

---

# **Software Requirements Specification**

**for**

## **NSTU Sportify**

**Prepared by**

**Mission Devnath (ASH2125002M)**

**Md. Al. Shariyar Shafi (ASH2125009M)**

**Mohammed Maruf Islam (MUH2125022M)**

**Institute of Information Technology**

**Noakhali Science and Technology University**

**05.12.2024**

## Table of Contents

|  |           |
|--|-----------|
| <b>1. Introduction</b>                         | <b>6</b>  |
| 1.1 Problem Statement                          | 6         |
| 1.2 Purpose                                    | 7         |
| 1.3 Project Scope                              | 7         |
| 1.4 Features                                   | 8         |
| 1.5 Glossary                                   | 8         |
| 1.6 References                                 | 9         |
| 1.7 Overview                                   | 9         |
| <b>2. Stakeholders and Characteristics</b>     | <b>10</b> |
| 2.1 Students                                   | 10        |
| 2.2 Faculty Members:                           | 10        |
| 2.3 Departments                                | 11        |
| 2.4 University Administration                  | 11        |
| <b>3. Design and Implementation Constrains</b> | <b>11</b> |
| 3.1 Scalability                                | 11        |
| 3.2 Availability                               | 11        |
| 3.3 User Experience                            | 12        |
| 3.4 Data Management                            | 12        |
| 3.5 Integration                                | 12        |
| 3.6 Technology Stack                           | 12        |
| 3.7 Testing                                    | 12        |
| 3.8 Compliance                                 | 13        |
| <b>4. Requirement Specification</b>            | <b>13</b> |
| 4.1 Functional Requirement                     | 13        |
| 4.1.1 Update schedule                          | 13        |
| 4.1.2 Notices                                  | 13        |
| 4.1.3 Representative Registration              | 14        |
| 4.1.4 Update Results                           | 14        |
| 4.1.5 Update events                            | 14        |
| 4.1.6 Admin Login                              | 15        |
| 4.1.7 Events                                   | 15        |
| 4.1.8 Schedules                                | 15        |
| 4.1.9 Results                                  | 16        |
| 4.1.10 Notices                                 | 16        |
| 4.1.11 Previous year results                   | 16        |
| 4.1.12 Assign Department Representatives       | 17        |
| 4.2 Data Requirement                           | 17        |
| 4.2.1 Update Events                            | 17        |
| 4.2.2 Update Schedule                          | 17        |
| 4.2.3 Update Results                           | 17        |
| 4.2.4 Post Notices                             | 17        |

|         |  |    |
|---------|--|----|
| 4.2.5   | Check Events .....   | 18 |
| 4.2.6   | Manage Team Details .....                                  | 18 |
| 4.2.7   | Check Schedule.....  | 18 |
| 4.2.8   | Check Live Scores .....                                    | 18 |
| 4.2.9   | View Results .....   | 18 |
| 4.2.10  | Check Notices .....  | 18 |
| 4.2.11  | View Previous Year Results .....                           | 18 |
| 4.3     | Performance Requirement.....                               | 18 |
| 4.3.1   | Response Time for Event Updates.....                       | 19 |
| 4.3.2   | Response Time for Match Schedule Updates .....             | 19 |
| 4.3.3   | Live Score Update Frequency.....                           | 19 |
| 4.3.4   | Results and Standings Update Time .....                    | 19 |
| 4.3.5   | Notice Posting Response Time .....                         | 19 |
| 4.3.6   | Team Details Update Time .....                             | 19 |
| 4.3.7   | Event Calendar Load Time .....                             | 19 |
| 4.3.8   | Match Schedule Load Time .....                             | 19 |
| 4.3.9   | Live Scores Display Update Time .....                      | 19 |
| 4.3.10  | Results and Standings Display Time .....                   | 19 |
| 4.3.11  | Notice Display Time .....                                  | 20 |
| 4.3.12  | Previous Year Event Results Load Time .....                | 20 |
| 4.3.13  | Event Sharing Response Time .....                          | 20 |
| 4.3.14  | Data Backup Frequency .....                                | 20 |
| 4.3.15  | Page Load Time .....                                       | 20 |
| 4.3.16  | Capacity Requirement.....                                  | 20 |
|         | Here are some capacity requirements for our project: ..... | 20 |
| 4.3.2.1 | Bandwidth .....  | 20 |
| 4.3.2.2 | Storage Capacity .....                                     | 20 |
| 4.3.2.3 | Processing Power .....                                     | 21 |
| 4.3.2.4 | Database Capacity.....                                     | 21 |
| 4.3.2.5 | Server Capacity .....                                      | 21 |
| 4.3.2.6 | Network Capacity .....                                     | 21 |

|  |           |
|--|-----------|
| 4.3.2.7 Memory Capacity.....                           | 21        |
| 4.3.3 Safety Critical Requirement.....                 | 22        |
| 4.3.4 Robustness or Fault-Tolerance Requirements ..... | 22        |
| 4.4 Maintainability and Supportability .....           | 23        |
| 4.4.1 Maintenance Requirements.....                    | 23        |
| 4.4.2 Supportability Requirements .....                | 23        |
| 4.5 Security Requirements .....                        | 23        |
| 4.5.1 Authentication and Authorization.....            | 24        |
| 4.5.2 Access Control .....                             | 24        |
| 4.5.3 Network Security .....                           | 24        |
| 4.5.4 Audit Trails .....                               | 24        |
| 4.5.5 Data Backup and Recovery.....                    | 24        |
| 4.5.6 System Monitoring.....                           | 24        |
| 4.5.7 Incident Response .....                          | 24        |
| 4.6 Usability and Human Integrity Requirements.....    | 24        |
| 4.6.1 Ease of Use Requirements .....                   | 24        |
| 4.6.2 Accessibility Requirements .....                 | 25        |
| 4.7 Look and Feel Requirements .....                   | 25        |
| 4.7.1 Appearance Requirements .....                    | 25        |
| 4.8 Style Requirements .....                           | 26        |
| <b>5. Requirement Engineering Process .....</b>        | <b>26</b> |
| 5.1 Requirement Elicitation Techniques.....            | 26        |
| 5.1.1 Hold Interviews.....                             | 26        |
| 5.1.2 System Interface Analysis.....                   | 27        |
| 5.1.3 Distribute Questionnaires.....                   | 27        |
| 5.2 Sample of requirement collection .....             | 27        |
| 5.2.1 Requirement collection -1 .....                  | 27        |
| 5.2.2 Requirement collection -2.....                   | 28        |
| 5.3 Requirement Validation .....                       | 30        |
| 5.3.1 Review the Requirements .....                    | 30        |
| 5.3.2 Test the Requirements.....                       | 30        |
| 5.3.3 Simulate the requirements.....                   | 30        |
| <b>Use Case Diagram .....</b>                          | <b>31</b> |
| <b>6. Use Case Description .....</b>                   | <b>32</b> |
| 6.1 UC-01: Update Event Calendar .....                 | 32        |
| 6.2 UC-02: Update Schedules.....                       | 32        |
| 6.3 UC-03: Update Notices.....                         | 33        |

|           |   |           |
|-----------|---|-----------|
| 6.4       | UC-04: Update Results .....                   | 33        |
| 6.5       | UC-05: Post Notices .....                     | 34        |
| 6.6       | UC-06: View Events .....                      | 35        |
| 6.7       | UC-07: Check Schedules .....                  | 35        |
| 6.8       | UC-08: Request Registration .....             | 35        |
| 6.9       | UC-09: View Results .....                     | 36        |
| 6.10      | UC-10: Check Notices .....                    | 37        |
| 6.11      | UC-11: View Previous Year Results .....       | 37        |
| 6.12      | UC-12: Review Request Registration .....      | 37        |
| 6.13      | UC-13: Manage Team Details .....              | 38        |
| 6.14      | UC-14: Assign Department Representative ..... | 39        |
| <b>7.</b> | <b>Activity Diagram.....</b>                  | <b>40</b> |
| 7.1       | UC-01: Update Event Calendar: .....           | 40        |
| 7.2       | UC-02: Update Match Schedule: .....           | 41        |
| 7.3       | UC-03: Update live Scores: .....              | 42        |
| 7.4       | UC-04: Update Result and Standings: .....     | 43        |
| 7.5       | UC-05: Post Notices: .....                    | 44        |
| 7.6       | UC-06: View Event Calendar .....              | 45        |
| 7.7       | UC-07: Check Match Schedule.....              | 45        |
| 7.8       | UC-08: Check Live Scores .....                | 46        |
| 7.9       | UC-09: View Results and Standings.....        | 46        |
| 7.10      | UC-10: Check Notices .....                    | 47        |
| 7.11      | UC-11: View Previous Year Event Results ..... | 47        |
| 7.12      | UC-12: Share Event .....                      | 48        |
| 7.13      | UC-13: Manage Team Details .....              | 49        |
| 7.14      | UC-14: Assign Department Representative ..... | 50        |
| <b>8.</b> | <b>Sequence Diagram .....</b>                 | <b>51</b> |
| 8.1       | UC-01: Update event calendar: .....           | 51        |
| 8.2       | UC-02: Update Match schedule: .....           | 52        |
| 8.3       | UC-03: Update live scores: .....              | 53        |
| 8.4       | UC-04: Update result and standing: .....      | 54        |
| 8.5       | UC-05: Post Notices: .....                    | 55        |
| 8.6       | UC-06: View Event Calendar .....              | 56        |
| 8.7       | UC-07: Check Match Schedule.....              | 56        |
| 8.8       | UC-08: Check Live Scores .....                | 57        |
| 8.9       | UC-09: View Results and Standings.....        | 57        |
| 8.10      | UC-10: Check Notices .....                    | 58        |
| 8.11      | UC-11: View Previous Year Event Results ..... | 58        |
| 8.12      | UC-12: Share Event .....                      | 59        |
| 8.13      | UC-13: Manage Team Details .....              | 60        |
| 8.14      | UC-14: Assign Department Representative ..... | 61        |
| <b>9.</b> | <b>Swim lanes: .....</b>                      | <b>62</b> |
| 9.1       | UC-01: Update Event Calendar .....            | 62        |
| 9.2       | UC-02: Update Match Schedule .....            | 63        |
| 9.3       | UC-03: Update Live Scores .....               | 64        |
| 9.4       | UC-04: Update Results and Standings.....      | 65        |
| 9.5       | UC-05: Post Notices .....                     | 66        |
| 9.6       | UC-06: View Event Calendar .....              | 67        |
| 9.7       | UC-07: Check Match Schedule.....              | 67        |
| 9.8       | UC-08: Check Live Scores .....                | 68        |
| 9.9       | UC-09: View Results and Standings.....        | 68        |
| 9.10      | UC-10: Check Notices .....                    | 69        |
| 9.11      | UC-11: View Previous Year Event Results ..... | 69        |
| 9.12      | UC-12: Share Event .....                      | 70        |
| 9.13      | UC-13: Manage Team Details .....              | 71        |
| 9.14      | UC-14: Assign Department Representative ..... | 72        |

# **1. Introduction**

**NSTU Sportify** is a cutting-edge platform tailored for the efficient management of sports events at Noakhali Science and Technology University (NSTU). With a focus on enhancing user experience and engagement, NSTU Sportify offers a comprehensive suite of functionalities catering to both administrators and users.

Administrators have access to tools for updating event calendars, match schedules, live scores, results, standings, and team details. Additionally, they can post notices to communicate important information effectively. For users, NSTU Sportify provides a seamless experience to check event calendars, match schedules, live scores, results, and notices. Users can also view historical event results from previous years and share event details with others. With NSTU Sportify, the goal is to foster a vibrant sports community within NSTU by facilitating easy access to information, encouraging participation, and promoting communication among administrators, teams, and users.

## **1.1 Problem Statement**

The current sports event management system at Noakhali Science and Technology University (NSTU) faces several challenges. There is a lack of a centralized platform for efficiently managing various aspects of sports events, including event calendars, match schedules, live scores, results, standings, and team details. This leads to disorganization and inefficiency in coordinating sports activities.

Users also encounter difficulties in accessing essential information such as event schedules, live scores, results, and notices due to the absence of a centralized system. This results in inconvenience and missed opportunities for participation.

Moreover, administrators find it challenging to communicate important notices and updates to participants and teams promptly, leading to miscommunication and a lack of engagement among stakeholders. Additionally, the unavailability of historical event results from previous years hinders the ability to track progress and analyze trends over time.

To address these challenges, we propose the development of NSTU Sportify, a comprehensive platform designed to streamline sports event management at NSTU. NSTU Sportify will offer a range of features tailored to the needs of administrators, teams, and participants, including event calendar management, real-time updates, communication tools, historical data accessibility, and event promotion capabilities.

## 1.2 Purpose

The purpose of NSTU Sportify is to revolutionize sports event management at Noakhali Science and Technology University (NSTU) by providing a centralized platform for administrators, teams, and participants. It aims to streamline event organization, enhance user experience, facilitate effective communication, provide access to historical data, and promote sports events within the NSTU community. Ultimately, NSTU Sportify seeks to create a vibrant and inclusive sports culture at NSTU by encouraging active participation and fostering community engagement.

## 1.3 Project Scope

The scope of NSTU Sportify encompasses the development and implementation of a comprehensive platform for managing sports events at Noakhali Science and Technology University (NSTU). The key components within the project scope include:

### 1. Administrative Features:

- Development of tools for administrators to update event calendars, match schedules, live scores, results, standings, and team details.
- Implementation of functionalities for administrators to post notices and communicate with teams and participants.
- Integration of user management capabilities to ensure secure access and permissions for administrators.

### 2. User Features:

- Designing user-friendly interfaces for participants to check event calendars, match schedules, live scores, results, and standings.
- Implementation of features for users to view notices, access historical event results from previous years, and share event details.
- Integration of real-time updates to provide users with the latest information on sports events and activities.

### 3. Communication Tools:

- Development of communication channels such as notifications, emails, and announcements to facilitate effective communication between administrators, teams, and participants.
- Implementation of messaging functionalities to enable direct communication between users within the platform.

**4. Historical Data Access:**

- Designing databases and data storage mechanisms to store historical event results and other relevant data.
- Developing interfaces for users to access and analyze historical event data for tracking progress and evaluating trends over time.

**5. Event Promotion:**

- Integration of features for sharing event details through social media platforms and other channels to promote sports events within the NSTU community and beyond.
- Implementation of promotional tools such as event banners, posters, and announcements to increase awareness and participation.

**6. Scalability and Flexibility:**

- Designing the platform with scalability in mind to accommodate future growth and expansion of sports events and activities at NSTU.
- Ensuring flexibility in the system architecture to adapt to changing requirements and accommodate customization based on user feedback.

## 1.4 Features

- **Update Event:** Administrators can create, update, and delete events.
- **Update Schedule:** Administrators can create, update, and delete match schedules.
- **Update Notices:** Administrators can provide real-time updates about notices.
- **Update Results:** Administrators can update match results and overall standings.
- **Notices:** Administrators can post important notices related to sports activities.
- **Representative Registration:** By registering they can make their own team.
- **Admin Login:** Admin can create, update, delete events, notices, schedules.
- **Event:** Users can view upcoming events and their details.
- **Schedule:** Users can view scheduled matches and their details.
- **Results:** Users can view match results and current standings.
- **Notices:** Users can view important notices.
- **Previous Year Results:** Users can access results from previous years' events.

## 1.5 Glossary

This section provides definitions for all document names, acronyms, and abbreviations. The application domain's terms and concepts are defined.

GUI - Graphical User Interface

SRS – Software Requirement Specification

UI – User Interface



SDLC – Software Development Life Cycle

RESTful – Representational State Transfer

## 1.6 References

- IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications*. IEEE Computer Society, 1998.
- Elsevier-*A systematic literature review on agile requirements engineering practices and challenges*.
- Bjarnason, E., Wnuk, K., & Regnell, B. (2011b). *Requirements are slipping through the gaps—A case study on causes & effects of communication gaps in large-scale software development*. In 2011 IEEE 19th international requirements engineering conference (pp. 37–46)
- Wolfgang, E. (2011). *Working with user stories*. In *Agile requirements engineering workshop*, July 2011.
- Bjarnason, E., Wnuk, K., & Regnell, B. (2011b). *Requirements are slipping through the gaps—A case study on causes & effects of communication gaps in large-scale software development*. In 2011 IEEE 19th international requirements engineering conference (pp. 37–46).

## 1.7 Overview

NSTU Sportify is a comprehensive sports management and information platform tailored for the Noakhali Science and Technology University (NSTU) community. This project is designed to centralize all sports-related activities, updates, and information, enhancing the sports experience for students, faculty, and sports enthusiasts. By integrating multiple functionalities into a single, user-friendly interface, NSTU Sportify aims to streamline the management and dissemination of sports information and foster greater engagement in university sports activities.

The platform allows administrators to manage and update the event calendar with ease, ensuring that all upcoming sports events are accurately scheduled and displayed. This feature supports real-time updates, ensuring the calendar reflects the most current information. Additionally, it includes a comprehensive match scheduling feature, which ensures that all matches, including their start times, venues, and participating teams, are precisely scheduled and easily accessible.

NSTU Sportify also offers real-time live score updates, keeping users informed about the progress of matches as they happen. This is particularly beneficial for those unable to attend matches in person but who still want to stay updated on their favorite teams and events. Administrators can update match results and standings, ensuring that the leaderboard reflects the latest outcomes and overall team performance.

The platform facilitates the posting of important announcements and notices related to sports activities, ensuring that all relevant stakeholders receive timely updates and critical information. It also includes features for comprehensive team management, allowing for the updating and management of team rosters, player details, and team statistics.

Users can access the event calendar easily, providing a clear overview of all scheduled sports events, helping them plan their participation and attendance. The platform also allows users to check the match schedule, view live scores, and access results and standings. Notices and announcements can be quickly checked, ensuring users are always informed about the latest updates. Additionally, users can view previous years' event results, which provides a historical context and enhances the engagement experience.

NSTU Sportify is designed to be an all-encompassing solution for managing university sports, making it easier for administrators to handle logistics and for users to stay informed and engaged. This platform not only centralizes information but also enhances communication and transparency, fostering a vibrant sports culture at NSTU.

## **2. Stakeholders and Characteristics**

**2.1 Students:** Students are primary stakeholders of the platform. They actively participate in and attend sports events at NSTU.

- They comprise both players and spectators who require easy access to event information such as schedules, live scores, results, and notices.
- They engage with the sports community through social interactions and event sharing.
- They seek a platform to view historical event results and track their team's performance.
- They require user-friendly interfaces and intuitive features due to varying levels of technical expertise.

**2.2 Faculty Members:** Faculty members support and mentor sports teams and encourage student participation in sports events.

- They require access to match schedules, live scores, results, and standings to stay informed about their teams' performance.
- They use the platform to post notices and updates related to sports activities and events.

- They facilitate communication and coordination between students and the administration regarding sports events.
- They aim to promote student engagement and participation in sports activities.

**2.3 Departments:** Departments include various academic and administrative units involved in organizing and supporting sports events.

- They require tools to manage event calendars, match schedules, live scores, results, and team details.
- They need to post notices and updates relevant to their departments and teams.
- They coordinate and manage sports events efficiently within their respective departments.
- They track and analyze the performance of their departmental teams over time.

**2.4 University Administration:** The university administration oversees the overall functioning of NSTU, including the management of sports activities.

- They have a vested interest in promoting a vibrant sports culture and fostering community engagement.
- They require tools to manage the event calendar, schedule matches, update live scores, and maintain results and standings.
- They need to communicate important notices and updates effectively to all stakeholders.
- They utilize data and analytics to track progress, evaluate trends, and make informed decisions for future sports programs and initiatives.

## 3. Design and Implementation Constrains

To ensure the success of NSTU Sportify, we will follow several key design and implementation constraints. These cover scalability, availability, user experience, data management, integration, technology stack, testing, and compliance.

### 3.1 Scalability

We need NSTU Sportify to handle a growing number of users and data efficiently. We will:

- Architecture: Use a modular design so that we can scale different parts independently.
- Load Balancing: Implement load balancers to distribute traffic across multiple servers.
- Database Scaling: Use MySQL to manage large datasets and ensure high performance.

### 3.2 Availability

We need the platform to be reliable and always accessible. We will:

- Redundancy: Implement backup systems to ensure service continuity.
- Monitoring and Alerts: Use tools to monitor performance and alert us to issues.

### **3.3 User Experience**

Providing a seamless and intuitive user experience is essential. We will:

- Responsive Design: Ensure the platform works well on all devices (desktops, tablets, smartphones).
- Intuitive Navigation: Design easy-to-use interfaces with consistent layouts.
- Performance Optimization: Ensure fast load times and smooth interactions.

### **3.4 Data Management**

Efficient and secure data management is critical. We will:

- Database Design: Use MySQL for a reliable and efficient database solution.
- Data Backup: Regularly back up data to prevent loss.

### **3.5 Integration**

Seamless integration with other services is essential. We will:

- Interoperability: Ensure compatibility with existing systems and standards.
- Extensibility: Design the system to accommodate future integrations and enhancements.

### **3.6 Technology Stack**

Choosing the right technology stack is important. We will:

- Frontend: Use Flutter for a consistent experience across iOS and Android devices.
- Backend: Use Django for building robust and scalable software.
- Database: Use MySQL for its reliability and performance.

### **3.7 Testing**

Thorough testing is vital for ensuring functionality and security. We will

- Unit Testing: Test individual components to ensure they work correctly.
- Integration Testing: Ensure different parts of the system work together seamlessly.
- Acceptance Testing: Verify that the platform meets user requirements and provides a good user experience.

### 3.8 Compliance

The platform must comply with relevant laws and regulations. We will:

- Data Protection: Ensure data management practices comply with local data protection laws.

By following these constraints, we will create a robust and user-friendly platform for managing sports events at Noakhali Science and Technology University.

## 4. Requirement Specification

All the requirements based on the elicitation process are described in this section.

### 4.1 Functional Requirement

Functional requirements are those requirements that are used to illustrate the internal working nature of the system, the description of the system, and explanation of each subsystem. It consists of what task the system should perform, the processes involved, which data the system should hold and the interfaces with the user.

