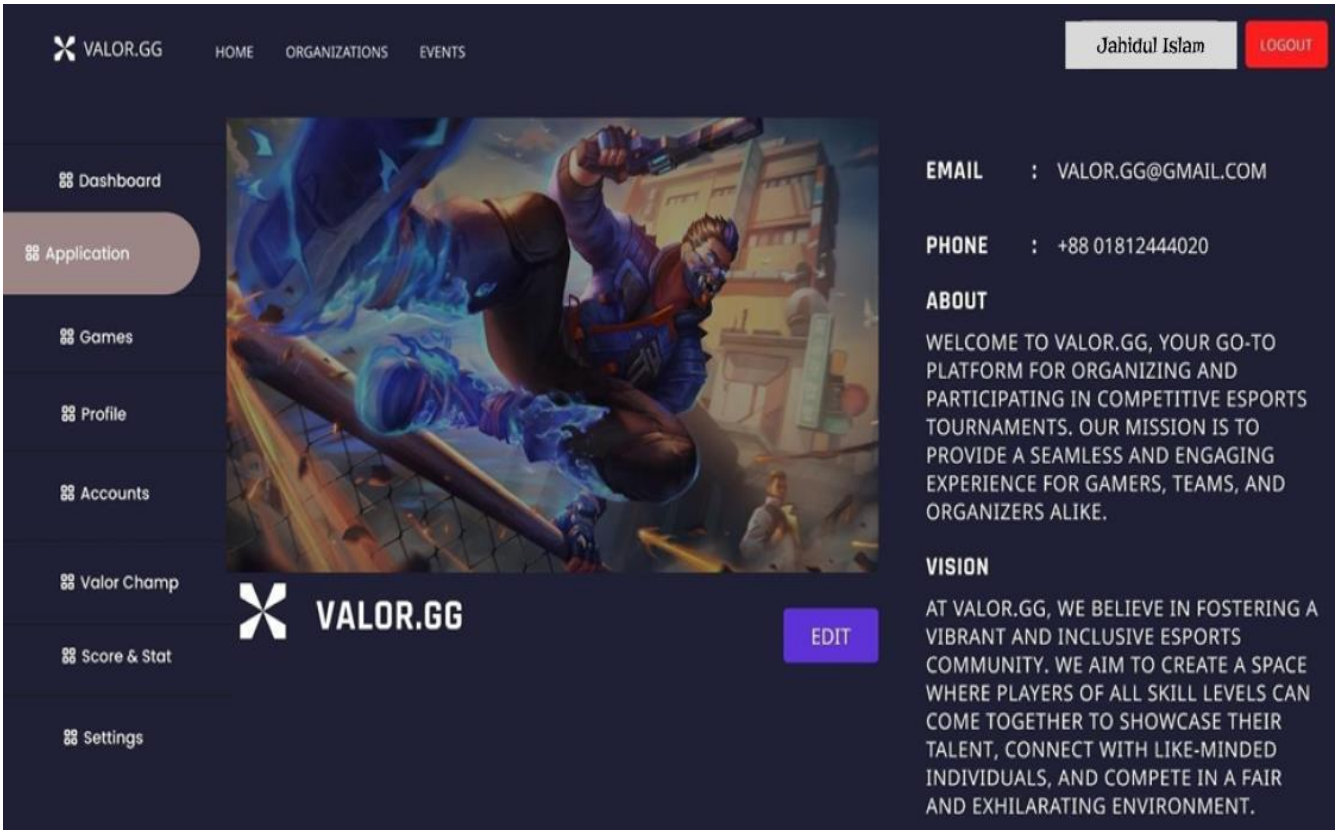



FIGURE: SYSTEM CONTROL PAGE

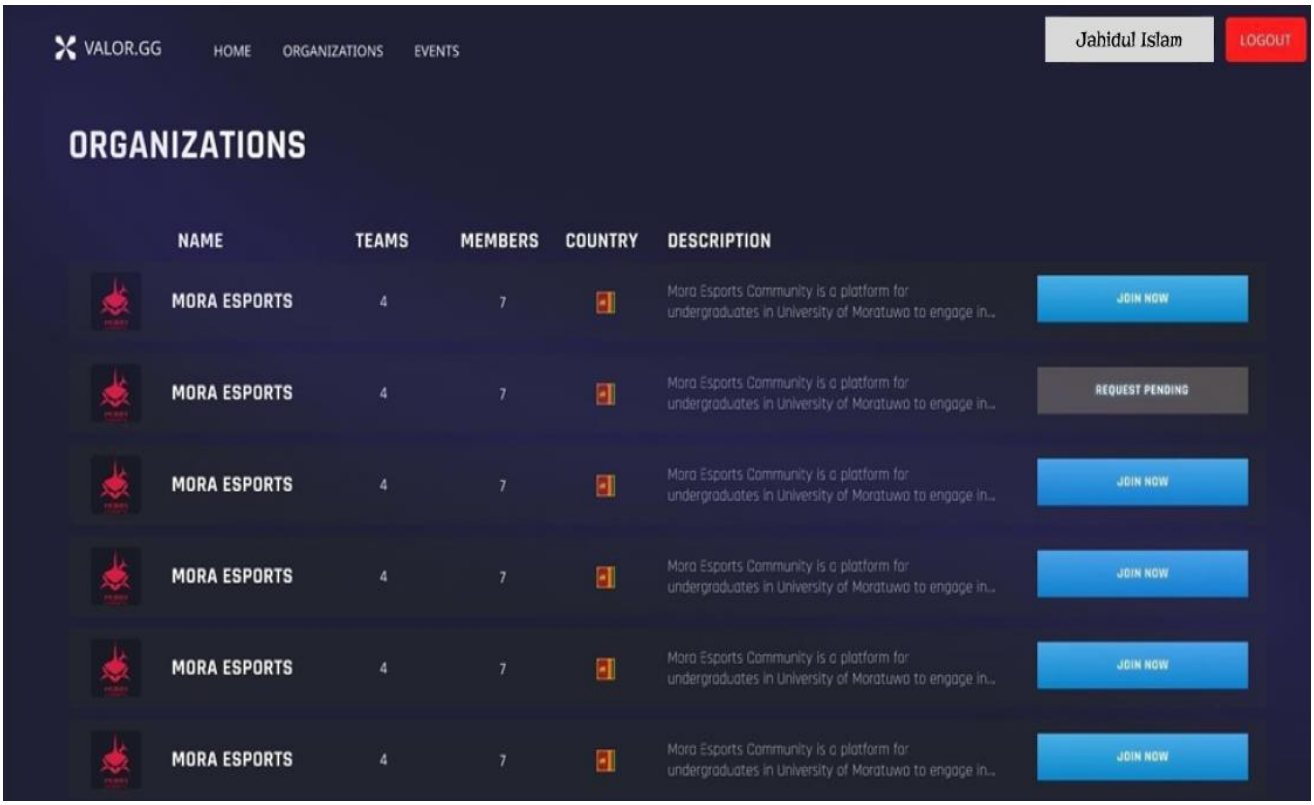



FIGURE: ORGANIZATION BROWSING PAGE

## 4.4 Project Requirements

**Environment & resource needs**

- Microsoft Windows 10 with Service Pack 1 or higher.
- Intel Core i5 or higher. (The 7th generation)
- 4 GB of RAM (minimum).
- 128 GB of free disk space for installation, plus extra space for temporary files during test runs.
- Mouse and keyword.

**Software Requirements:**

- Automated Testing tool: selenium.
- Web Browsers: Internet Explorer/ Mozilla Firefox/ Google Chrome.
- Database: MySQL 5.7.12, 8.0.12+ (see note below on MySQL 8 support. PostgreSQL 10.0+.
- Eclipse, Microsoft Visual Studio code.

**Effort Estimation:**

We assumed that as per our project requirement the source code can be 18000 lines.

So, our SLOC = 18000

So, we can find out how much time and how many people are required for complete this project by

COCOMO Rule

SLOC = 18000

Coefficient = 2.4

Project complexity, P = 1.05

SLOC-dependent coefficient, T = 0.38 (for Organic type)

Effort = PM = Coefficient<Effort Factors> *($SLOC$/1000) p

= 2.4 * (18000/1000) ^1.05

= 49.92

≅ 50

Development time = DM = 2.50*(PM)T

= 2.50 * (PM) ^ 0.38

= 11 Months

Required number of people, ST = PM / DM

Page 13 of 16

= 50/11

= 4.55

≅ 5

So, we need 11 months (44 weeks) and 5 members for development this software.

**Budget Estimation:**

i. Project development time = 11 Months
Number of developers will work = 5
Working days= 5 Day
Working hour per day= 8 Hours
Working hour in 1 week= (5*8) = 40 Hours
Charge for each developer per hour = 300 TK

Charge for each developer
Per week = (300*40) = 12,000 TK
For a month = (12000*4) = 48,000 TK
For 11 months = (48000*11) = 528,000 TK
Charge for 6 developers for 11 months = (528,000 * 5) = 2,640,000 TK
