Club Management System

**Software Requirements Specification**

Version 3.0

Danang, February 2025

**TABLE OF CONTENTS**

**TABLE OF CONTENTS**. 2

[**1. DOCUMENT HISTORY**](#_6fjx8adf9a3p) **3**

[**2. REFERENCE DOCUMENTS**](#_jrnxi0z80tsy) **3**

[**3. INTRODUCTION**](#_nd9tt24jub99) **3**

[3.1 Purpose](#_g3gah9nlcrrh) 3

[3.2 In Scope](#_a6mf6uqioisp) 4

[**4. OVERVIEW**](#_mnrij95yz507) **4**

[4.1 Actors](#_q1mabmypqkth) 4

[4.2 System Use Case Diagram](#_jzpxcwe5s51f) 5

[4.3 System Class Diagram](#_7e6kktsngcvt) 6

[**5. USE CASE DESCRIPTION**](#_a6a2rrg56rss) **7**

[5.1. Sign Up](#_4xv11oq6hnxp) 7

[5.2. Sign In](#_3m9dnl5u1oma) 9

[5.3. Sign Out](#_3ua3uvgc2r4o) 10

[5.4. Edit User Profile](#_9l2sjc93c36h) 11

[5.5. Create Club](#_i7v8bzm0n5rq) 12

[5.6.Update Club Detail Information](#_i5f91usi31lh) 13

[5.7. Join Club Request](#_hu2h0h7dke3w) 15

[5.8. Assign Club Roles](#_ixf11wr3uz0y) 16

[5.9. Create Post](#_c6r8es2zys59) 18

[5.10 Update Post](#_mrndmi585kjd) 19

[5.11 Delete Post](#_vw1ah0dgjy9x) 20

[5.12 Interact Post](#_x7s4nvtlb19f) 21

[5.13 Approve Post](#_jvqfo4nf9h8) 22

[5.14 View Notification](#_8lcpmyqv9a9s) 24

[5.15 Approve/Reject Membership Request](#_63tq7y9nmzqs) 25

[5.16 Remove Members](#_knxoqwy8wv8l) 26

[5.17 Create Club Event](#_vqt7vhkglem5) 27

[5.18 Update Club Event](#_i73bxcapojwx) 28

[5.19 Delete Club Event](#_2jsebhbib6ub) 29

[5.20 Create Task](#_acmiu1dop0jn) 29

[5.21 Delete Task](#_jr7q0nwu273t) 30

[5.22 Update Task](#_y7vobd7rnbhi) 31

[5.23 View Task](#_6apb6bqppjqh) 32

[5.24 Assign Task for Club Member](#_5vgqoitrdd) 33

[5.25 Approve/Reject Create Club Requests](#_xuluzkl9ecwy) 34

[5.26 Delete Club](#_3q1jb2wmagha) 36

[**6. DATA STRUCTURE**](#_sdhvcmijd48o) **37**

[6.1 Database](#_xy2k9q1748rp) 37

# **1. DOCUMENT HISTORY**

| **Date** | **Summary of Changes** | **Version** |
| --- | --- | --- |
| 25/01/2025 | Analyse software requirements for Cub management system | CMS\_SRS\_sample\_v0.5 |
| 10/02/2025 | Design data structure for hotel management system | CMS\_SRS\_sample\_v1.0 |
| 30/03/2025 | Finalizing document | CMS\_SRS\_sample\_v2.0 |

# 

# **2.** **REFERENCE DOCUMENTS**

| **Document Name** | **Description** |
| --- | --- |
| [1] What is Class Diagram? - visual-paradigm | Reference document |
| [2] Slide bài giảng SWD392 | Reference document |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# **3.** **INTRODUCTION**

The following subsections of the Software Requirements Specifications (SRS) document provide an overview of the entire SRS.

## **3.1** **Purpose**

- The Software Requirements Specification (SRS) defines the requirements for the Club Management System (CMS). It ensures the system automates manual tasks, provides accurate and efficient operations, and supports the storage and management of club-related information.

- This SRS will help developers build the system effectively and ensure it meets stakeholder expectations. Feedback will be incorporated to deliver a system tailored to the needs of school clubs.

## **3.2** **In Scope**

- The CMS will serve as a specialized social media platform for school clubs, enabling features like club creation, role and assignment management, and member applications.

- The system will support efficient management, generate detailed reports, ensure data accuracy, and facilitate secure role-based access for staff and managers.

# **4.** **OVERVIEW**

## **4.1** **Actors**

| **Actor** | **Description** |
| --- | --- |
| Guest | A Guest is a person who accesses the CMS without logging in. They can view the public information about clubs, events, and activities but cannot interact with or join clubs. |
| Member | A Member is a person with an account in the CMS. They can view club details, participate in events, submit assignments, and interact with other members. |
| Moderator | A Moderator is responsible for maintaining a positive and organized environment within a club. |
| Club Admin | A Club Admin is responsible for managing a specific club. They can create and manage events, assign roles, approve/reject member applications, and oversee club activities. |
| System Admin | The System Administrator maintains the CMS platform. They handle technical issues, system updates, security, user management, and overall platform maintenance |
| User | A Registered User is a person with an account in the CMS. They can view the public information about clubs, events, activities, and join clubs. |

