**Software Requirements Specification**

**for**

**Student Club Management System with Budget and Venue Integration**

**Version <1.0>**

**Table of contents**

Contents

[1 Introduction 4](#_Toc198489330)

[1.1 Purpose 4](#_Toc198489331)

[1.2 Scope 4](#_Toc198489332)

[1.3 Product Overview 5](#_Toc198489333)

[1.3.1 Product Perspective 5](#_Toc198489334)

[1.3.2 Product Function 6](#_Toc198489335)

[1.3.3 User Characteristics 6](#_Toc198489336)

[1.3.4 Limitations 7](#_Toc198489337)

[1.4 Definitions 7](#_Toc198489338)

[2 References 8](#_Toc198489339)

[3 Requirements 9](#_Toc198489340)

[3.1 Functions 9](#_Toc198489341)

[3.2 Performance Requirements 9](#_Toc198489342)

[3.3 Usability Requirements 9](#_Toc198489343)

[3.4 Interface Requirements 9](#_Toc198489344)

[3.5 Logical Database Requirements 9](#_Toc198489345)

[3.6 Design Constraints 9](#_Toc198489346)

[3.7 Software System Attributes 9](#_Toc198489347)

[3.8 Supporting Information 9](#_Toc198489348)

[4 Verification 10](#_Toc198489349)

[4.1 Verification Approach 10](#_Toc198489350)

[4.2 Verification Criteria 10](#_Toc198489351)

[5 Appendices 11](#_Toc198489352)

[5.1 Assumptions and Dependencies 11](#_Toc198489353)

[5.2 Acronyms and Abbreviations 11](#_Toc198489354)

[5.3 Glossary (Optional Section) 11](#_Toc198489355)

# Introduction

## Purpose

The purpose of the Student Club Management System with Budget and Venue Integration is to streamline and automate the management of student clubs within a university setting. This system aims to centralize key club activities such as event planning, budget tracking, and venue booking into a single, user-friendly platform.   
  
The system will serve as a bridge between student organizations, faculty coordinators, and administrative staff, ensuring smooth communication and accountability. Ultimately, it is designed to enhance organizational efficiency, promote student engagement, and support the growth of extracurricular activities on campus.

## Scope

The proposed software is a web-based Student Club Management System designed for use within a university environment. The goal of the system is to provide a centralized platform for managing student clubs, memberships, budgets, and venue reservations efficiently. The application is to be developed over a 3-month period by a team of 4 developers.

The system will support the following scope:

* The platform must support a minimum of 30 clubs.
* The system must be scalable to accommodate at least 10,000 students.
* The system will involve 3 internal user roles:
  + System Administrator – managing the overall system, users, and configurations.
  + Club Administrator – manages specific club activities, members, events, voting, and requests.
  + Student – can view/join clubs, view/join events and voting

Also, the system will integrate with two external systems:

* University Financial Management System – for tracking fund transparency, submitting budget, approving, and tracking club budgets.
* Campus Space Reservation Database – for checking availability and booking university venues for club activities.

Key features of the system will include:

* Role-Based Access Control with clearly defined permissions, UI access, and features based on user roles.
* Real-time integration with external systems to reflect the latest data for financials and venue availability.
* Responsive and accessible web interface for users with a stable internet connection.

## Product Overview

The Student Club Management System is a web-based application designed for management of student clubs within a university. The system is integrated with the university's University Financial Management System and Campus Space Reservation Database. Allow club admin and members to manage club’s detail, checking financial flow and booking of venue.

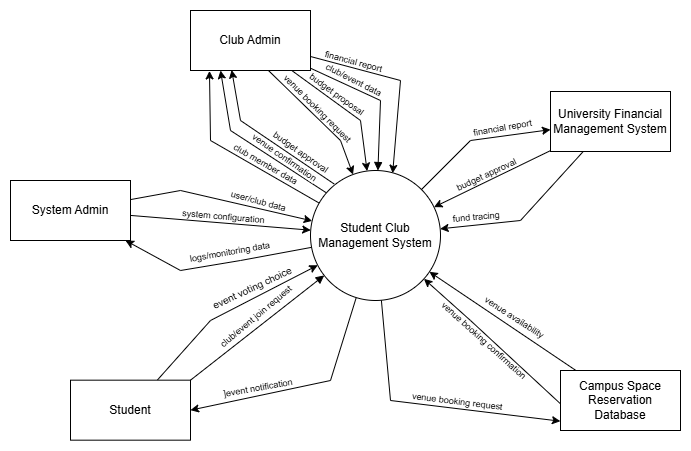
### Product Perspective

The Student Club Management System is a web-based application designed for the management of student clubs within the university. The platform is for handling club’s membership, event planning, budget tracking, and venue booking for a minimum of 30 clubs and 10,000 students of the traffic. The system interacting with external systems and users to streamline club operations while maintaining clear boundaries. It is focuses on club-related activities.

The system interfaces with two external systems which are University Financial Management System and the Campus Space Reservation Database. The Financial Management System enables clubs to submit budgets, receive approvals, and track funds, ensuring transparency and compliance with university financial policies. The Campus Space Reservation Database allows club administrators to view real-time venue availability and submit booking requests to university for club’s events, simplifying the event scheduling. These interactions assume a stable internet connection and API connection with the corresponding external system to provide real-time updates to users.

The system supports three internal actors which are System Admin, Club Admin, and Student. These actors are accessing the platform via a web interface with role-based access control to perform their respective features. System Admins can manage the platform’s configuration and user accounts, Club Admins manage club-specific operations like membership, budgets, events, and event voting, Students engage with clubs through membership, event voting and event participation. External actors include the Financial Management System and Campus Space Reservation Database, which connecting data with the system to support its functionalities.

A context diagram, provided below, illustrates the system’s interactions. The diagram depicts the Student Club Management System as the central entity, with bidirectional data flows to the Financial Management System (fund tracing, budget submissions and approvals), the Campus Space Reservation Database (venue availability, venue booking requests and confirmations), and the three user actors (login, data input, and output). This visual representation clarifies the system’s scope and external dependencies.

Figure 1.0 Context Diagram - Student Club Management with Budget & Venue Integration

### Product Function

The Student Club Management System provides the following primary functions:

* **Membership Management**: Allow adding, viewing, and removing members and managing membership status.
* **Event Management**: Allow creating, proposing, announcing, voting, and joining events.
* **Venue Booking**: Allow viewing available venue, booking venue for club’s events through university’s API.
* **Financial Management:** Allow tracing fund, requesting budget, allocate club budget and view financial report.
* **System Management**: Allow system admin to manage user accounts, monitor system data, create, edit, and remove clubs.

### User Characteristics

The Student Club Management System is designed for:

* **System Admin**: University staff with advanced technical skills, know about the system and computer state, responsible for system configuration and user management, monitor the system daily and provide help to users with troubles.
* **Club Admin**: Student or lecturer who has basic computer skills, manage club operations, manage members, manage events and manage event voting.
* **Student**: University student with minimal computer skills, joining clubs and events, using the system with stable internet access, interesting in explorer clubs.

### Limitations

The Student Club Management System has the following limitations:

* User must be university staff/student, also comply with university policies
* User must have minimum with a device that can access web browser (computer / smart phone)
* Some function like booking venue, financial tracking relies on University Financial Management System and Campus Space Reservation Database, which may impose integration constraints.
* Booking venue and budget request are limited by sequential process as its depend on the API of University Financial Management System and Campus Space Reservation Database.
* Basic logging on user login, perform action and external call to API.
* Lack of direct control on financial approval and venue booking confirmation because its depend on the external system.
* Constrained to web-compatible languages like JavaScript, HTML and CSS, limiting the backend complexity.
* Support around 30 clubs and 10000 students to use
* Vulnerable to external system and misuse of admin role.
* Limited to university staff to become an admin role to avoid leak to student data to other party
* Require user to have minimal physical effort and assume user can handle web navigation.
* Subject to real-time update delay from University Financial Management System and Campus Space Reservation Database, affecting incorrect data/outdate data
* Support only for English language UI
* Lack of automated backup systems, require manual backup
* Require online internet connectivity, no offline functionality provided