**Total Developer Cost**: 2,640,000 TK

ii. Project manager charge for 11 months = (11*60000) = 660,000 TK
Other employees charge for 11 month = 300,000 TK
**Total Management Cost:** 960,000 TK

iii. Office rent for 11 months= (11*30000) = 330,000 TK
Server and Hosting for 1 years = 200,000 TK
Electricity and other bills = 200,000 TK
**Total infrastructure Cost:** 730,000 TK

iv. Development Tools = 60,000 TK
Third-Party APIs = 100,000 TK
Software Licenses = 50,000 TK
**Total Software and Licensing Cost:** 210,000 TK

v. **Total Estimated Cost** = Developer Cost + Management Cost + Infrastructure Cost + Software/Licensing Cost = 2,640,000 TK + 960,000 TK + 730,000 TK + 210,000 TK = 4,540,000 TK

vi. **Total Estimated Budget:** 4,540,000 TK

## 5. FEATURES NOT TO BE TESTED

The following is a list of the areas that will not be specifically addressed. All testing in these areas will be indirect as a result of other testing efforts. For example:

- **Third-Party Integrations**: Features that rely on third-party services, such as weather widgets or external maps. As these integrations are often provided and maintained by external entities, their behavior is not directly under our control and does not require testing within our system.
- **Standardized UI Elements**: UI components that follow established design patterns and conventions, such as buttons, navigation menus, and headers. These elements are consistently present throughout the system and are unlikely to exhibit unexpected behavior.

- **System Start-Up/Shut Down**: Basic start-up and shut-down processes of the application might not need extensive testing unless they are known to have complex dependencies.
- **Common Browser Compatibility**: If your application is being developed for specific browsers and devices and follows modern web standards, you might not need to test every minor browser version for compatibility.