#### 4.1.1 Update schedule

|                     |   |                 |      |
|---------------------|---|-----------------|------|
| <b>FR-1</b>         | Update schedule   |                 |      |
| <b>Description</b>  | Admins will be able to create and modify the match schedule. Matches should include information such as participating teams, date, time, and venue. |                 |      |
| <b>Stakeholders</b> | Admin   | <b>Priority</b> | High |

#### 4.1.2 Notices

|                    |   |  |  |
|--------------------|---|--|--|
| <b>FR-2</b>        | Post Notices  |  |  |
| <b>Description</b> | This functional requirement describes the process of allowing admin to update any kind of information about sports. Admins should be able to post notices for |  |  |

|                     |   |                 |      |
|---------------------|---|-----------------|------|
|                     | upcoming events, changes in schedule, or other important information. Notices should be categorized and time-stamped. |                 |      |
| <b>Stakeholders</b> | Admin   | <b>Priority</b> | High |

#### 4.1.3 Representative Registration

|                     |  |                 |      |
|---------------------|--|-----------------|------|
| <b>FR-3</b>         | <b>Representative Registration</b>   |                 |      |
| <b>Description</b>  | This functional requirement describes the process of registering a representative so that they create a team for their own department. |                 |      |
| <b>Stakeholders</b> | Faculties  | <b>Priority</b> | High |

#### 4.1.4 Update Results

|                     |  |                 |      |
|---------------------|--|-----------------|------|
| <b>FR-4</b>         | Results and Standings will update after schedule match   |                 |      |
| <b>Description</b>  | After ending every match admins will enter and update match results. The system will automatically update team standings based on match results. |                 |      |
| <b>Stakeholders</b> | Admin  | <b>Priority</b> | High |

#### 4.1.5 Update events

|             |   |  |  |
|-------------|---|--|--|
| <b>FR-5</b> | Event calendar will organize according to upcoming events |  |  |
|-------------|---|--|--|

|                     |  |                 |      |
|---------------------|--|-----------------|------|
| <b>Description</b>  | Admins can add, edit, and delete events on the calendar. Events should have details such as date, time, location, and description. |                 |      |
| <b>Stakeholders</b> | Admin  | <b>Priority</b> | High |

#### 4.1.6 Admin Login

|                     |   |                 |      |
|---------------------|---|-----------------|------|
| <b>FR-6</b>         | Admin Login   |                 |      |
| <b>Description</b>  | By this functional requirements admin can create, read, update and delete matches, events, schedules and so on. |                 |      |
| <b>Stakeholders</b> | Admin   | <b>Priority</b> | High |

#### 4.1.7 Events

|                     |  |                 |      |
|---------------------|--|-----------------|------|
| <b>FR-7</b>         | Check event Calendar   |                 |      |
| <b>Description</b>  | System will provide users an updated calendar where events are categorized based on time and date. It will help determine how long it will take for events to occur. |                 |      |
| <b>Stakeholders</b> | Users  | <b>Priority</b> | High |

#### 4.1.8 Schedules

|             |                |  |  |
|-------------|----------------|--|--|
| <b>FR-8</b> | Check schedule |  |  |
|-------------|----------------|--|--|

|                     |  |                 |        |
|---------------------|--|-----------------|--------|
| <b>Description</b>  | System will provide users match schedule of a running tournament. From this user can see the fixed time of a scheduled match easily. |                 |        |
| <b>Stakeholders</b> | Users  | <b>Priority</b> | Medium |

#### 4.1.9 Results

|                     |   |                 |        |
|---------------------|---|-----------------|--------|
| <b>FR-10</b>        | View results and standings after ending match   |                 |        |
| <b>Description</b>  | User will able to view a list of sports sections like football, cricket, chess and carom etc. Then they can select any one of them and can view past matches results and current standings. User also can view scoreboard of every sports section's previous match from here. |                 |        |
| <b>Stakeholders</b> | Users   | <b>Priority</b> | Medium |

#### 4.1.10 Notices

|                     |   |                 |        |
|---------------------|---|-----------------|--------|
| <b>FR-11</b>        | Check notices   |                 |        |
| <b>Description</b>  | User will be able to view posted notices. Notices should be listed chronologically and be searchable. |                 |        |
| <b>Stakeholders</b> | Users   | <b>Priority</b> | Medium |

#### 4.1.11 Previous year results

|              |                                  |  |  |
|--------------|----------------------------------|--|--|
| <b>FR-12</b> | View previous year event results |  |  |
|--------------|----------------------------------|--|--|



|                     |   |                 |     |
|---------------------|---|-----------------|-----|
| <b>Description</b>  | User should be able to access the results of previous year's events. Data should be archived and retrievable by year. |                 |     |
| <b>Stakeholders</b> | Users   | <b>Priority</b> | Low |

#### 4.1.12 Assign Department Representatives

|                     |  |                 |        |
|---------------------|--|-----------------|--------|
| <b>FR-14</b>        | Assign Department Representatives  |                 |        |
| <b>Description</b>  | Admin will assign a department representative from every department, who can lead own department team. He will also paly a role of a coach of this team. |                 |        |
| <b>Stakeholders</b> | Admin  | <b>Priority</b> | Medium |

## 4.2 Data Requirement

| <b>Requirement Name</b>      | <b>Requirement Description</b>  | <b>Stakeholders</b>       | <b>Priority</b> |
|------------------------------|---|---------------------------|-----------------|
| <b>4.2.1 Update Events</b>   | Allow administrators to add, edit, and delete events in the event calendar, including event names, dates, times, and locations. | University Administration | High            |
| <b>4.2.2 Update Schedule</b> | Enable administrators to create, modify, and remove match schedules, specifying dates, times, and participating teams.          | University Administration | High            |
| <b>4.2.3 Update Results</b>  | Allow administrators to record and update final match results and overall team standings after events.                          | University Administration | High            |
| <b>4.2.4 Post Notices</b>    | Enable administrators to post important notices and announcements related to sports events and activities.                      | University Administration | High            |

|  |   |  |        |
|--|---|--|--------|
| <b>4.2.5 Check Events</b>                | Provide access for students, faculty, and departments to view the event calendar with details of upcoming and past events.  | Students, Faculty Members, Departments | High   |
| <b>4.2.6 Manage Team Details</b>         | Allow administrators to add, edit, and delete team information, including team names, members, roles, and contact details.  | Departments                            | High   |
| <b>4.2.7 Check Schedule</b>              | Allow students, faculty, and departments to view detailed match schedules, including dates, times, and participating teams. | Students, Faculty Members, Departments | High   |
| <b>4.2.8 Check Live Scores</b>           | Enable students, faculty, and departments to view live scores of ongoing matches in real-time..                             | Students, Faculty Members, Departments | High   |
| <b>4.2.9 View Results</b>                | Allow users to view final match results and team standings after events have concluded.                                     | Students, Faculty Members, Departments | High   |
| <b>4.2.10 Check Notices</b>              | Provide access for users to view posted notices and announcements related to sports events and activities.                  | Students, Faculty Members, Departments | High   |
| <b>4.2.11 View Previous Year Results</b> | Enable users to access archived results from previous years' events for reference and analysis.                             | Students, Faculty Members, Departments | Medium |

In addition to the above, the system would also require data on the usage of the platform, such as the number of users, active sessions, and engagement metrics, to help to understand how the system is being used and to make improvements over time. It's important to ensure that all the data is collected ethically, stored securely, and used in compliance with privacy regulations.

### 4.3 Performance Requirement

It is important to maintain the performance of the software system. To ensure performance we maintain these steps:

| Requirement Name                                      | Requirement Description   | Stakeholders  | Priority |
|---|---|---|----------|
| <b>4.3.1 Response Time for Event Updates</b>          | The system should update the event calendar within 2 seconds of an administrator's input.         | University Administration   | High     |
| <b>4.3.2 Response Time for Match Schedule Updates</b> | The system should update match schedules within 2 seconds of an administrator's input.            | University Administration   | High     |
| <b>4.3.3 Live Score Update Frequency</b>              | Live scores should be updated within 1 second of receiving new data..                             | University Administration   | High     |
| <b>4.3.4 Results and Standings Update Time</b>        | Final results and team standings should be updated within 5 seconds after input.                  | University Administration   | High     |
| <b>4.3.5 Notice Posting Response Time</b>             | Notices should be posted and visible to users within 2 seconds of submission by an administrator. | University Administration   | High     |
| <b>4.3.6 Team Details Update Time</b>                 | Team details should be updated within 3 seconds of administrator input.                           | Departments   | High     |
| <b>4.3.7 Event Calendar Load Time</b>                 | The event calendar should load and display within 2 seconds for users.                            | Students, Faculty Members, Departments, University Administration | High     |
| <b>4.3.8 Match Schedule Load Time</b>                 | Match schedules should load and display within 2 seconds for users.                               | Students, Faculty Members, Departments, University Administration | High     |
| <b>4.3.9 Live Scores Display Update Time</b>          | Live scores should refresh every 1 second to display the most recent data.                        | Students, Faculty Members, Departments, University Administration | High     |
| <b>4.3.10 Results and Standings Display Time</b>      | Results and standings should load and display within 3 seconds for users.                         | Students, Faculty Members, Department, University Administration  | High     |

| Requirement Name                                    | Requirement Description  | Stakeholders  | Priority |
|---|--|---|----------|
| <b>4.3.11 Notice Display Time</b>                   | Notices should be visible to users within 2 seconds of accessing the notice section.     | Students, Faculty Members, Departments, University Administration | High     |
| <b>4.3.12 Previous Year Event Results Load Time</b> | Previous year event results should load and display within 3 seconds for users.          | Students, Faculty Members, Departments, University Administration | Medium   |
| <b>4.3.13 Event Sharing Response Time</b>           | Sharing event details should be processed within 2 seconds of user action.               | Students, Faculty Members, Administrators                         | Medium   |
| <b>4.3.14 Data Backup Frequency</b>                 | Data backups should occur every 24 hours to ensure data integrity and availability.      | University Administration   | High     |
| <b>4.3.15 Page Load Time</b>                        | All pages within the platform should load within 2 seconds under normal load conditions. | All Stakeholders  | High     |

### 4.3.16 Capacity Requirement

Here are some capacity requirements for our project:

#### 4.3.2.1 Bandwidth

- **Requirement:** The system must have sufficient bandwidth to support high traffic volume during peak periods, such as holiday seasons or promotional events.
- **Description:** The system's network infrastructure should be capable of handling increased traffic loads without degradation in performance, ensuring that users can access the system and its resources without delays or interruptions during peak periods.

#### 4.3.2.2 Storage Capacity

- **Requirement:** The system must have sufficient storage capacity to store event information, participants data.
- **Description:** The system's storage infrastructure should be able to accommodate the growing volume of data generated by event notices, participants details, and history records. Adequate storage capacity ensures that the system can maintain comprehensive records and scale to meet increasing data storage needs over time.

#### 4.3.2.3 Processing Power

- **Requirement:** The system must have sufficient processing power to handle large volumes of data, including searches.
- **Description:** The system's processing capabilities should be able to efficiently process incoming data streams, execute complex queries, and perform real-time analytics to support various system functionalities. Sufficient processing power ensures that the system can handle concurrent user requests and maintain responsiveness under varying workloads.

#### 4.3.2.4 Database Capacity

- **Requirement:** The system must have sufficient database capacity to store customer information, transaction data, and inventory information.
- **Description:** The system's database infrastructure should be capable of storing and managing vast amounts of structured and unstructured data, including dept names, event histories, and participants names and details, Representatives names. Adequate database capacity ensures that the system can maintain data integrity, support data-driven decision-making, and scale to accommodate growing data requirements.

#### 4.3.2.5 Server Capacity

- **Requirement:** The system must have sufficient database capacity to store event information, participants names, previous years winners.
- **Description:** The system's database infrastructure should be capable of storing and managing vast amounts of structured and unstructured data, including all event details and event histories. Adequate database capacity ensures that the system can maintain data integrity, support data-driven decision-making, and scale to accommodate growing data requirements.

#### 4.3.2.6 Network Capacity

- **Requirement:** The system must have sufficient network capacity to support multiple user connections and ensure accessibility from different locations.
- **Description:** The system's network infrastructure should be able to handle incoming and outgoing network traffic, support concurrent user connections, and provide reliable connectivity across different geographical regions. Adequate network capacity ensures that users can access the system seamlessly and experience minimal latency or downtime.

#### 4.3.2.7 Memory Capacity

- **Requirement:** The system must have sufficient memory capacity to handle multiple user requests without any delay or interruption.
- **Description:** The system's memory resources should be able to store and manipulate data in memory, perform calculations, and cache frequently accessed information to improve performance. Adequate memory capacity ensures that the system can efficiently process user requests and maintain responsiveness under varying workloads.

#### 4.3.3 Safety Critical Requirement

There are no safety critical requirements for our project.

#### 4.3.4 Robustness or Fault-Tolerance Requirements

##### 4.3.4.1 Error Handling

The system must have proper error handling mechanisms in place to handle unexpected errors or exceptions and prevent the system from crashing or becoming unresponsive.

##### 4.3.4.2 Redundancy

The system must have redundant components, such as servers, databases, or network connections, to ensure that critical functions are available even in case of failures.

##### 4.3.4.3 Failover

The system must have a failover mechanism that automatically switches to a backup system or component in case of a failure to ensure that the system remains operational.

##### 4.3.4.4 Load Balancing

The system must have load balancing mechanisms that distribute the load across multiple servers to ensure that no single server is overwhelmed with requests and that the system remains responsive.

##### 4.3.4.5 Performance Monitoring

The system must have performance monitoring mechanisms that track system performance and alert administrators if the system falls below defined thresholds.

##### 4.3.4.6 Recovery Time Objectives (RTO)

The system must have defined RTOs for each critical component or system function, specifying the maximum acceptable downtime and the required recovery time in case of a failure.

##### 4.3.4.7 Backups

The system must have a backup mechanism that regularly backs up critical data to ensure that data can be recovered in case of a catastrophic failure.

## **4.4 Maintainability and Supportability**

### **4.4.1 Maintenance Requirements**

#### **4.4.1.1 Regular Software Updates**

The system must have a mechanism to update the software to the latest version, which includes bug fixes, performance improvements, and security enhancements.

#### **4.4.1.2 Regular Hardware Maintenance**

The system must have regular hardware maintenance, which includes cleaning, inspection, and repair or replacement of faulty components.

#### **4.4.1.3 Data Backup and Recovery**

The system must have a regular data backup mechanism to ensure that critical data is not lost in case of hardware or software failures.

#### **4.4.1.4 System Monitoring**

The system must have a system monitoring mechanism that tracks system performance, usage patterns, and error logs to identify potential issues and optimize system performance.

#### **4.4.1.5 User Support**

The system must have a user support mechanism to help users troubleshoot issues, provide guidance, and resolve user complaints or issues.

#### **4.4.1.6 Training**

The system must provide training to administrators and users to ensure that they are familiar with the system's features and capabilities, and can use the system effectively.

#### **4.4.1.7 Documentation**

The system must have documentation that outlines the system architecture, design, and usage, which is helpful for troubleshooting, maintenance, and training purposes.

### **4.4.2 Supportability Requirements**

This system meets Testability, Maintainability, Compatibility, Configurability, Serviceability, and install ability which are related to supportability requirements.

## **4.5 Security Requirements**

Securing information is much more important for our system to get users dependability. Here are some of them:

#### **4.5.1 Authentication and Authorization**

The system must have a mechanism to authenticate users and authorize access to system resources based on the user's role and level of access.

#### **4.5.2 Access Control**

The system must have access control mechanisms to restrict access to sensitive data and system resources to authorized personnel only.

#### **4.5.3 Network Security**

The system must have network security mechanisms, such as firewalls and intrusion detection systems, to protect against network attacks and unauthorized access to the system.

#### **4.5.4 Audit Trails**

The system must have audit trail mechanisms that log user activity and system events, to enable the identification and investigation of security breaches and unauthorized access attempts.

#### **4.5.5 Data Backup and Recovery**

The system must have a data backup and recovery mechanism to ensure that critical data can be restored in case of data loss or corruption.

#### **4.5.6 System Monitoring**

The system must have system monitoring mechanisms that detect and alert administrators of any suspicious or malicious activity on the system.

#### **4.5.7 Incident Response**

The system must have an incident response plan in place to handle security breaches or attacks and minimize the impact of such incidents.

### **4.6 Usability and Human Integrity Requirements**

This system will provide more user-friendly environment

#### **4.6.1 Ease of Use Requirements**

Our system will be easier to use by any type of people and they don't need any training to use the system.



## 4.6.2 Accessibility Requirements

To get access to the application, the application provides authorization/authentication. This application will use various modules.

### 4.6.2.1 Navigation

The system must have a clear and consistent navigation mechanism, which includes keyboard shortcuts, to help users navigate the system effectively.

### 4.6.2.2 Text Size and Font

The system must have options to increase or decrease the text size and change the font style to help users with visual impairments read the content easily.

### 4.6.2.3 Color Contrast

The system must have sufficient color contrast between the background and foreground elements to help users with visual impairments distinguish between different elements on the screen.

### 4.6.2.8 Standards Compliance

The system must comply with accessibility standards, such as Web Content Accessibility Guidelines (WCAG) 2.1, to ensure that the system is accessible to users with disabilities.

## 4.7 Look and Feel Requirements

Look and feel requirements mainly refer to how the system will look.

### 4.7.1 Appearance Requirements

|                     |  |                 |      |
|---------------------|--|-----------------|------|
| <b>AR-1</b>         | Text color and font  |                 |      |
| <b>Description</b>  | The platform's text color and font should be chosen to ensure readability, consistency, and aesthetic appeal across all pages and devices. The primary text color should provide sufficient contrast with the background to ensure accessibility for all users, including those with visual impairments. |                 |      |
| <b>Stakeholders</b> | Developer, User, Administrator   | <b>Priority</b> | High |

## 4.8 Style Requirements

There are no style requirements in our system.

## 4.8 Legal Requirements

Legal requirements normally refer to the terms and conditions or privacy policy of any organization. The terms and condition of our application is that, no third-party software or person is allowed to use our data for their business purpose.

# 5. Requirement Engineering Process

Requirements Engineering (RE) determines software requirements according to customer requirements or needs. Requirements engineering process includes requirements elicitation, needs modeling, requirements analysis, requirements assurance & validation, and requirements management.

## 5.1 Requirement Elicitation Techniques

Requirement's elicitation is the practice of researching and finding system requirements for users, customers, and other stakeholders, also referred to as "requirement gathering". Requirement elicitation can be done by contacting participants directly or by doing some research, analysis and testing.

### 5.1.1 Hold Interviews

We hold discussions that can be conducted individually or with a small group of participants. They are an effective way to gather detailed requirements without spending a lot of time with participants because we meet with people to discuss only certain important aspects of this program. Interviews are useful for obtaining individual requirements from members and organizing workshops where those members of the program come together to resolve any issues or conflicts. We primarily base our interviews on specific criteria such as:

- Short description about goals and objectives
- Updating the events
- Updating the schedule
- Updating results
- Notices
- Checking events
- Checking schedule

- Viewing results
- Checking notices
- Viewing previous year event results

### 5.1.2 System Interface Analysis

The first thing to do is to identify which systems the system-to-be shall communicate with. It could be a server on the Internet, a piece of software on the same host as the system-to-be, some hardware or something completely different.

### 5.1.3 Distribute Questionnaires

The questionnaire is a useful way to investigate styles, changes in attitudes and users' ideas, and user satisfaction with priorities and preferences. Our lists of questions were as short as possible. The respondent may be tired or frustrated. Had a basic reason for all the questions as well as group the topic areas together for the respondent to focus on. The main advantage of this survey responses was that they were collected in the usual way. Information was summarized by a large number of people.

## 5.2 Sample of requirement collection

### 5.2.1 Requirement collection -1

This report summarizes the results of the questionnaires distribution conducted to gather requirements for our system. The objective of the surveys was to identify the key needs and expectations of the stakeholders and to use this information to develop a comprehensive set of requirements for the system.

### Interview Methodology:

The interviews were conducted with stakeholders from various departments of NSTU. The interviews were conducted in a one-to-one format, each lasting approximately 7-10 minutes.

### Participants:

A total of 12 & 15 responses were collected for project requirements & functional requirements respectively.

### Findings:

The following are the key findings from the interviews:

- Mobile application is preferred.
- Users want to view event information without logging in.
- Users want the ability to search for events and schedules.
- Filtering events based on categories should be available.
- Real-time live score updates should be provided.
- Notices and updates should be easily accessible.
- Users should be able to share events on social media.
- Historical event results should be viewable.

**Key Requirements:**

Based on the findings from the interviews, the following are the key requirements for NSTU Sportify System:

- User-friendly interface with easy navigation.
- Ability to search for events and schedules.
- Real-time live score updates.
- Filter events based on categories.
- Post and view notices.
- Share events on social media.
- View previous year event results.

**Assumptions:**

It was assumed during the survey process that the NSTU Sportify system will be accessible via the web and a mobile application will be developed.

**Limitations:**

Responses collected were fewer than expected.

**Conclusion:**

The stakeholders' surveys provided valuable insights into the requirements for the NSTU Sportify system. The key findings and requirements will be used to develop a comprehensive set of requirements for the system.

**5.2.2 Requirement collection -2**

This report summarizes the results of the stakeholder interviews conducted to gather requirements for our system. The objective of the interviews was to identify the key needs and expectations of the stakeholders and to use this information to develop a comprehensive set of requirements for the system.

**Distribute Questionnaires Methodology:**

The Questionnaires were distributed to stakeholders from various cities and roles. It takes 3-4 minutes to complete the questionnaire.

**Participants:**

A total of 13-14 were interviewed, including:  
10-12 students,  
Some players of NSTU football main squad.

**Findings:**

The following are the key findings from the interviews:

- Mobile application is more comfortable.
- User-friendly interface with easy navigation.
- Users can search and view sports information.
- Users can manage team details.
- Users can upload and manage event details.
- Real-time updates for match schedules and live scores.
- Notifications for updates and notices should be available.

**Key Requirements:**

Based on the findings from the interviews, the following are the key requirements for the System:

- User-friendly interface with easy navigation.
- Search and view sports information.
- Manage team details.
- Real-time match schedule and live scores updates.
- Notification system for updates and notices.

**Assumptions:**

It was assumed during the interview process that the NSTU Sportify system will be accessible via the web and a mobile application will be developed.

**Limitations & Challenges:**

- Meet the stakeholders on their free schedules.
- 

**Conclusion:**

The stakeholder interviews provided valuable insights into the requirements for the NSTU Sportify system. The key findings and requirements will be used to develop a comprehensive set of requirements for the system.

## **5.3 Requirement Validation**

Requirement validation ensures that the requirements are correct and reflect the quality desired from this program. Initially, our requirements seemed sufficient, but upon deeper inspection and practical application, ambiguities and gaps were discovered.

### **5.3.1 Review the Requirements**

Peer reviews, particularly rigorous evaluations, are essential among high-quality software processes. A team of reviewers representing different perspectives carefully examined written requirements, analysis models, and related information to identify inconsistencies and gaps.

### **5.3.2 Test the Requirements**

Testing creates another perspective on the requirements. We conducted tests to ensure the expected performance was achieved. User testing was also performed to document the expected product behavior under specified conditions.

### **5.3.3 Simulate the requirements**

To simulate requirements, tools are available that were used to mimic a proposed system or to add details to written requirements. Simulation takes prototyping to the next level, allowing us to visualize how the system will operate and identify potential issues before implementation.

.

## Use Case Diagram

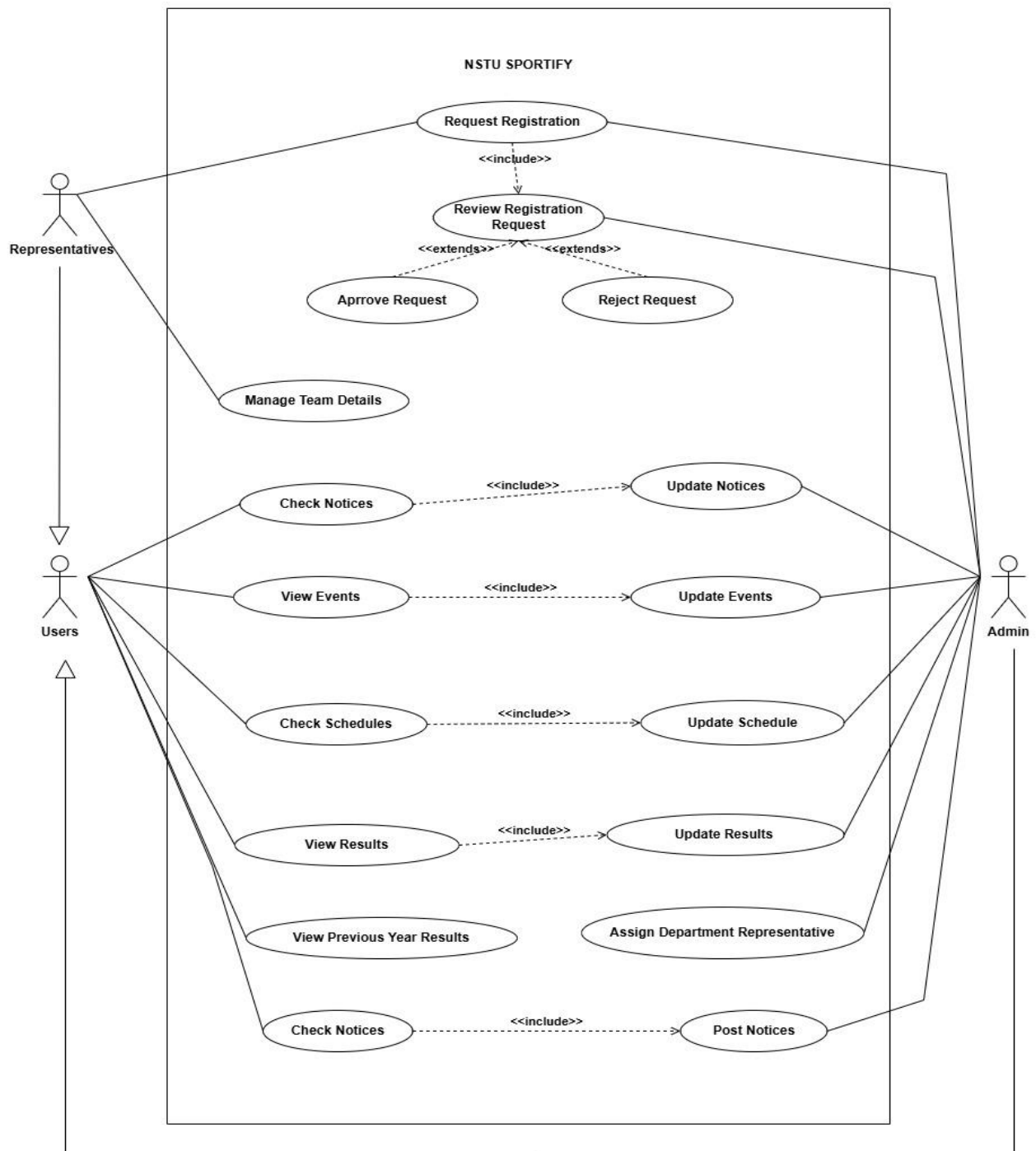


Figure 1: Use case Diagram

## 6. Use Case Description

### 6.1 UC-01: Update Event Calendar

|                              |   |
|------------------------------|---|
| <b>Use Case</b>              | <b>Update Events</b>  |
| <b>Goal</b>                  | Update the events with new events or modifications to existing events.  |
| <b>Preconditions</b>         | The user must be logged in with administrative privileges.  |
| <b>Success end condition</b> | The event calendar is successfully updated, and the changes are visible to all users.   |
| <b>Failed end condition</b>  | The event calendar is not updated due to validation errors or system issues.  |
| <b>Primary Actor</b>         | Admin   |
| <b>Secondary Actor</b>       | System  |
| <b>Trigger</b>               | Admin initiates an update to the events.  |
| <b>Main Success Flows</b>    | <ol style="list-style-type: none"> <li>1. Admin logs into the system.</li> <li>2. Admin navigates to the "Event Calendar" section.</li> <li>3. Admin selects the option to add or edit an event.</li> <li>4. Admin enters or updates event details.</li> <li>5. Admin submits the changes.</li> <li>6. System validates the input and updates the calendar.</li> <li>7. System confirms the update to the Admin.</li> </ol> |
| <b>Alternative Flow</b>      | If validation fails, the system displays an error message and prompts the admin to correct the input.   |
| <b>Quality Requirements</b>  | <ol style="list-style-type: none"> <li>1. The system must validate all input fields.</li> <li>2. Changes should reflect in the event calendar in real-time.</li> </ol>  |