## Definitions

|  |  |
| --- | --- |
| Term | Definition |
| Student Club Management System | A web-based application to manage student’s club |
| Web-based Application | An application running on browser like google chrome, Firefox or Microsoft edge |
| Student | University students that use the program for joining clubs |
| Club Admin | University student/lecturer with basic computer skills to manage the club operations |
| System Admin | University staff with advance computer skill for monitor/configure the system, manage user account and manage clubs. |
| Member Status | The status of a student on specific club (active, pending, quit, suspended) |
| Role-Based Access Control | A security mechanism to restrict user access the unauthorized page/features. |
| Event Proposal | An event suggestion for club member to approve, joining and voting. |
| Venue Booking | The proses to reserve a venue from university campus for event purpose. |
| University Financial Management System | An external system which process the financial related request to university. |
| Campus Space Reservation Database | An external database providing university venue availability and booking service. |

# References

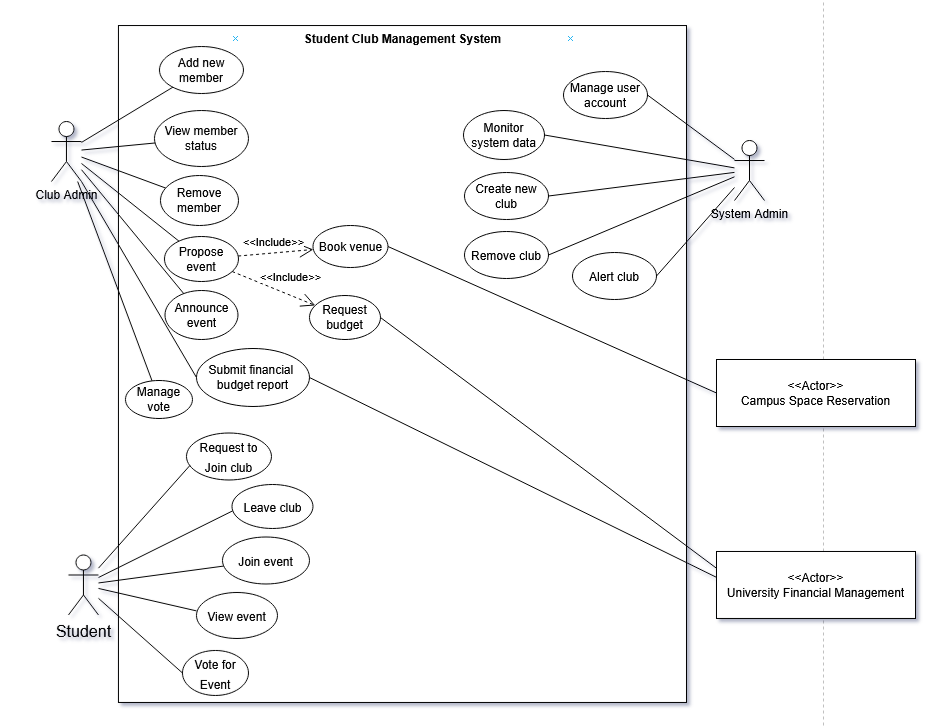
ISO/IEC/IEEE. (2018). Systems and software engineering—Life cycle processes—Requirements engineering. *ISO/IEC/IEEE 29148:2018*.- [ISO/IEC/IEEE 29148:2018 - Systems and software engineering — Life cycle processes — Requirements engineering](https://www.iso.org/standard/72089.html)

Requirements

# Requirements

### Functions

The **Use Case Diagram** for the Student Club Management System illustrates the interactions between **actors** and the **core functionalities** of the system. It provides a visual overview of how different roles interact with the system to achieve specific goals.