## 6. TESTING APPROACH

### 6.1 Testing Levels

- In the testing approach for the Esports management system, we will implement a multi-level testing strategy to ensure the comprehensive validation and verification of the system's functionality, performance, and usability. This strategy encompasses several testing levels that together contribute to the overall quality of the system.
- **Unit Testing:** At the foundational level, unit testing will be performed to scrutinize individual components and modules of the system. This involves testing the smallest units of code to ensure their correctness and functionality. Unit tests will be written to verify that each module operates as intended and handles various inputs and scenarios appropriately.
- **Integration Testing:** Moving beyond individual units, integration testing comes into play. This level of testing focuses on examining how different modules interact and collaborate within the system. The goal is to identify any inconsistencies or miscommunications between modules and ensure that they integrate seamlessly.
- **Functional Testing:** Functional testing forms a core component of our approach. This level of testing assesses the system's functional requirements, ensuring that each feature and functionality works as specified. It encompasses various scenarios, inputs, and interactions to validate that the system behaves as expected.
- **Usability Testing:** Usability testing is crucial to gauging how well the system meets user expectations and needs. Real users will interact with the system to assess its overall usability, identifying any pain points, confusing elements, or areas that require improvement.
- **Performance Testing:** Performance testing will assess the system's responsiveness and stability under different load conditions. This level of testing includes load testing, stress testing, and scalability testing to determine how the system performs under varying levels of user traffic.
- **Security Testing:** Security testing will be employed to identify vulnerabilities, safeguard sensitive data, and ensure the system's resilience against potential threats. This level of testing aims to maintain the confidentiality, integrity, and availability of user and system data.
- **User Acceptance Testing (UAT):** UAT is the final level of testing and involves real users testing the system in a real-world environment. Users will assess whether the system meets their requirements, ensuring that it aligns with their expectations and needs.

## 6.2 Test Tools

The Selenium IDE Tool will be used for automated testing. We utilize this instrument to discover errors and verify that our systems are of high quality, responsive, progressive, and consistent.

