Software Requirements Specification

for

Club Management System

Prepared by

Nur Fatin Nabilah Binti Mohd Shofe

Nur Fakhira Binti Rosdi

Muhamad Amir Hassan Bin Hamzan

Mohamad Danial Effendi Bin Ariff Fadzilah

13 May 2020

Contents

1. Introduction

1.1 Problem Statement

Some of the students in Universiti Putra Malaysia (UPM) are actively involved in the clubs that exist in the university. It is time-consuming for them to join a club that they want as they need to go to the place to register as a member. Also, it is an inconvenience technique since students need to register as a member using the paper-based technique. Then, the problem is there is a limited medium to spread the information about the club to the members and cause the student missed the opportunity to participate in the activity.

This project will deal with this problem and it will ease the members' task to join a club without the need to request being a member in a manual way which is queuing up for becoming a member.

1.2 Purpose

The purpose of this document is to present a detailed description of the Club Management System (CMS). It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli.

1.3 Scope

The final product for CMS is a web-platform system that provides a new management system for clubs in Universiti Putra Malaysia. This system will allow the user to manage the club in a proper technique and minimize the number of errors in managing a club. CMS users will be able to:

- Register as a member
- View the activities of the registered club
- Make payment for the fee
- View annually reports for the club

1.4 Definitions, Acronyms, and Abbreviations

Terms	Definition
CMS	Club Management System
UPM	University Putra Malaysia
Instructors	The instructor for the clubs in the system
Members	The members of the clubs in the system
UC	Use Case

1.5 References

1. IEEE Software Engineering Standards Committee, "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications", October 20, 1998.
2. *https://www.cse.msu.edu/~cse435/Handouts/SRSExample-webapp.doc*
3. http://www.cse.chalmers.se/~feldt/courses/reqeng/examples/srs_example_2010_group2.pdf
4. *https://trello.com/c/39YU00db/12-completed-srs-example*
5. *https://www.scribd.com/book/55958794*
6. *http://www.radford.edu/team14/requirements.doc*

2. Overall Description

2.1 Assumptions and Dependencies

Assumption

- Instructor or members of CMS has an active internet connection or has access to one to view the website
- Instructors or members run an operating system which supports internet browsing.
- Instructors or members run the latest browsing software to access our website.
- Instructors or members have experience in using web application features.
- Instructors or members can understand English language as the main language in this web application.

Dependencies

- Users need an internet connection of at least 500Kb per second.
- Users need to run an operating system that supports internet browsing.
- Our system only relies on the student Id which is matric no. and password to log in, there is no alternative way to log in into our system.
- For the payment process it depends on the efficiency of the bank web application system.

2.2 Existing System and Limitations

Club Management System

- System is only available in English.
- Provides too many unnecessary features that most university clubs will never use.
- Club managers/administrators need to be trained thoroughly before implementing the system.

GymMaster

- Expensive fees needed to pay using the software
- Located in New Zealand, different time zones- limited time for getting support.

3. External Interface Requirements

The system will be implemented using the XHTML language. NetBeans will be used as an IDE to implement.

3.1 User Interfaces

Several user interfaces have been developed for our system to ease the users of this system. Moreover, these user interfaces are developed in English languages as our targeted market are students of Universiti Putra Malaysia which consists of multiracial.

3.2 Hardware Interfaces

This system needs smartphones and desktop computers to access the system. It uses a keyboard to input text data. It also uses a mouse or touch screen to allow the cursor to move on a screen.

3.3 Software Interfaces

Since this system is a web-based system, this system can operate on Windows, Safari, Android, and iOS operating systems. Furthermore, this system will use phpMyAdmin as the database management system to store the data of instructor, member, and club details. To develop this system, we will choose XHTML language to implement it.

4. System Analysis

4.1 Product Function

4.1.1 Register

After the user browses our system, the user must register to the system to use it. There will be a register button in the login page which will be redirected to the user registration page. User needs to fill all the required fields in the interface. After the user registered to the system, the user can proceed to login to use the system.

4.1.2 Login

The user that has registered in the system can log in to the system using their username and password.

4.1.3 Update activity

An instructor can update the activities for the club including the name of activities, date, location, and detail of the activity. After adding the and update the activity, the activity will be listed in the activity column.

4.1.4 Update profile

Both instructor and member can update their profile based on the field provided in the update profile interface.

4.1.5 Access payment record

The members of the club need to make annual payments when they join a certain club. The instructor can view the payment made by the members to follow-up which members have already made the payment or still not made the payment.

4.1.6 Generate report

Instructor generates a report for the club based on the name, date, attendance for the club and details about the activity that has been conducted.