Figure 2.0 Use Case for Student Club Management with Budget & Venue Integration

1) Student

F001 Request to Join a club

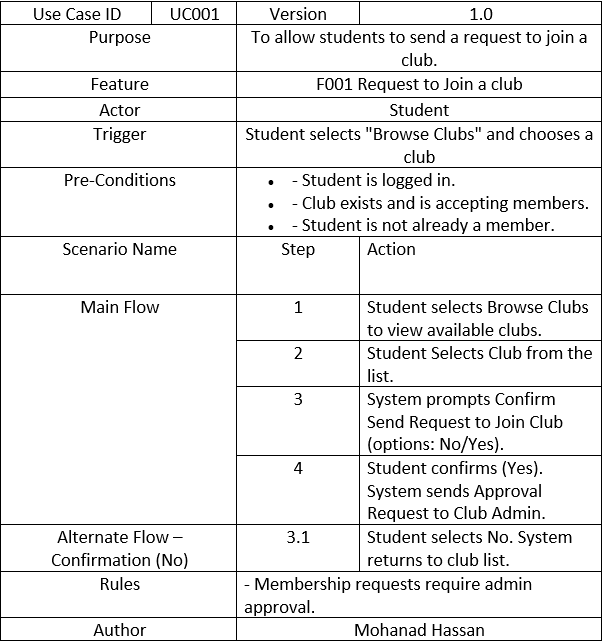


Table 1.0 Use Case Specification – Request to Join a Club

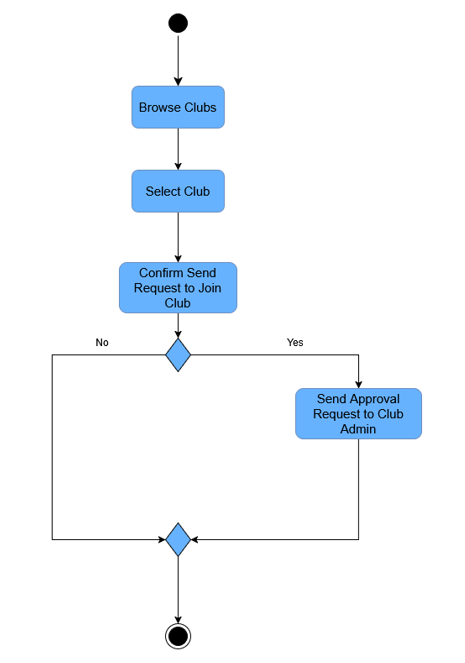


Figure 3.0 Activity Diagram – Request to Join a Club

F002 Leave Club

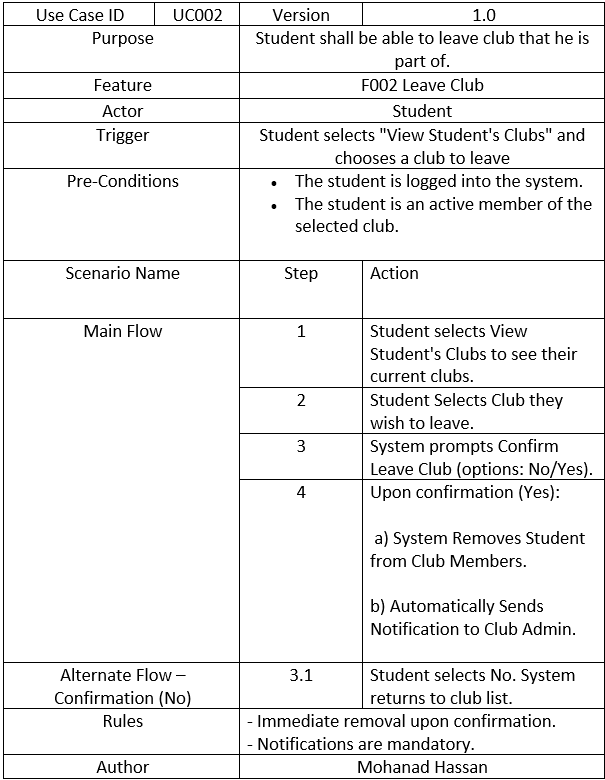


Table 2.0 Use Case Specification – Leave Club

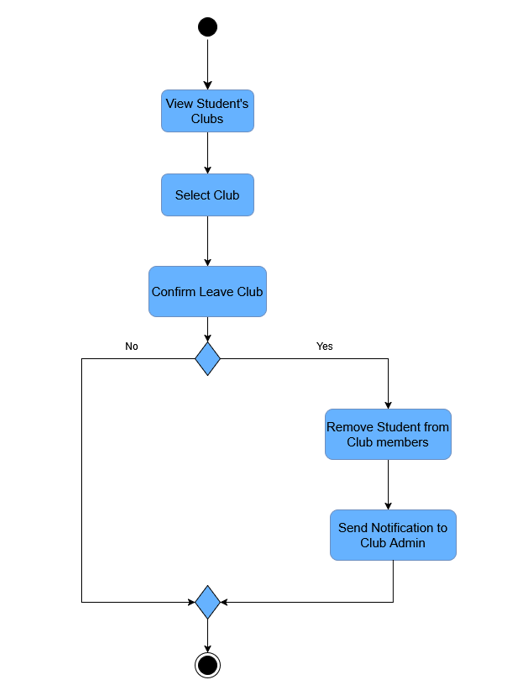
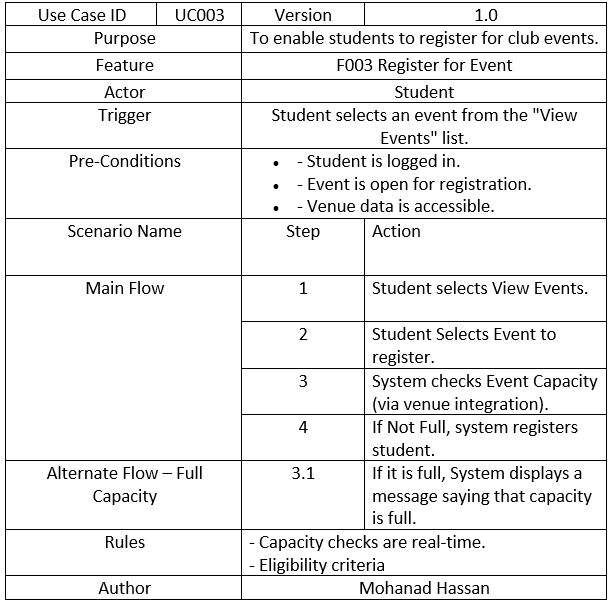


Figure 4.0 Activity Diagram – Leave Club

F003 Register for Event

Table 3.0 Use Case Specification - Register for Event

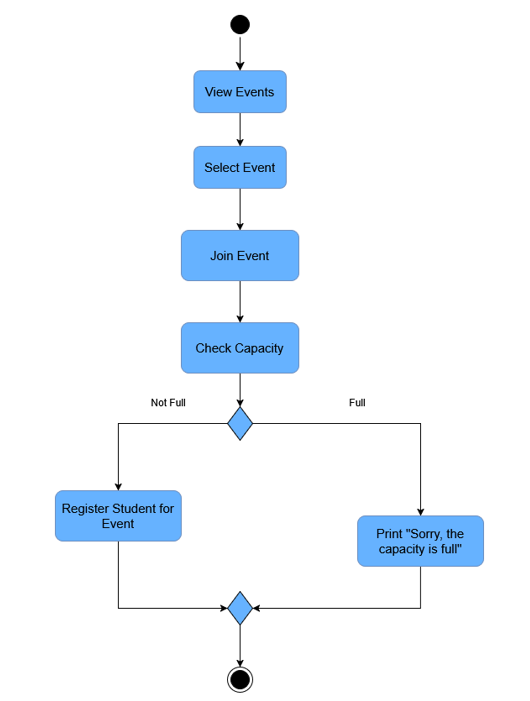
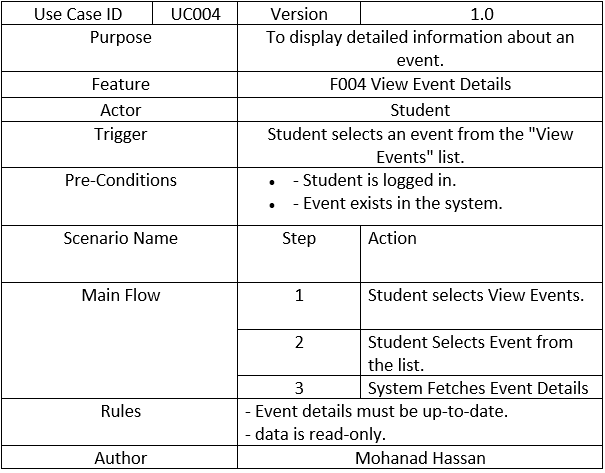


Figure 5.0 Activity Diagram – Register for Event

F004 View Event Details

Table 4.0 Use Case Specification- View Event Details

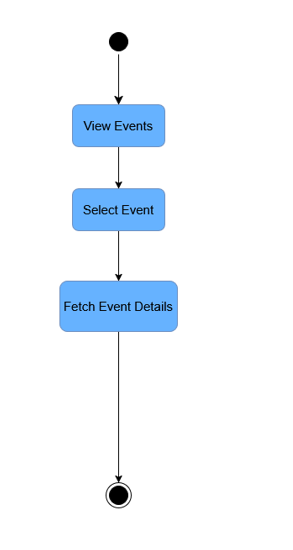
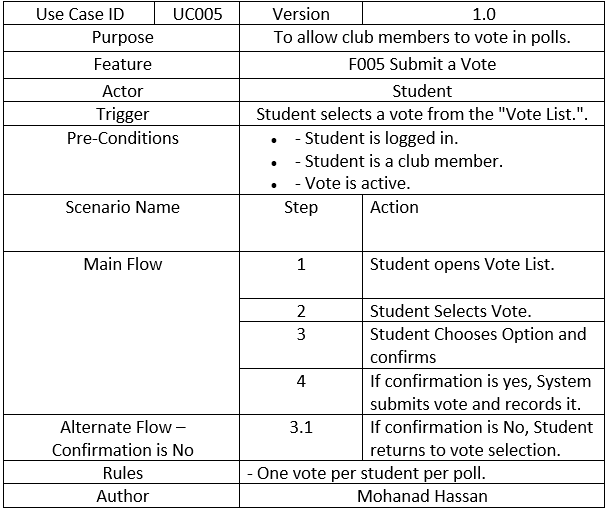


Figure 6.0 Activity Diagram – View Event Details

F005 Submit a Vote

Table 5.0 Use Case Specification – Submit a Vote

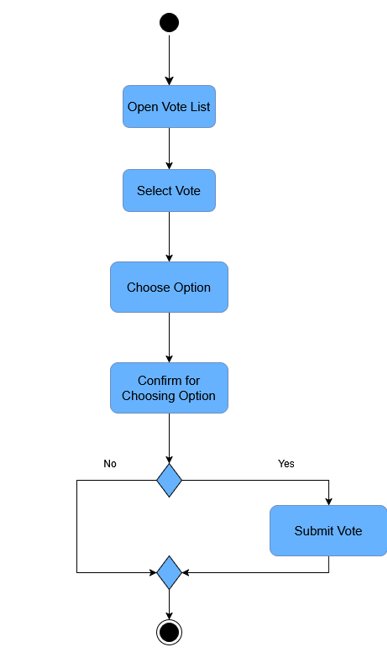


Figure 7.0 Activity Diagram - Submit a Vote

2) Club Admin

F006 Submit Financial Budget Report

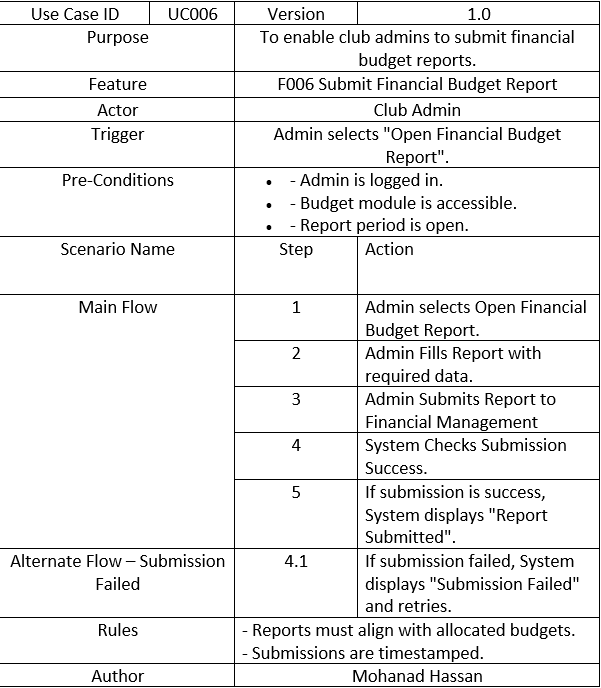
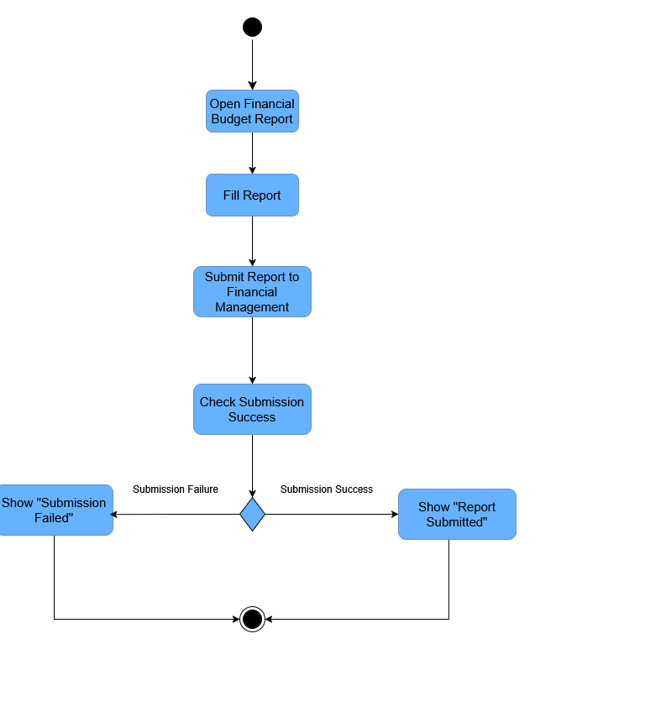