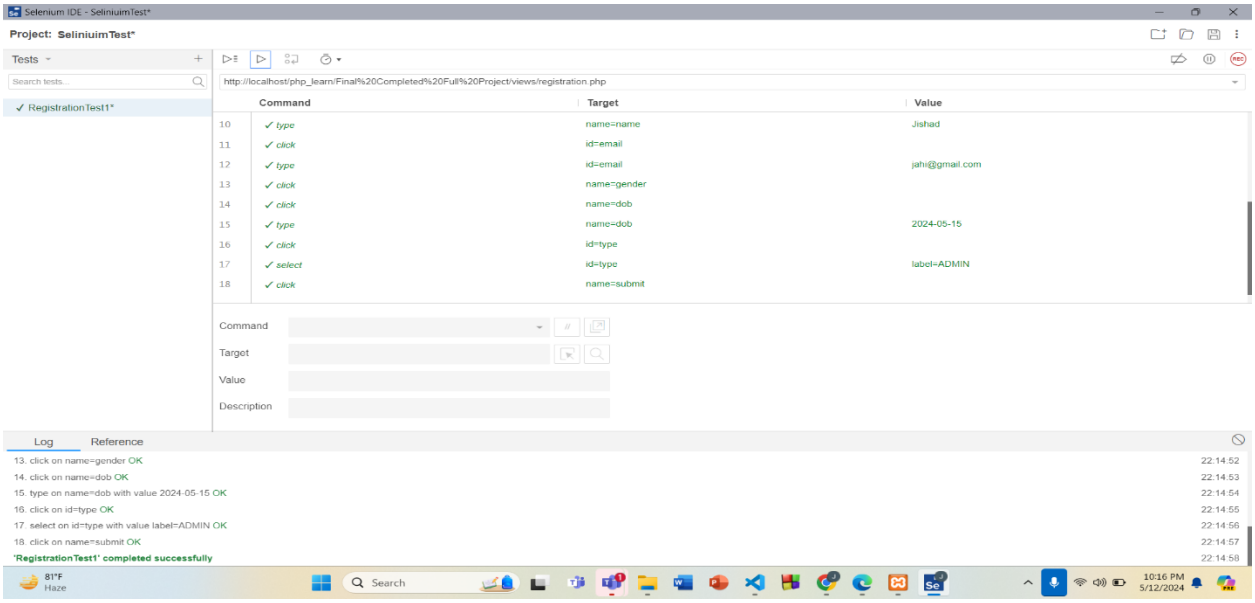



FIGURE: REGESTRATION MODULE

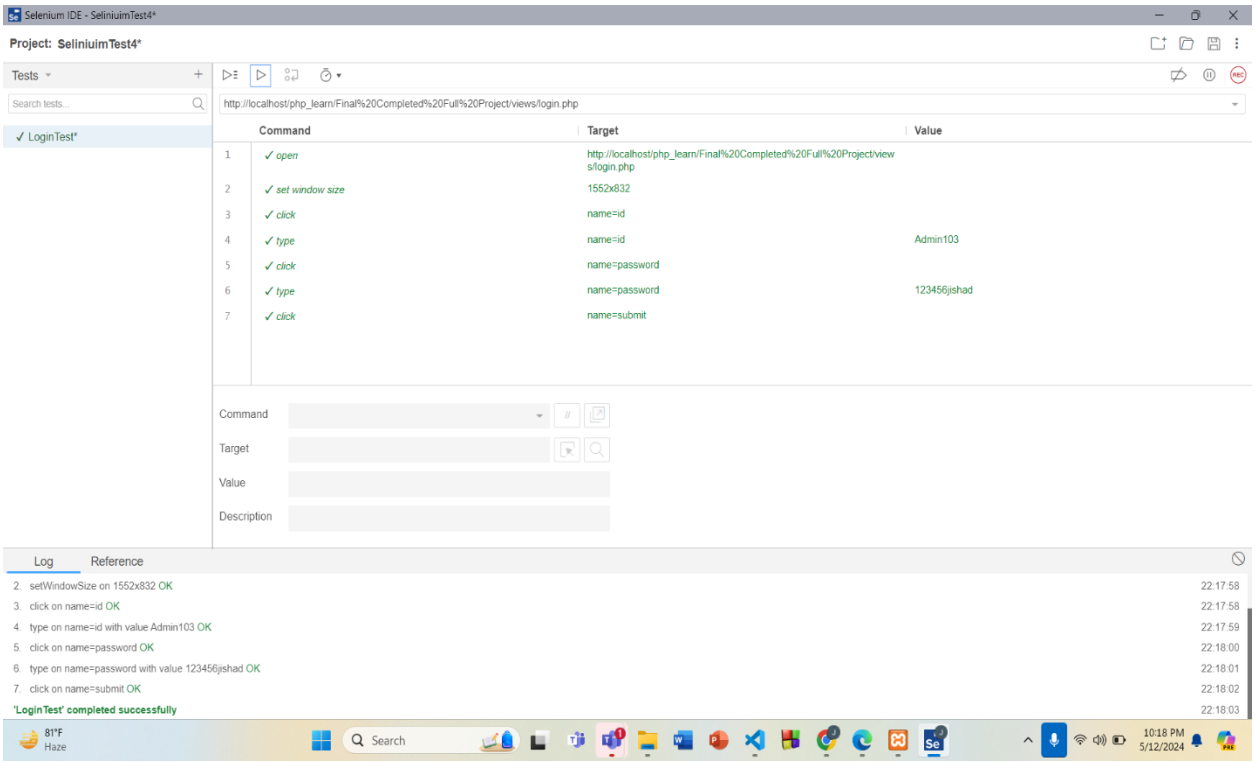



FIGURE: LOG IN MODULE

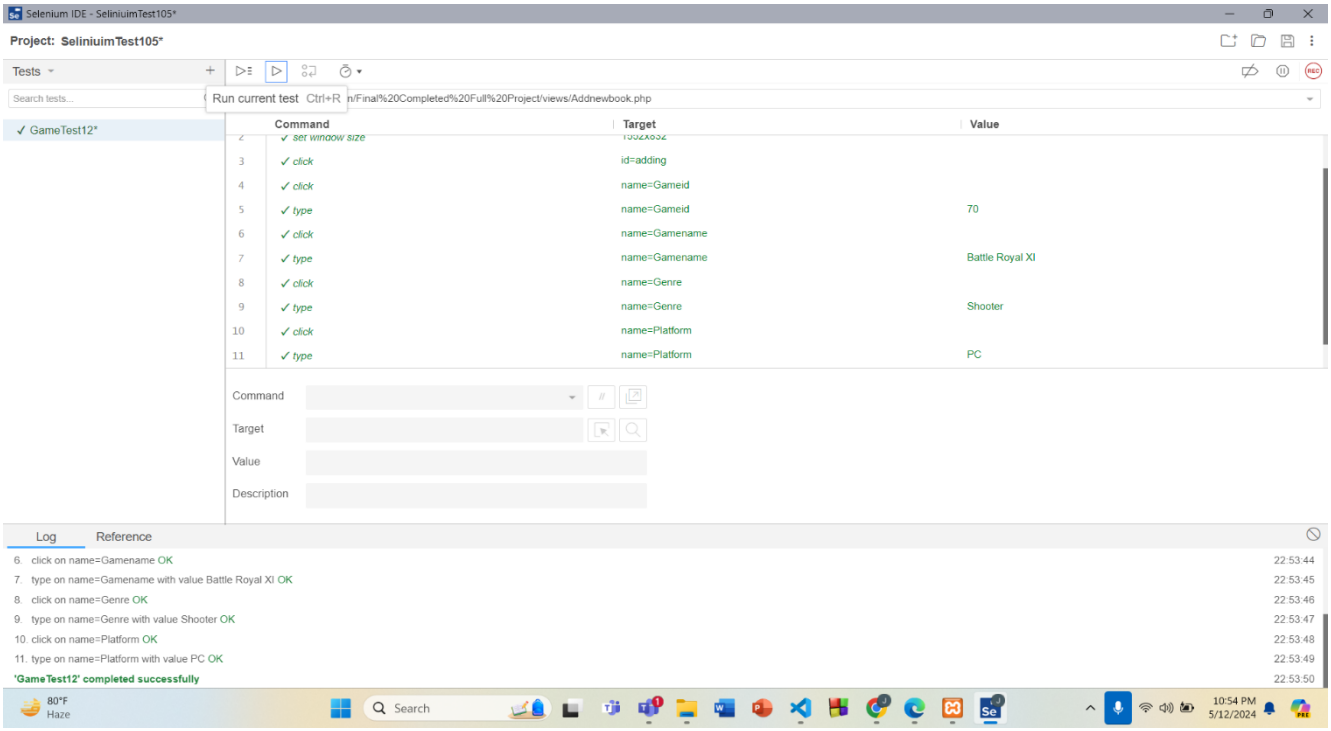



FIGURE: GAME MODULE

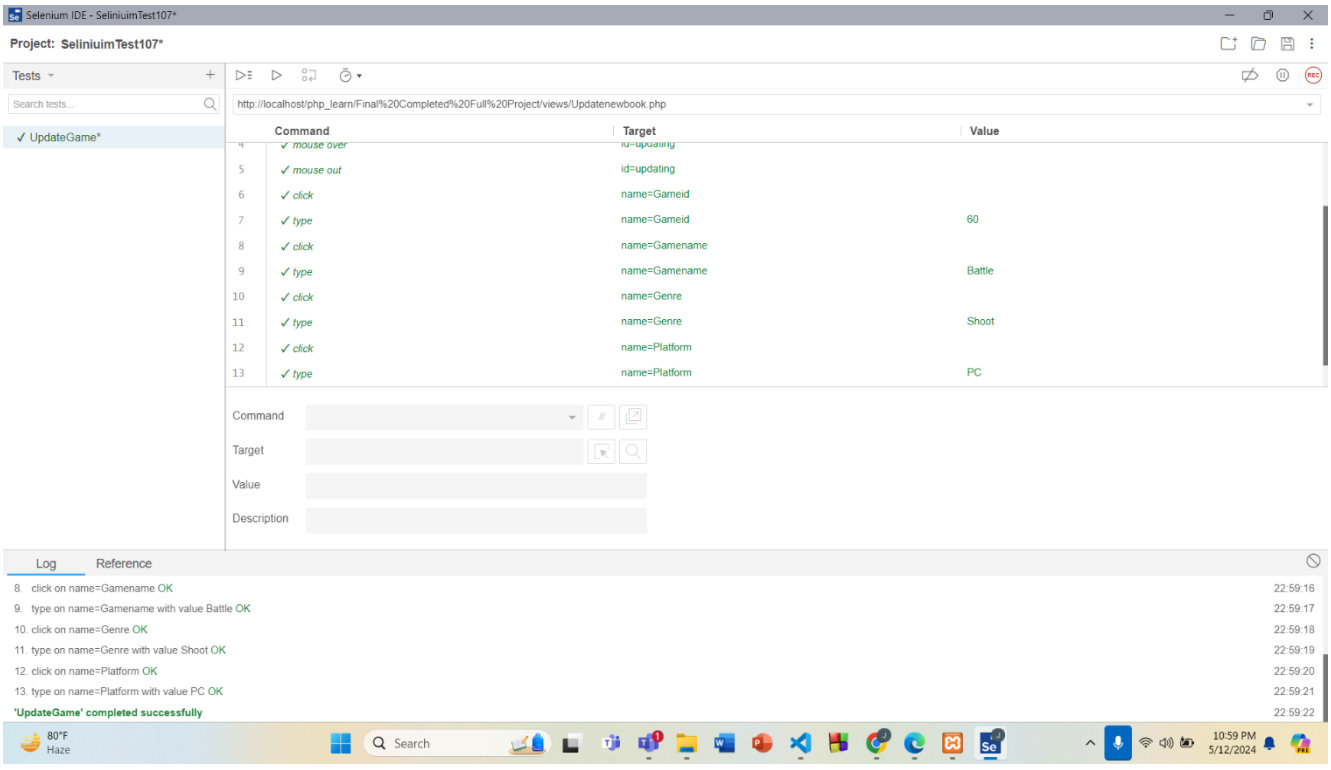



FIGURE: UPDATE MODULE

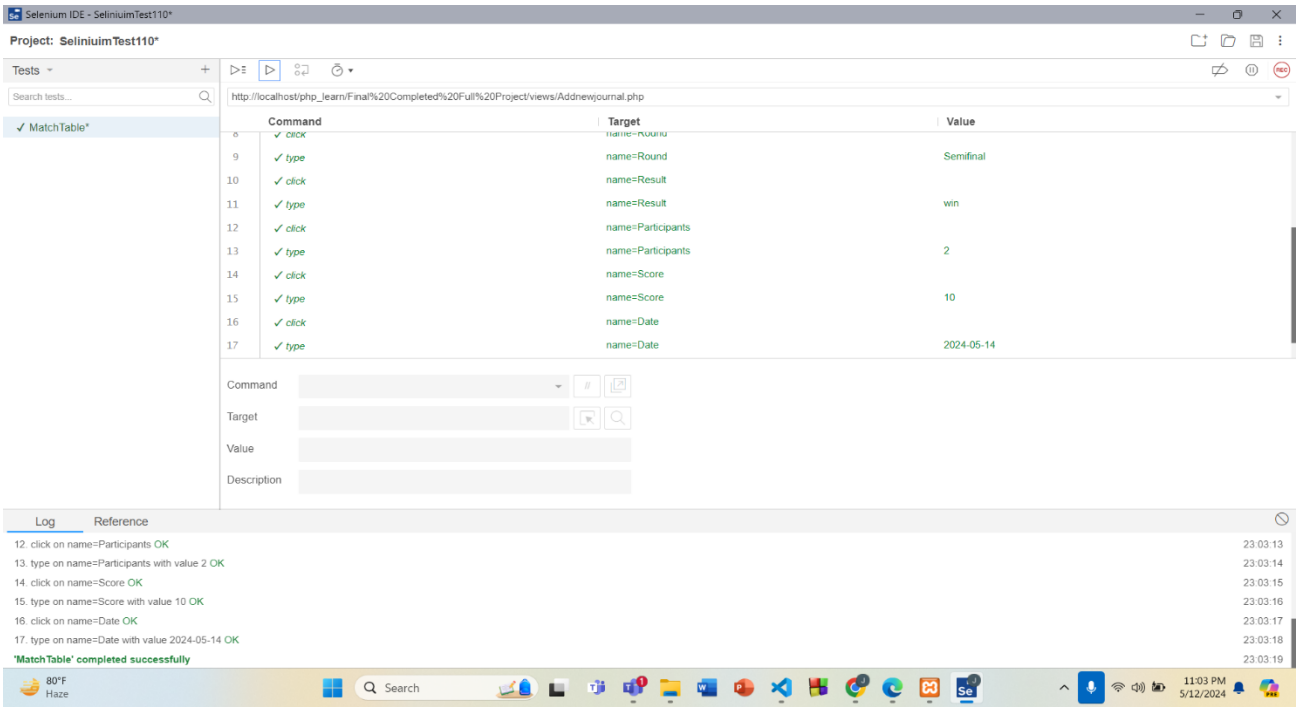



FIGURE: MATCH MODULE

## 6.3 Meetings

The test team will meet once in every week to evaluate progress to date and to identify error trends and problems as early as possible. The test team leader will meet with development and the project manager once every two weeks as well. These two meetings will be scheduled on different weeks. Additional meetings can be called as required for emergency situations.