4.1.7 View activity

Members of the club can view the activity that has been added by the instructor.

4.2 Use Case Diagram

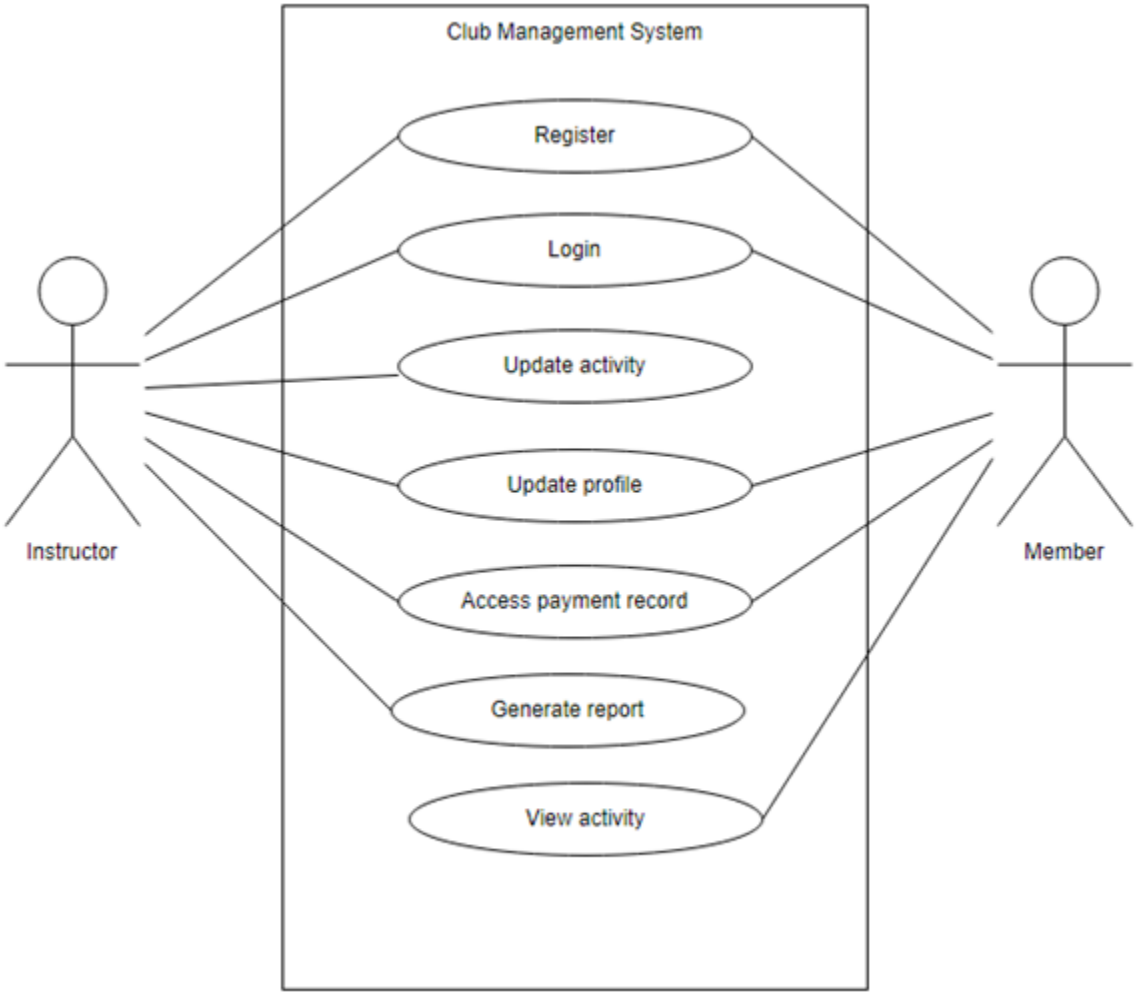

Figure 4.2: Club Management System Use Case Diagram

Identifier	Use case	Description of use case
UC1	Register	Instructors and members must register in order to use the system.
UC2	Login	Instructors and members have to login before using the system.
UC3	Update activity	The instructor can update the activities for the club.
UC4	Update profile	Instructors and members can update their own profile when necessary.
UC5	Access payment record	Members can access their payment history after they succeed making the payment for the club.
UC6	Generate report	The instructor can generate a report for each of the activities that has been conducted.
UC7	View activity	Members can view the activity that has been added by the instructor.