### 6.2 UC-02: Update Schedules

|                              |  |
|------------------------------|--|
| <b>Use Case</b>              | <b>Update Schedules</b>  |
| <b>Goal</b>                  | Update the schedule of matches.  |
| <b>Preconditions</b>         | The user must be logged in with administrative privileges.   |
| <b>Success end condition</b> | The match schedule is updated and displayed correctly.   |
| <b>Failed end condition</b>  | The schedule is not updated due to validation errors or system issues.   |
| <b>Primary Actor</b>         | Admin  |
| <b>Secondary Actor</b>       | System   |
| <b>Trigger</b>               | Admin initiates an update to the match schedule.   |
| <b>Main Success Flows</b>    | <ol style="list-style-type: none"> <li>1. Admin logs into the system.</li> <li>2. Admin navigates to the "Match Schedule" section.</li> <li>3. Admin selects the option to add or edit a match.</li> <li>4. Admin enters or updates match details.</li> <li>5. Admin submits the changes.</li> <li>6. System validates the input and updates the schedule.</li> <li>7. System confirms the update to the Admin.</li> </ol> |



|                             |  |
|-----------------------------|--|
| <b>Use Case</b>             | <b>Update Schedules</b>  |
| <b>Alternative Flow</b>     | If validation fails, the system displays an error message and prompts the admin to correct the input.  |
| <b>Quality Requirements</b> | <ol style="list-style-type: none"> <li>1. The system must validate all input fields.</li> <li>2. Changes should reflect in the match schedule in real-time.</li> </ol> |

### 6.3 UC-03: Update Notices

|                              |  |
|------------------------------|--|
| <b>Use Case</b>              | <b>Update Notices</b>  |
| <b>Goal</b>                  | Publish or modify notices for users effectively.   |
| <b>Preconditions</b>         | The user must be logged in with administrative privileges.   |
| <b>Success end condition</b> | Notices are successfully updated or published and visible to all users.  |
| <b>Failed end condition</b>  | Notices are not updated due to validation errors or system issues.   |
| <b>Primary Actor</b>         | Admin  |
| <b>Secondary Actor</b>       | System   |
| <b>Trigger</b>               | Admin initiates the process to update or publish a notice.   |
| <b>Main Success Flows</b>    | <ol style="list-style-type: none"> <li>1. Admin logs into the system.</li> <li>2. Admin navigates to the "Notices" section.</li> <li>3. Admin selects an existing notice to edit or initiates the creation of a new notice.</li> <li>4. Admin enters or modifies the notice details (e.g., title, description, expiry date).</li> <li>5. Admin submits the notice for publishing.</li> <li>6. System validates the input and saves the changes.</li> <li>7. The system confirms the update to the Admin and displays the updated notice to all users.</li> </ol> |
| <b>Alternative Flow</b>      | <b>Validation Failure:</b> <ul style="list-style-type: none"> <li>• If the input fields do not pass validation (e.g., missing title, invalid date), the system displays an error message.</li> <li>• The admin is prompted to correct the input and resubmit.</li> </ul>   |
| <b>Quality Requirements</b>  | <ul style="list-style-type: none"> <li>• The system must validate all input fields before submission.</li> <li>• Updated notices should be visible to users immediately after confirmation.</li> </ul>   |

### 6.4 UC-04: Update Results

|                      |  |
|----------------------|--|
| <b>Use Case</b>      | <b>Update Results</b>                                      |
| <b>Goal</b>          | Update match results and standings.                        |
| <b>Preconditions</b> | The user must be logged in with administrative privileges. |

|                              |  |
|------------------------------|--|
| <b>Use Case</b>              | <b>Update Results</b>  |
| <b>Success end condition</b> | Match results and standings are updated and displayed correctly.   |
| <b>Failed end condition</b>  | Results and standings are not updated due to validation errors or system issues.   |
| <b>Primary Actor</b>         | Admin  |
| <b>Secondary Actor</b>       | System   |
| <b>Trigger</b>               | Admin initiates an update to the match results and standings.  |
| <b>Main Success Flows</b>    | <ol style="list-style-type: none"> <li>1. Admin logs into the system.</li> <li>2. Admin navigates to the "Results and Standings" section.</li> <li>3. Admin selects the option to update results and standings.</li> <li>4. Admin enters the results and updates standings.</li> <li>5. Admin submits the changes.</li> <li>6. System validates the input and updates the results and standings.</li> <li>7. System confirms the update to the Admin.</li> </ol> |
| <b>Alternative Flow</b>      | If validation fails, the system displays an error message and prompts the admin to correct the input.  |
| <b>Quality Requirements</b>  | <ol style="list-style-type: none"> <li>1. The system must validate all input fields.</li> <li>2. Changes should reflect in the results and standings in real-time.</li> </ol>  |

## 6.5 UC-05: Post Notices

|                              |   |
|------------------------------|---|
| <b>Use Case</b>              | <b>Post Notices</b>   |
| <b>Goal</b>                  | Post important notices for users.   |
| <b>Preconditions</b>         | The user must be logged in with administrative privileges.  |
| <b>Success end condition</b> | Notices are posted and visible to all users.  |
| <b>Failed end condition</b>  | Notices are not posted due to validation errors or system issues.   |
| <b>Primary Actor</b>         | Admin   |
| <b>Secondary Actor</b>       | System  |
| <b>Trigger</b>               | Admin initiates a post for a notice.  |
| <b>Main Success Flows</b>    | <ol style="list-style-type: none"> <li>1. Admin logs into the system.</li> <li>2. Admin navigates to the "Notices" section.</li> <li>3. Admin selects the option to post a new notice.</li> <li>4. Admin enters the notice details.</li> <li>5. Admin submits the notice.</li> <li>6. System validates the input and posts the notice.</li> <li>7. System confirms the posting to the Admin.</li> </ol> |
| <b>Alternative Flow</b>      | If validation fails, the system displays an error message and prompts the admin to correct the input.   |
| <b>Quality Requirements</b>  | <ol style="list-style-type: none"> <li>1. The system must validate all input fields.</li> <li>2. Notices should be posted immediately and visible to users in real-time.</li> </ol>   |

## 6.6 UC-06: View Events

|                              |  |
|------------------------------|--|
| <b>Use Case</b>              | <b>View Events</b>   |
| <b>Goal</b>                  | Allow users to view the event calendar.  |
| <b>Preconditions</b>         |  |
| <b>Success end condition</b> | The event calendar is displayed with all the events.   |
| <b>Failed end condition</b>  | The event calendar is not displayed due to system issues.  |
| <b>Primary Actor</b>         | User   |
| <b>Secondary Actor</b>       | System   |
| <b>Trigger</b>               | User navigates to the event calendar page.   |
| <b>Main Success Flows</b>    | <ol style="list-style-type: none"> <li>1. User navigates to the "Event Calendar" section.</li> <li>2. System retrieves and displays the event calendar.</li> </ol> |
| <b>Alternative Flow</b>      | If the system fails to retrieve the calendar, an error message is displayed.   |
| <b>Quality Requirements</b>  | <ol style="list-style-type: none"> <li>1. The event calendar should load quickly.</li> <li>2. Calendar updates should be reflected in real-time.</li> </ol>        |

## 6.7 UC-07: Check Schedules

|                              |  |
|------------------------------|--|
| <b>Use Case</b>              | <b>Check Schedules</b>   |
| <b>Goal</b>                  | Allow users to check the schedule of matches.  |
| <b>Preconditions</b>         | N/A  |
| <b>Success end condition</b> | The match schedule is displayed.   |
| <b>Failed end condition</b>  | The match schedule is not displayed due to system issues.  |
| <b>Primary Actor</b>         | User   |
| <b>Secondary Actor</b>       | System   |
| <b>Trigger</b>               | User navigates to the match schedule page.   |
| <b>Main Success Flows</b>    | <ol style="list-style-type: none"> <li>1. User navigates to the "Match Schedule" section.</li> <li>2. System retrieves and displays the match schedule.</li> </ol> |
| <b>Alternative Flow</b>      | If the system fails to retrieve the schedule, an error message is displayed.   |
| <b>Quality Requirements</b>  | <ol style="list-style-type: none"> <li>1. The match schedule should load quickly.</li> <li>2. Schedule updates should be reflected in real-time.</li> </ol>        |

## 6.8 UC-08: Request Registration

|                              |   |
|------------------------------|---|
| <b>Use Case</b>              | <b>Request Registration</b>   |
| <b>Goal</b>                  | Allow a representative to request registration in the system.                                       |
| <b>Preconditions</b>         | The representative must have access to the registration form via the system interface.              |
| <b>Success end condition</b> | The representative's registration request is successfully submitted and pending review by the Admin |

|                             |  |
|-----------------------------|--|
| <b>Use Case</b>             | <b>Request Registration</b>  |
| <b>Failed end condition</b> | The registration request is not submitted due to incomplete or invalid information.  |
| <b>Primary Actor</b>        | Representative   |
| <b>Secondary Actor</b>      | System   |
| <b>Trigger</b>              | Representative initiates the registration process by accessing the registration page.  |
| <b>Main Success Flows</b>   | <ul style="list-style-type: none"> <li>• Representative navigates to the registration page.</li> <li>• Representative fills out the registration form with required details (e.g., name, contact information, organization details).</li> <li>• Representative uploads any required supporting documents.</li> <li>• Representative submits the registration request.</li> <li>• System validates the input fields and stores the request in the database.</li> <li>• System notifies the Representative that the request has been submitted successfully and is awaiting review.</li> </ul> |
| <b>Alternative Flow</b>     | <b>Validation Failure:</b> <ul style="list-style-type: none"> <li>• If the system detects missing or invalid information (e.g., incorrect email format), it displays an error message.</li> <li>• The representative is prompted to correct the input and resubmit the form.</li> </ul>  |
| <b>Quality Requirements</b> | <ul style="list-style-type: none"> <li>• The system must validate all input fields before submission.</li> <li>• The registration request must be stored securely in the system.</li> </ul>  |

## 6.9 UC-09: View Results

|                              |  |
|------------------------------|--|
| <b>Use Case</b>              | <b>View Results and Standings</b>  |
| <b>Goal</b>                  | N/A  |
| <b>Preconditions</b>         | Allow users to view match results and standings.   |
| <b>Success end condition</b> | Match results and standings are displayed.   |
| <b>Failed end condition</b>  | Results and standings are not displayed due to system issues.  |
| <b>Primary Actor</b>         | User   |
| <b>Secondary Actor</b>       | System   |
| <b>Trigger</b>               | User navigates to the results and standings page.  |
| <b>Main Success Flows</b>    | <ol style="list-style-type: none"> <li>1. User navigates to the "Results and Standings" section.</li> <li>2. System retrieves and displays the results and standings.</li> </ol> |
| <b>Alternative Flow</b>      | If the system fails to retrieve the results and standings, an error message is displayed.  |
| <b>Quality Requirements</b>  | <ol style="list-style-type: none"> <li>1. Results and standings should load quickly.</li> <li>2. Updates should be reflected in real-time.</li> </ol>                            |

## 6.10 UC-10: Check Notices

|                              |  |
|------------------------------|--|
| <b>Use Case</b>              | <b>Check Notices</b>   |
| <b>Goal</b>                  | Allow users to check recent notices  |
| <b>Preconditions</b>         | N/A  |
| <b>Success end condition</b> | Notices are displayed.   |
| <b>Failed end condition</b>  | Notices are not displayed due to system issues.  |
| <b>Primary Actor</b>         | User   |
| <b>Secondary Actor</b>       | System   |
| <b>Trigger</b>               | User navigates to the notices page.  |
| <b>Main Success Flows</b>    | <ol style="list-style-type: none"> <li>1. User navigates to the "Notices" section.</li> <li>2. System retrieves and displays the notices.</li> </ol> |
| <b>Alternative Flow</b>      | If the system fails to retrieve notices, an error message is displayed.  |
| <b>Quality Requirements</b>  | <ol style="list-style-type: none"> <li>1. Notices should load quickly.</li> <li>2. Updates should be reflected in real-time.</li> </ol>              |

## 6.11 UC-11: View Previous Year Results

|                              |  |
|------------------------------|--|
| <b>Use Case</b>              | <b>View Previous Year Event Results</b>  |
| <b>Goal</b>                  | Allow users to view results of events from the previous year.  |
| <b>Preconditions</b>         | N/A  |
| <b>Success end condition</b> | Previous year event results are displayed.   |
| <b>Failed end condition</b>  | Previous year event results are not displayed due to system issues.  |
| <b>Primary Actor</b>         | User   |
| <b>Secondary Actor</b>       | System   |
| <b>Trigger</b>               | User navigates to the previous year event results page.  |
| <b>Main Success Flows</b>    | <ol style="list-style-type: none"> <li>1. User navigates to the "Previous Year Event Results" section.</li> <li>2. System retrieves and displays the previous year event results.</li> </ol> |
| <b>Alternative Flow</b>      | If the system fails to retrieve previous year event results, an error message is displayed.  |
| <b>Quality Requirements</b>  | <ol style="list-style-type: none"> <li>1. Previous year event results should load quickly.</li> <li>2. Data should be accurate and complete.</li> </ol>                                      |

## 6.12 UC-12: Review Request Registration

|                 |   |
|-----------------|---|
| <b>Use Case</b> | <b>Review Request Registration</b>  |
| <b>Goal</b>     | Allow the admin to review, approve, or reject registration requests submitted by representatives. |

|                              |   |
|------------------------------|---|
| <b>Use Case</b>              | <b>Review Request Registration</b>  |
| <b>Preconditions</b>         | <ul style="list-style-type: none"> <li>• The admin must be logged into the system with administrative privileges.</li> <li>• There must be pending registration requests in the system.</li> </ul>  |
| <b>Success end condition</b> | The registration request is either approved or rejected, and the representative is notified of the outcome.   |
| <b>Failed end condition</b>  | The registration request remains in a pending state due to system errors or incomplete actions.   |
| <b>Primary Actor</b>         | Admin   |
| <b>Secondary Actor</b>       | System  |
| <b>Trigger</b>               | Admin initiates the review of a pending registration request.   |
| <b>Main Success Flows</b>    | <ul style="list-style-type: none"> <li>• Admin logs into the system.</li> <li>• Admin navigates to the "Pending Registration Requests" section.</li> <li>• Admin selects a specific registration request to review.</li> <li>• The system displays the details of the registration request, including personal and organizational information.</li> <li>• Admin reviews the information and decides to approve or reject the request.</li> <li>• Admin submits the decision (approve/reject).</li> <li>• The system updates the status of the registration request and notifies the representative of the outcome.</li> </ul> |
| <b>Alternative Flow</b>      | <p><b>Missing or Invalid Data:</b></p> <ul style="list-style-type: none"> <li>• If required information in the request is incomplete or invalid, the system flags the issue.</li> <li>• Admin can request additional information or clarification from the representative before making a decision.</li> </ul>  |
| <b>Quality Requirements</b>  | <ul style="list-style-type: none"> <li>• The system must ensure that all required fields in the registration request are filled in before submission.</li> <li>• Notifications of approval or rejection should be sent to the representative immediately.</li> <li>• The system must log the admin's decision for auditing purposes.</li> </ul>   |

## 6.13 UC-13: Manage Team Details

|                              |   |
|------------------------------|---|
| <b>Use Case</b>              | <b>Manage Team Details</b>  |
| <b>Goal</b>                  | Allow department representatives to create and manage team details.         |
| <b>Preconditions</b>         | Department representative must be logged in.                                |
| <b>Success end condition</b> | Teams are created and updated with the correct player information.          |
| <b>Failed end condition</b>  | Teams are not created or updated due to validation errors or system issues. |

|                             |  |
|-----------------------------|--|
| <b>Use Case</b>             | <b>Manage Team Details</b>   |
| <b>Primary Actor</b>        | Department Representative  |
| <b>Secondary Actor</b>      | System   |
| <b>Trigger</b>              | Department representative initiates the team management process.   |
| <b>Main Success Flows</b>   | <ol style="list-style-type: none"> <li>1. Department representative logs into the system.</li> <li>2. Representative navigates to the team management section.</li> <li>3. Representative enters or updates team details.</li> <li>4. Representative adds players to the team.</li> <li>5. Representative submits the team information.</li> <li>6. System validates and saves the team information.</li> <li>7. System confirms the updates to the representative.</li> </ol> |
| <b>Alternative Flow</b>     | If validation fails, the system displays an error message and prompts the representative to correct the input.   |
| <b>Quality Requirements</b> | <ol style="list-style-type: none"> <li>1. The system should ensure data integrity and consistency.</li> <li>2. Updates should be reflected immediately.</li> </ol>   |

## 6.14 UC-14: Assign Department Representative

|                              |   |
|------------------------------|---|
| <b>Use Case</b>              | <b>Assign Department Representative</b>   |
| <b>Goal</b>                  | Assign a representative for each department responsible for managing team details and other department-specific tasks.  |
| <b>Preconditions</b>         | The user must be logged in with administrative privileges.  |
| <b>Success end condition</b> | A department representative is successfully assigned and notified.  |
| <b>Failed end condition</b>  | The representative is not assigned due to validation errors or system issues.   |
| <b>Primary Actor</b>         | Admin   |
| <b>Secondary Actor</b>       | System  |
| <b>Trigger</b>               | Admin initiates the process to assign a department representative.  |
| <b>Main Success Flows</b>    | <ol style="list-style-type: none"> <li>1. Admin logs into the system.</li> <li>2. Admin navigates to the "Assign Department Representative" section.</li> <li>3. Admin selects a department from the list.</li> <li>4. Admin chooses a representative from the list of users.</li> <li>5. Admin assigns the representative to the department.</li> <li>6. System validates the input and updates the department representative information.</li> <li>7. System notifies the newly assigned representative.</li> </ol> |
| <b>Alternative Flow</b>      | If validation fails, the system displays an error message and prompts the admin to correct the input.   |
| <b>Quality Requirements</b>  | <ol style="list-style-type: none"> <li>1. The system must validate all input fields.</li> <li>2. Changes should reflect in the system in real-time.</li> <li>3. The system should notify the representative immediately upon assignment.</li> </ol>   |