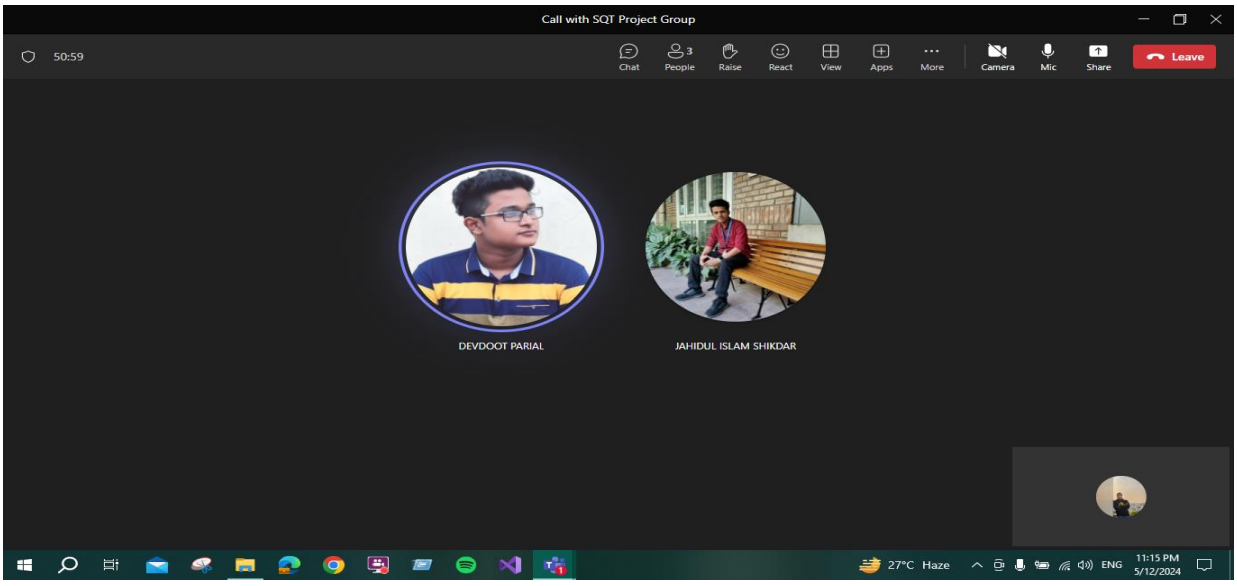



FIGURE: GROUP MEETING

## 7. TEST CASES/TEST ITEMS

| | | | | |
|---|---|---|---|---|
| Project Name: User Registration and authentication | | | Test Designed by: Md. Aminul Islam | |
| Test Case_ID: UA_1 | | | Test Designed date: 07/05/2024 | |
| Test Priority (Low,Medium,High): High | | | Test Executed by: Jahidul Islam Shikdar | |
| Test Title: Verify user registration and authentication process | | | Test Execution date: 07/05/2024 | |
| Description: Test the user registration and authentication functionality of the system. | | | | |
| Precondition: The user is on the registration/authentication page. | | | | |
| Test Steps | Test Data | Expected Results | Actual Result | Status (Pass/Fail) |
| 1. Navigate to the registration page.<br>2. Enter valid user information.<br>3. Click on the "Register" button.<br>4. Verify that the user is authenticated to the system. | ID: Admin103<br>PASSWORD: 123jishad | After registration, the user should be redirected to the login page. | The user is successfully registered and redirected to the login page. | Pass |

FIGURE : TEST CASE FOR AUTHENTICATION

| Project Name: Tournament Creation and Management | | | Test Designed by: Jahidul Islam Shikdar | |
|---|---|---|---|---|
| Test Case_ID: TM_1 | | | Test Designed date: 07/05/2024 | |
| Test Priority (Low,Medium,High): High | | | Test Executed by: Devdoot Parial | |
| Test Title: Verify tournament creation and management functionalities | | | Test Execution date: 07/05/2024 | |
| Description: Test the functionality of creating, managing, and scheduling esports tournaments. | | | | |
| Precondition: The system is accessible, and the user has organizer privileges. | | | | |
| Test Steps | Test Data | Expected Results | Actual Result | Status (Pass/Fail) |
| 1. Log in to the system as a tournament organizer.<br>2. Navigate to the tournament creation section and Enter tournament details.<br>3. Click on the "Create Tournament" button and verify that the tournament is successfully created.<br>4. Navigate to the tournament management section and Set up brackets for the tournament. | ID: Admin103<br>PASSWORD: 123jishad | The tournament is successfully created and listed in the system.<br><br>Brackets are set up correctly and reflect the tournament structure. | . The tournament is successfully created and listed in the system.<br><br>Brackets are set up correctly and reflect the tournament structure. | Pass |