4.2.1 Register (UC1)

Identifier	Use Cases	Description
UC1	Register	1. **Objective** – New members and instructors need to register into the system in order to use this system. 2. **Priority** - High 3. **Actors –** Member, instructor 4. **Flow of Events** 4.1. **Basic Flow** 4.1.1. Members and instructors will fill the registration form. 4.1.2. Their data will be saved in the database of the system. 5. **Preconditions –** Members and instructors need to click the register button. 6. **Post conditions –** Member and instructor is successfully registered. 7. **Notes –** None

Table 4.2.1: Description for register use case (UC1)

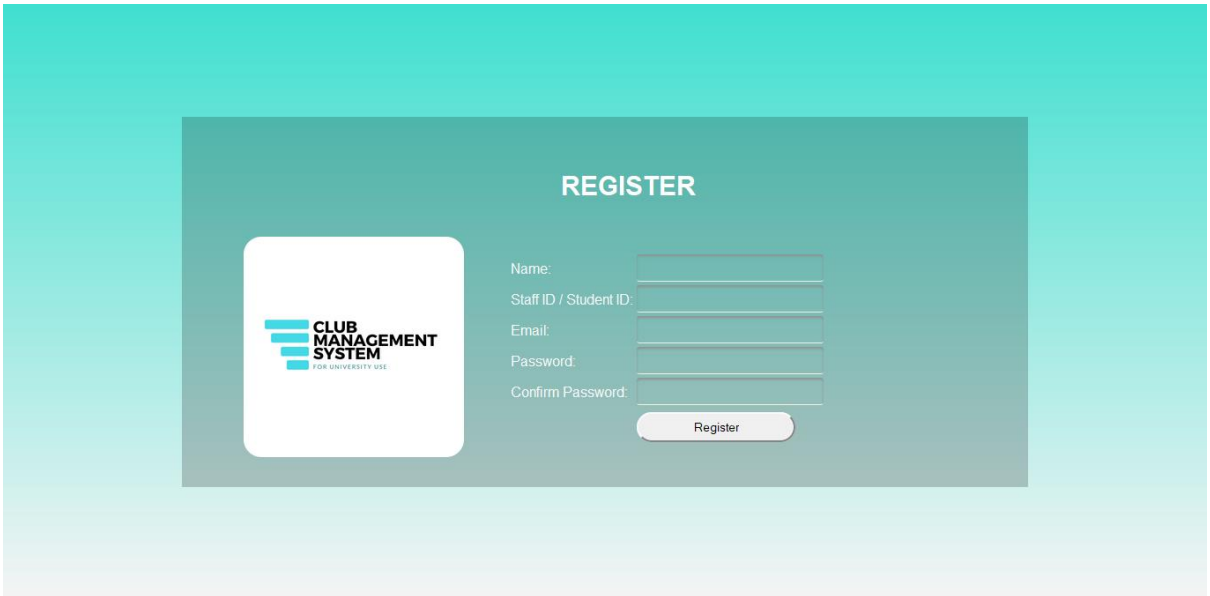

Figure 4.2.1 shows the register User Interface for CMS. Register allows instructor and member to register into CMS before using the features in it.

4.2.2 Login (UC2)

Identifier	Use Cases	Description
UC2	Login	1. **Objective** – New member and instructor login into the system if they already have an account. 2. **Priority** - High 3. **Actors –** Member, instructor 4. **Flow of Events** 4.2. **Basic Flow** 4.1.1. Members and instructors will login into the system. 5. **Preconditions –** Member and instructor open CMS webpage. 6. **Post conditions –** Member and instructor is successful login. 7. **Notes –** None

Table 4.2.2: Description for login use case (UC2)

