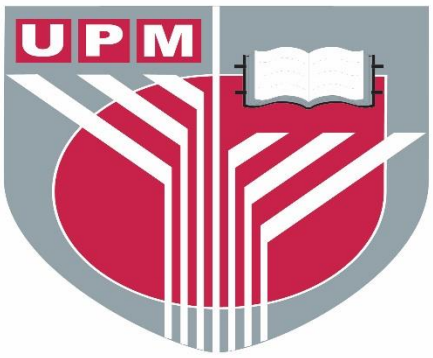

SSE4353

COMPONENT-BASED SOFTWARE DEVELOPMENT

REPORT 1

CLUB MANAGEMENT SYSTEM

LECTURER: DR. NOVIA INDRIATY ADMODISASTRO

TEAM MEMBERS:

MATRIC NUMBER	NAME
191974	NUR FATIN NABILAH BINTI MOHD SHOFE
192628	NUR FAKHIRA BINTI ROSDI
193577	MUHAMAD AMIR HASSAN BIN HAMZAN
194093	MOHAMAD DANIEL EFFENDI BIN ARIFF FADZILAH

Table of Contents

1. Team Member 3

2. Project background 3

2.1. Purpose 3

2.2. Scope 3

User Registration 4

Friendly Booking 4

3. System analysis 5

3.1. User classes and characteristic 5

3.1.1. Operating Environment 5

3.1.2. Design and Implementation Constraints 5

3.1.3. Assumptions and Dependencies 6

3.2. Use case name and identifier 7

3.2.1. Register (UC1) 8

3.2.2. Login (UC2) 9

3.2.3. Update Activity (UC2) 10

3.2.4. Update Profile (UC4) 11

3.2.5. Access payment records (UC5) 12

3.2.6. Generate report (UC6) 14

3.2.7. View activity (UC7) 15

4. Other Non-functional Requirements 16

4.1. Performance Requirements 16

4.2. Safety Requirements 16

4.3. Security Requirements 16

4.4. Software Quality Attributes 16

1. Team Member

Name	Organization
Nur Fatin Nabilah binti Mohd Shofe	Leader
Nur Fakhira binti Rosdi	Designer
Muhamad Amir Hassan bin Hamzan	Programmer
Mohamad Danial Effendi bin Ariff Fadzilah	Programmer

2. Project background

2.1. 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.

2.2. Scope

CMS project provides and manages various club activities such as member registration, registration for various regular and vacation batches and more. The CMS software is a net-built system that manages the entire club activities and provides respective functionality for various types of visitors (for the students specifically). This system is built with respect managing all the clubs offered in Universiti Putra Malaysia (UPM). It allows normal users to avail for club membership, book the ground for desired days and even enrol for various activities in the club that they found interesting. The CMS is built keeping in mind various daily activities of all club and the software automates all these club functionalities for easy operation for the students. There are actually a numerous amount of club and organizations offered in UPM but the students wouldn't enrol because they weren't that happy signing up using the manual way, which is the paper based operations.

Currently, the process of managing the club is file based and manual. This obsolete management system slows down functionality of the club. For example, a new user wants to enrol in a training batch he or she must visit the club and fill up the registration form. The form

then passes through a hierarchy of club members before approval. It takes time as well as effort form a user's perspective. This is just a single case. Same problem persists in all the major operation of the club. To solve this problem, we come up with a system name CMS. It is a web-based application. This system uses Java Server Pages as a programming language. This system interacts with user through the webpage. The scope of the project is to design an interface for student to choose clubs and activities that they want to join. Besides, this project also provide student to register the club they want without queuing and waiting at physical club. The CMS is fully automated and requires just one person from the club to maintain the functionalities of the club. The user can register for new membership, view activity of the club, and view payment history that they made. CMS will eliminate paper oriented, manual processing and queue for club registration. This system also easy to use and saves human efforts and cost. Here are some other things that the project can deliver:-

User Registration

This project takes care of registering a new user for club membership. A user can request for club membership by filling up a form and register to the club.

Friendly Booking

In this system, when the user books clubs for specific day, and the club chosen is full, the system will display another alternative club for user to join. This ensures that there is no negotiation and complain between user.

3. System analysis

3.1. User classes and characteristic

Instructors: They are the core users and can update the activities for the clubs, view the payment made by the members and generate report for members' absentees.
Club members: They can register into the system, can update and view their profile based on their needs and view the activity added by the instructor of their club.

3.1.1. Operating Environment

The software will operate with the following software components and applications:
Hardware
Ram: Minimum of 256 Mb
Software
Operating System Windows XP or above, Mac OS
Browser: Internet Explorer, Mozilla Firefox, Google Chrome or Safari
Other services: PHP 5.4.4, JavaFX

3.1.2. Design and Implementation Constraints

This software is design to manage several students who want to join club in university. The members can choose by themselves what and how many clubs they want to join in a specific time. Each member must register and log into the system as a club member with their username which is student id and password. Each member must keep their password as confidential. They can also change their password to secure it.