FIGURE : TEST CASE FOR TOURNAMENT CREATION AND MANAGEMENT

| Project Name: Team Creation and Management | | | Test Designed by: Devdoot Parial | |
|---|---|---|---|---|
| Test Case_ID: TC_2 | | | Test Designed date: 07/05/2024 | |
| Test Priority (Low,Medium,High): Medium | | | Test Executed by: Md. Aminul Islam | |
| Test Title: Verify team creation and management functionalities | | | Test Execution date: 07/05/2024 | |
| Description: Test the functionality of creating and managing teams, including team profiles, member invitations, and roster management. | | | | |
| Precondition: The system is accessible, and the user has player or team manager privileges. | | | | |
| Test Steps | Test Data | Expected Results | Actual Result | Status (Pass/Fail) |
| 1. Log in to the system as a player or team manager.<br>2. Navigate to the team creation section and Enter team details.<br>3. Click on the "Create Team" button and verify that the team is successfully created.<br>4. Navigate to the team management section and manage team members, including handling substitutions and updating member information. | ID: User103<br>PASSWORD: 123jishad | The team is successfully created and listed in the system.<br><br>Team members are managed effectively, and changes are updated accordingly in the team profile. | The team is successfully created and listed in the system.<br><br>Team members are managed effectively, and changes are updated accordingly in the team profile. | Pass |

FIGURE : TEST CASE FOR TEAM CREATION AND MANAGEMENT

| | | | | |
|---|---|---|---|---|
| Project Name: Payment and Registration Processing | | | Test Designed by: Jahidul Islam Shikdar | |
| Test Case_ID: PR_1 | | | Test Designed date: 09/05/2024 | |
| Test Priority (Low,Medium,High): High | | | Test Executed by: Md. Aminul Islam | |
| Test Title: Verify payment and registration processing functionalities | | | Test Execution date: 09/05/2024 | |
| Description: Test the functionality of processing tournament registration fees and confirming participant registrations upon successful payment. | | | | |
| Precondition: A tournament with registration fees is open for registration and Integrated payment gateways are configured and operational. | | | | |
| Test Steps | Test Data | Expected Results | Actual Result | Status (Pass/Fail) |
| 1. Log in to the system as a participant.<br>2. Navigate to the tournament registration section and Enter registration details.<br>3. Choose a payment method and complete the payment process by confirming the transaction.<br>4. Verify that the payment is processed successfully, and registration is confirmed. | ID: Admin103<br>PASSWORD: 123jishad | The payment is processed successfully, and registration is confirmed upon successful payment. | The payment is processed successfully, and registration is confirmed upon successful payment. | Pass |

FIGURE : TEST CASE FOR PAYEMENT

| | | | | |
|---|---|---|---|---|
| Project Name: Communication and Notifications | | | Test Designed by: Devdoot Parial | |
| Test Case_ID: CN_1 | | | Test Designed date: 09/05/2024 | |
| Test Priority (Low,Medium,High): Low | | | Test Executed by: Md. Aminul Islam | |
| Test Title: Verify communication and notifications functionalities | | | Test Execution date: 09/05/2024 | |
| Description: Test the functionality of in-app messaging, voice calling system, and automated notifications. | | | | |
| Precondition: Match schedules, tournament updates, and other relevant information are available for notifications. | | | | |
| Test Steps | Test Data | Expected Results | Actual Result | Status (Pass/Fail) |
| 1. Log in to the system as a participant, team manager, or organizer.<br>2. Navigate to the messaging and notifications section and verify that the in-app messaging system allows users to send messages and initiate voice calls within the system.<br>3. Check the automated notifications configuration.<br>4. Verify that the notification is sent to the intended recipients. | Check communication section | The in-app messaging system allows users to send messages and initiate voice calls effectively.<br><br>Notifications are sent to the intended recipients promptly. | The in-app messaging system allows users to send messages and initiate voice calls effectively.<br><br>Notifications are sent to the intended recipients promptly. | Pass |

FIGURE : TEST CASE FOR COMMUNICATION AND NOTIFICATIONS

## 8. ITEM PASS/FAIL CRITERIA

The entrance criteria for each step of testing must be met before proceeding to the subsequent phase. The

criteria for passing and failing are listed here.

• In accordance with the stated scenario, the expected outcome must occur for the design to be deemed successful. Otherwise, this criterion must be failed.

• If an item any feature is tested over 10 times, but fails then it will be told as a failed case.

• Crashing of the system will be deemed a failure scenario

• After submitting any query if the system doesn't show the results, then it will be counted as a failed case.

## 9. TEST DELIVERABLES

- Test Plan
- Test Cases
- Test Scripts
- Test Data
- Execution Log
- Defect Report
- Test Summary Report
- Screen prototypes

## 10. STAFFING AND TRAINING NEEDS

This project is also implemented for personal and test job integration. To generate everything decently at least we are recommended to have one – full timer tester for the purpose. The job will find challenging because of its execution process and working dedications. Some challenging stuffs and job task are discussed here below:

**o Project manager:** Responsible for the overall project execution. The project manager leads the team to work from every side. Even in any critical situation, the project manager has to play a vital role to safe the project from getting dismissed. The project manager is included for the

feasibility study to ending the project and delver to the client. So the project manager must have some quality experience and training in this sector.