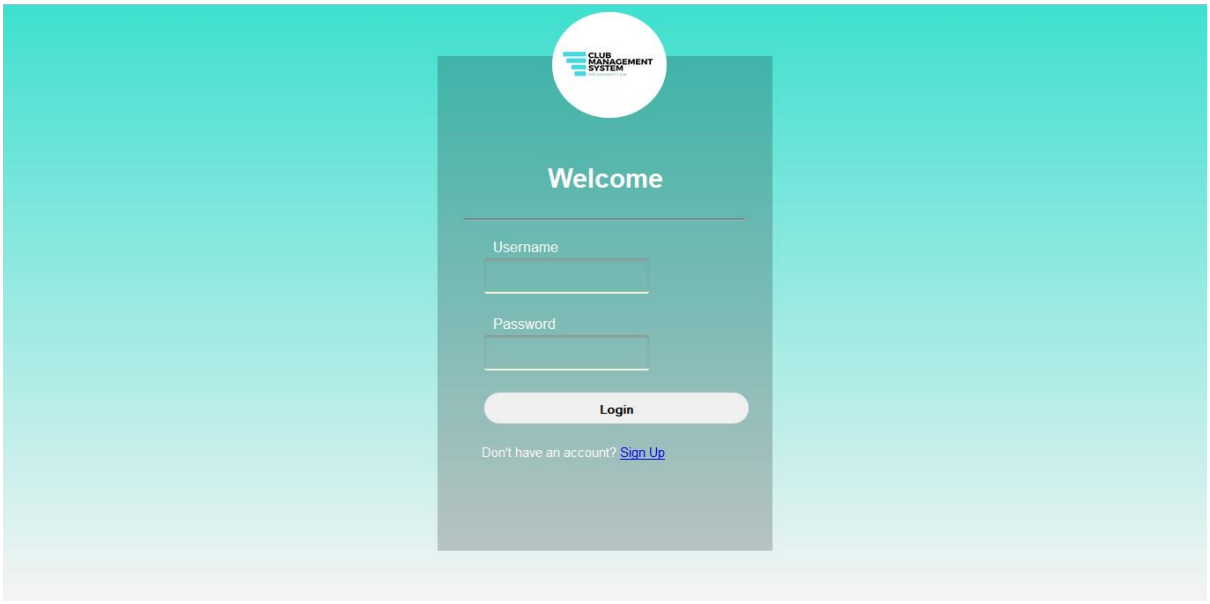

Figure 4.2.2 shows login User Interface for CMS. Members or instructors need to enter their username and password to login into the system.

4.2.3 Update Activity (UC3)

Identifier	Use Cases	Description
UC3	Update activity	1. **Objective** – The instructor can update activities of the club inside the system. 2. **Priority** - Medium 3. **Actors –** Instructor 4. **Flow of Events** 4.1. **Basic Flow** 4.1.1. The instructor will login into their account 4.1.2. Instructors update the activities into the system. 5. **Pre-conditions –** Instructors need to login into the system and click the update activity button. 6. **Post conditions –** The system will show new activities on the main page 7. **Notes –** None

Table 4.2.3: Description for update activity use case (UC3)

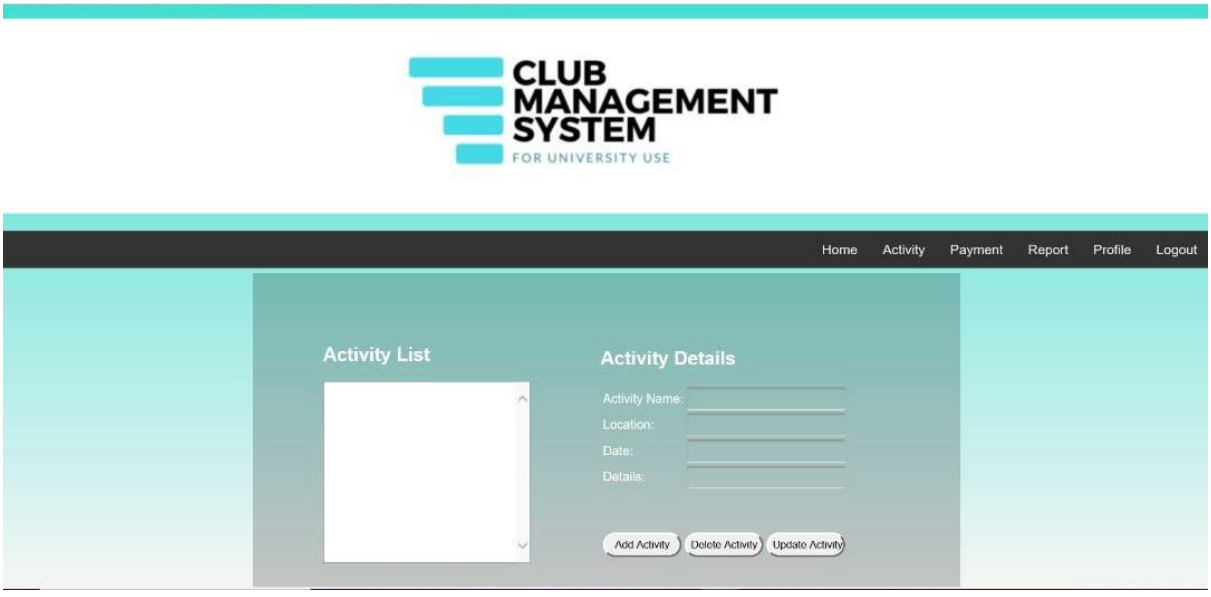

Figure 4.2.3 shows update activity User Interface. Update activity allows instructors to add and update about the activity for the club.