## 

## 

## 

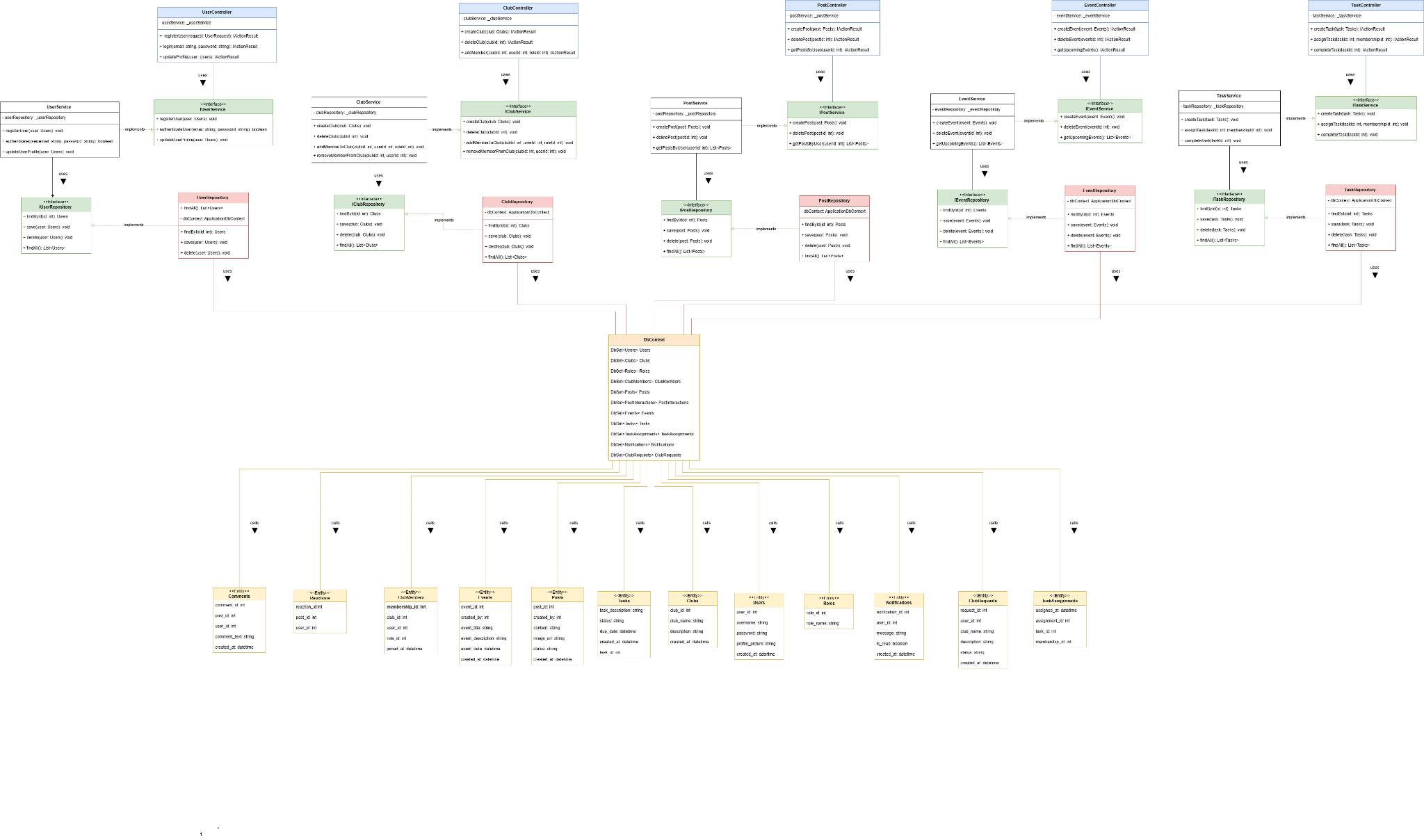
## 

## 

## **4.2** **System Use Case Diagram**

## 

## **4.3** **System Class Diagram**

******

# **5.** **USE CASE DESCRIPTION**

## **5.1. Sign Up**

| Use Case ID: | **UC-01** | USE CASE NAME | **SIGN-UP** |
| --- | --- | --- | --- |
| Create By: | MinhDQ | Date Created: |  |
| Primary Actor: | Guest | Secondary Actor | None |
| Description | This use case describes the process of a new visitor signing up for an account, providing their personal details such as name, email, and password to become a registered user. | | |
| Priority: | High | | |
| Trigger | The guest clicks on the "Sign Up" button on the login page or accesses the registration page via a link. | | |
| Preconditions: | PRE-01**:** The guest has a valid email address.  PRE-02**:** The guest provides a valid password. | | |
| Post-Conditions | POST-01**:** The system works stably without errors.  POST-02**:** A new user account is created, and the guest receives a unique identifier upon successful verification.  POST-03**:** The guest can log in using their newly created account | | |
| Basic Flow: | 1. The guest accesses the home page. 2. Clicks on the avatar icon, then selects the “Sign In” button. 3. The system redirects to the sign-in page. 4. Clicks on “Sign Up.” 5. Fills in all required fields for account creation. 6. Clicks the “Create Account” button. 7. Verifies email with the code sent to their registered email. 8. The system redirects to the sign-in page. | | |
| Alternative Flow: | None | | |
| Exception: | **1a.** The guest enters an invalid username or email.   * **1a1.** The system displays an error message and prompts the guest to enter a valid username, email, or password.   **1b.** The guest enters a username or email that already exists.   * **1b1.** The system displays an error message and prompts the guest to enter a different username or email. | | |

## **5.2. Sign In**

| Use Case ID: | **UC-02** | USE CASE NAME | **SIGN-IN** |
| --- | --- | --- | --- |
| Create By: | AnNNH | Date Created: |  |
| Primary Actor: | Guest | Secondary Actor | None |
| Description | This use case describes how a user can sign in to their account. The system verifies the credentials and grants access to the user's account. | | |
| Priority: | High | | |
| Trigger | User clicks on the "Sign In" Button in Home Page. | | |
| Preconditions: | PRE-01: User has a created and valid account  PRE-02: User account has been authorised. | | |
| Post-Conditions | POST-01: The system works stably without errors.  POST-02:User successfully logged into the website. | | |
| Basic Flow: | 1. User access to the sign-in page.  2. User to choose login with their account.  3. User type in account info and press sign in button.  4. System verifies the user's account information and allows the user  to access the system.  5. System redirects to Home page. | | |
| Alternative Flow: | 2a. User to login with Gmail method.  2a1. System redirects to the Google sign in page.  3a. Users type in their Google account and click log in.  4a. Google verifies the account’s information and allows the user access to the system. | | |
| Exception: | 4c. The system fails to verify the user's account information and display messages.  4c1. Users stop the login process.  Use case stops.  4c2. Users click on Forgot password | | |

## **5.3. Sign Out**

| Use Case ID: | **UC-03** | USE CASE NAME | **SIGN-OUT** |
| --- | --- | --- | --- |
| Create By: | AnNNH | Date Created: |  |
| Primary Actor: | Member,Club admin,System admin | Secondary Actor | None |
| Description | This use case describes how a user can sign out of their account, which clears any active cookies and logs them out of the system | | |
| Priority: | High | | |
| Trigger | User clicks on the "Logout" Button in the sidebar. | | |
| Preconditions: | PRE-01: User has logged in into website | | |
| Post-Conditions | POST-01: The system works stably without errors.  POST-2: System navigates the users back to the home page. | | |
| Basic Flow: | 1. User click on “mark 3 line” to expand the sidebar .  2. Click on the log out option.  3. The user is now no longer determined as any role but Guest.  4. The guest will be navigated back to the home page. | | |
| Alternative Flow: | None | | |
| Exception: | None | | |