Table 6.0 Use Case Specification – Submit Financial Budget Report

Figure 8.0 Activity Diagram – Submit Financial Budget Report

F007 Propose Event

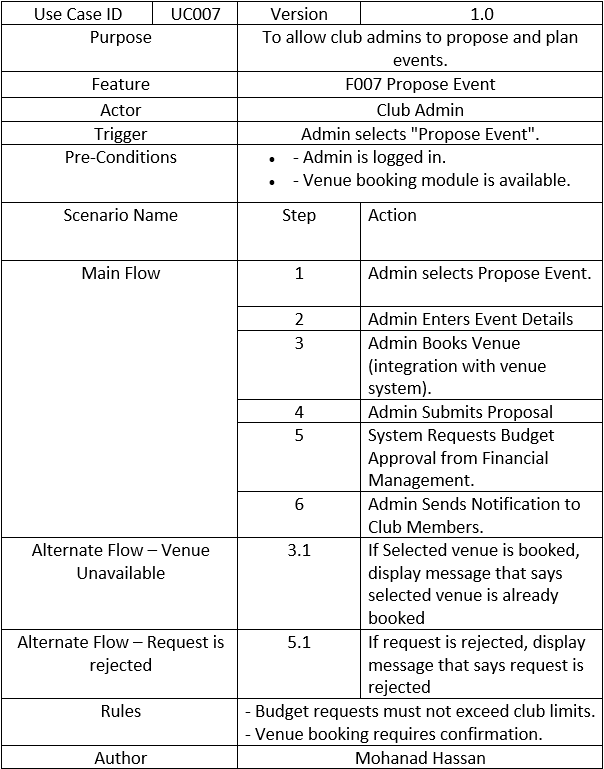


Table 7.0 Use Case Specification – Propose Event

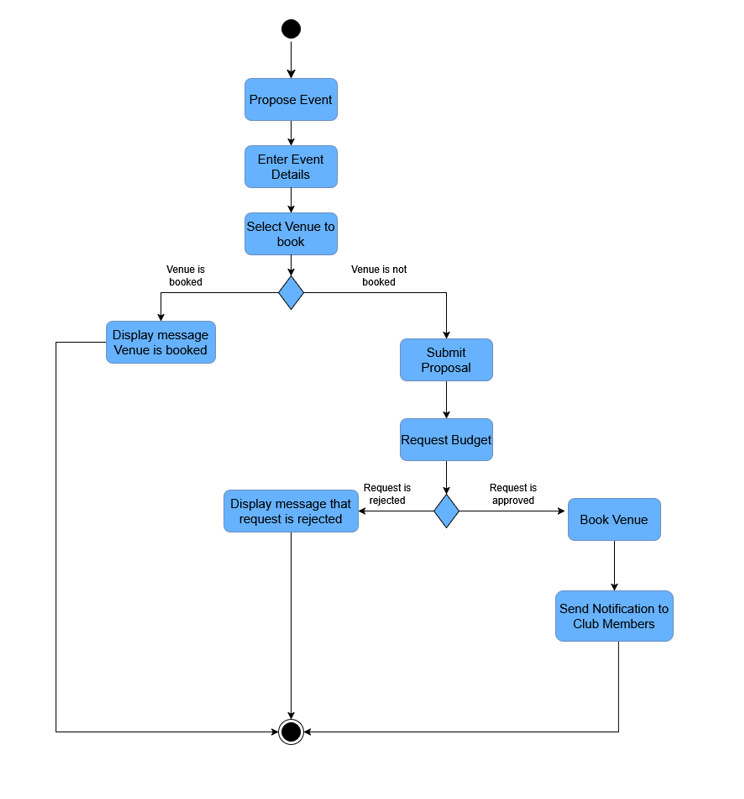
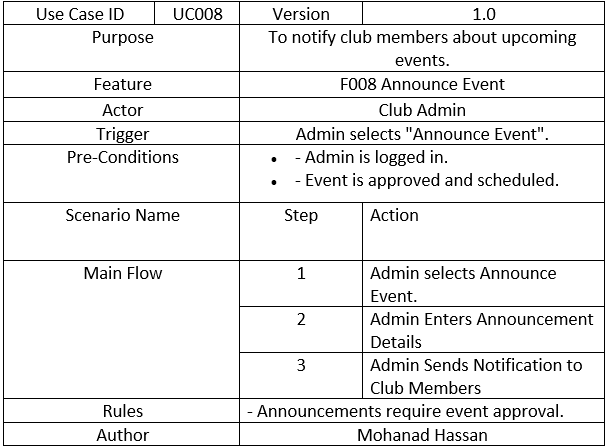


Figure 9.0 Activity Diagram – Propose Event

F008 Announce Event

Table 8.0 Use Case Specification – Announce Event

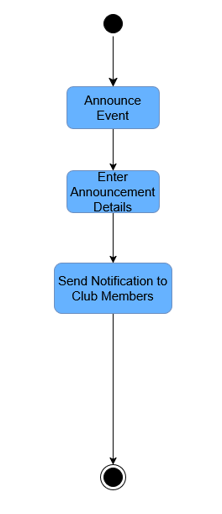
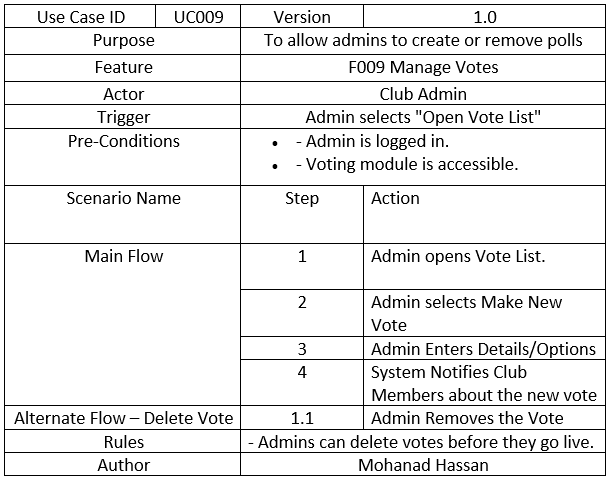
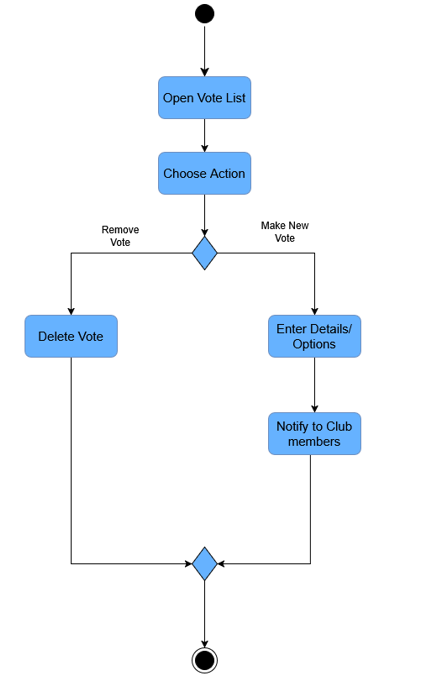