4.2.4 Update Profile (UC4)

Identifier	Use cases	Description
UC4	Update profile	1. **Objective** – Member and instructor can update their profile and the administrator can save in the database 2. **Priority** - Medium 3. **Actors –** Member, instructor 4. **Flow of Events** **4.1. Basic flow** 4.1.1. The member will update their profile 4.1.2. Administrator will receive the update and save it into the system database 5. **Pre-conditions –** Member or instructor click profile button 6. **Post conditions –** The new profile will be saved in the database 7. **Notes –** None

Table 4.2.4: Description for update profile use case (UC4)

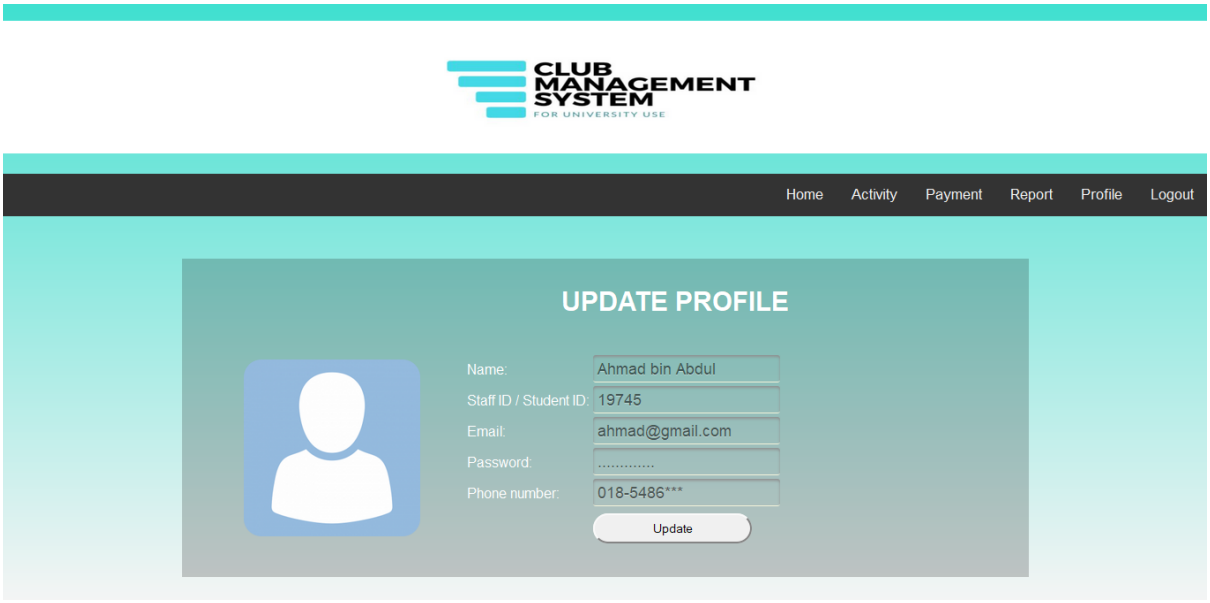

Figure 4.2.4 shows update profile User Interface for CMS. Update profile allows the instructor and member to update their email, password, and phone number when it is necessary.

4.2.5 Access payment records (UC5)

Identifier	Use cases	Description
UC5	Access payment records	1. **Objective** – The instructor can see all the user payment record while the member can see their own payment record 2. **Priority** - High 3. **Actors –** Instructor, Member 4. **Flow of Events** 4.1. **Basic Flow** 4.1.1. The member can see their payment record to make sure there is no error 4.1.2. The instructor can see all the payment records to make sure all of it are successful 5. **Pre-conditions –** Instructor and member click on the payment button. 6. **Post conditions –** Instructor and member can view payment record. 7. **Notes –** None

Table 4.2.5: Description for payment record use case (UC5)