## **5.4. Edit User Profile**

| Use Case ID: | **UC-04** | USE CASE NAME | **EDIT-PROFILE** |
| --- | --- | --- | --- |
| Create By: | MinhDQ | Date Created: |  |
| Primary Actor: | Registered User | Secondary Actor | None |
| Description | This use case describes how a registered user can view and update their personal details. | | |
| Priority: | High | | |
| Trigger | The user clicks on the profile section. | | |
| Preconditions: | PRE-01: The user must be logged into the system. | | |
| Post-Conditions | POST-01: The system updates and saves the user’s profile changes.  POST-02: The user sees the updated profile information. | | |
| Basic Flow: | 1. The user logs into the system. 2. Clicks on the profile section. 3. Edits profile details (e.g., name, email, profile picture). 4. Clicks the "Save" button. 5. The system updates and displays the changes. | | |
| Alternative Flow: | None | | |
| Exception: | **4a.** The user enters invalid information.   * **4a1.** The system displays an error message and prompts the user to correct the information. | | |

## **5.5. Create Club**

| Use Case ID: | **UC-05** | USE CASE NAME | **Create Club** |
| --- | --- | --- | --- |
| Create By: | AnNNH | Date Created: |  |
| Primary Actor: | Register User | Secondary Actor | None |
| Description | This use case Describes Register create request create a club | | |
| Priority: | High | | |
| Trigger | User clicks on the "Create Club" Button in the Home page. | | |
| Preconditions: | PRE-01: The Registered User is logged into the system. | | |
| Post-Conditions | POST-01: The system works stably without errors.  POST-02: Create request sent successfully and awaiting approval from the System Admin.  POST-03:Registered User receives notification of request status. | | |
| Basic Flow: | 1. The Registered User accesses the club creation page.  2. The Registered User enters the required details, including club name, description, location, and contact information.  3. The Registered User submits the form.  4.The system records the request and displays a success message.  5. The System Admin receives the request and can approve or reject it.  6. The Registered User receives a notification when the request is processed. | | |
| Alternative Flow: | **1.The Registered User cancels the creation process:**The Registered User exits the club creation page without submitting.No data is stored, and the process is aborted.  **2.The Registered User enters incomplete or invalid information:**The system detects missing or incorrect data and displays appropriate error messages.The Registered User must correct the information before resubmitting. | | |
| Exception: | 1.System error or connectivity issues:The system displays an error message and requests the Registered User to try again later.  2.The Registered User lacks the required permissions:The system denies access and displays an authorization error message. | | |

## 

## **5.6.Update Club Detail Information**

| Use Case ID: | **UC-03** | USE CASE NAME | **Update Club Detail Information** |
| --- | --- | --- | --- |
| Create By: | AnNNH | Date Created: |  |
| Primary Actor: | Club admin | Secondary Actor | None |
| Description | This use case allows a Club admin to update the detailed information of a club, including name, description, location, contact details, and other relevant information. | | |
| Priority: | High | | |
| Trigger | Club admin clicks on the "Update Club" Button in the Club management dashboard. | | |
| Preconditions: | PRE-01: Club admin has logged in into website | | |
| Post-Conditions | POST-01: The system works stably without errors.  POST-02: The updated club information is successfully saved in the system.  POST-03:The system reflects the new details for users viewing the club profile. | | |
| Basic Flow: | 1. The Club Admin accesses the club management dashboard.  2. The system displays the current details of the club.  3. The Club Admin modifies the necessary fields.  4.The Club Admin submits the updated information.  5.The system validates the input and saves the changes.  6.The system confirms the update with a success message. | | |
| Alternative Flow: | **The Club Admin decides not to save changes:**   * The Club Admin exits the editing page without submitting updates. * The system retains the previous information without modification.   **The Club Admin attempts to enter invalid data:**   * The system detects errors (e.g., missing required fields, invalid format) and displays appropriate error messages. | | |
| Exception: | **System error or connectivity issues:**The system displays an error message and prompts the Club Admin to try again later.  **The Club Admin lacks the required permissions:**The system denies access and displays an authorization error message. | | |

## **5.7. Join Club Request**

| Use Case ID: | **UC-03** | USE CASE NAME | **Join-Club-Request** |
| --- | --- | --- | --- |
| Create By: | AnNNH | Date Created: |  |
| Primary Actor: | Register User | Secondary Actor | None |
| Description | This use case describes how a register user can create request to join a club | | |
| Priority: | High | | |
| Trigger | User clicks on the "Join Club" Button in club list page. | | |
| Preconditions: | PRE-01: User has logged in into website | | |
| Post-Conditions | POST-01: The system works stably without errors.  POST-02: Join request sent successfully and awaiting approval from the Club Admin.  POST-03:Registered User receives notification of request status. | | |
| Basic Flow: | 1.The Registered User accesses the club list page.  2.The Registered User selects the desired club to join.  3.The system displays the club details and a "Submit Join Request" button.  4.The Registered User clicks the "Submit Join Request" button.  5.The system records the request and displays a success message.  6.The Club Admin receives the request and can approve or reject it.  7.The Registered User receives a notification when the request is processed. | | |
| Alternative Flow: | The club is not accepting new members:The system displays a message: "The club is not currently accepting new members."  The Registered User has already submitted a request:The system displays a message: "Your request is pending approval." | | |
| Exception: | None | | |

## **5.8. Assign Club Roles**

| Use Case ID: | **UC-08** | USE CASE NAME | **ASSIGN-CLUB**  **-ROLES** |
| --- | --- | --- | --- |
| Create By: | MinhDQ | Date Created: |  |
| Primary Actor: | Club Admin | Secondary Actor | Club Member |
| Description | This use case describes how a club admin assigns roles and permissions to members. | | |
| Priority: | High | | |
| Trigger | The club admin accesses the member management section. | | |
| Preconditions: | PRE-01**:** The club admin must be logged into the system.  PRE-02**:** The user receiving the role must be a member of the club. | | |
| Post-Conditions | POST-01**:** The system updates the assigned roles and permissions.  POST-02**:** The assigned member has updated access based on their new role. | | |
| Basic Flow: | 1. The club admin logs into the system. 2. Accesses the club management section. 3. Selects a member to assign a role. 4. Chooses a role and sets permissions. 5. Clicks "Save." 6. The system updates the member’s role and permissions. | | |
| Alternative Flow: | None | | |
| Exception: | **8a.** The admin selects an invalid user.   * **8a1.** The system displays an error message when the admin clicks “Save”. | | |