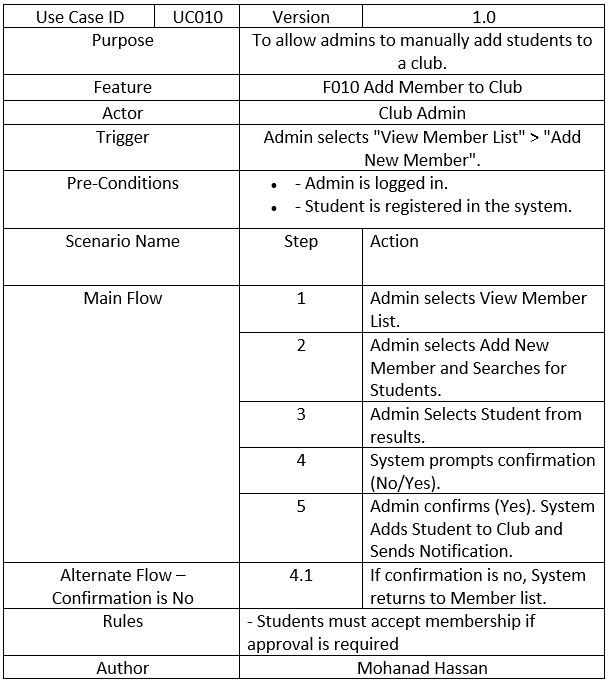
Figure 10.0

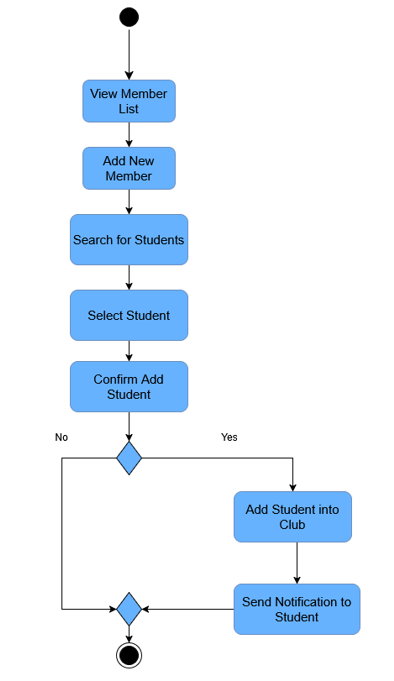
F009 Manage Votes



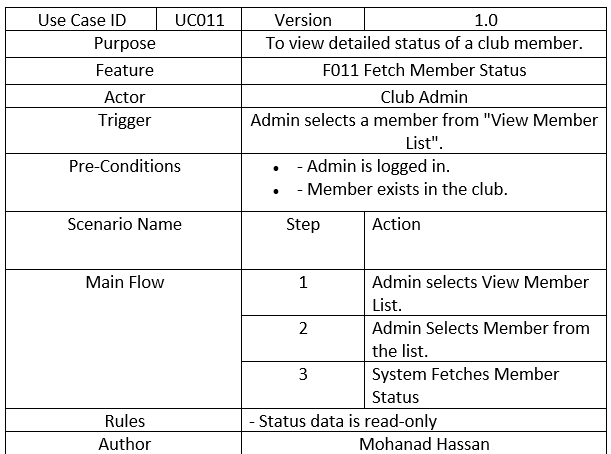


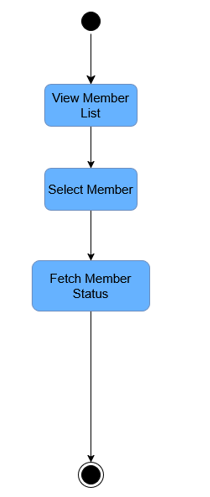
F010 Add Member to Club



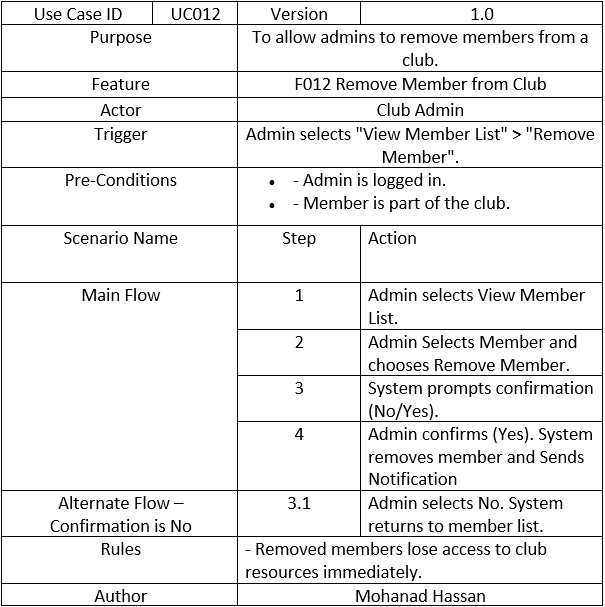


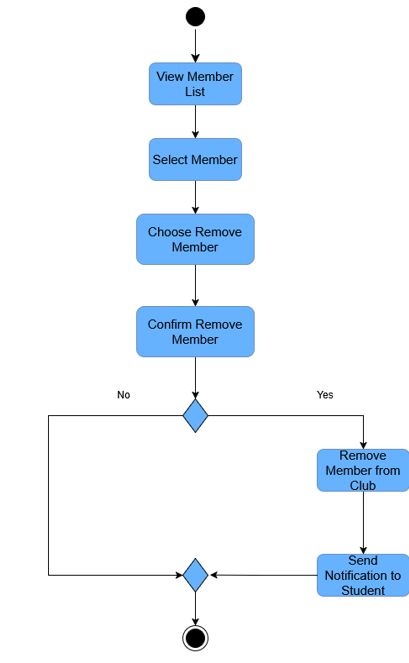
F011 Fetch Member Status





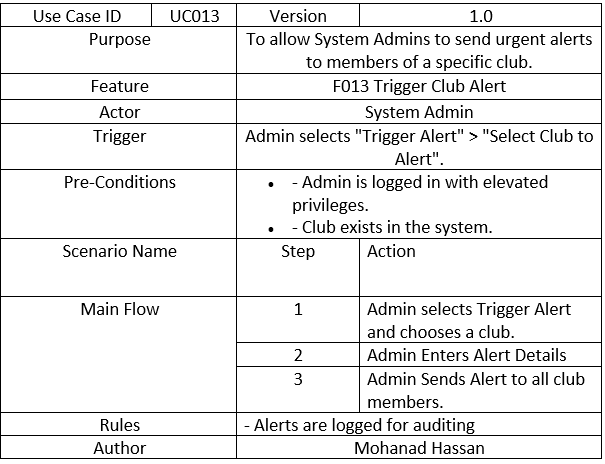
F012 Remove Member from Club

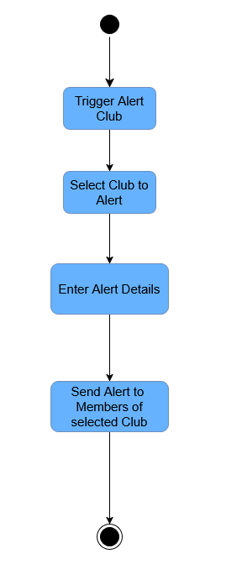




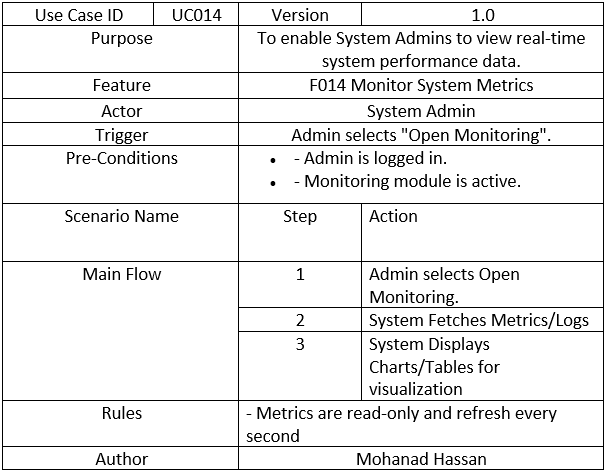
3) System Admin

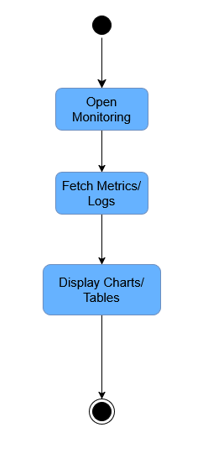
F013 Trigger Club Alert



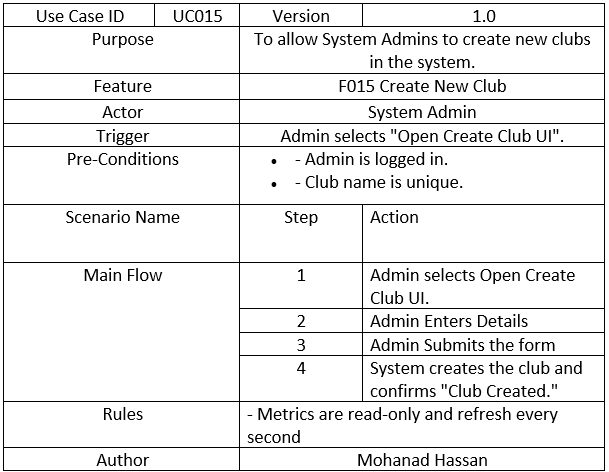


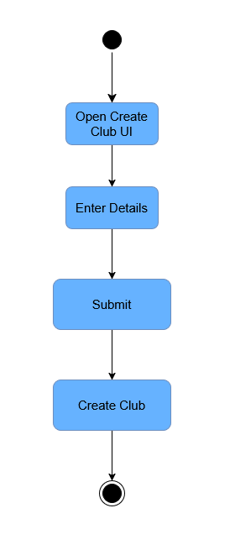
F014 Monitor System Metrics



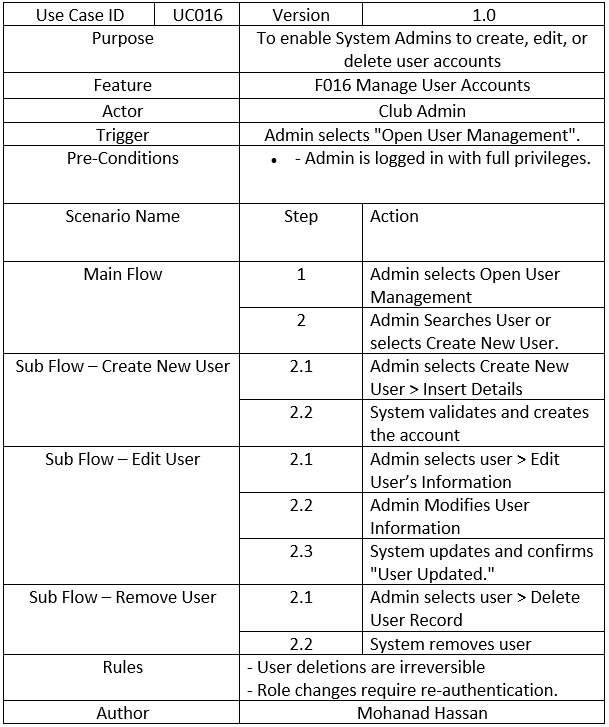


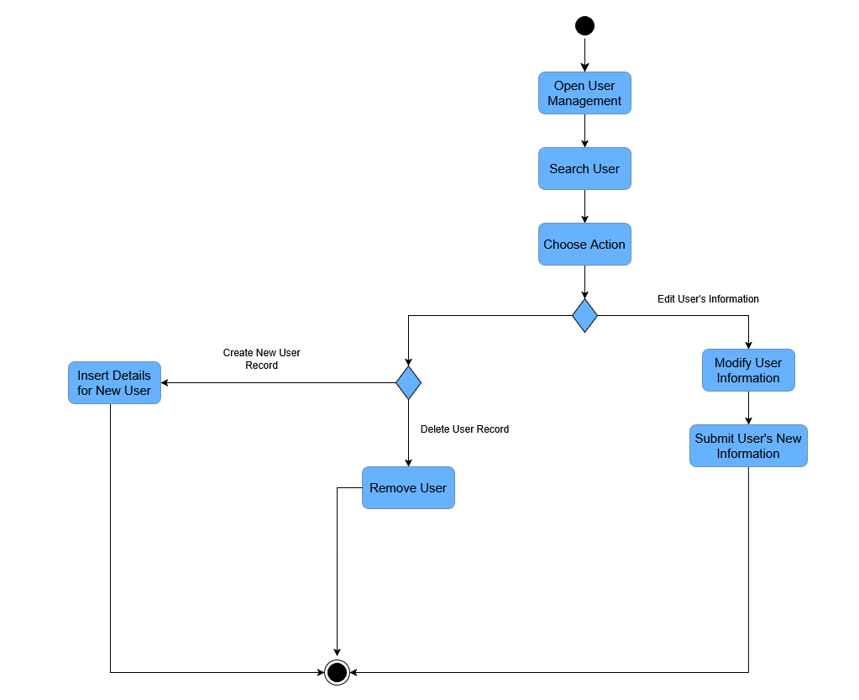
F015 Create New Club



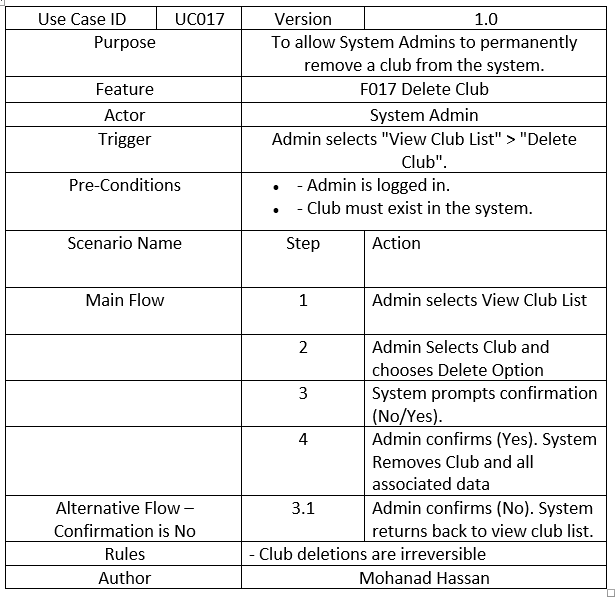


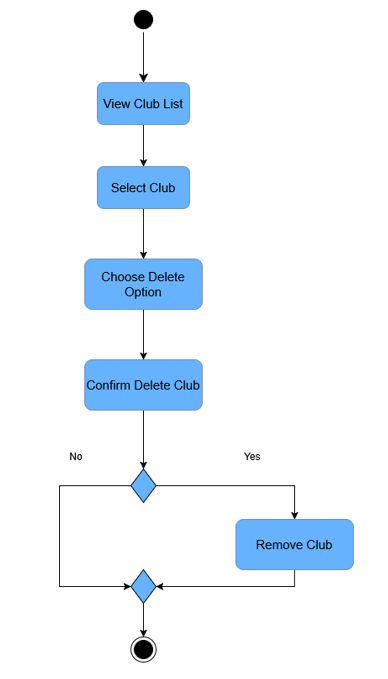
F016 Manage User Accounts





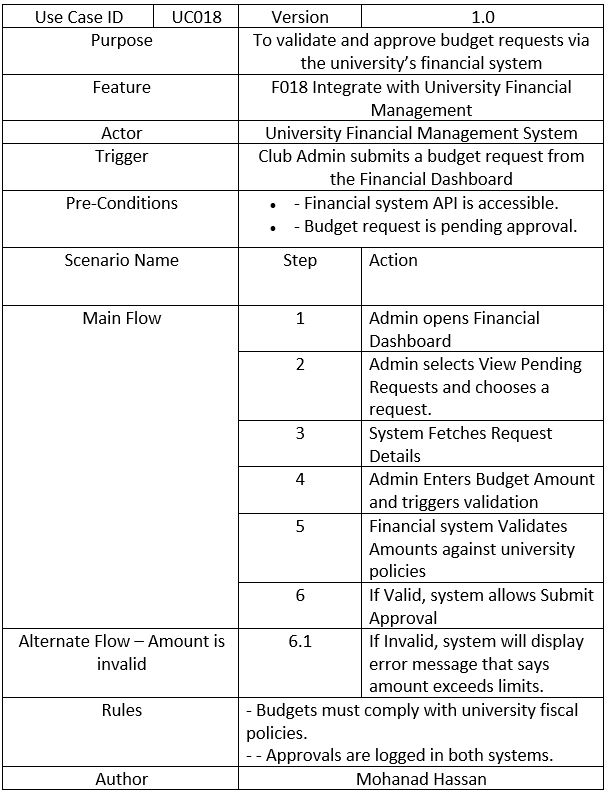
F017 Delete Club

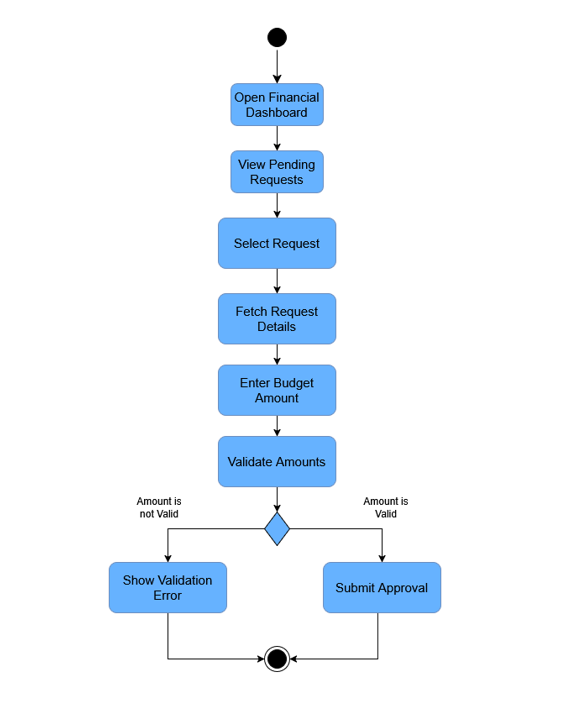




4) Financial Management System

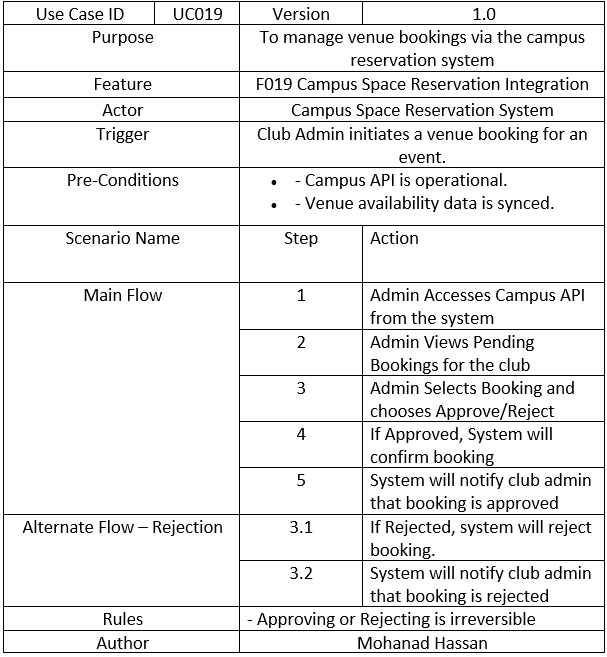
F018 Integrate with University Financial Management

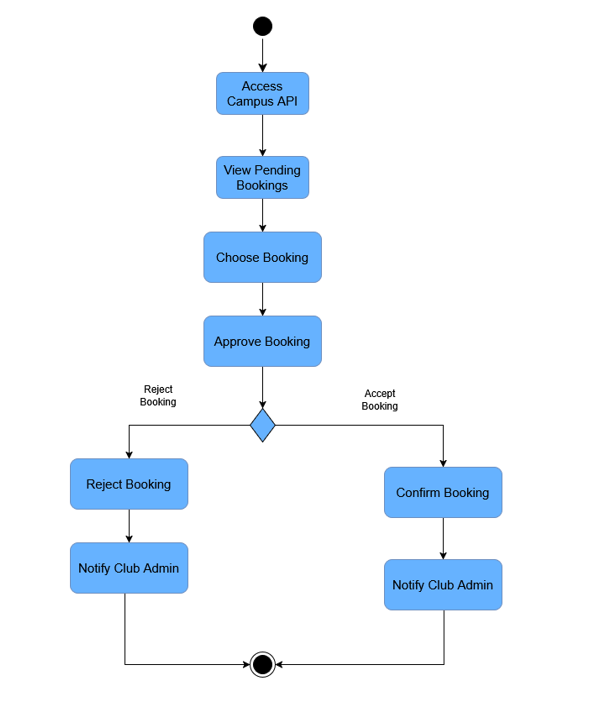




5) Campus Space Reservation

F019 Campus Space Reservation Integration