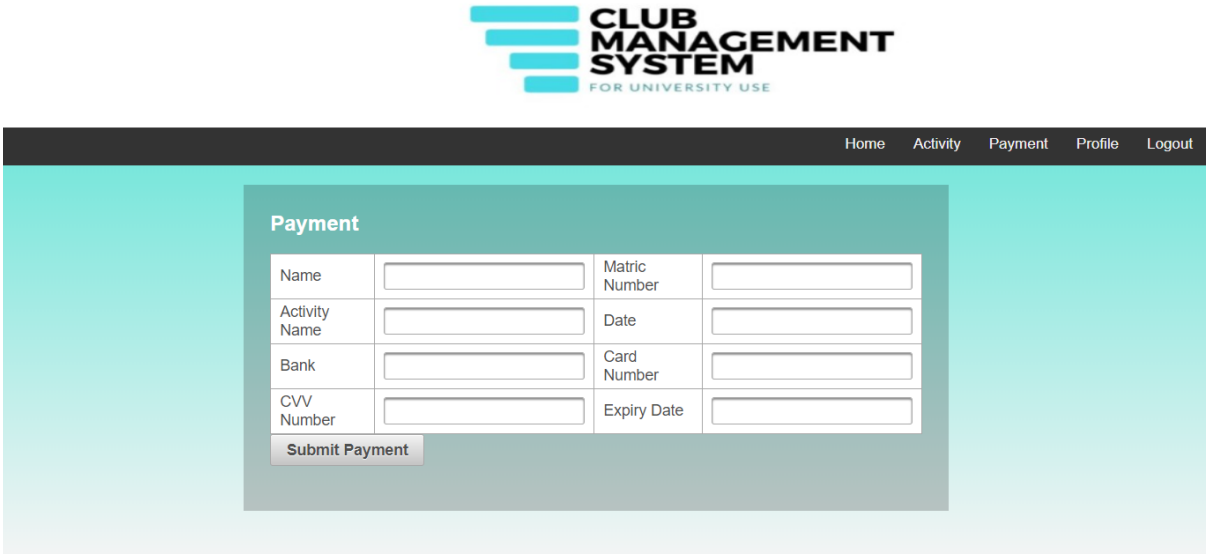

Figure 4.2.5 (a) shows payment User Interface for members. This payment interface allows members to make any payment needed for each activity.

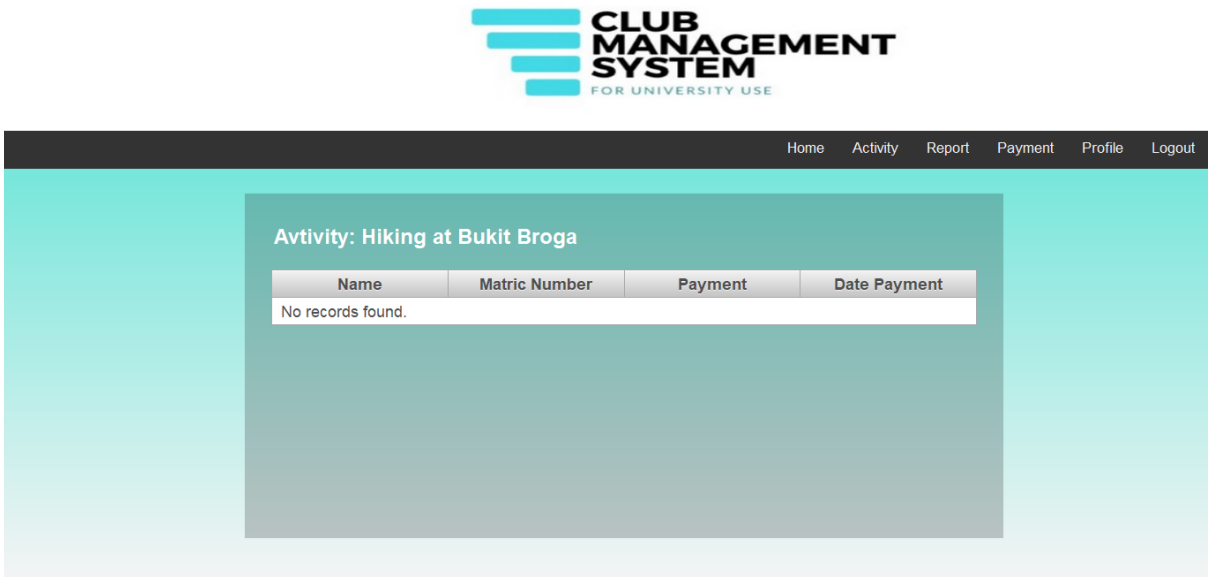

Figure 4.2.5 (b) shows payment record User Interface. Payment record allows instructors to monitor members that make payment for certain activities.

4.2.6 Generate report (UC6)

Identifier	Use cases	Description
UC6	Generate report	1. **Objective** – The instructor can generate report about the activity of the club 2. **Priority** - High 3. **Actors –** Instructor 4. **Flow of Events** 4.1. **Basic Flow** 4.1.1 Instructor will generate report about specific activity 5. **Pre-conditions –** Instructor will click report button 6. **Post conditions –** A report will be created that can help in future reference 7. **Notes –** None

Table 4.2.6: Description for generate report use case (UC6)