## 7. Activity Diagram

### 7.1 UC-01: Update Event Calendar:

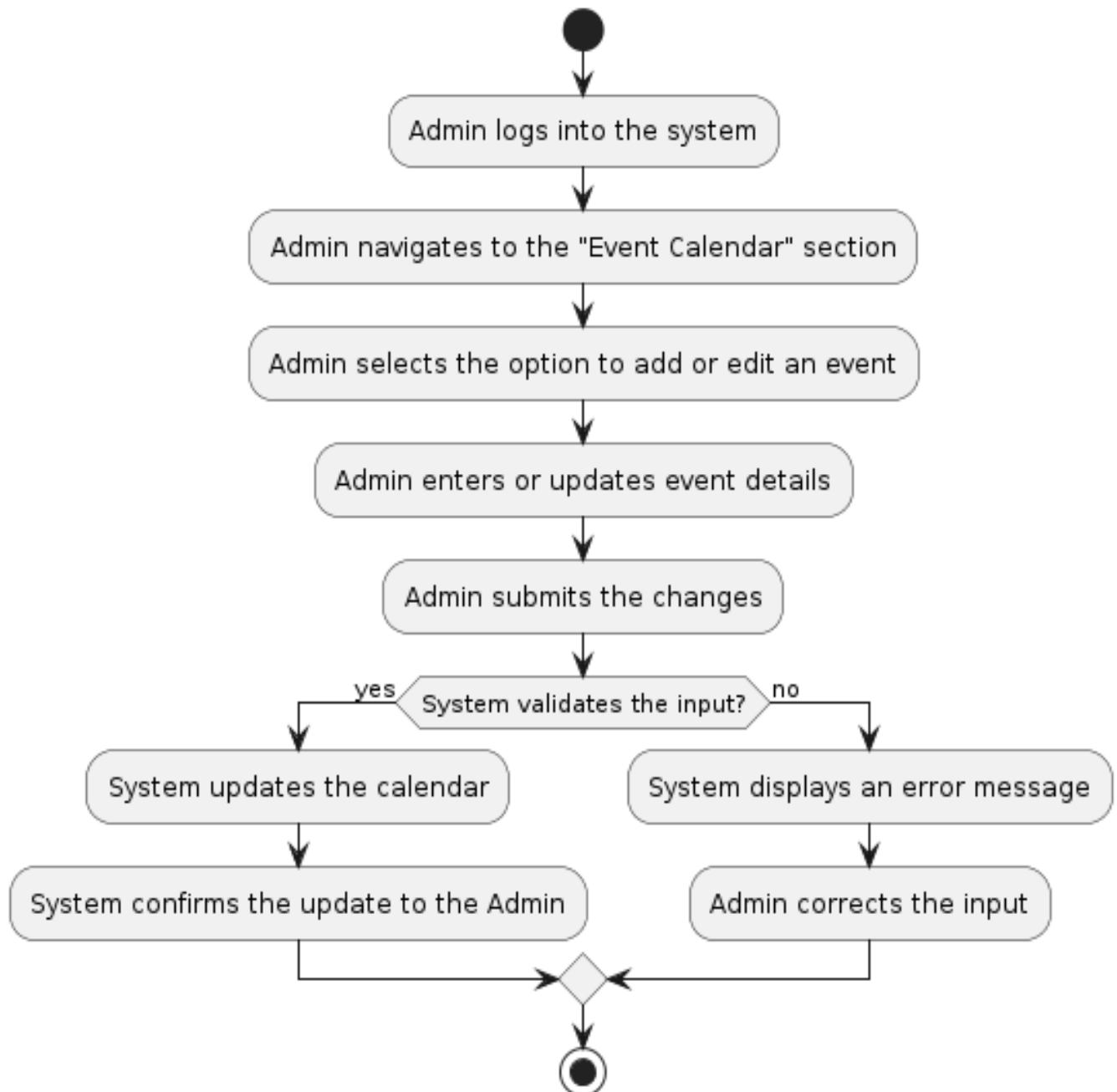


Figure 2: Update Event Calendar



## 7.2 UC-02: Update Match Schedule:

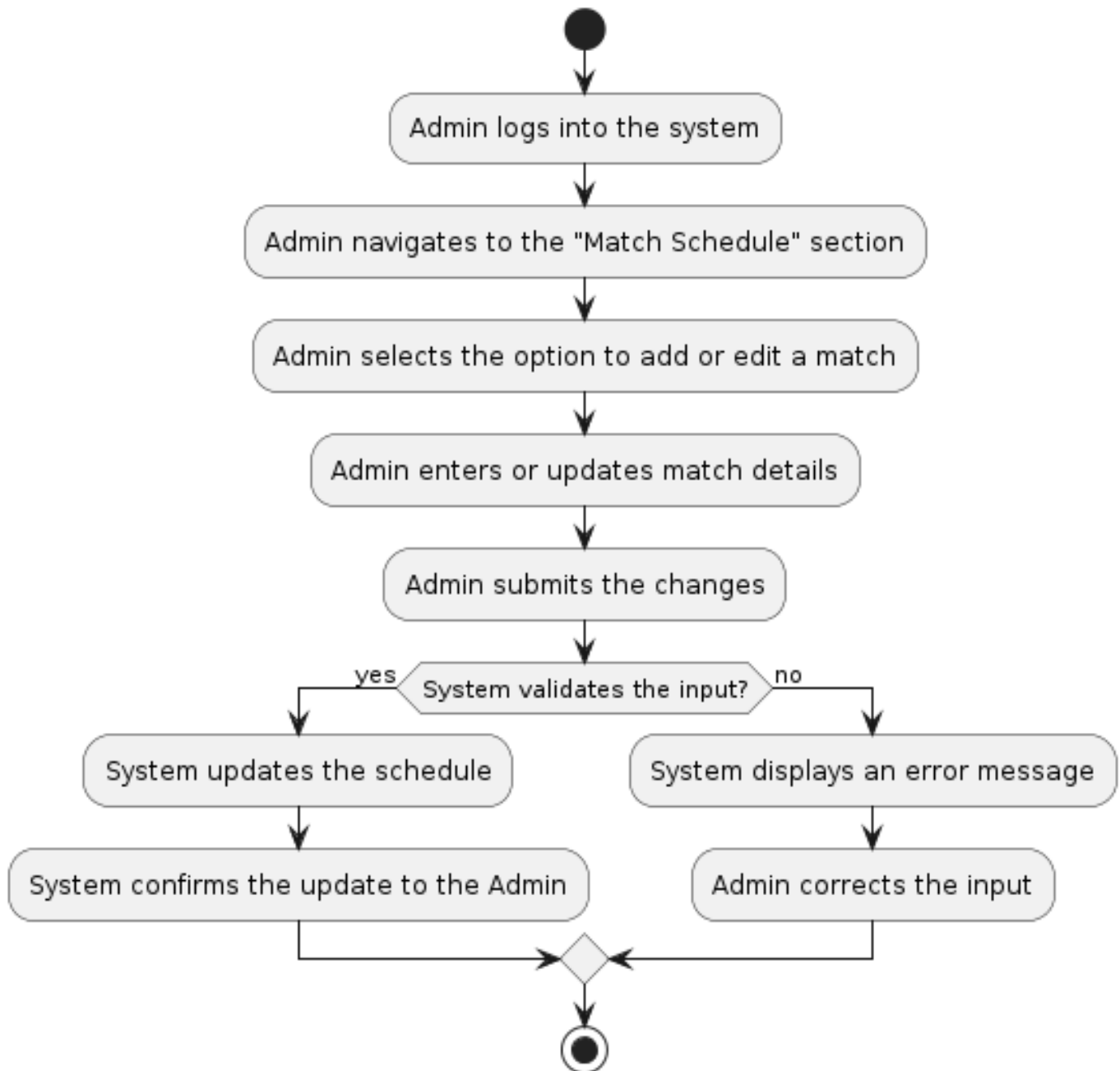


Figure 3: Update Match Schedule

### 7.3 UC-03: Update live Scores:

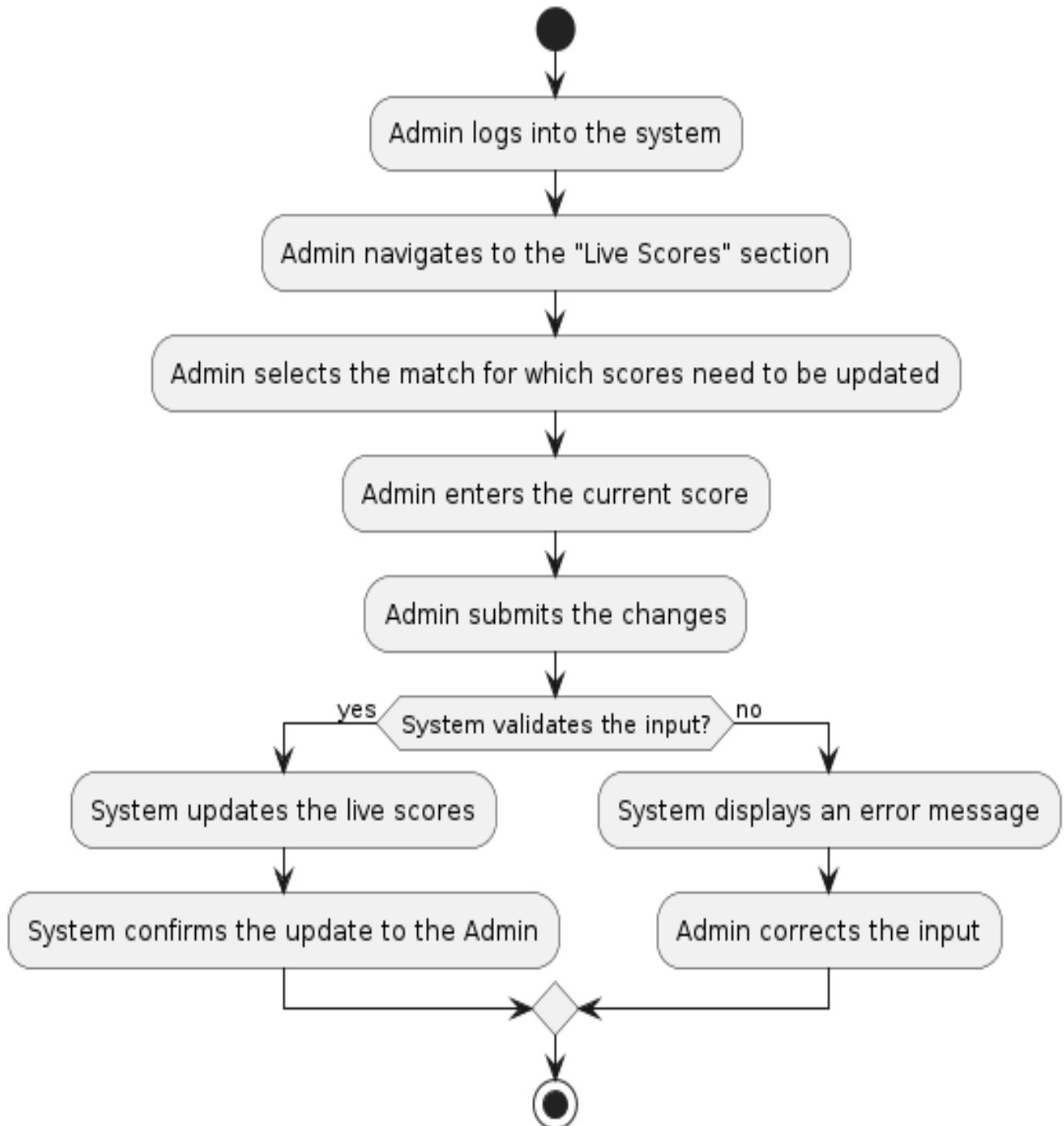


Figure 4: Update live Scores

#### 7.4 UC-04: Update Result and Standings:

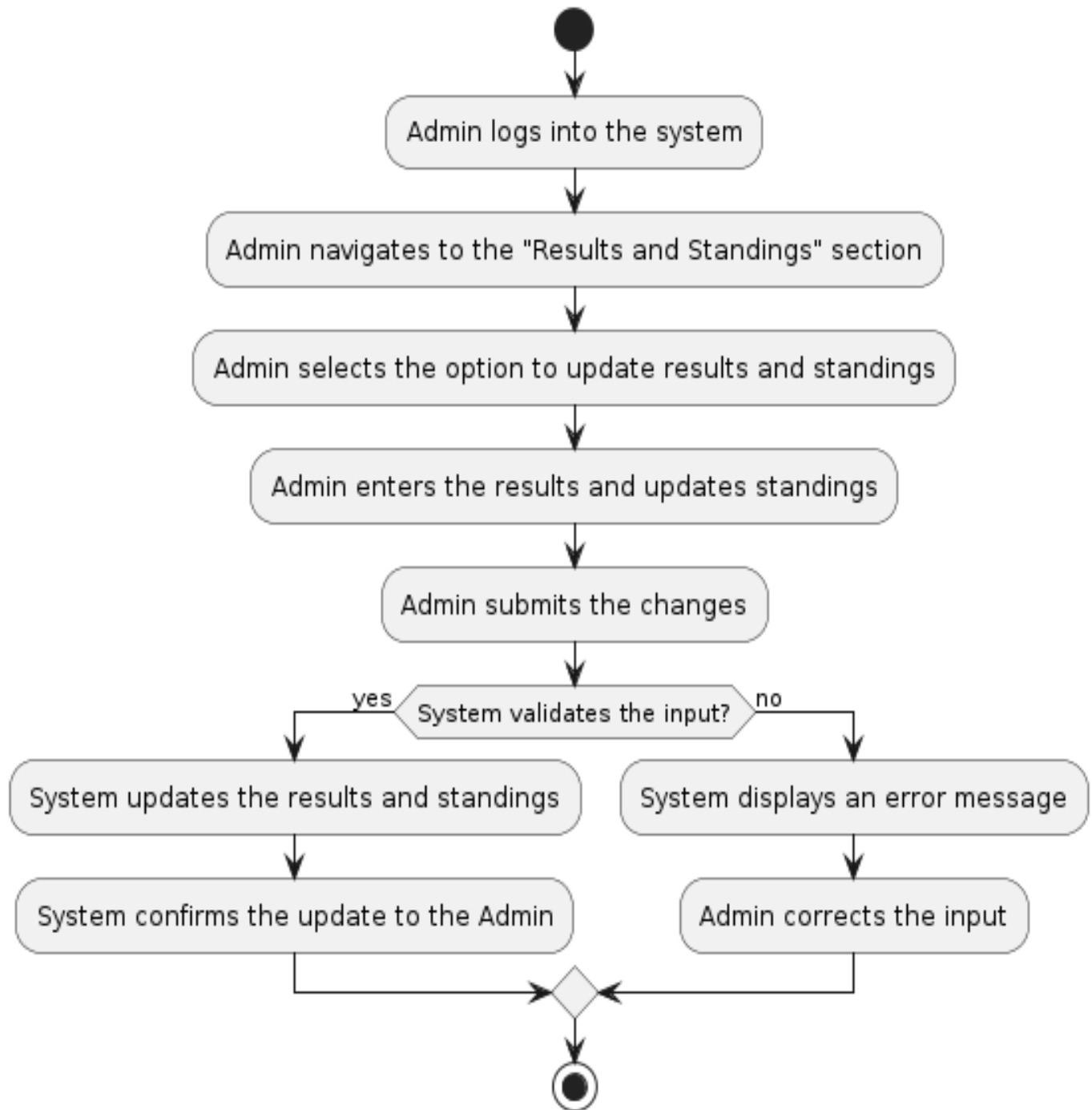


Figure 5: Update Result and Standings

## 7.5 UC-05: Post Notices:

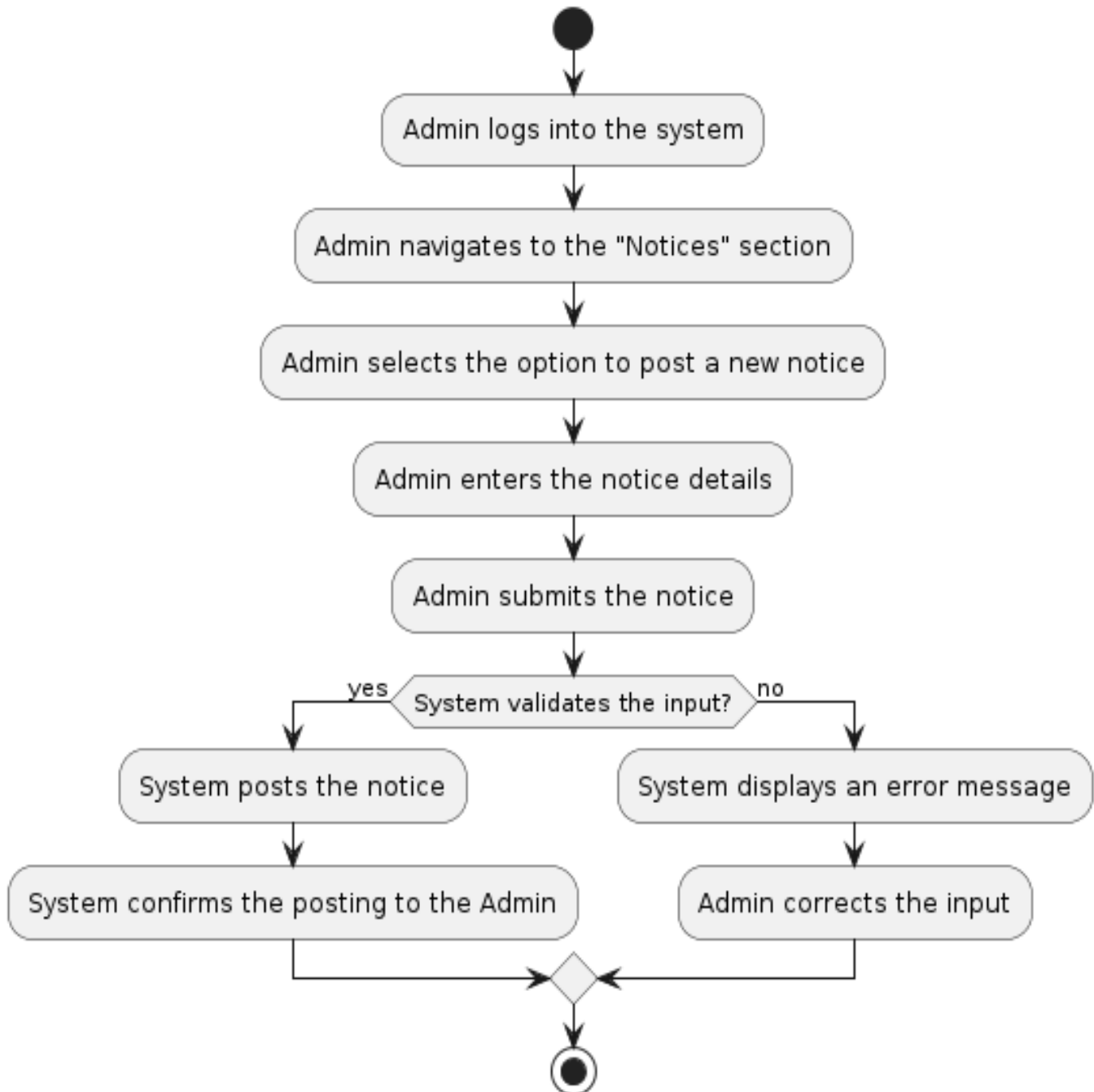


Figure 6: Post Notices

## 7.6 UC-06: View Event Calendar

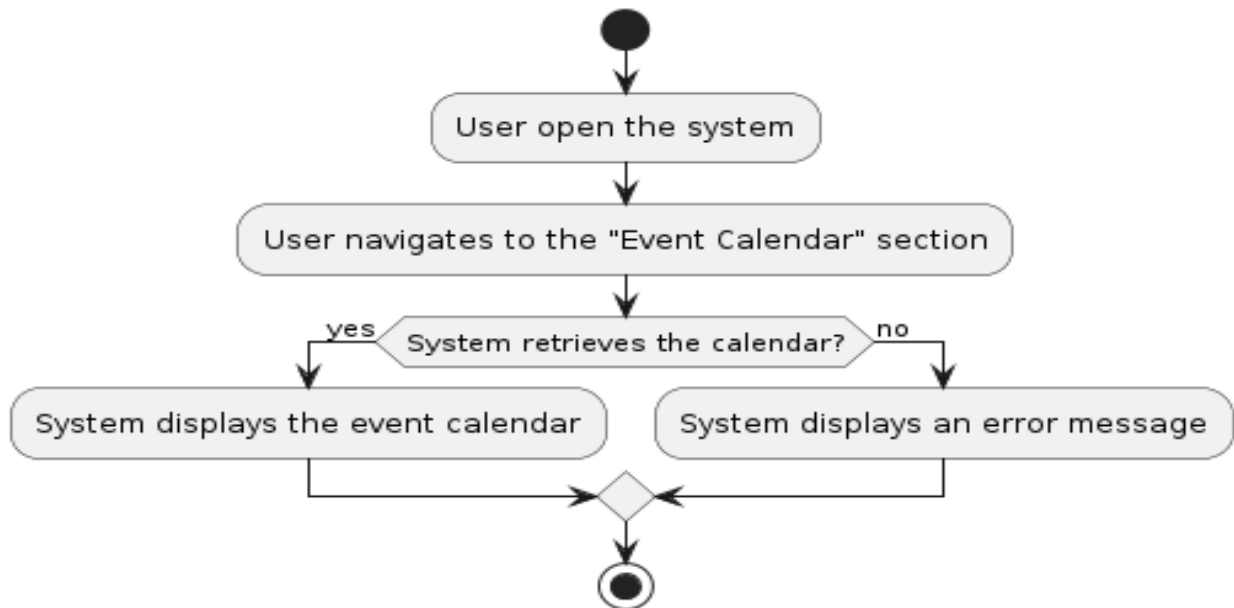


Figure 7: View Event Calendar

## 7.7 UC-07: Check Match Schedule

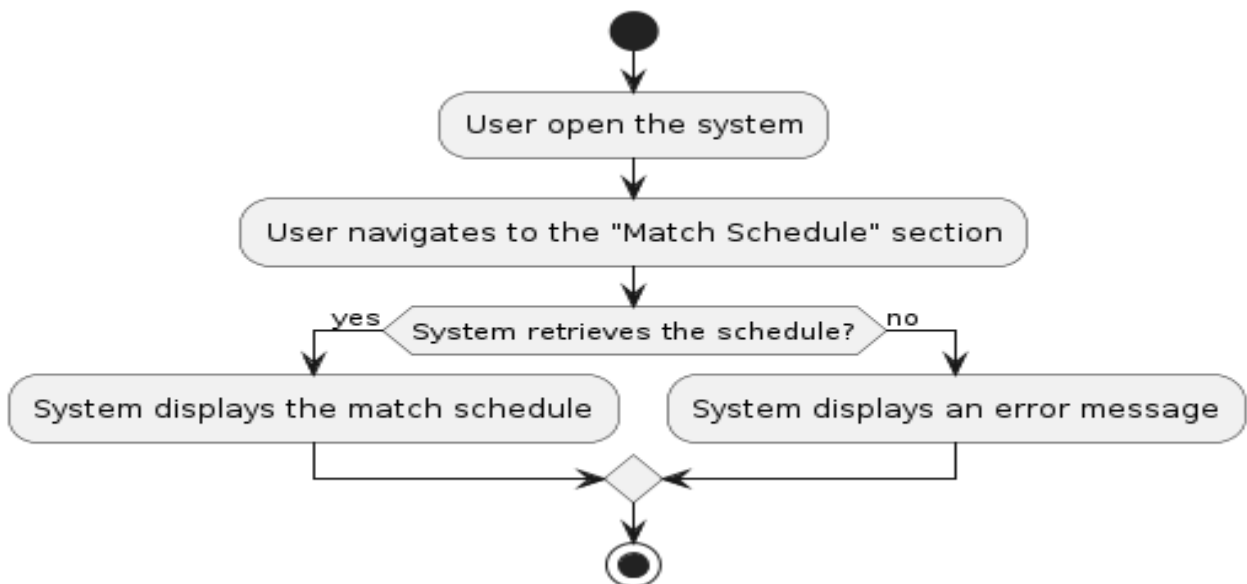