### Performance Requirements

The performance requirements for Student Club Management System are:

|  |  |  |
| --- | --- | --- |
| **Requirement ID** | **Description** | **Priority** |
| REQ\_P001 | The average respond time of system shall less than 2 seconds. | High |
| REQ\_P002 | The system shall support at least 500 users performing action at same time. | High |
| REQ\_P003 | System should be at least 90% of uptime every day. | High |
| REQ\_P004 | The average responds time of system on external interface (eg. query venue availability) shall less than 5 seconds. | High |
| REQ\_P005 | The average responds time of system on peak load scenarios (eg. deadline of voting) shall less than 10 seconds. | High |

### Usability Requirements

### Interface Requirements

#### System Interfaces

#### User Interfaces

#### Hardware Interfaces

#### Software Interfaces

#### Communications Interfaces

### Logical Database Requirements

The Student Club Management System shall utilize a relational database to logically organize and manage data related to users, clubs, events, votes, budgets, reports, and external systems. The database will support core operations and maintain relationships between the entities as represented in below and the class diagram below.

1. **User:**

* Attribute: userID, name, email, password
* Description: Base class for system admin, club admin and student.

1. **Student:**

* Attribute: userID, name, email, password (inherit from user)
* Description: Aggregation to Club but can exist independently

1. **SystemAdmin:**

* Attribute: userID, name, email, password (inherit from user)

1. **ClubAdmin:**

* Attribute: userID, name, email, password (inherit from user)
* Description: Aggregation to Club but can exist independently

1. **Club:**

* Attribute: clubID, name, description, members, events, votes
* Description: Composes multiple Event, aggregates multiple Budget, FinancialReport, ClubAdmin, Student

1. **Event:**

* Attribute: eventID, title, description, date, status, club, venue, participants
* Description: - Composition to Club. Composes multiple votes, aggregates multiple venue

1. **Venue:**

* Attribute: venueID, name, location, capacity, isAvailable
* Description: Aggregation to Event but can exist independently.

1. **Vote:**

* Attribute: voteID, voters, event, votesForYes, votesForNo
* Description: Composition to Event.

1. **Budget:**

* Attribute: budgetID, amount, purpose, status, requestDate
* Description: Aggregation to Club but can exist independently.