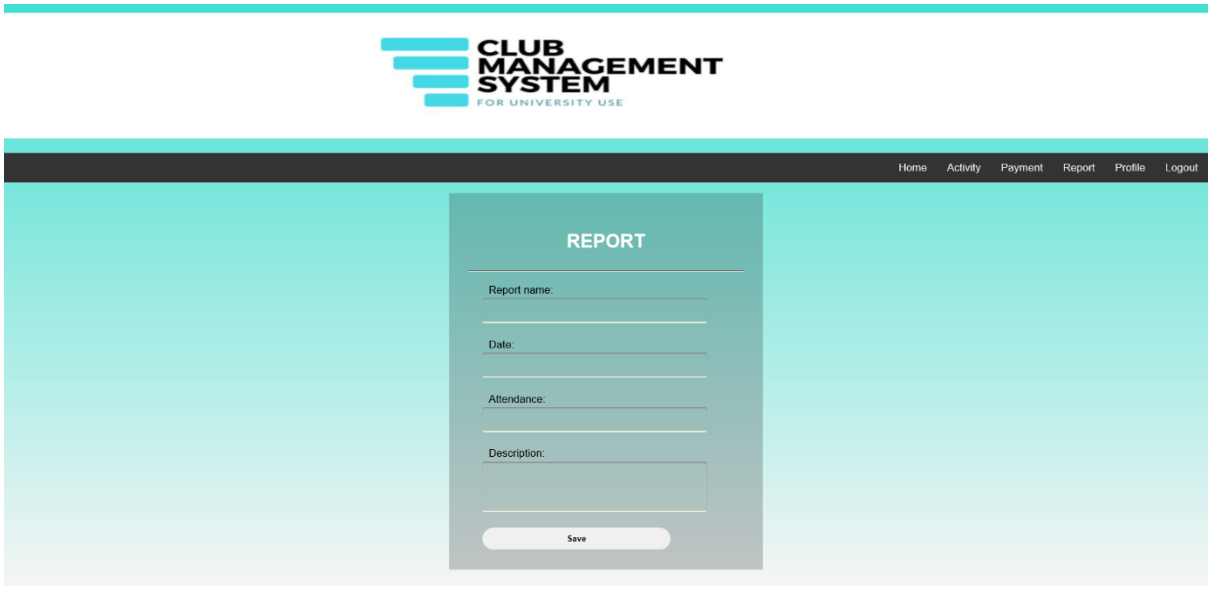

Figure 4.2.6 shows the generated report User Interface. Generate report allows instructors to create a report for certain activities based on the activity name, date, attendance, and description.

4.2.7 View activity (UC7)

Identifier	Use cases	Description
UC7	View activity	1. **Objective** – Member can view the activity for the club that they join 2. **Priority** - Medium 3. **Actors –** Member 4. **Flow of Events** 4.1. **Basic Flow** 4.1.1. Member view activity that provided for the club 5. **Pre-conditions –** Member click view button 6. **Post conditions –** Member view list of activity 7. **Notes –** None

Table 4.2.7: Description for view activity use case (UC7)

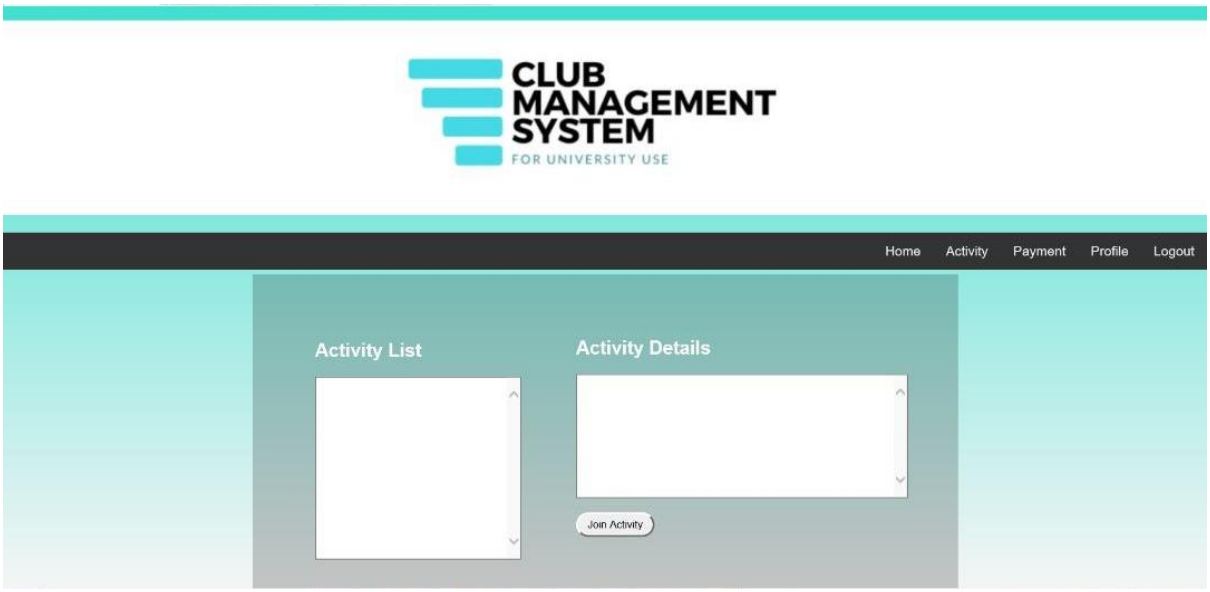

Figure 4.2.7 shows view activity User Interface. View activity interface allows members to view activity of their club including the details of the activity.