Figure 8: Check Match Schedule

## 7.8 UC-08: Check Live Scores

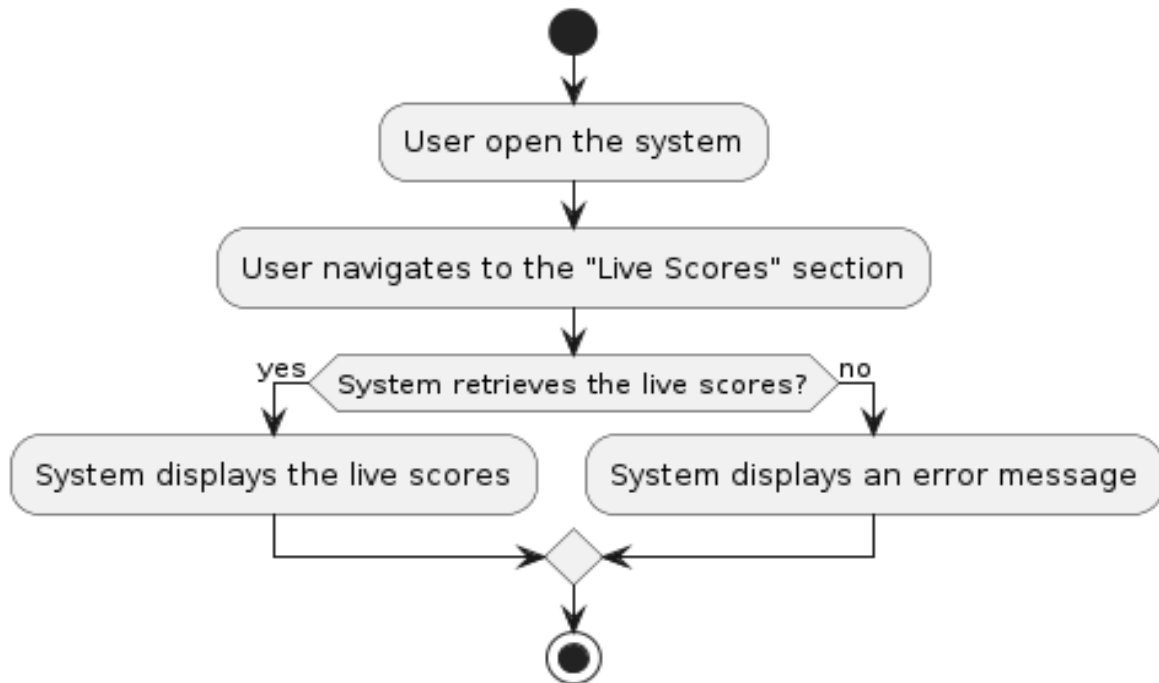


Figure 9: Check Live Scores

## 7.9 UC-09: View Results and Standings

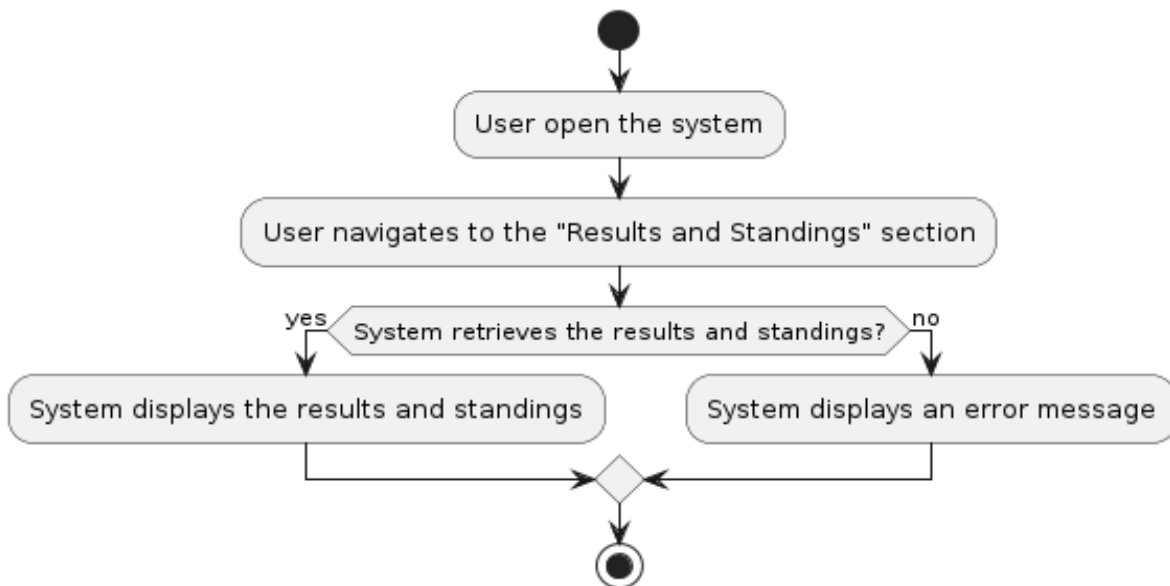
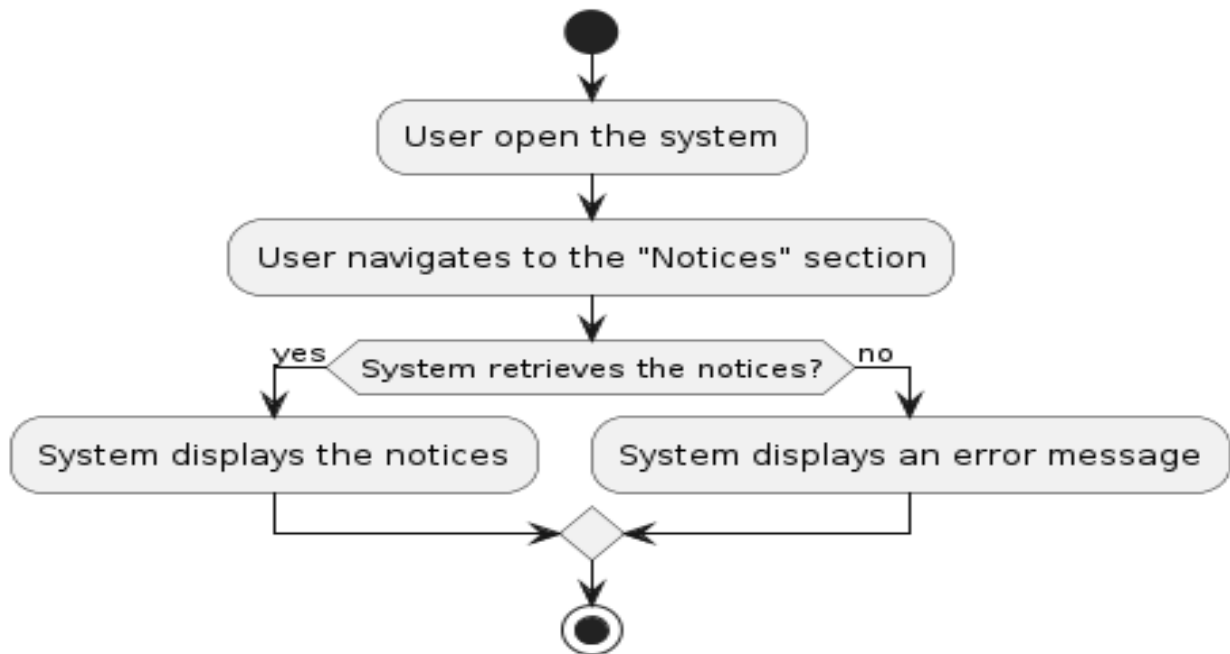
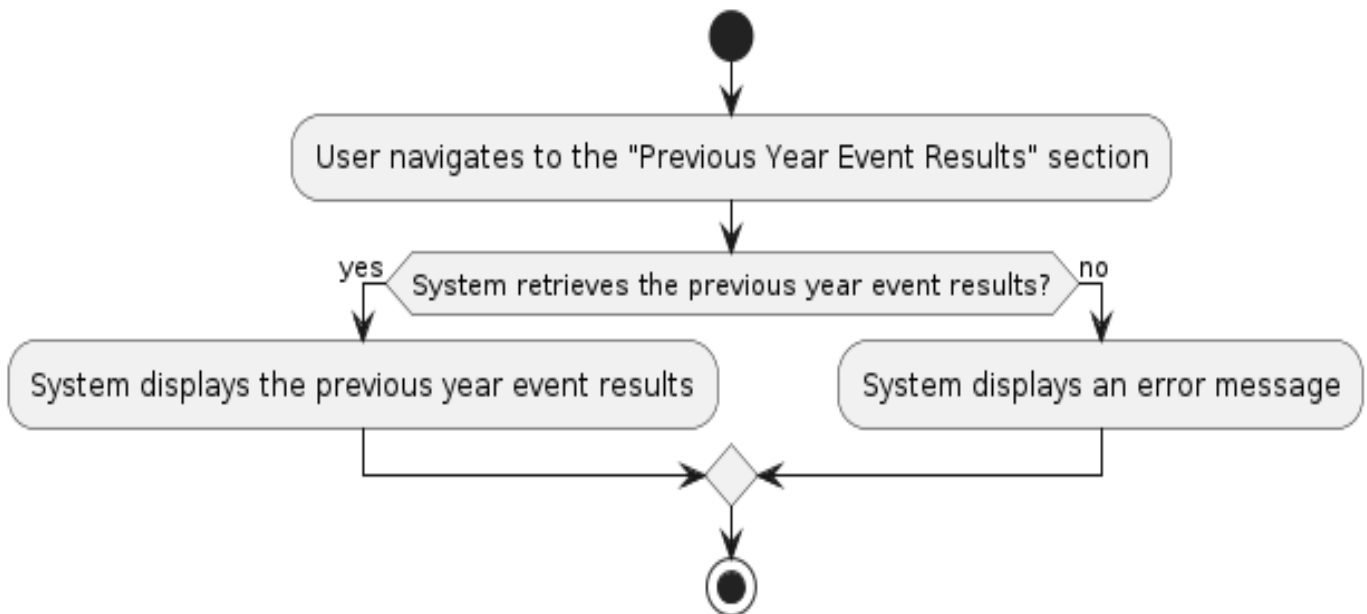


Figure 10: View Results and Standings

**7.10 UC-10: Check Notices***Figure 11: Check Notices***7.11 UC-11: View Previous Year Event Results***Figure 12: View Previous Year Event Results*

## 7.12 UC-12: Share Event

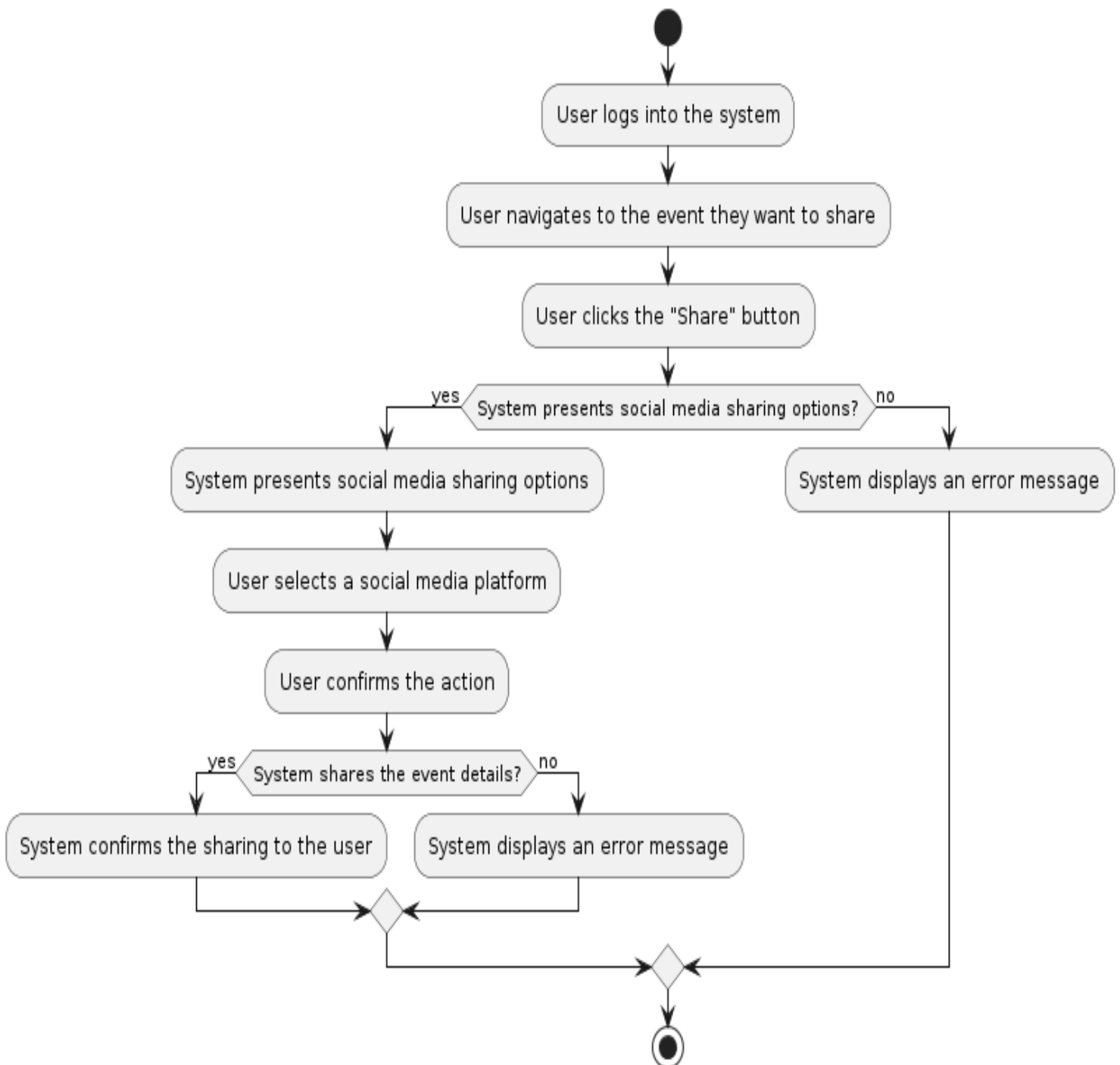


Figure 13: Share Event



### 7.13 UC-13: Manage Team Details

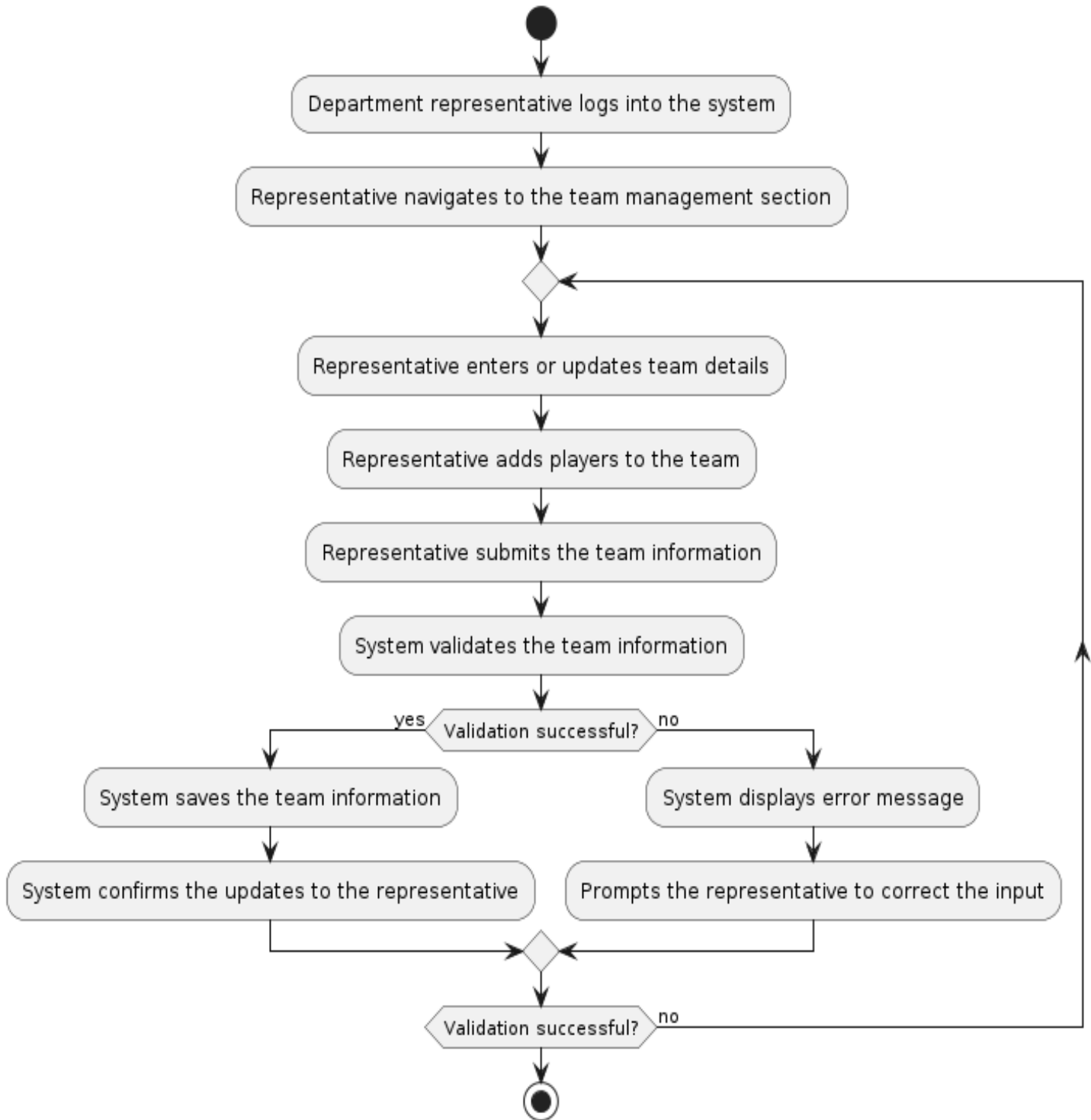
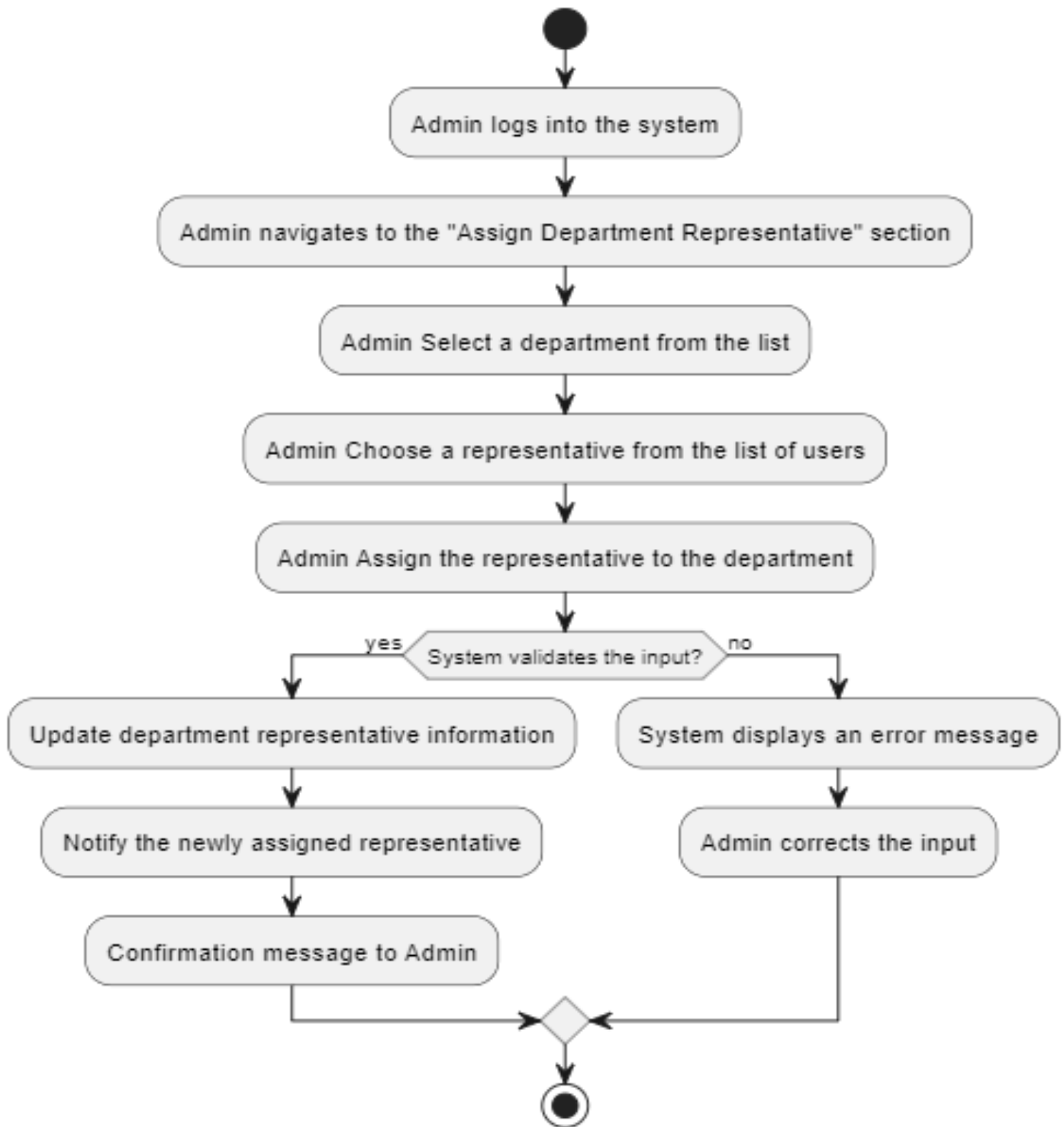