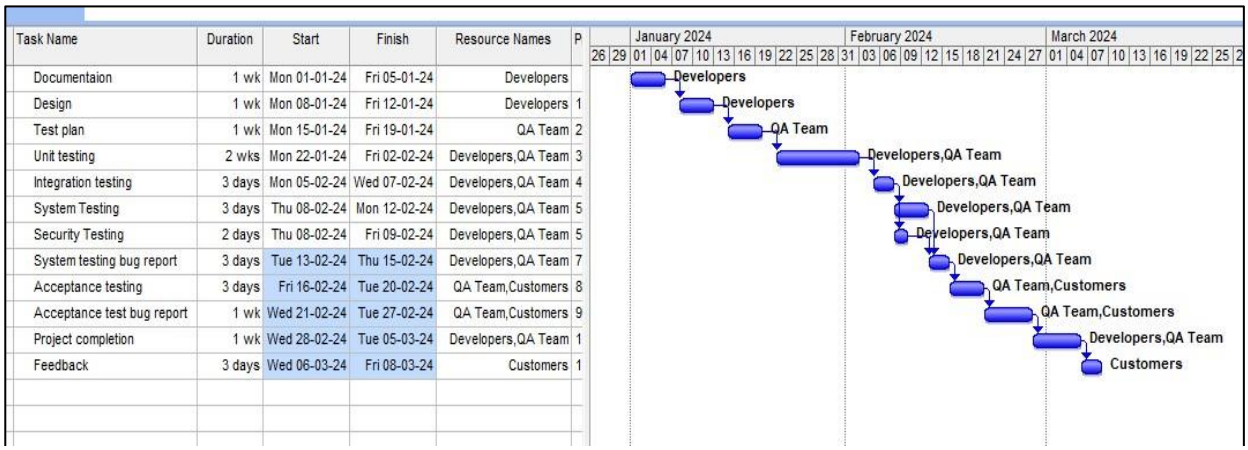
**o The Manager:** Responsible for creating expert test strategies, evaluating managing test cycles and recommending testing completion. The Manager must be qualified to evaluate professional standard test design and find out the needs of client.

**o The developers and Engineers:** Response for designing and implementing the knowledge for design tests, creating methods, preparing test data's, executing the tests, Conducting automated test strategies and providing the test administrator with measurement information. Test engineers should be able to plan and execute any test case using automated technologies.

## 11. RESPONSIBILITIES

| | TM | PM | Dev Team | Test Team | Client |
|---|---|---|---|---|---|
| Test cases documentation | X | X | X | X | |
| Test Procedures and rules | X | | X | X | |
| Unit test documentation & execution | | | X | X | |
| Integration test Documentation & Execution | X | | X | X | |
| System test Documentation & Execution | | X | | X | |
| System Design Reviews | X | X | X | X | X |
| Details Design Reviews | X | X | X | X | |
| Screen & Report prototype reviews | X | X | | X | X |
| Change Control and regression testing | X | X | X | X | X |
| Acceptance test Documentation & Execution | X | X | | X | X |

## 12. TESTING SCHEDULE



## 13. PLANNING RISKS AND CONTINGENCIES

As we embark on the development of the Esports management system, it's essential to recognize potential risks that could impact the project's timeline, budget, and overall success. Mitigating these risks with well-thought-out contingencies will help ensure a smoother project execution. Here, we outline some of the key planning risks and the corresponding contingency strategies:

**Technical Challenges:**

- **Risk:** Unforeseen technical complexities or integration issues might arise during the development of the Esports management system.
- **Contingency:** Allocate additional time in the project schedule for comprehensive technical assessments and testing. Maintain close collaboration with technical experts to address challenges promptly.

**Scope Creep:**

- **Risk:** As the project progresses, new feature requests or changes might emerge, expanding the system's scope beyond the initial plan.
- **Contingency:** Establish a robust change management process that requires formal approval for scope changes. Regularly review and prioritize new requests to prevent scope creep.

**Resource Constraints:**

- **Risk:** Shortages in human resources, technical expertise, or budget constraints might hinder project progress.
- **Contingency:** Cross-train team members to handle various roles, explore outsourcing options for specialized tasks, and maintain open communication with stakeholders regarding budget adjustments.

**Vendor or Third-Party Dependencies:**

- **Risk:** The project's progress might be affected if external vendors or third-party services experience delays or disruptions.
- **Contingency:** Identify backup vendors or alternative solutions in case of vendor-related delays. Maintain constant communication with third parties to monitor progress and anticipate potential issues.

**Stakeholder Misalignment:**

- **Risk:** Disagreements or misunderstandings among stakeholders regarding project goals, requirements, or priorities could lead to project delays.

- **Contingency:** Establish a robust communication plan that facilitates continuous stakeholder engagement. Conduct regular progress reviews to ensure alignment and promptly address any concerns.

**Unanticipated Regulatory Changes:**

- **Risk:** Changes in regulatory requirements or compliance standards might necessitate adjustments to the Esports management system.
- **Contingency:** Stay informed about relevant regulations and design the system to accommodate potential changes. Maintain flexibility in the project plan to address regulatory adjustments.

**User Adoption Challenges:**

- **Risk:** Users might encounter difficulties in adopting the new system, leading to lower satisfaction and utilization rates.
- **Contingency:** Develop a comprehensive user training and onboarding plan. Conduct usability testing with real users to identify pain points and refine the user experience.

**Data Security Breaches:**

- **Risk:** Data breaches or security vulnerabilities could compromise sensitive user information and damage the system's reputation.
- **Contingency:** Implement robust security measures, conduct regular security audits, and establish incident response protocols to address and mitigate security breaches promptly.

**External Factors (Pandemic, Natural Disasters, etc.):**

- **Risk:** Unforeseen external events, such as a pandemic or natural disaster, could disrupt project progress and resource availability.
- **Contingency:** Develop a remote work plan, ensure data backups, and establish a crisis management strategy to handle such situations effectively.

## 14. APROVALS

| | |
|---|---|
| Project Sponsor – Devdoot | Approved |
| Development Management- Jahid | Approved |
| EDI Project Manager- Aminul | Approved |
| RS Test Manager- Devdoot | Approved |
| RS Development Team Manager- Jahid | Approved |
| Reassigned Sales- Aminul | Approved |
| Order Entry EDI Team Manager- Devdoot | Approved |