4.3 Entity Relationship Diagram

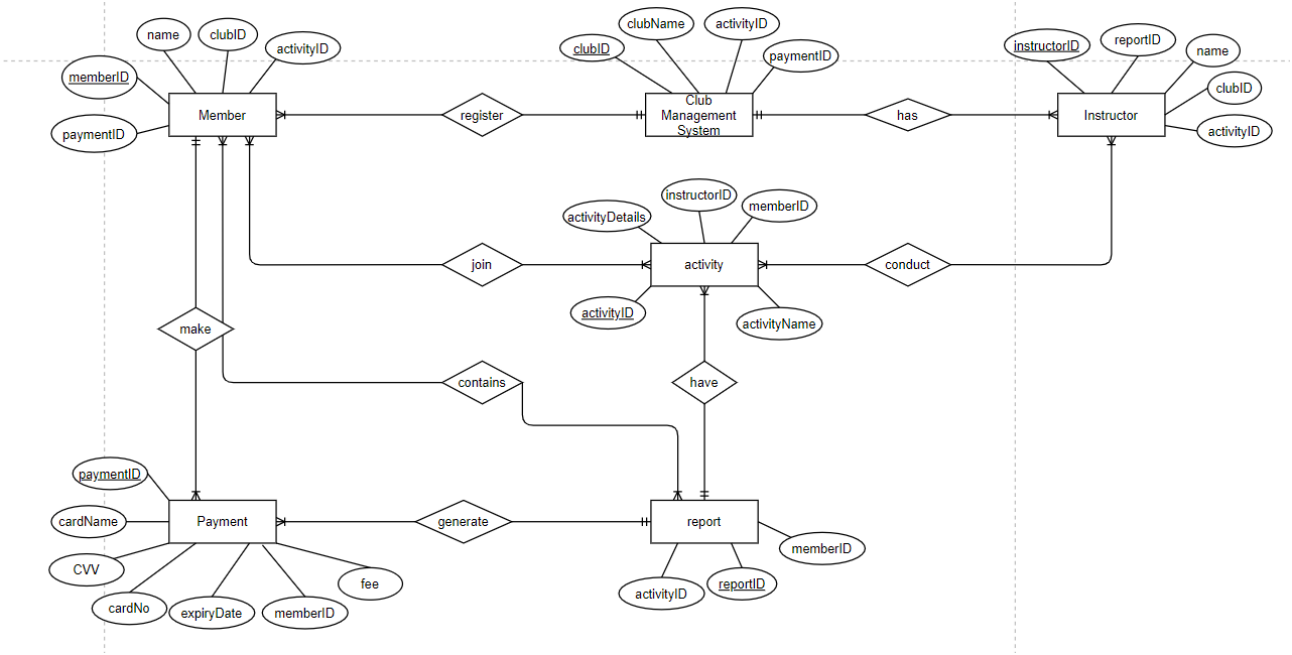

Figure 4.3: Club Management System Entity Relationship Diagram

5. Non-functional Requirements

5.1 Performance Requirements

The application should be portable, and users should be possible to key in the data without facing any technical problem. Since the application will hold many members, the database should be scalable. To use the software for a long time, the capacity of users that can be entered into the system will be large. The number of connections to the system should not slow down the application due to the usage of users considering their location, bandwidth, and latency. The response time for the admin to access the user data in the database should not be more than 5 seconds. The application should be flexible for future enhancements such as the additional form for the user to enter.

5.2 Safety Requirements

The system shall not have a single point of failure that can cause an accident. The requirements that are concerned with possible loss, damage or harm that could affect the system will be back up every day. The system should not allow strangers to access the data that only the admin can access it.

5.3 Security Requirements

The system will ensure the unauthorized access to the system and its data is not allowed. The access permission for the system data may only be charged by the system's admin. To prevent the data loss, all system data must be backed up every 24 hours and the backup copies stored in a secure place. The data also will be encrypted. The application should be password protected by entering the username and password.

5.4 Reliability

For reliability, the system can be used by multiple users concurrently. Any user can access the system even though they use a low-performance PC. The system is also available for 24 hours of the day. The admin can access the data that has been entered by the user as soon as possible. The system will not go down easily and will not wholly be affected by a single application failure.