Figure 14: Manage Team Details

**7.14 UC-14: Assign Department Representative***Figure 15: Assign Department Representative*

## 8. Sequence Diagram

### 8.1 UC-01: Update event calendar:

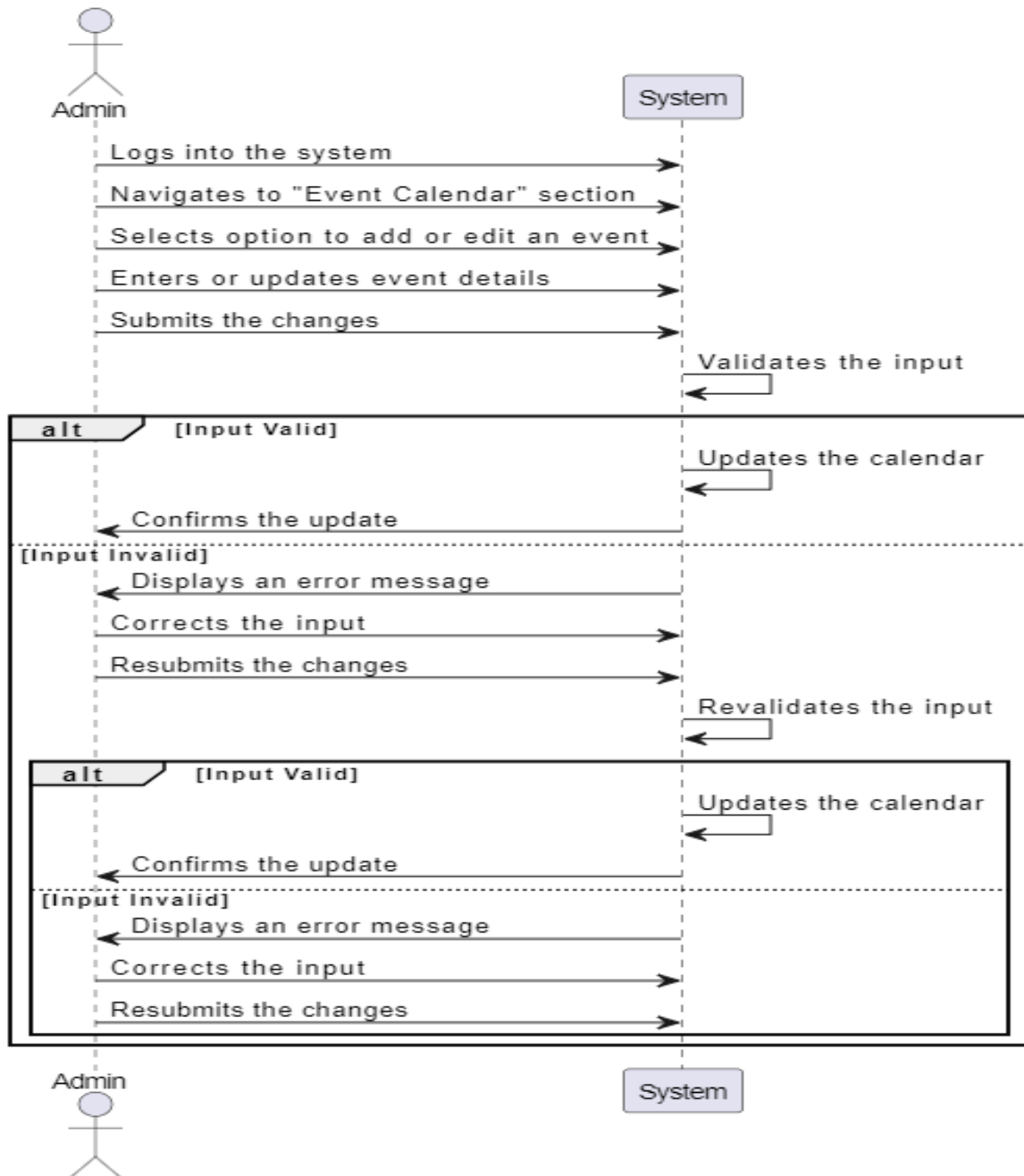


Figure 16: Update Event Calendar

## 8.2 UC-02: Update Match schedule:

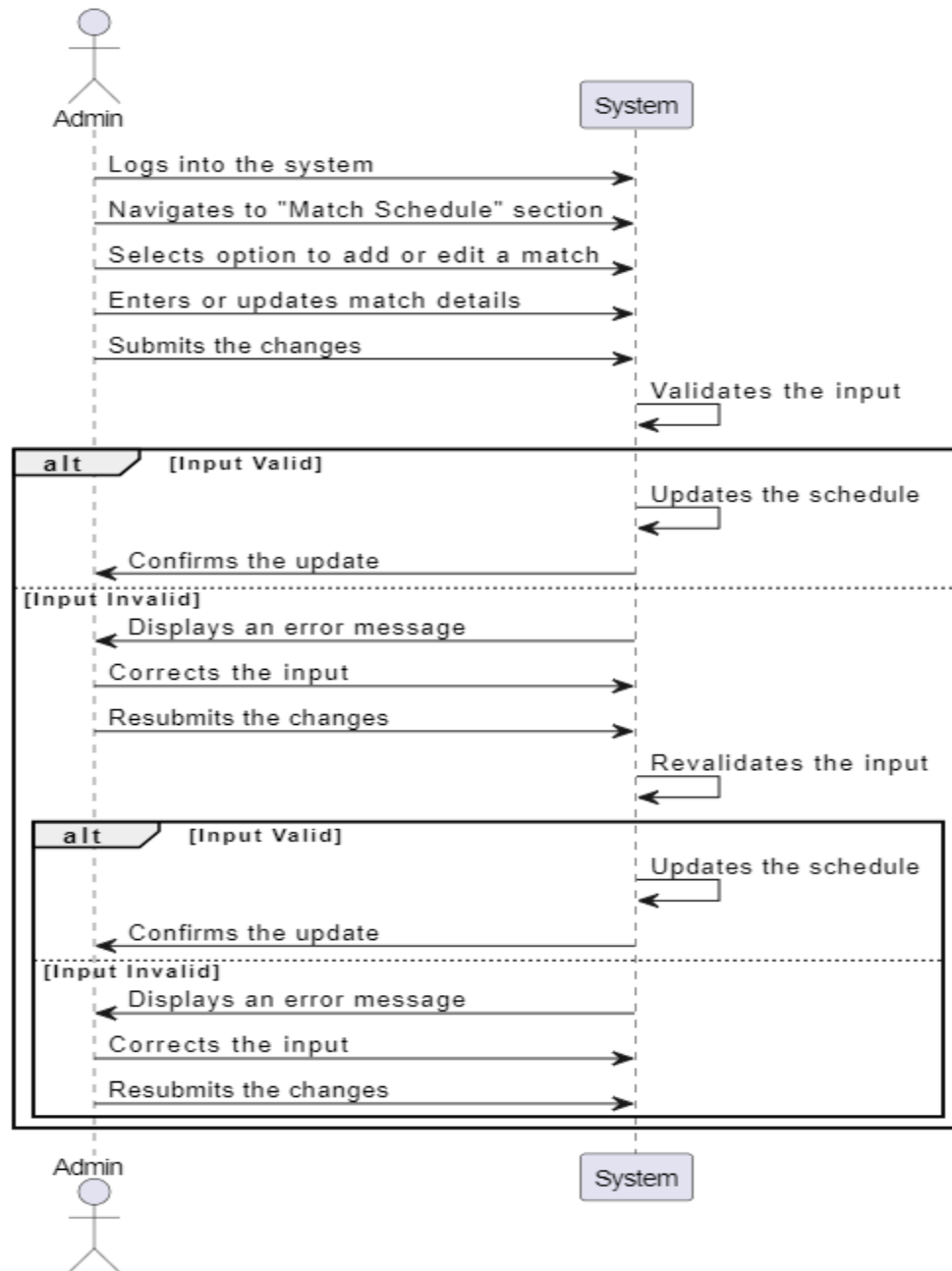


Figure 17: Update Match Schedule

### 8.3 UC-03: Update live scores:

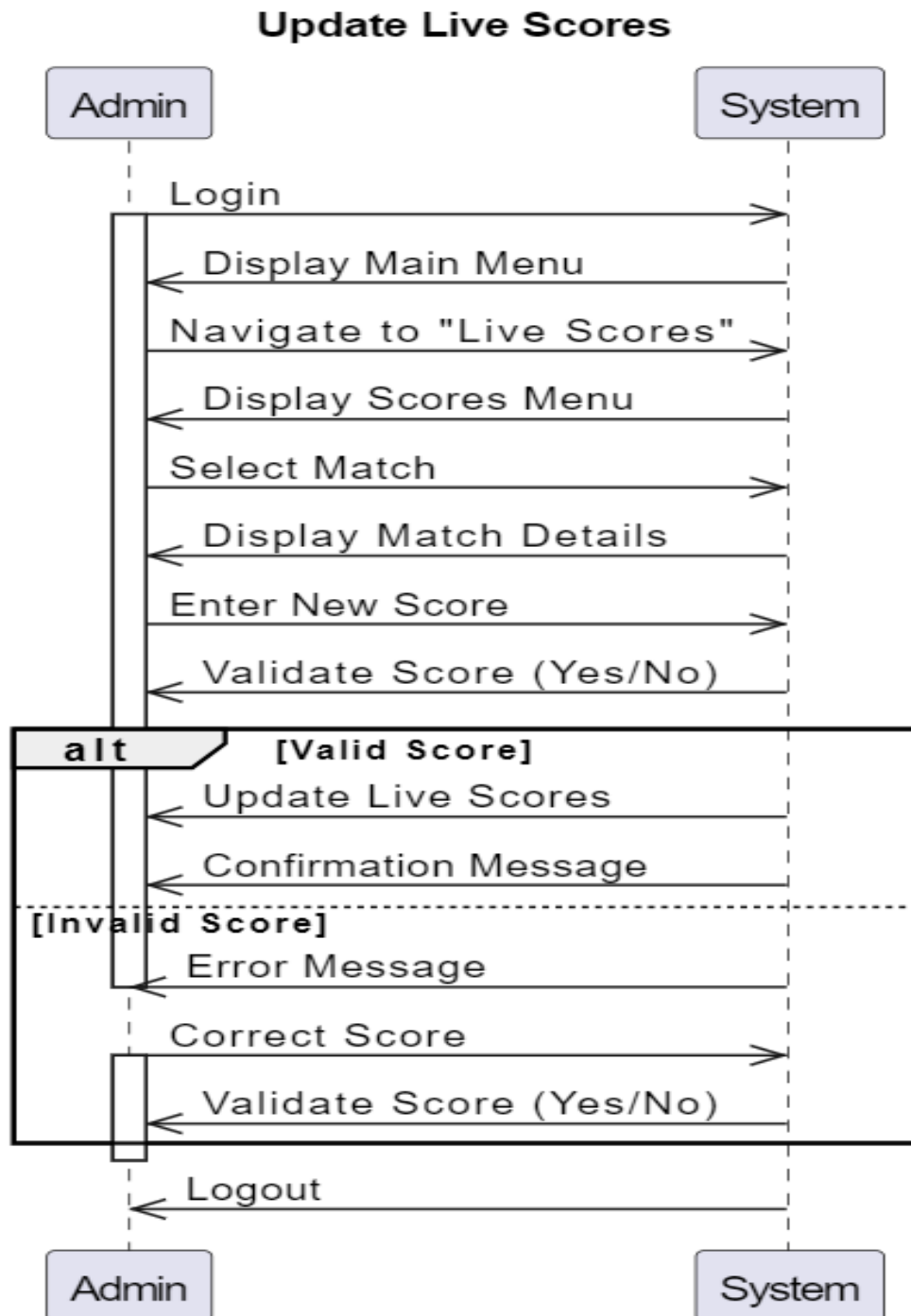


Figure 18: Update Live Scores

## 8.4 UC-04: Update result and standing:

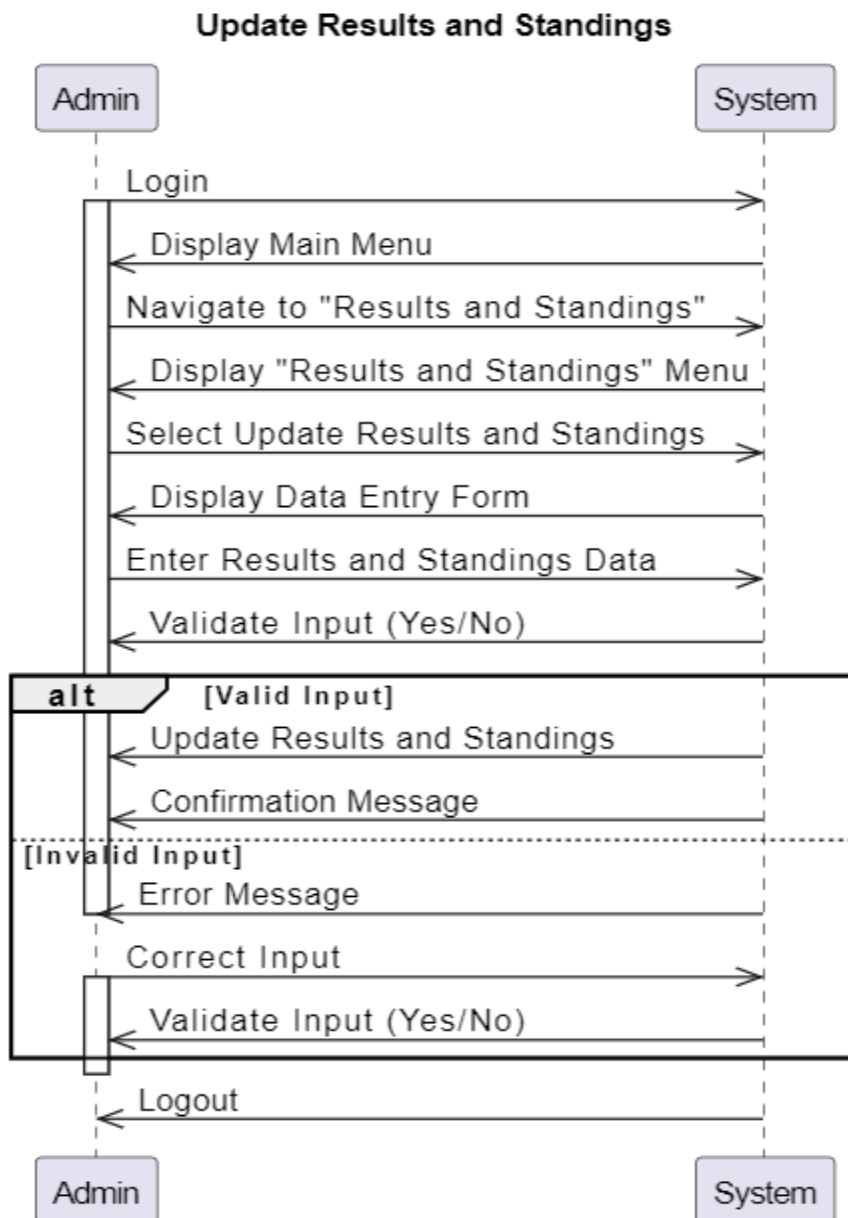


Figure 19: Update Results and Standings

## 8.5 UC-05: Post Notices:

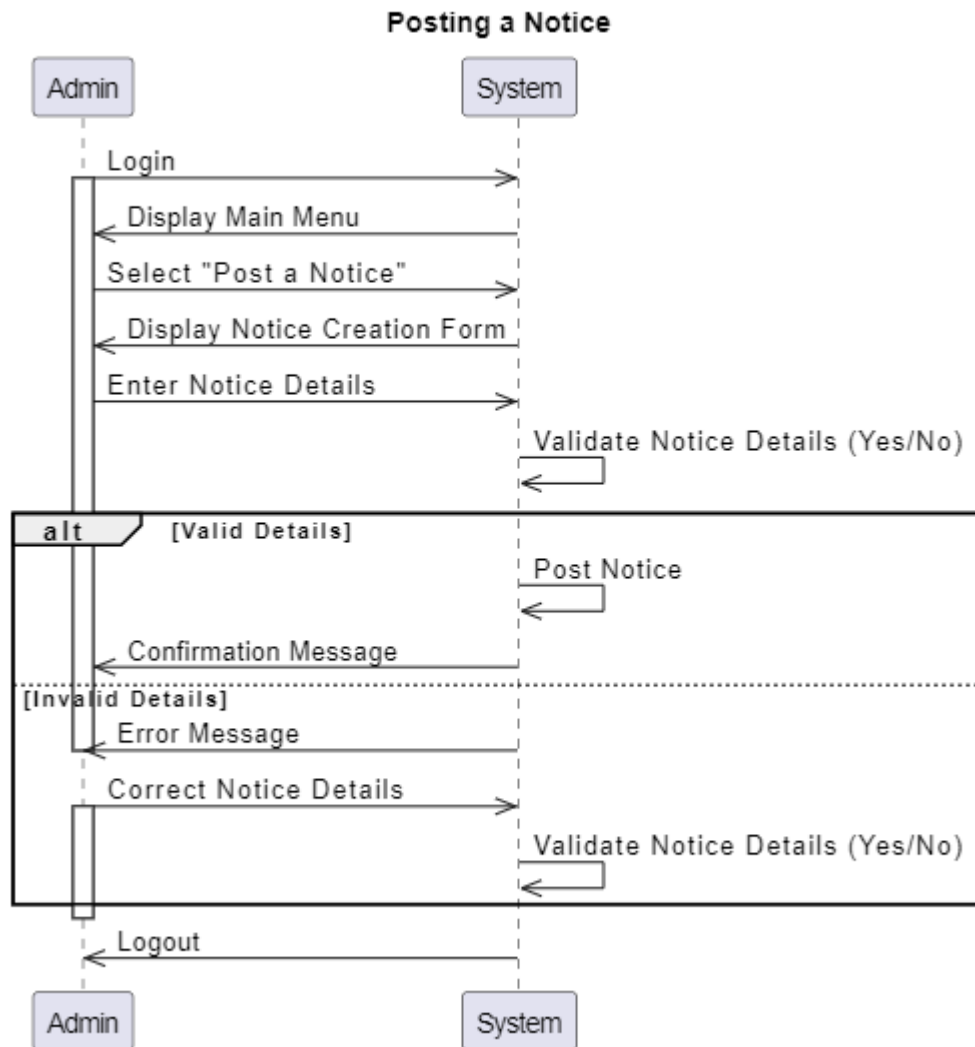


Figure 20: Post Notices

## 8.6 UC-06: View Event Calendar

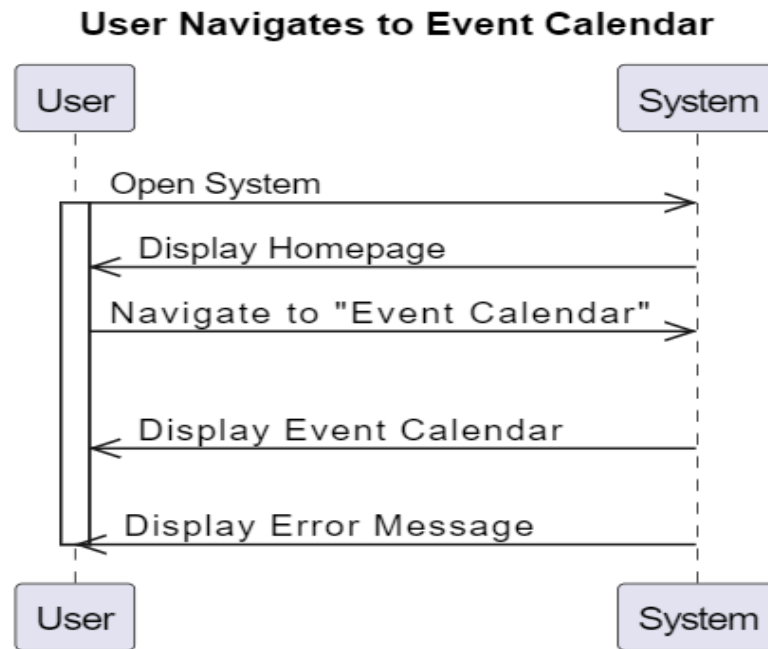


Figure 21: View Event Calendar

## 8.7 UC-07: Check Match Schedule

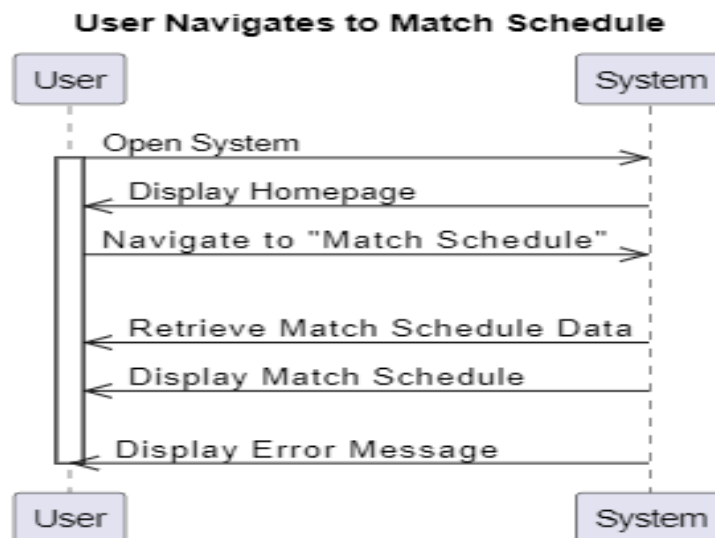


Figure 22: Check Match Schedule



## 8.8 UC-08: Check Live Scores

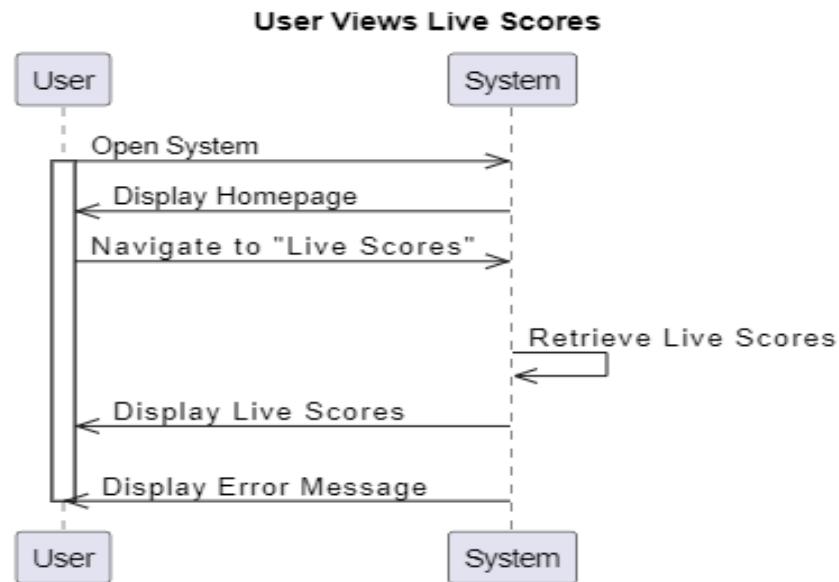


Figure 23: Check Live Scores

## 8.9 UC-09: View Results and Standings

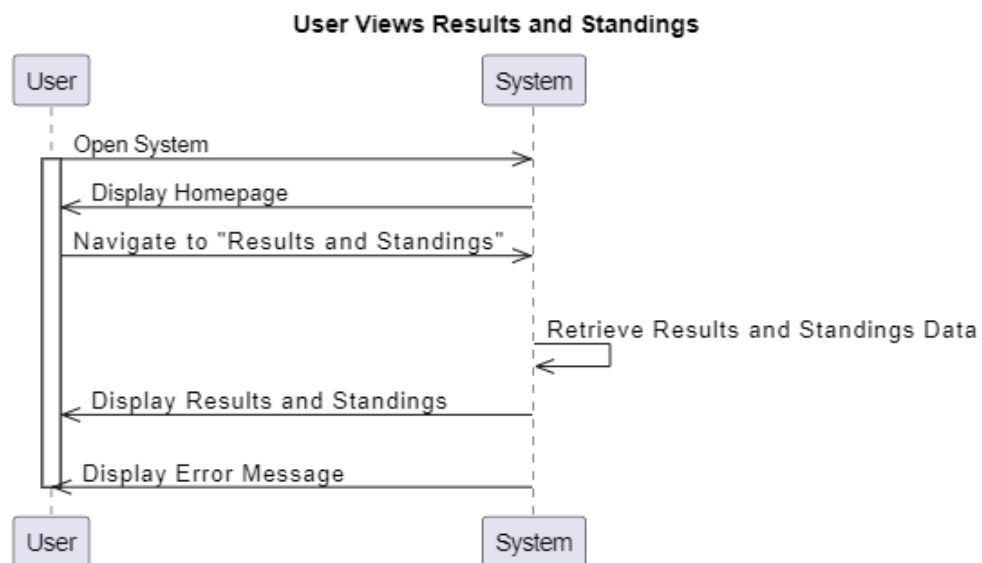