## 

## **5.9. Create Post**

| Use Case ID: | **UC-09** | USE CASE NAME | **CREATE-POST** |
| --- | --- | --- | --- |
| Create By: | MinhDQ | Date Created: |  |
| Primary Actor: | Club Member | Secondary Actor | Club Admin |
| Description | This use case describes how users create posts within a club. | | |
| Priority: | Medium | | |
| Trigger | A club member creates a new post. | | |
| Preconditions: | PRE-01**:** The user must be a member of the club. | | |
| Post-Conditions | POST-01**:** The post is created successfully.  POST-02**:** The post is on the approving list of club admin. | | |
| Basic Flow: | 1. The user logs into the system. 2. Accesses a club’s discussion section. 3. Clicks “Create Post.” 4. Enters content and submits. 5. The system creates the post on the approval list. | | |
| Alternative Flow: | None | | |
| Exception: | **9a.** The user submits empty content.   * **9a1.** The system displays an error message. | | |

## **5.10 Update Post**

| Use Case ID: | **UC-10** | USE CASE NAME | **UPDATE-POST** |
| --- | --- | --- | --- |
| Create By: | MinhDQ | Date Created: |  |
| Primary Actor: | Club Member | Secondary Actor | None |
| Description | This use case describes how users can update their posts. | | |
| Priority: | Medium | | |
| Trigger | A user wants to edit a post. | | |
| Preconditions: | PRE-01**:** The post must exist in the system.  PRE-02**:** The user must be the creator or have permissions. | | |
| Post-Conditions | POST-01**:** The post is updated successfully. | | |
| Basic Flow: | 1. The user logs into the system. 2. Accesses the post they created. 3. Clicks “Edit” to modify content. 4. Clicks “Save.” 5. The system updates the post. | | |
| Alternative Flow: | None | | |
| Exception: | **10a.** The user tries to edit a post they don’t have permission for.   * **10a1.** The system displays an error message. | | |

## **5.11 Delete Post**

| Use Case ID: | **UC-11** | USE CASE NAME | **DELETE-POST** |
| --- | --- | --- | --- |
| Create By: | MinhDQ | Date Created: |  |
| Primary Actor: | Club Member | Secondary Actor | Club Admin |
| Description | This use case describes how users can delete their posts. | | |
| Priority: | Medium | | |
| Trigger | A user wants to delete a post. | | |
| Preconditions: | PRE-01**:** The post must exist in the system.  PRE-02**:** The user must be the creator or have permissions. | | |
| Post-Conditions | POST-01: The post is deleted successfully. | | |
| Basic Flow: | 1. The user logs into the system. 2. Accesses the post they created. 3. Clicks “Delete.” 4. Confirms the action. 5. The system removes the post. | | |
| Alternative Flow: | None | | |
| Exception: | **10a.** The user tries to delete post they don’t have permission for.   * **10a1.** The system displays an error message. | | |

## **5.12 Comment Post**

| Use Case ID: | **UC-12** | USE CASE NAME | **INTERACT-POST** |
| --- | --- | --- | --- |
| Create By: | MinhDQ | Date Created: |  |
| Primary Actor: | Club Member | Secondary Actor | Club Admin |
| Description | This use case describes how users add comments to an existing post in a club. | | |
| Priority: | Low | | |
| Trigger | A user submits a comment on a post. | | |
| Preconditions: | PRE-01: The post must be visible to the user.  PRE-02: The user must be logged in. | | |
| Post-Conditions | POST-01: The comment is recorded and displayed under the post. | | |
| Basic Flow: | 1. The user logs into the system. 2. Accesses a club’s discussion section. 3. Selects a post to comment on. 4. Enters a comment and submits it. 5. The system records and displays the comment under the post. | | |
| Alternative Flow: | None | | |
| Exception: | **12a.** The post has been deleted or is no longer available.   * **12a1**. The system displays an error message. | | |

## 

## **5.13 Edit Comment**

| Use Case ID: | **UC-13** | USE CASE NAME | **EDIT-COMMENT** |
| --- | --- | --- | --- |
| Create By: | MinhDQ | Date Created: |  |
| Primary Actor: | Club Member | Secondary Actor | Club Admin |
| Description | This use case describes how users edit their existing comments on a post. | | |
| Priority: | Low | | |
| Trigger | A user selects the option to edit a comment. | | |
| Preconditions: | PRE-01: The comment must belong to the user.  PRE-02: The post must be visible to the user.  PRE-03: The user must be logged in. | | |
| Post-Conditions | POST-01: The edited comment replaces the original comment. | | |
| Basic Flow: | 1. The user logs into the system. 2. Accesses a post where they have commented. 3. Clicks on the "Edit" option for their comment. 4. Updates the comment content. 5. Clicks "Save." 6. The system updates the comment with the new content. | | |
| Alternative Flow: | None | | |
| Exception: | **12.4a.** The post has been deleted or is no longer available.   * **12.4a1.** The system displays an error message.   **12.4b.** The comment has been deleted or is no longer available.   * **12.4b1.** The system displays an error message. | | |

## 

## **5.14 Delete Comment**

| Use Case ID: | **UC-14** | USE CASE NAME | **DELETE-COMMENT** |
| --- | --- | --- | --- |
| Create By: | MinhDQ | Date Created: |  |
| Primary Actor: | Club Member | Secondary Actor | Club Admin |
| Description | This use case describes how users delete their comments from a post. | | |
| Priority: | Low | | |
| Trigger | A user selects the option to delete a comment. | | |
| Preconditions: | PRE-01: The comment must belong to the user.  PRE-02: The post must be visible to the user.  PRE-03: The user must be logged in. | | |
| Post-Conditions | POST-01: The edited comment is removed the original comment. | | |
| Basic Flow: | 1. The user logs into the system. 2. Accesses a post where they have commented. 3. Clicks on the "Delete" option for their comment. 4. Confirms the deletion. 5. The system removes the comment. | | |
| Alternative Flow: | None | | |
| Exception: | **12.5a.** The post has been deleted or is no longer available.   * **12.5a1.** The system displays an error message.   **12.5b.** The comment has already been deleted.   * **12.5b1.** The system displays an error message. | | |