1. **FinancialReport:**

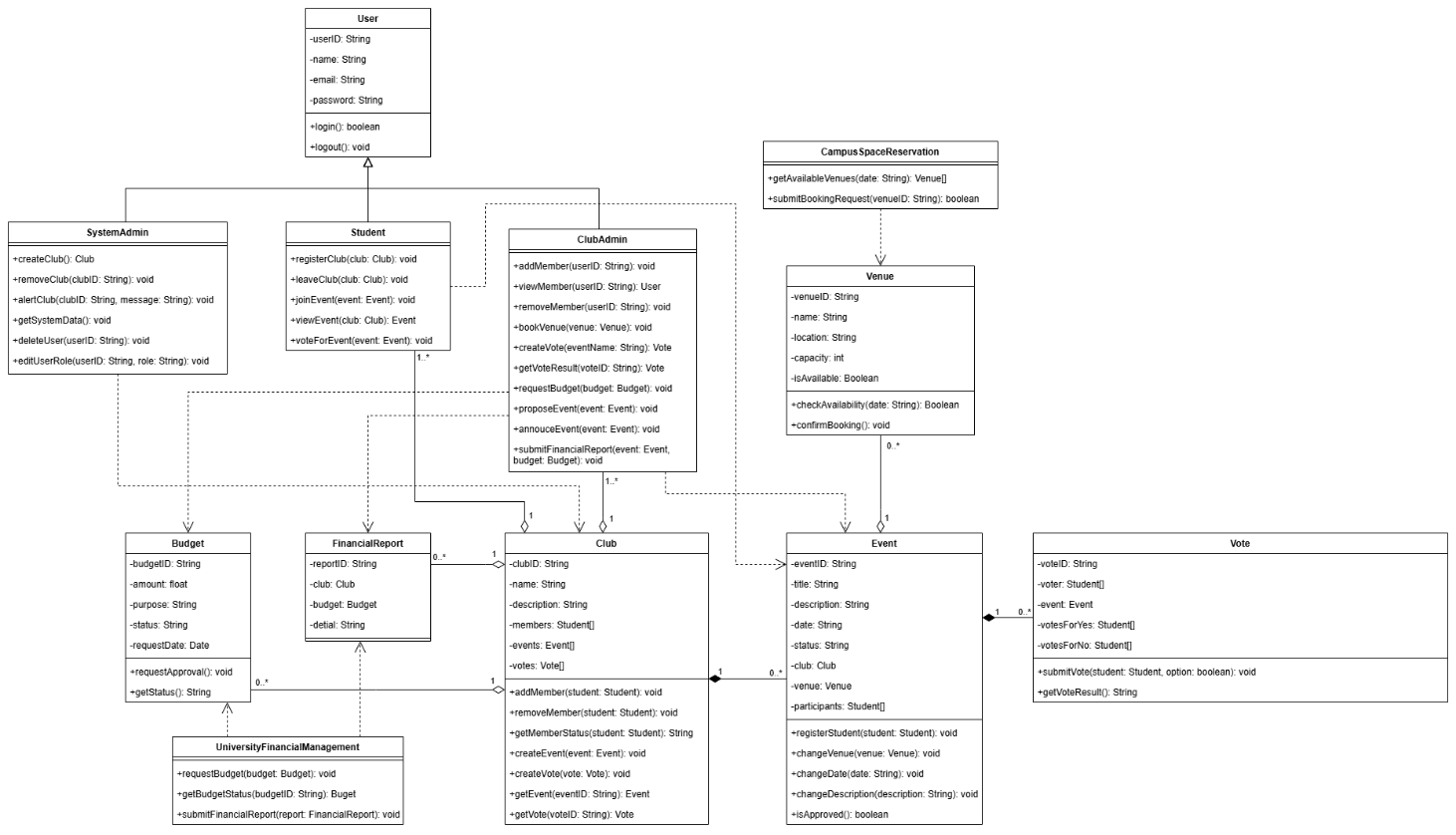
* Attribute: reportID, club, budget, detail
* Description: Aggregation to Club but can exist independently.

1. **UniversityFinancialManagement:**

* Description: External system for financial tracking and budget submission.

1. **CampusSpaceReservation:**

* Description: External system for venue availability checking and venue booking.



### Design Constraints

1. **University Branding Compliance**: The user interface design must follow the university’s branding guidelines, including color schemes and logo placement.
2. **Web-Based Interface**: The platform must be accessible via web browsers and must be implemented to support both desktop and mobile views.
3. **Scalability Limit**: The system must be designed to handle a minimum of 30 clubs and 10,000 students, with database and server architecture constrained to scale efficiently within these limits without requiring a complete re-development of the system.
4. **Data Privacy Regulation**: The system must comply with applicable data privacy laws, ensuring that student and club admin data (e.g., names, emails) is encrypted and access is restricted to authorized users only.
5. **Performance Constraint**: The system must maintain average response times under 2 seconds, limiting the design to lightweight frameworks and optimizing database queries to meet this requirement.
6. **Authentication Security**: The system must use secure authentication mechanisms (e.g., password hashing, session management) to protect user accounts.
7. **Integration with University System**: The system must integrate with existing university systems (e.g., UniversityFinancialManagement and CampusSpaceReservation), constraining the design to use compatible APIs.
8. **Language and Localization**: The system must support the main language(s) of university and be designed to allow future localization (e.g., for international students).
9. **Hosting Constraint:** The system must be hosted on the university’s network, and all data must store on servers located within the university’s network for security.

### Software System Attributes

**Reliability:**

* The system shall aim for 99.9% uptime**,** excluding planned maintenance and downtime of external integrated systems.
* Daily backups of the primary database shall be performed
* Backup success and integrity shall be monitored and logged.

**Availability:**

* The web server shall have a maximum downtime of 2 hours per 24-hour period.
* External APIs (Financial, Venue) are expected to have a maximum downtime of 2 hours per 24-hour period each.
* The system shall be accessible via modern browsers and responsive across desktop and mobile devices.

**Security:**

* The system shall support 2-Factor Authentication.
* User sessions shall expire after 30 minutes to prevent unauthorized access.
* The system shall implement **Role-Based Access Control (RBAC)** to ensure users can only access features and data appropriate to their role.
* API integrations shall use **API key authentication** with key rotation policies.
* All communication with the web server shall occur over **HTTPS** (TLS 1.2 or higher)
* Student data privacy shall comply with **GDPR/FERPA** and be continuously monitored by **Microsoft Defender for Cloud**

**Maintainability:**

* The system shall be developed using well-documented code and a modular design facilitated by the Django framework.
* Configuration settings should be manageable by System Admins where appropriate.

**Portability:**

* Being a web-based application, the client-side is portable across modern web browsers and operating systems (Windows, macOS, Android, iOS)
* Source code shall be maintained in **GitHub** with version control and GitHub Actions for CI/CD, enabling environment-independent builds.

### Supporting Information

The following supporting information is provided to assist with understanding, developing, and maintaining the Student Club Management System. Unless explicitly stated, these items are **not considered part of the formal system requirements**, but serve to supplement the specification:

* **User Manual:**

Detailed user manuals for each system role(Student, Club Admin, and System Admin) will be developed and distributed upon system deployment to support operational training and onboarding.

* **API Documentation:**

Technical API documentation for internal components will be maintained to facilitate system development, integration, and future maintenance.

* **Input/Output Sample:**

Examples of input forms (e.g., event proposal, budget request) and system-generated outputs (e.g., financial reports, event listings) are included to illustrate system interactions.

* **Problem Description:**

The system addresses the lack of a centralized digital platform for managing student clubs, which currently results in fragmented event coordination, manual budgeting processes, and limited visibility of club activities.

* **Security & Packaging instructions:**

Implement Two-Factor Authentication (2FA) for all administrative users, enhancing protection against unauthorized access. Source code and configuration files will be version-controlled and packaged via GitHub repositories, with access restricted through organization-based role permissions.

# Verification

### Verification Approach

### Verification Criteria

# Appendices

### Assumptions and Dependencies

**Assumptions:**

| **Dependency** | **Purpose** | **Risk if Unavailable** |
| --- | --- | --- |
| University Financial Management System | For real-time budget tracking, approval workflows, and fund disbursement. | Budget features will not function if the API fails. |
| Campus Space Reservation Database (CSRD) | For checking venue availability and submitting booking requests. | Venue booking functionality becomes unusable. |
| Microsoft Entra ID | Used for secure authentication and role-based access control (RBAC). | Unauthorized access or denial of access to legitimate users. |
| Azure Cloud Services | Includes hosting (App Service), storage (Azure SQL), and auto-scaling (Load Balancer). | Extended downtime in case of Azure service outages. |
| IBM DB2 | Production database for core entities like member lists, budgets, and bookings. | Risk of data loss, corruption, or system-wide disruption. |
| Development Timeline | The project assumes a 3-month completion timeline with no critical delays in API availability or testing. | Risk of feature cutbacks or reduced quality if deadlines slip. |
| Microsoft Outlook API | Sends booking confirmations and event notifications. | Users may miss critical updates or reminders. |

### Acronyms and Abbreviations

**SRS:** Software Requirements Specifications

**UI:** User Interface

**API:** Application Programming Interface

**CSRD:** Campus Space Reservation Database

**2FA:** Two-Factor Authentication

**RBAC:** Role-Based Access Control

**GDPR:** General Data Protection Regulation

**FERPA:** Family Educational Rights and Privacy Act

**SLA:** Service-Level Agreement

### Glossary