Figure 24: View Results and Standings

## 8.10 UC-10: Check Notices

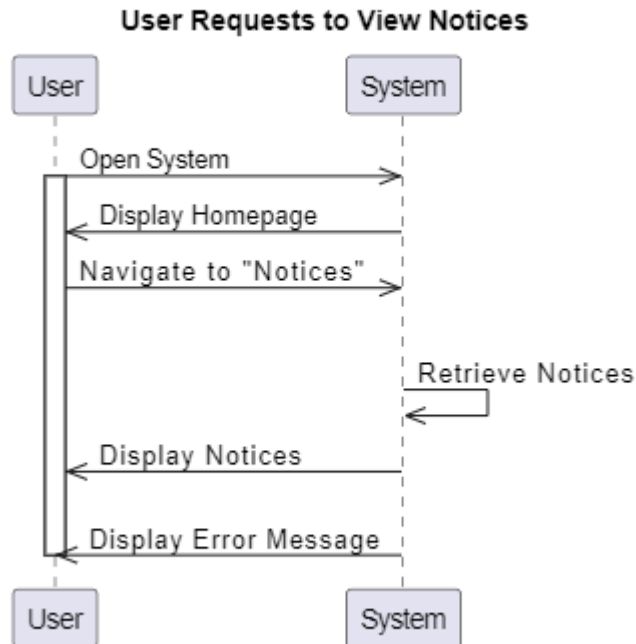


Figure 25: Check Notices

## 8.11 UC-11: View Previous Year Event Results

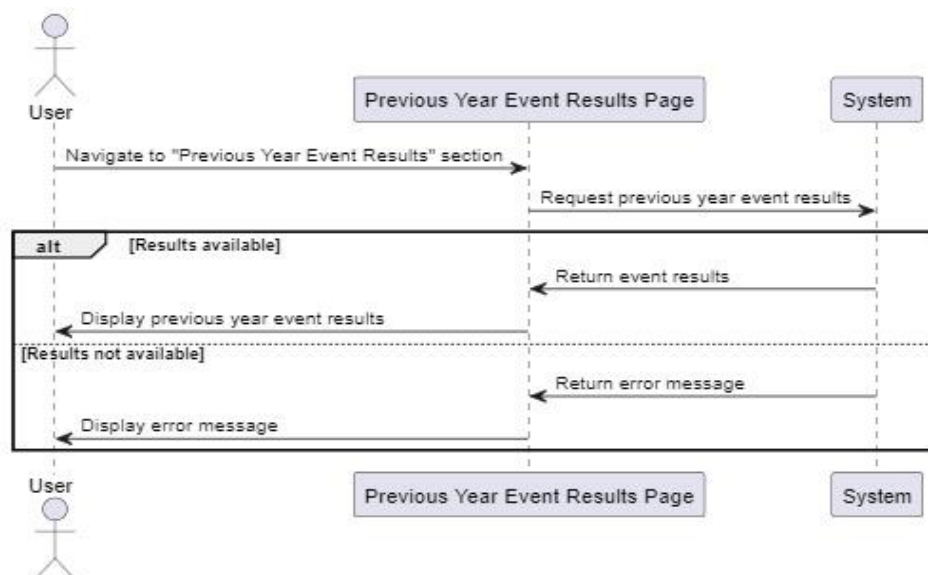


Figure 26: View Previous Year Event Results

## 8.12 UC-12: Share Event

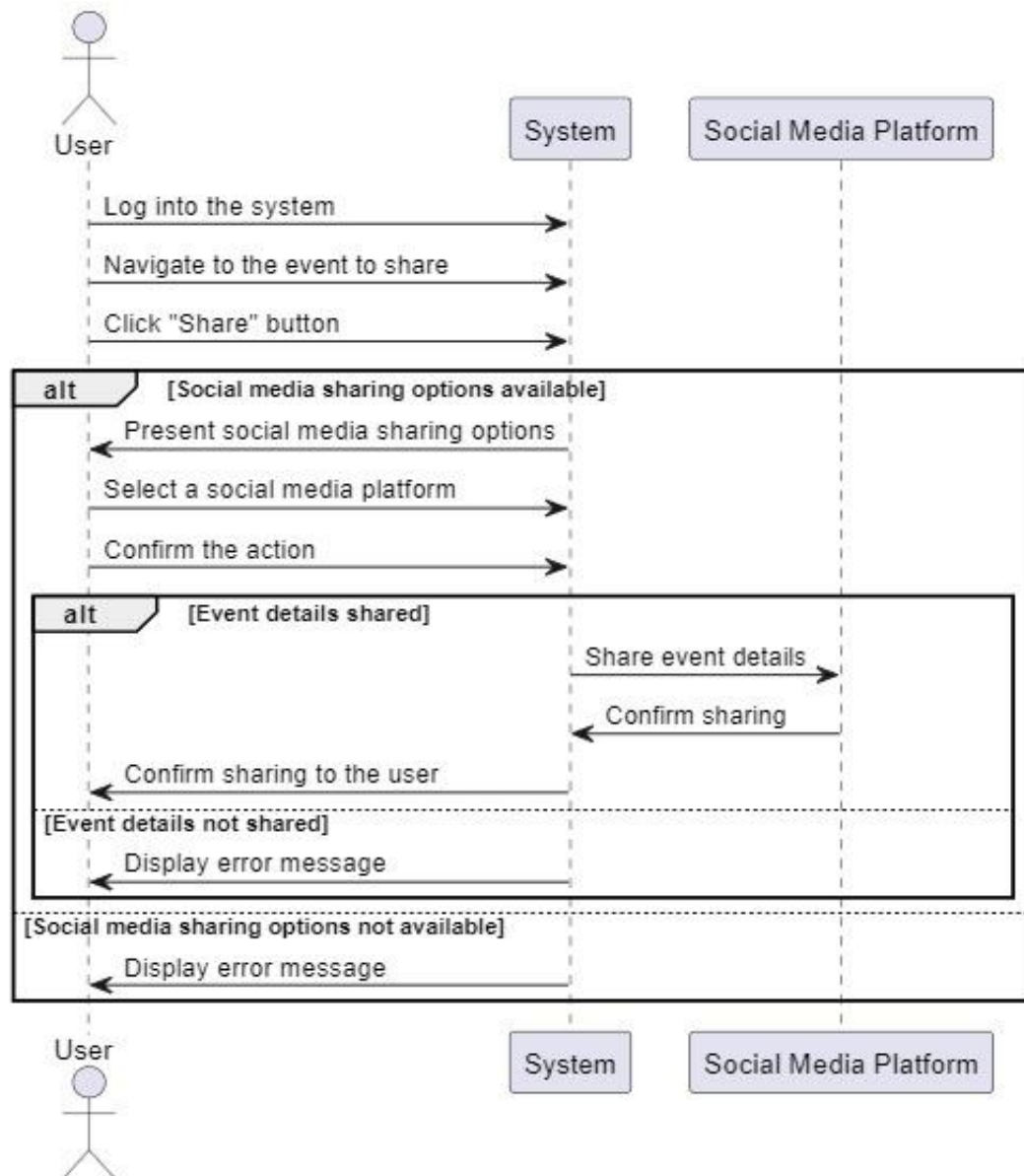


Figure 27: Update Event Calendar

### 8.13 UC-13: Manage Team Details

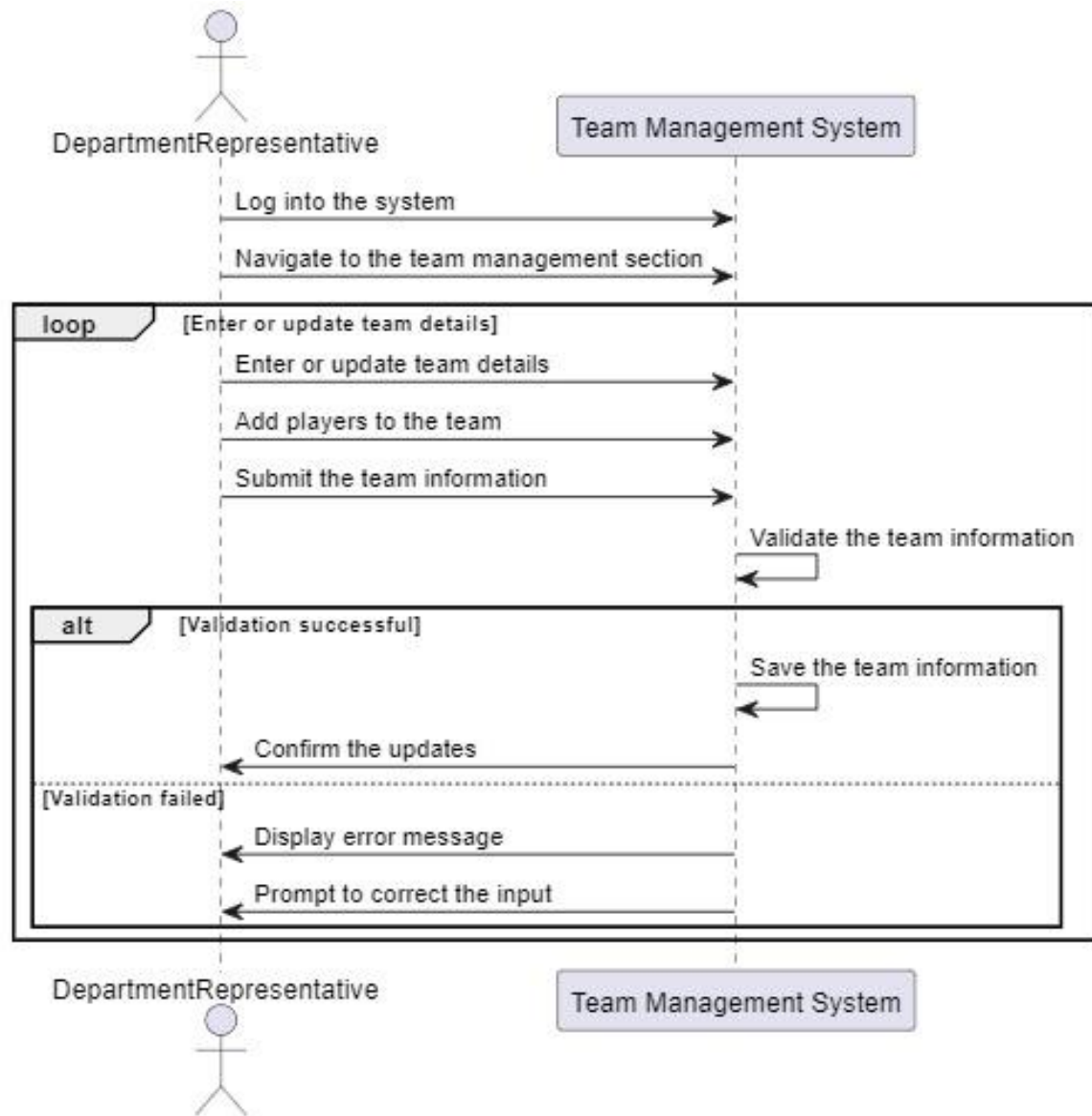


Figure 28: Manage Team Details

## 8.14 UC-14: Assign Department Representative

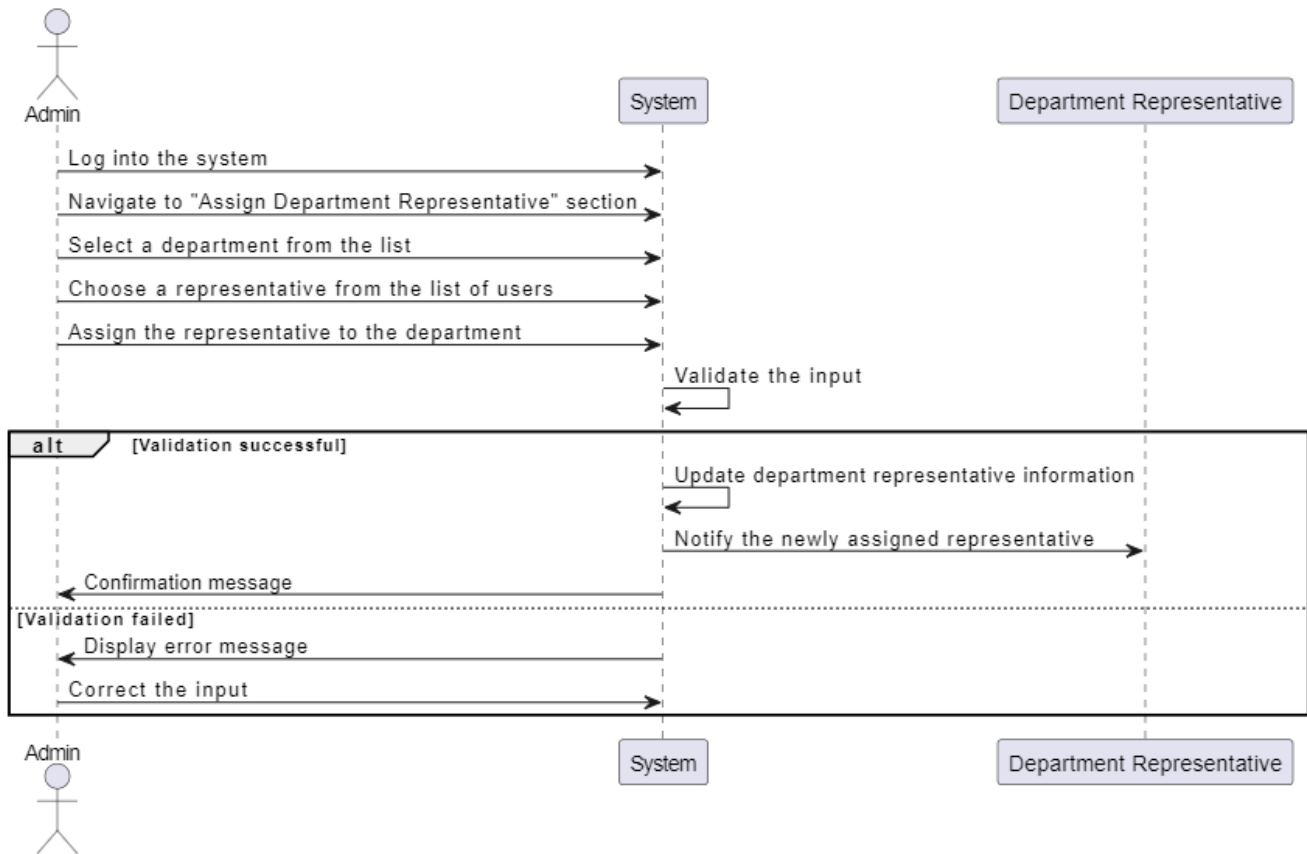


Figure 29: Assign Department Representatives

## 9. Swim lanes:

### 9.1 UC-01: Update Event Calendar

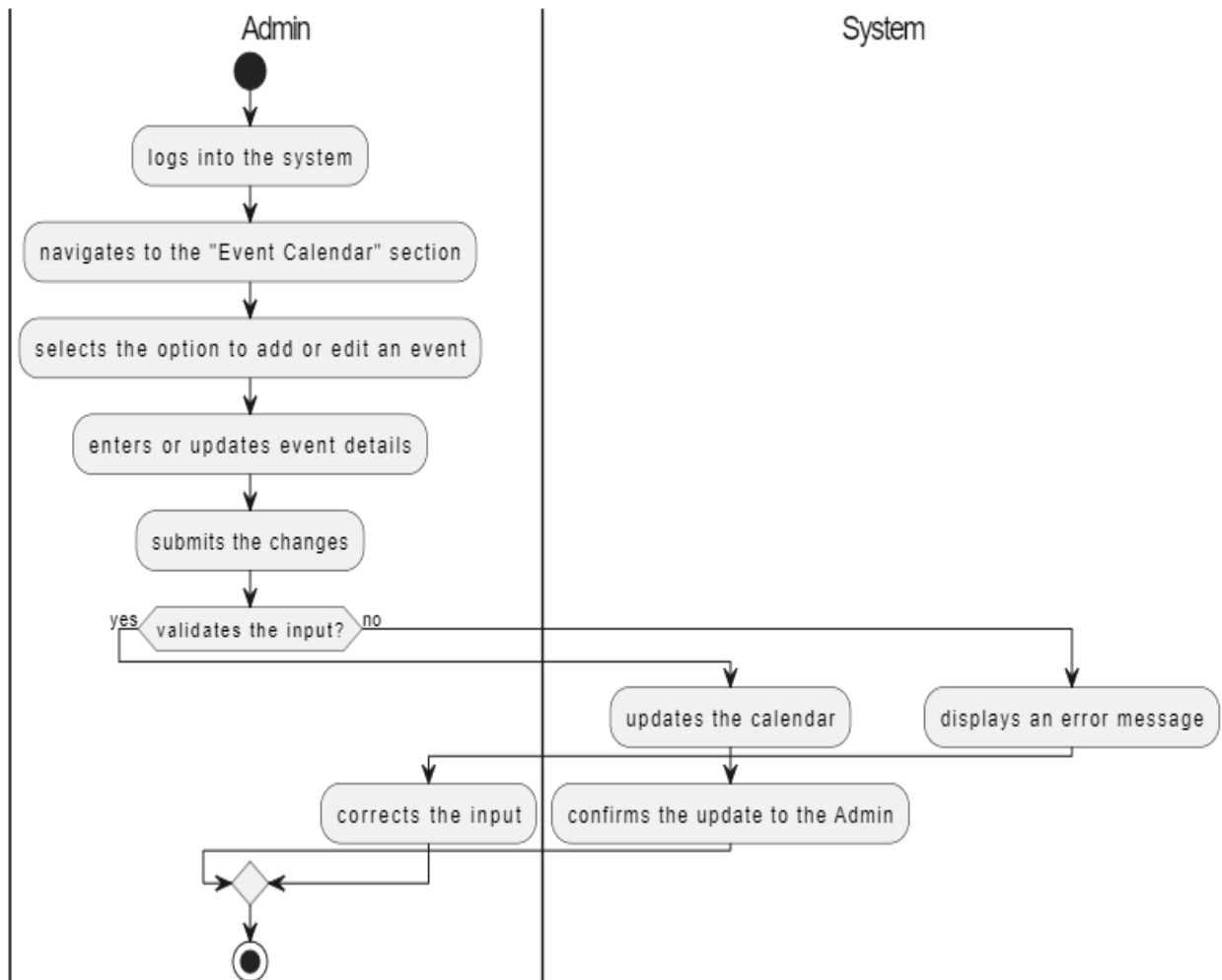


Figure 30: Update Event Calendar

## 9.2 UC-02: Update Match Schedule

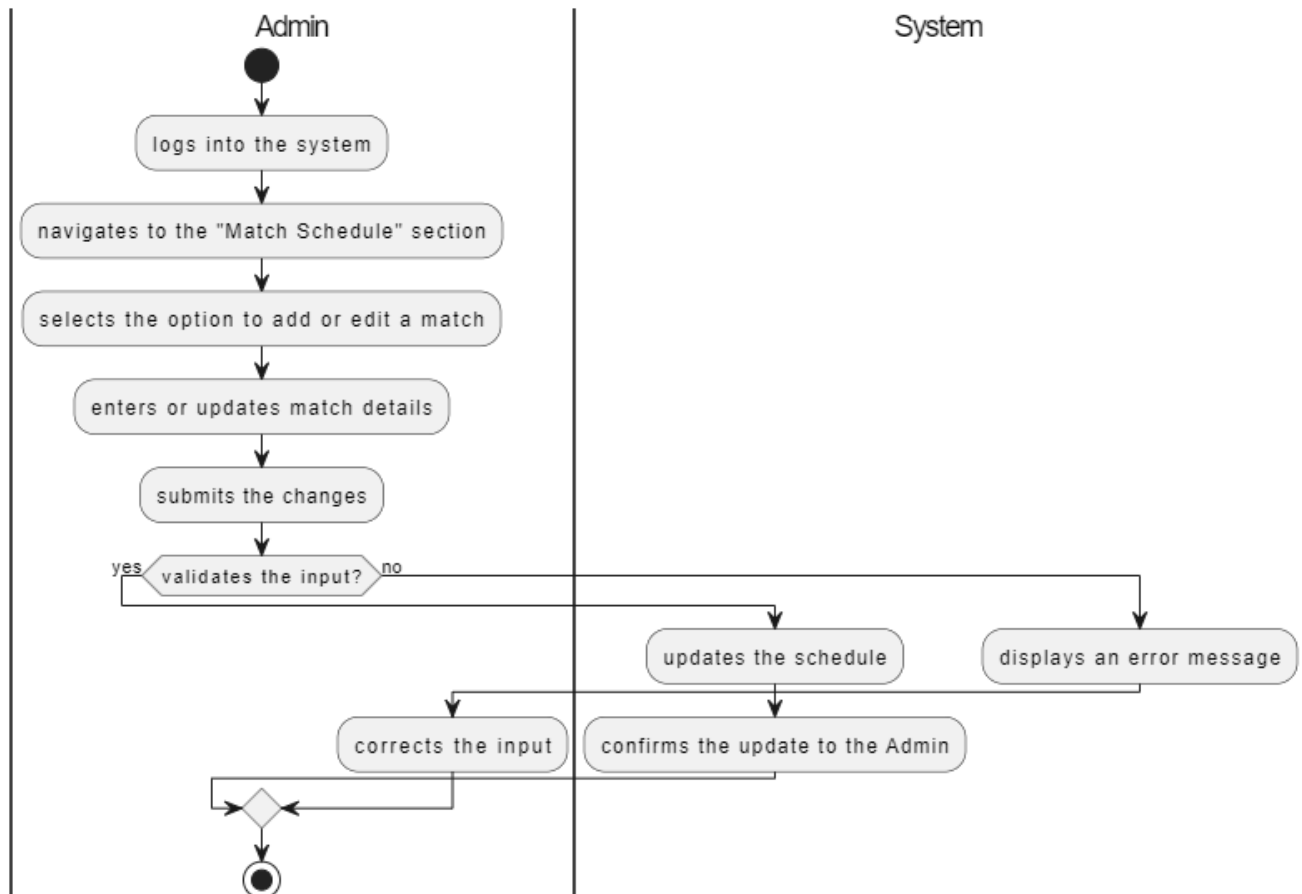


Figure 31: Update Match Schedule

### 9.3 UC-03: Update Live Scores

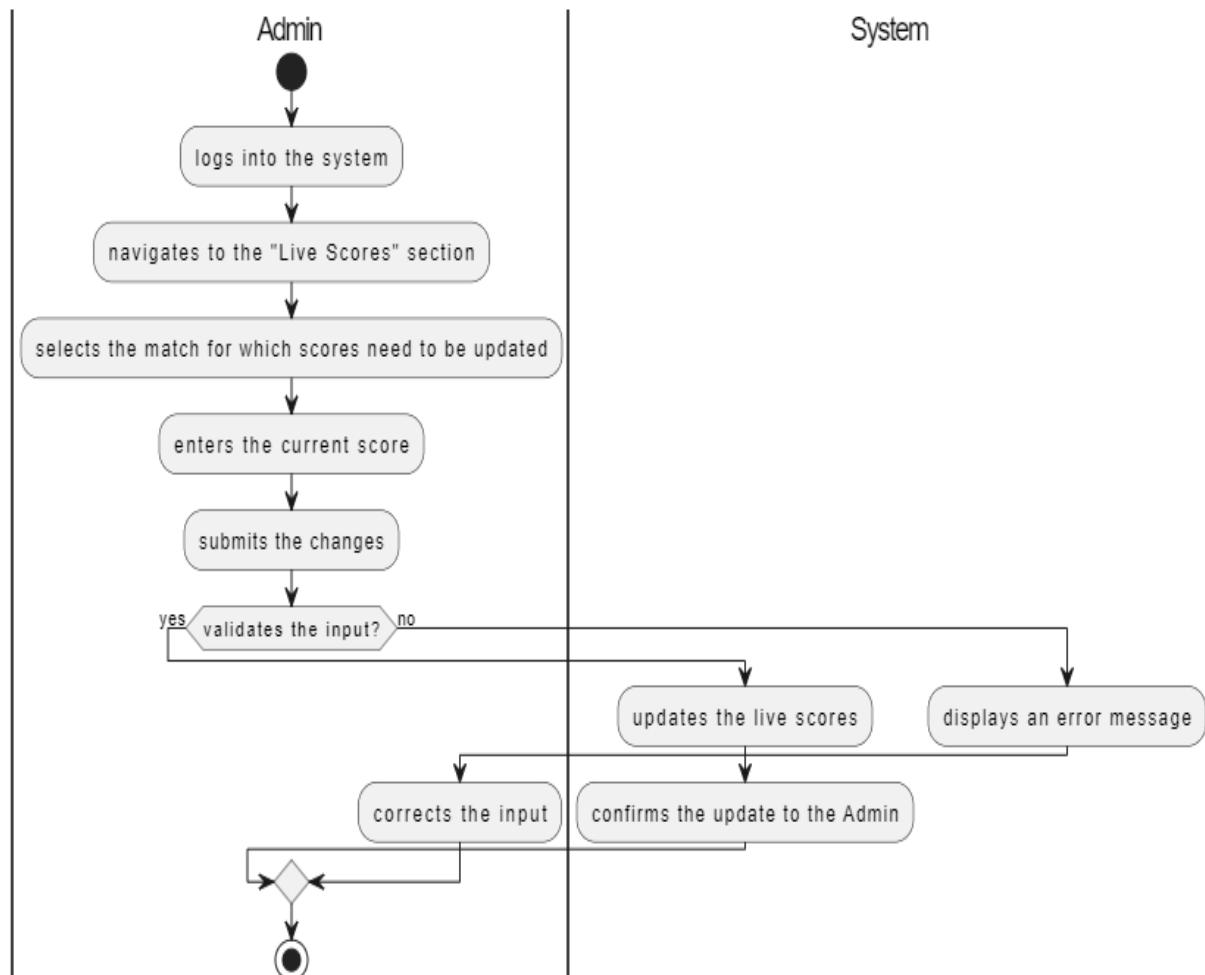


Figure 32: Update Live Scores



## 9.4 UC-04: Update Results and Standings

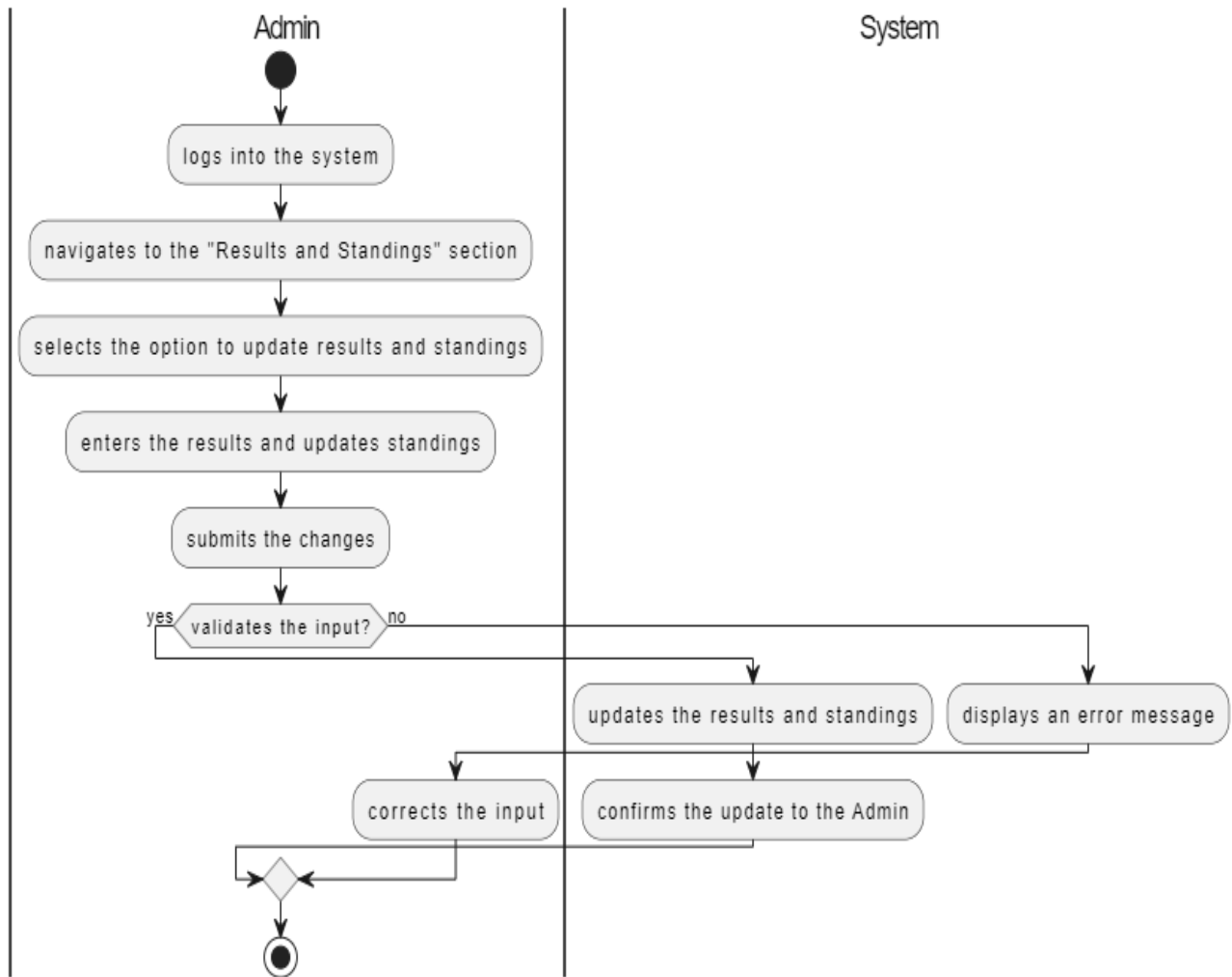


Figure 33: Update Results and Standings