## 

## **5.15 Like Post**

| Use Case ID: | **UC-15** | USE CASE NAME | **LIKE-POST** |
| --- | --- | --- | --- |
| Create By: | MinhDQ | Date Created: |  |
| Primary Actor: | Club Member | Secondary Actor | Club Admin |
| Description | This use case describes how users like an existing post in a club. | | |
| Priority: | Low | | |
| Trigger | A user submits a comment on a post. | | |
| Preconditions: | PRE-01: The post must be visible to the user.  PRE-02: The user must be logged in. | | |
| Post-Conditions | POST-01: The like is recorded and reflected in the system. | | |
| Basic Flow: | 1. The user logs into the system. 2. Accesses a club’s discussion section. 3. Selects a post to like. 4. Clicks the "Like" button. 5. The system records and updates the like count. | | |
| Alternative Flow: | None | | |
| Exception: | **12a.** The post has been deleted or is no longer available.   * **12a1**. The system displays an error message. | | |

## 

## **5.16 Share Post**

| Use Case ID: | **UC-16** | USE CASE NAME | **SHARE-POST** |
| --- | --- | --- | --- |
| Create By: | MinhDQ | Date Created: |  |
| Primary Actor: | Club Member | Secondary Actor | Club Admin |
| Description | A user clicks the "Share" button on a post. | | |
| Priority: | Low | | |
| Trigger | A user clicks the "Share" button on a post. | | |
| Preconditions: | PRE-01: The post must be visible to the user.  PRE-02: The user must be logged in. | | |
| Post-Conditions | POST-01: The shared post is available for other users to see. | | |
| Basic Flow: | 1. The user logs into the system. 2. Accesses a club’s discussion section. 3. Selects a post to share. 4. Clicks the "Share" button. 5. Chooses the sharing method (e.g., within the club, external link). 6. The system records the share and displays it in the selected location. | | |
| Alternative Flow: | None | | |
| Exception: | **12a.** The post has been deleted or is no longer available.   * **12a1**. The system displays an error message. | | |

## 

## **5.17 Approve Post**

| Use Case ID: | **UC-17** | USE CASE NAME | **APPROVE-POST** |
| --- | --- | --- | --- |
| Create By: | MinhDQ | Date Created: |  |
| Primary Actor: | Club Admin | Secondary Actor | Club Member |
| Description | This use case describes the process of approving or rejecting posts before they are published. | | |
| Priority: | Medium | | |
| Trigger | A club member submits a post for approval. | | |
| Preconditions: | PRE-01: Post approval is enabled for the club.  PRE-02: The club admin/moderator is logged into the system. | | |
| Post-Conditions | POST-01: The post is either published or rejected.  POST-02: The user receives a notification about the decision. | | |
| Basic Flow: | 1. A club member submits a post. 2. The system sends the post to the moderation queue. 3. A club admin/moderator reviews the post. 4. If approved, the system publishes the post. 5. If rejected, the system notifies the user. | | |
| Alternative Flow: | None | | |
| Exception: | **13a.** The admin/moderator finds the post violates rules.   * **13a1.** The system rejects the post and notifies the user. | | |

## **5.18 View Notification**

| Use Case ID: | **UC-18** | USE CASE NAME | **View Notification** |
| --- | --- | --- | --- |
| Create By: | PhuongPH | Date Created: |  |
| Primary Actor: | Registered User | Secondary Actor | None |
| Description | This use case describes how a registered user can view notifications related to club activities, events, and updates. | | |
| Priority: | Medium | | |
| Trigger | User navigates to the "Notifications" section. | | |
| Preconditions: | PRE-01: User has a created and valid account  PRE-02: User account has been authorised. | | |
| Post-Conditions | POST-01: User successfully views the notifications. | | |
| Basic Flow: | 1. User navigates to the "Notifications" section. 2. System retrieves and displays all notifications. | | |
| Alternative Flow: | 2a. No new notifications are available.   * System displays a "No new notifications" message. | | |
| Exception: | 2c. System fails to retrieve notifications due to a server error. | | |

## **5.19 Approve/Reject Membership Request**

| Use Case ID: | **UC-19** | USE CASE NAME | **Approve/Reject Membership Request** |
| --- | --- | --- | --- |
| Create By: | PhuongPH | Date Created: |  |
| Primary Actor: | Club Admin | Secondary Actor | None |
| Description | This use case allows the club admin to approve or reject membership requests submitted by users. | | |
| Priority: | High | | |
| Trigger | Admin accesses the membership request list. | | |
| Preconditions: | PRE-01: The admin is logged in.  PRE-02: Pending membership requests exist. | | |
| Post-Conditions | POST-01: The membership status is updated based on admin action. | | |
| Basic Flow: | 1. Admin views the list of membership requests. 2. Admin selects a request and chooses to approve or reject it. 3. System updates the membership status accordingly. | | |
| Alternative Flow: | 2a. Admin wants to review member details before deciding.   * Admin clicks on the member profile. * System displays the member's information. | | |
| Exception: | 3c. System fails to process the request due to a database error. | | |

## **5.20 Remove Members**

| Use Case ID: | **UC-20** | USE CASE NAME | **Remove Members** |
| --- | --- | --- | --- |
| Create By: | PhuongPH | Date Created: |  |
| Primary Actor: | Club Admin | Secondary Actor | None |
| Description | Allows a club admin to remove a member from the club. | | |
| Priority: | Medium | | |
| Trigger | Admin accesses the member list and selects a member to remove. | | |
| Preconditions: | PRE-01: The admin is logged in.  PRE-02: At least one member exists in the club. | | |
| Post-Conditions | POST-01: The member is removed from the club. | | |
| Basic Flow: | 1. Admin accesses the member list. 2. Admin selects a member and chooses "Remove." 3. System removes the member from the club. | | |
| Alternative Flow: | 2a. Admin wants to reassign tasks before removing a member.   * System suggests reassigning tasks. * Admin selects another member for reassignment. | | |
| Exception: | 3c. System fails to remove the member due to a database error. | | |