## 9.5 UC-05: Post Notices

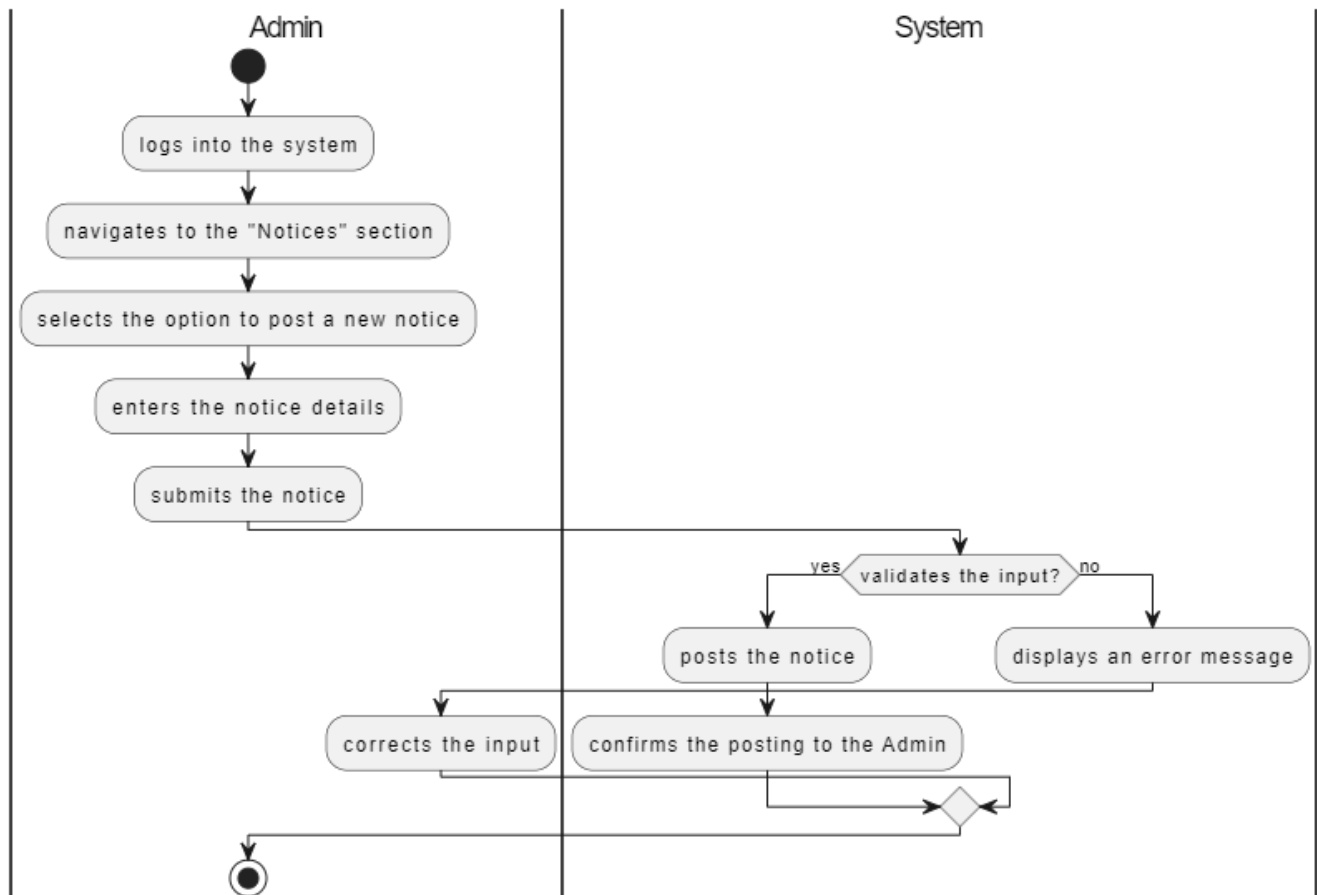


Figure 34: Post Notices

## 9.6 UC-06: View Event Calendar

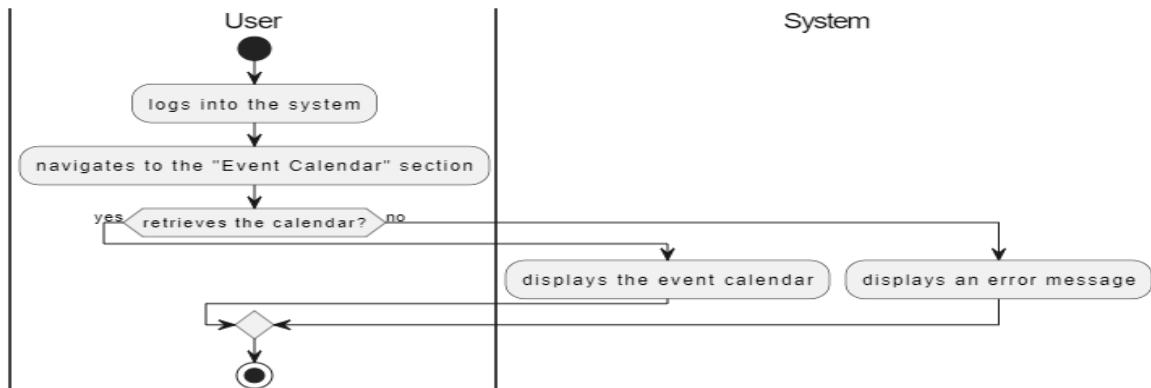


Figure 35: View Event Calendar

## 9.7 UC-07: Check Match Schedule

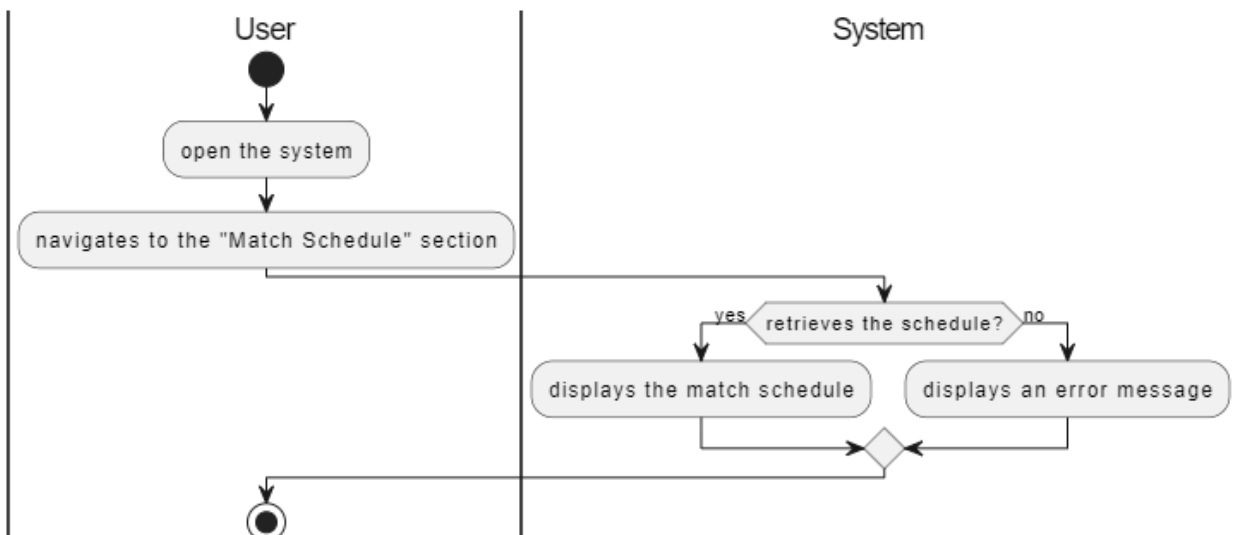


Figure 36: Check Match Schedule

## 9.8 UC-08: Check Live Scores

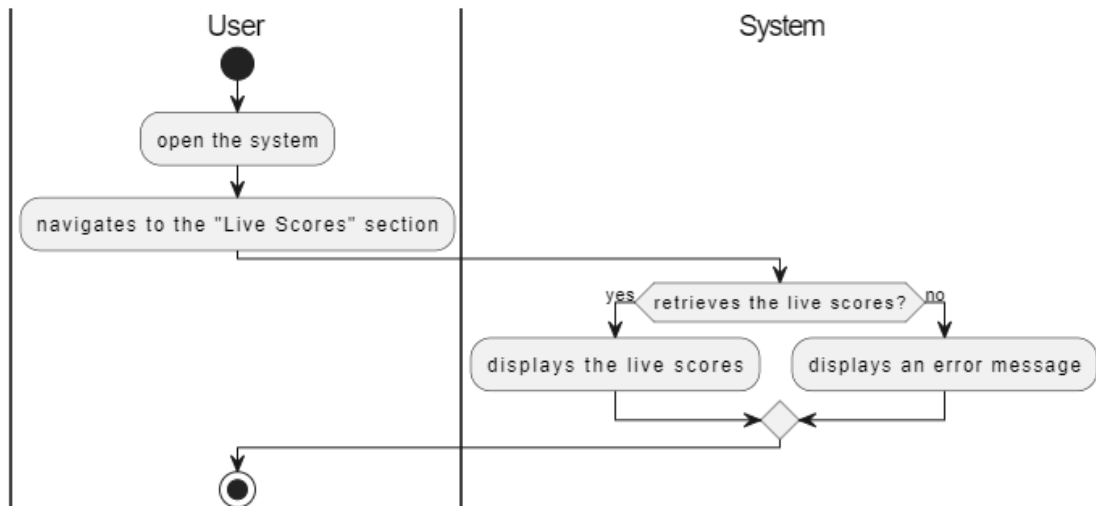


Figure 37: Check Live Scores

## 9.9 UC-09: View Results and Standings

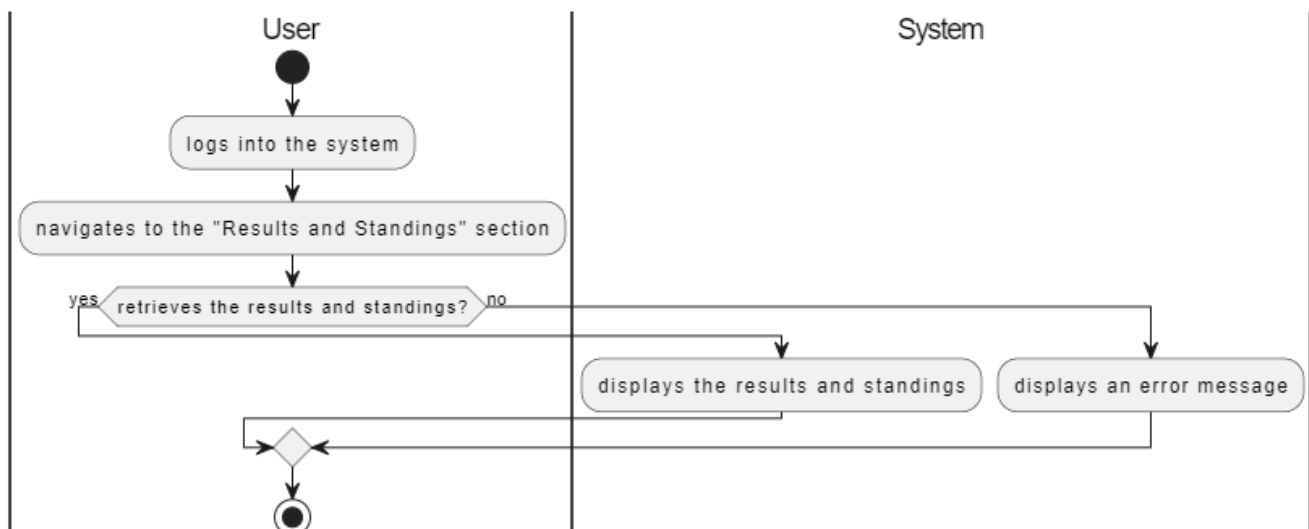


Figure 38: View Results and Standings

## 9.10 UC-10: Check Notices

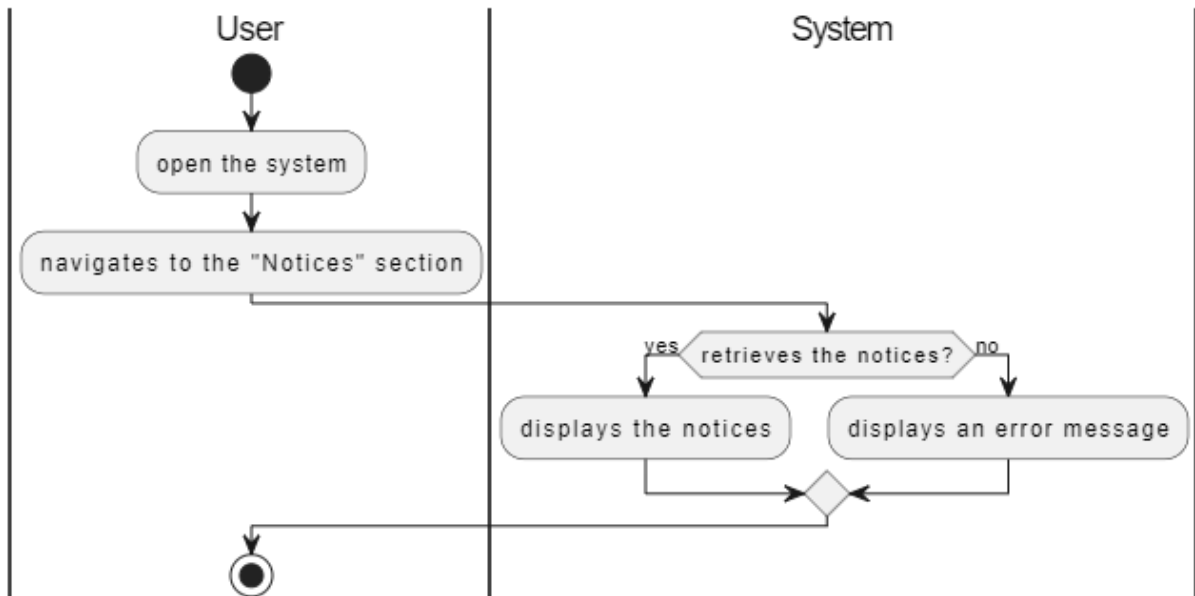


Figure 39: Check Notices

## 9.11 UC-11: View Previous Year Event Results

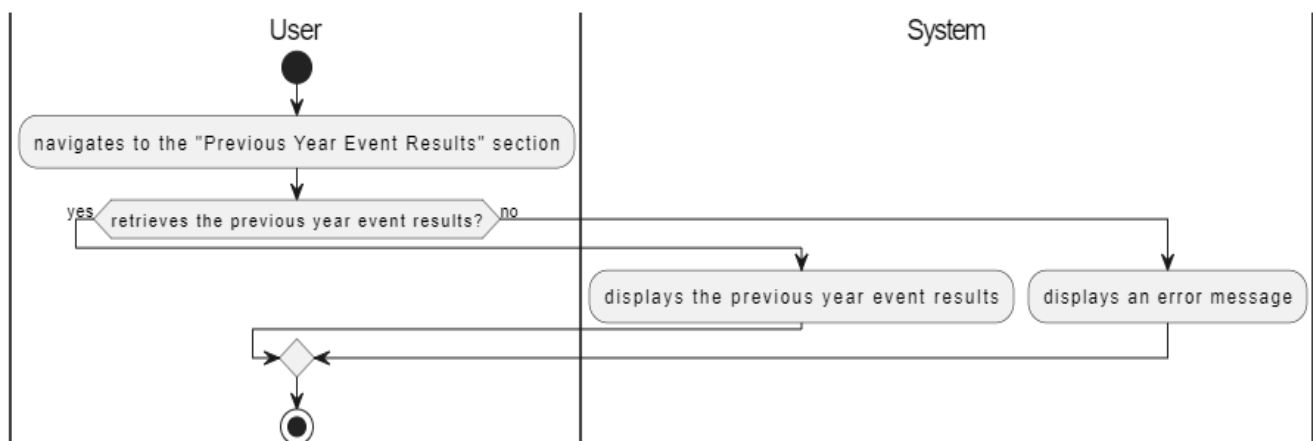


Figure 40: View Previous Year Event Results

## 9.12 UC-12: Share Event

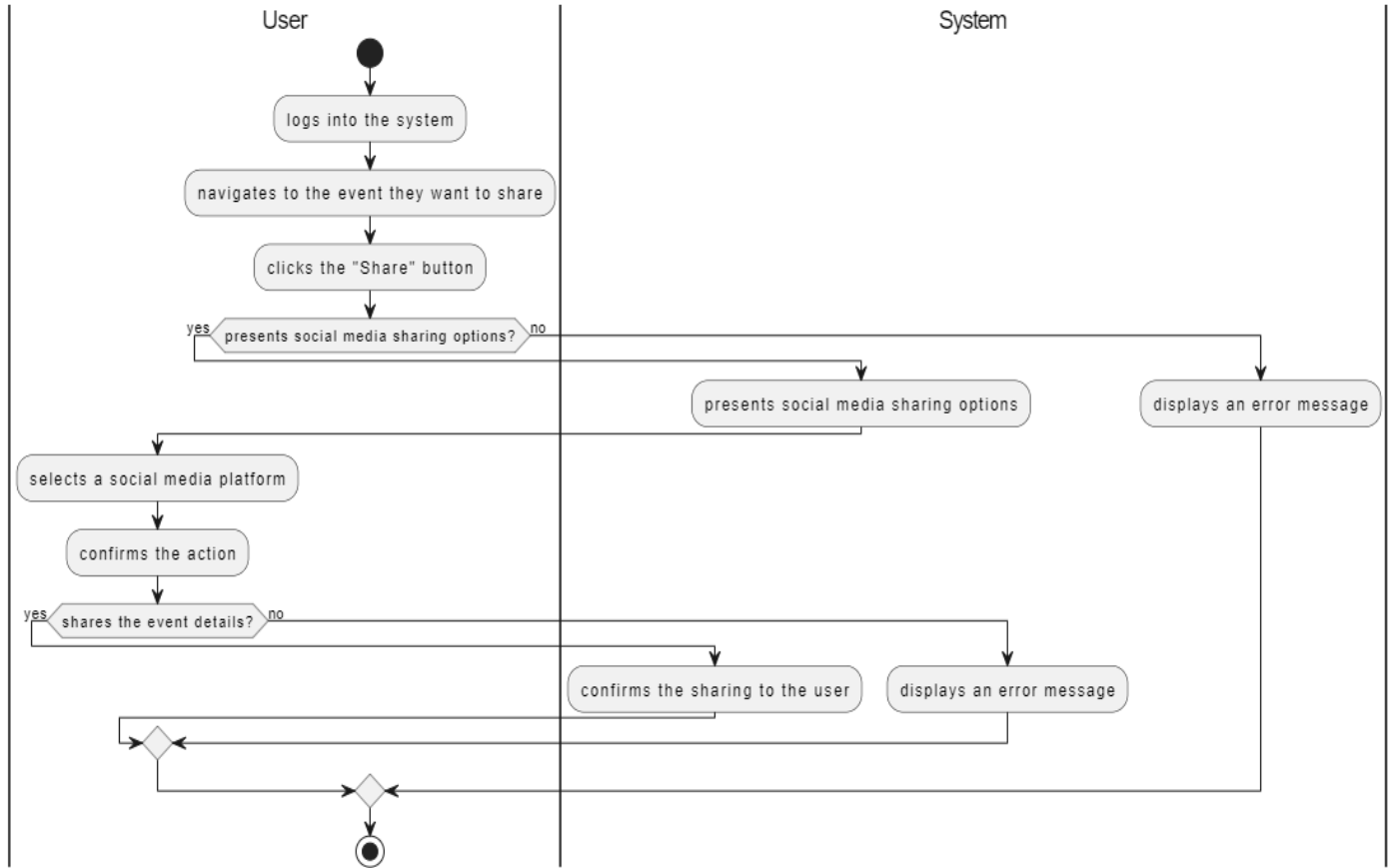


Figure 41: Share Event

### 9.13 UC-13: Manage Team Details

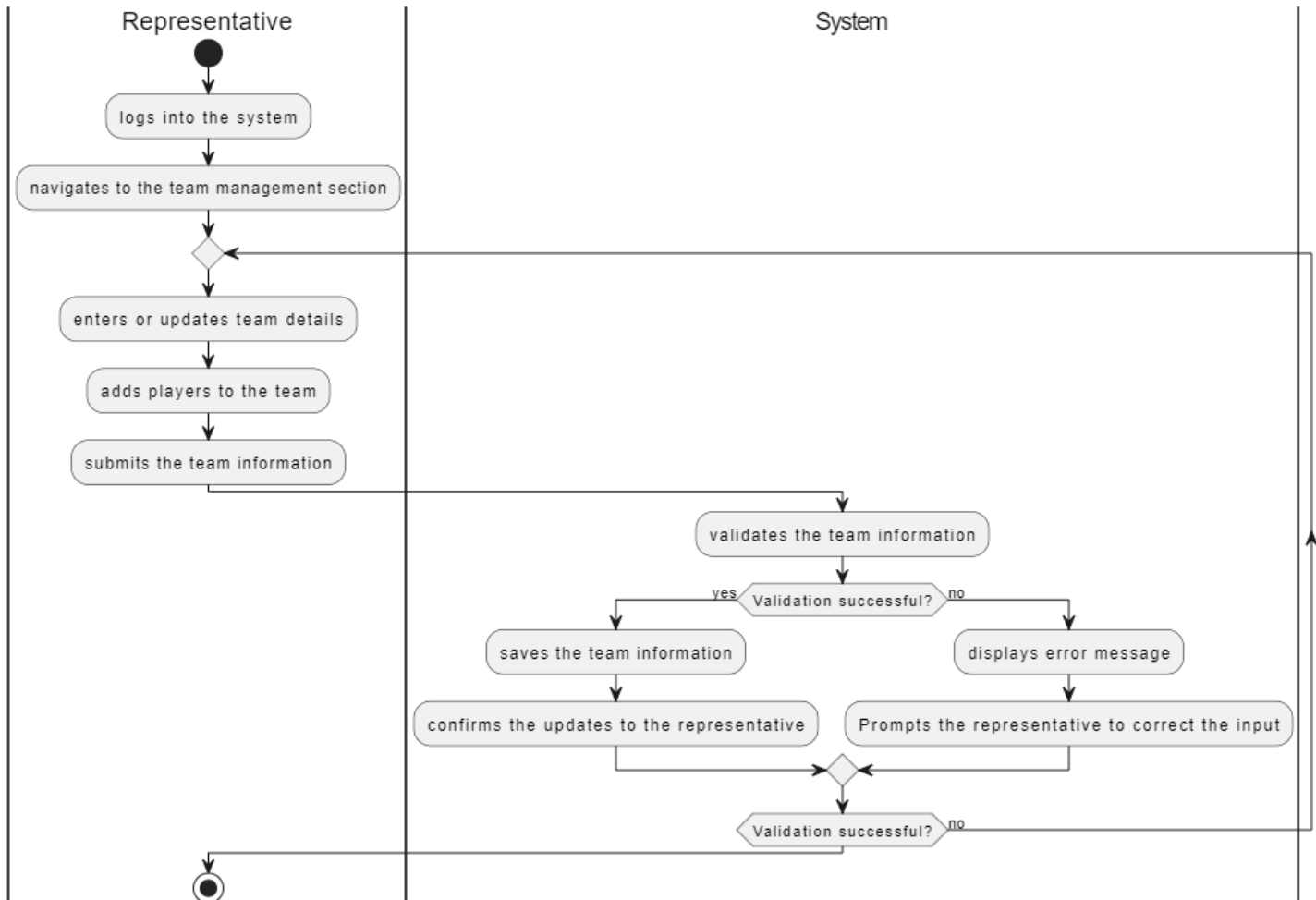


Figure 42: Manage Team Details

## 9.14 UC-14: Assign Department Representative

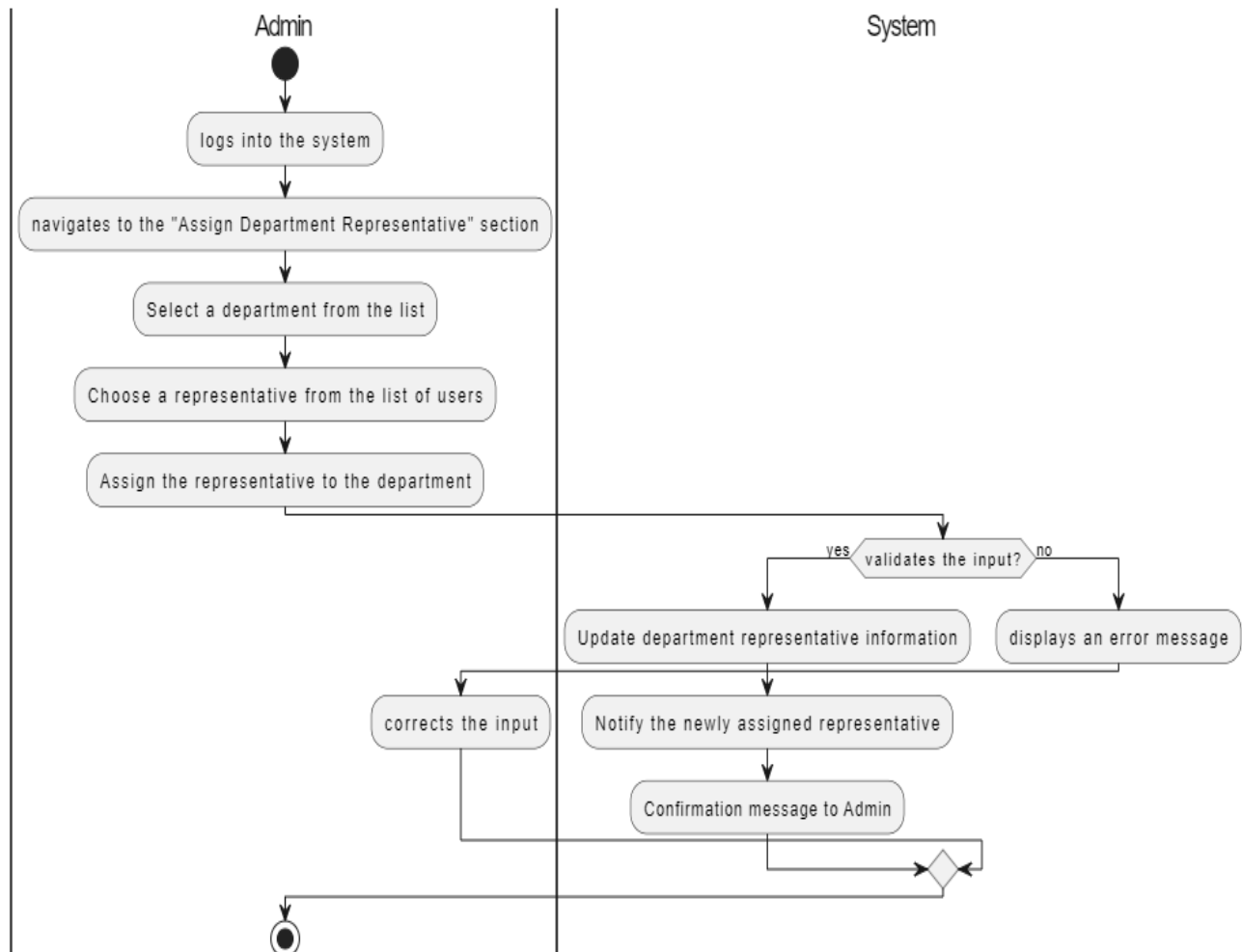


Figure 43: Assign Department Representatives