## **5.21 Create Club Event**

| Use Case ID: | **UC-21** | USE CASE NAME | **Create Club Event** |
| --- | --- | --- | --- |
| Create By: | PhuongPH | Date Created: |  |
| Primary Actor: | Club Admin | Secondary Actor | None |
| Description | Allows a club admin to create a new event. | | |
| Priority: | High | | |
| Trigger | Admin selects the "Create Event" option.. | | |
| Preconditions: | PRE-01: Admin is logged in. | | |
| Post-Conditions | POST-01: Event is created successfully. | | |
| Basic Flow: | 1. Admin navigates to the event creation page. 2. Admin enters event details and submits. 3. System saves the event and notifies club members. | | |
| Alternative Flow: | 2a. Admin saves the event as a draft. | | |
| Exception: | 3c. System fails due to validation errors. | | |

## **5.22 Update Club Event**

| Use Case ID: | **UC-22** | USE CASE NAME | **Update Club Event** |
| --- | --- | --- | --- |
| Create By: | PhuongPH | Date Created: |  |
| Primary Actor: | Club Admin | Secondary Actor | None |
| Description | Allows a club admin to modify event details. | | |
| Priority: | High | | |
| Trigger | Admin selects the "Create Event" option.. | | |
| Preconditions: | PRE-01: Event exists. | | |
| Post-Conditions | POST-01: Event details are updated successfully. | | |
| Basic Flow: | 1. Admin selects an event. 2. Admin modifies event details. 3. System saves changes. | | |
| Alternative Flow: | 2a. Admin cancels changes. | | |
| Exception: | 3c. System fails to update event. | | |

## **5.23 Delete Club Event**

| Use Case ID: | **UC-23** | USE CASE NAME | **Delete Club Event** |
| --- | --- | --- | --- |
| Create By: | PhuongPH | Date Created: |  |
| Primary Actor: | Club Admin | Secondary Actor | None |
| Description | Admin selects an event to delete. | | |
| Priority: | High | | |
| Trigger | Admin selects the "Create Event" option.. | | |
| Preconditions: | PRE-01: Event exists. | | |
| Post-Conditions | POST-01: Event is deleted. | | |
| Basic Flow: | 1. Admin selects an event. 2. Admin confirms deletion. 3. System removes the event. | | |
| Alternative Flow: | 2a. Admin cancels deletion. | | |
| Exception: | 3c. System fails to delete the event. | | |

## **5.24 Create Task**

| Use Case ID: | **UC-24** | USE CASE NAME | **Create Task** |
| --- | --- | --- | --- |
| Create By: | PhuongPH | Date Created: |  |
| Primary Actor: | Club Admin | Secondary Actor | None |
| Description | Allows an admin to create a task. | | |
| Priority: | Medium | | |
| Trigger | Admin selects "Create Task." | | |
| Preconditions: | PRE-01: Admin is logged in. | | |
| Post-Conditions | POST-01: Task is created successfully. | | |
| Basic Flow: | 1. Admin enters task details. 2. System saves tasks. | | |
| Alternative Flow: | 2a. Admin saves as a draft. | | |
| Exception: | 3c. System fails to save tasks. | | |

## **5.25 Delete Task**

| Use Case ID: | **UC-25** | USE CASE NAME | **Delete Task** |
| --- | --- | --- | --- |
| Create By: | PhuongPH | Date Created: |  |
| Primary Actor: | Club Admin | Secondary Actor | None |
| Description | Allows an admin to delete a task. | | |
| Priority: | Medium | | |
| Trigger | Admin selects "Delete Task." | | |
| Preconditions: | PRE-01: Admin is logged in.  PRE-02: Task exists. | | |
| Post-Conditions | POST-01: Task is deleted successfully. | | |
| Basic Flow: | 1. Admin selects a task. 2. Admin confirms deletion. | | |
| Alternative Flow: | 2a. Admin cancels deletion. | | |
| Exception: | 3c. System fails to delete. | | |

## **5.26 Update Task**

| Use Case ID: | **UC-26** | USE CASE NAME | **Update Task** |
| --- | --- | --- | --- |
| Create By: | PhuongPH | Date Created: |  |
| Primary Actor: | Club Admin | Secondary Actor | None |
| Description | Allows an admin to update a task. | | |
| Priority: | Medium | | |
| Trigger | Admin selects "Update Task." | | |
| Preconditions: | PRE-01: Admin is logged in.  PRE-02: Task exists. | | |
| Post-Conditions | POST-01: Task is updated successfully. | | |
| Basic Flow: | 1. Admin selects a task. 2. Admin confirms deletion. | | |
| Alternative Flow: | 2a. Admin cancels update. | | |
| Exception: | 3c. System fails to update. | | |

## **5.27 View Task**

| Use Case ID: | **UC-27** | USE CASE NAME | **View Task** |
| --- | --- | --- | --- |
| Create By: | PhuongPH | Date Created: |  |
| Primary Actor: | Club Member | Secondary Actor | None |
| Description | Allows a club member to view tasks. | | |
| Priority: | Medium | | |
| Trigger | Club member access task list | | |
| Preconditions: | PRE-01: The user is logged in.  PRE-02: the user is a member of a club | | |
| Post-Conditions | POST-01: The system displays a list of tasks assigned to the member. | | |
| Basic Flow: | 1. User navigates to the "Tasks" section. 2. System retrieves and displays a list of tasks assigned to the user. 3. User selects a task to view details. 4. System displays task details. | | |
| Alternative Flow: | 2a. No tasks are assigned to the user.   * System displays "No tasks available." | | |
| Exception: | 2c. System fails to retrieve tasks due to a server error. | | |

## **5.28 Assign Task for Club Member**

| Use Case ID: | **UC-28** | USE CASE NAME | **Assign Task for Club Member** |
| --- | --- | --- | --- |
| Create By: | PhuongPH | Date Created: |  |
| Primary Actor: | Club Admin | Secondary Actor | Club Member |
| Description | This use case allows a club admin to assign tasks to club members. | | |
| Priority: | High | | |
| Trigger | Admin selects "Assign Task" from the menu. | | |
| Preconditions: | PRE-01: The admin is logged in.  PRE-02: At least one member exists in the club.  PRE-03: At least one task exists. | | |
| Post-Conditions | POST-01: The task is successfully assigned to a club member. | | |
| Basic Flow: | 1. Admin navigates to the "Manage Tasks" section. 2. Admin selects a task to assign. 3. Admin chooses a club member and assigns the task. 4. System updates the task with the assigned member. 5. System notifies the assigned member. | | |
| Alternative Flow: | 3a. Admin assigns the task to multiple members.   * System updates the task for all selected members. | | |
| Exception: | 4c. System fails to assign the task due to a validation error (e.g., member does not exist). | | |

## **5.29 Approve/Reject Create Club Requests**

| Use Case ID: | **UC-29** | USE CASE NAME | **Approve/Reject Create Club Request** |
| --- | --- | --- | --- |
| Create By: | AnNNH | Date Created: |  |
| Primary Actor: | Club Admin | Secondary Actor | None |
| Description | This use case Describes system admin approve or reject Create club requests submitted by users. | | |
| Priority: | High | | |
| Trigger | System Admin Click on ”Create club request” button on home page. | | |
| Preconditions: | PRE-01: The system admin is logged in.  PRE-02: There are pending club creation requests in the system.. | | |
| Post-Conditions | POST-01:If approved, the club is officially created and available in the system.  POST-02:If rejected, the club request is removed, and the Registered User is notified of the decision | | |
| Basic Flow: | 1. The System Admin accesses the club creation request management page. 2. The System Admin views the list of pending club creation requests. 3. The System Admin selects a request to review. 4. The System Admin examines the details of the request, including the club name, description, and other relevant information. 5. The System Admin chooses to either approve or reject the request. 6. If approved, the system creates the club and assigns the requesting user as the Club Admin. 7. If rejected, the system removes the request and notifies the user of the rejection reason. 8. The System Admin receives confirmation of the action taken. | | |
| Alternative Flow: | **The System Admin postpones the decision:**   * The System Admin exits the review page without taking action. * The request remains in the pending list for future review. | | |
| Exception: | **System error or connectivity issues**:The system displays an error message and prompts the System Admin to try again later.  **The System Admin lacks the necessary permissions**:The system denies access and displays an authorization error message.  **The request has already been processed by another admin**:The system notifies the System Admin that the request has already been handled. | | |

## **5.30 Delete Club**

## 

| Use Case ID: | **UC-30** | USE CASE NAME | **DELETE-CLUB** |
| --- | --- | --- | --- |
| Create By: | AnNNH | Date Created: |  |
| Primary Actor: | System Admin | Secondary Actor | None |
| Description | This use case describes how System admin can delete a club. | | |
| Priority: | Medium | | |
| Trigger | System admin clicks on the “Delete club” button on Club detail page. | | |
| Preconditions: | PRE-01: The club must exist in the system.  PRE-02: The System Admin is logged into the system. | | |
| Post-Conditions | POST-01: The club is deleted successfully. | | |
| Basic Flow: | 1. The System Admin accesses the club management dashboard. 2. The System Admin views the list of existing clubs. 3. The System Admin selects the club to be deleted. 4. The system displays a confirmation prompt to confirm the deletion. 5. The System Admin confirms the deletion. 6. The system removes the club and all associated data. 7. The system notifies the Club Admin and members about the deletion. 8. The System Admin receives a confirmation message of the successful deletion. | | |
| Alternative Flow: | None | | |
| Exception: | **System error or connectivity issues**:The system displays an error message and prompts the System Admin to try again later.  **The club does not exist**:The system notifies the System Admin that the selected club is no longer available for deletion.  **Insufficient permission**s:The system denies access and displays an authorization error message. | | |

## **5.31 Accept/Decline Task**

| Use Case ID: | **UC-31** | USE CASE NAME | **Accept/Decline Task** |
| --- | --- | --- | --- |
| Create By: | PhuongPH | Date Created: | dit me may |
| Primary Actor: | Club Admin | Secondary Actor | None |
| Description | Allows an admin to create a task. | | |
| Priority: | High | | |
| Trigger | Admin selects "Create Task." | | |
| Preconditions: | PRE-01: Admin is logged in.  PRE-02: Task exists and is in a pending state. | | |
| Post-Conditions | POST-01: Task status is updated to "Accepted" or "Declined." | | |
| Basic Flow: | 1. Admin views the task. 2. Admin selects "Accept" or "Decline." 3. System updates the task status accordingly. | | |
| Alternative Flow: | 2a. Admin chooses to mark the task as "Pending Review." | | |
| Exception: | 3c. System fails to update task status. | | |

## **5.32 Join Event**

| Use Case ID: | **UC-32** | USE CASE NAME | **Join Event** |
| --- | --- | --- | --- |
| Create By: | AnNNH | Date Created: |  |
| Primary Actor: | Club Member | Secondary Actor | None |
| Description | Allows a club member to join an event. | | |
| Priority: | High | | |
| Trigger | Club member selects the "Join Event" option.. | | |
| Preconditions: | PRE-01: Club member is logged in.  PRE-02: Club member belong to a club  PRE-03: Club has event | | |
| Post-Conditions | POST-01: Club member join event successfully. | | |
| Basic Flow: | 1.Club member navigates to the event page.  2.Club member click an event  3.Club member click “Join event” button in event detail page | | |
| Alternative Flow: | none | | |
| Exception: | 3c. System fails due to validation errors. | | |

## **5.33 View Club Event**

| Use Case ID: | **UC-33** | USE CASE NAME | **View Club Event** |
| --- | --- | --- | --- |
| Create By: | AnNNH | Date Created: |  |
| Primary Actor: | User | Secondary Actor | None |
| Description | Allows a user to view events organized by clubs. | | |
| Priority: | High | | |
| Trigger | A user selects the "View Events" option in the club page. | | |
| Preconditions: | PRE-01: Club has event | | |
| Post-Conditions | POST-01: User views event successfully. | | |
| Basic Flow: | 1.User navigates to the event page.  2.User clicks an event  3.System displays a list of events associated with the club. | | |
| Alternative Flow: | none | | |
| Exception: | 3c. System fails to retrieve event details due to a server error. | | |

## 

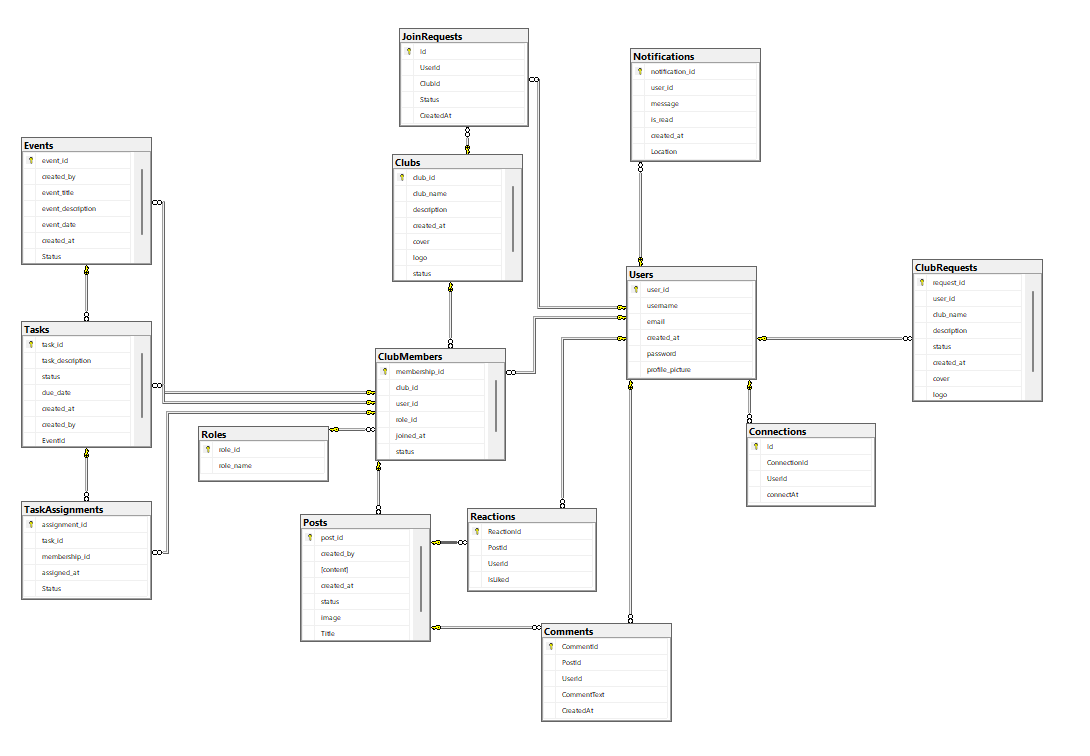
## 

## **5.34 View Club**

| Use Case ID: | **UC-33** | USE CASE NAME | **View Club Event** |
| --- | --- | --- | --- |
| Create By: | PhuongPH | Date Created: |  |
| Primary Actor: | User | Secondary Actor | None |
| Description | Allows a user to view events organized by clubs. | | |
| Priority: | High | | |
| Trigger | Allows a user to view general information about their club. | | |
| Preconditions: | PRE-01: Club is exist | | |
| Post-Conditions | POST-01: User views club successfully. | | |
| Basic Flow: | 1.User navigates to the clubs page.  2.User clicks a club.  3.System displays an overview of the club, including name, description, and members. | | |
| Alternative Flow: | none | | |
| Exception: | 3c. System fails to retrieve club details due to a server error. | | |

# **6.** **DATA STRUCTURE**

## **6.1** **Database**

****

# **7.** **NON-FUNCTIONAL**

## **1. External Interfaces**

### **1.1. User Interface**

* UI-01: The system must display Success Message or Error Message when completing an action.
* UI-02: When the user is doing the Delete option, there must be a Confirm (Y/N) dialog.
* UI-03: The UI must minimise horizontal scrolling.
* UI-04: All pages in the system must have a Header menu and a Navigation menu.

### **1.2. Software Interface**

* SI-01: Users can use Google services as Login, which will help them to authenticate their account.
* UI-02: When the user is doing the Delete option, there must be a Confirm (Y/N) dialog.
* UI-03: The UI must minimise horizontal scrolling.
* UI-04: All pages in the system must have a Header menu and a Navigation menu.

## **2. Quality Attributes**

### **2.1. Availability**

* The Club management System should be accessible for at least 95% of the time
* The Club management System is accessible 24/7 from Monday to Sunday, except for the maintenance hours of 23:00-24:00 on Sundays.

### **2. 2 Usability**

* An untrained user can use the payment function within a maximum of 15 minutes.
* Links and buttons are easily clickable.

### **2.3. Reliability**

* Under the worst conditions, the mean time between failures should be no more than 5%.
* The platform's ability to recover from any issues should take no longer than 5 hours.

### **2.4. Performance**

* The platform’s loading time should not exceed 2 seconds.
* The platform’s content should be both rendered and readable within a maximum of 3 seconds
* Processing times of the most complexes request must complete within 3s (not include timeout ones)
* Uploads of large images (>20mb) should be complete processing after 3s.
* The platform has the ability to handle up to 200 requests per second.

### **2.5. Security**

* The user’s password should be encrypted using the Bcrypt algorithm before being stored in the database.
* Data backup should be performed every month.

# **8.** **APPENDICES**

## Appendix A: Acronyms and Abbreviations

A list of commonly used acronyms and abbreviations in this document:

* **SDS**: Software Design Specification
* **UI**: User Interface
* **API**: Application Programming Interface
* **DB**: Database
* **RBAC**: Role-Based Access Control

## Appendix B: References

A list of documents, tools, and resources referenced in this specification:

* System architecture diagrams
* Database schema design documents
* Relevant API documentation
* Security and compliance guidelines

## Appendix C: Tools and Technologies

A summary of the tools and technologies used in the system:

* **Backend**: NET 9.0
* **Frontend**: HTML, CSS, JavaScript
* **Database**: SQL Server
* **Version Control**: Git, GitHub
* **Testing Framework**: JUnit

## Appendix D: Glossary

Definitions of key terms used in this document:

* **Scalability**: The ability of the system to handle growth efficiently.
* **Microservices**: An architectural style where applications are composed of loosely coupled services.
* **Authentication**: The process of verifying user identity.
* **Encryption**: The process of encoding information to protect it from unauthorized access.

## Appendix E: Contact Information

Relevant contact details for project stakeholders and support:

* **Team Member**: Phan Hoàng Phương - phuongphde180422
* **Team Leader**: Đặng Quang Minh - minhdqde180271
* **Team Member**: Nguyễn Nghĩa Hải An - annnhde180150