3.1.3. Assumptions and Dependencies

Assumption

- Instructor or members of CMS has an active internet connection or has access to one to view the website
- Instructor or members runs an operating system which support internet browsing.
- Instructor or members runs latest browsing software to access our website.

Dependencies

1. User need the connection of internet at minimum of 500Kb per second.
2. User need to run an operating system that support internet browsing.
3. 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.

3.2. Use case name and identifier

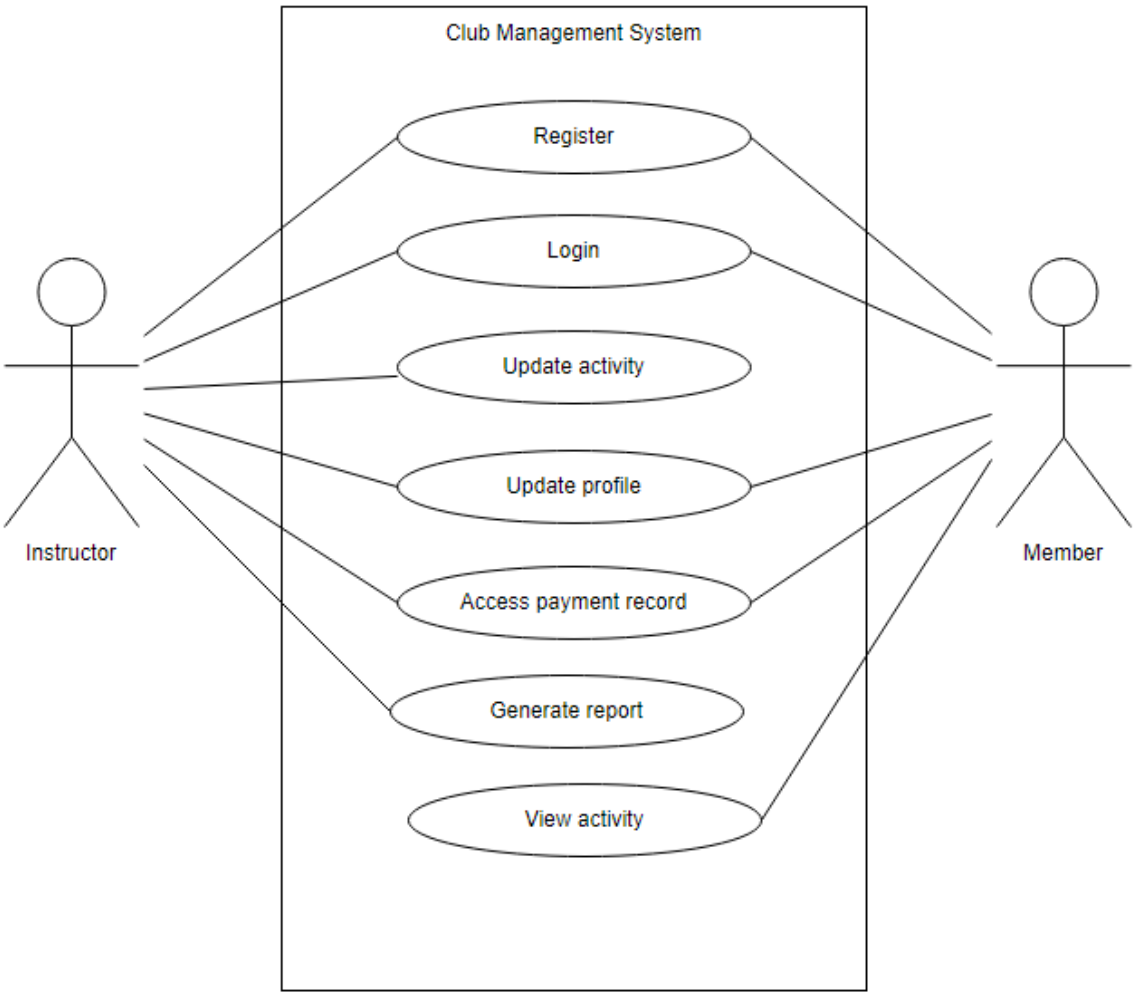

Diagram 3.2 shows the use cases for CMS.

3.2.1. Register (UC1)

1. **Objective** – New member and instructor 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. Member and instructor will fill the registration form.
     - 4.1.2. Their data will be saved in the database of the system.
5. **Preconditions –** Member and instructor need to click register button.
6. **Post conditions –** Member and instructor is successful registered.
7. **Notes –** None

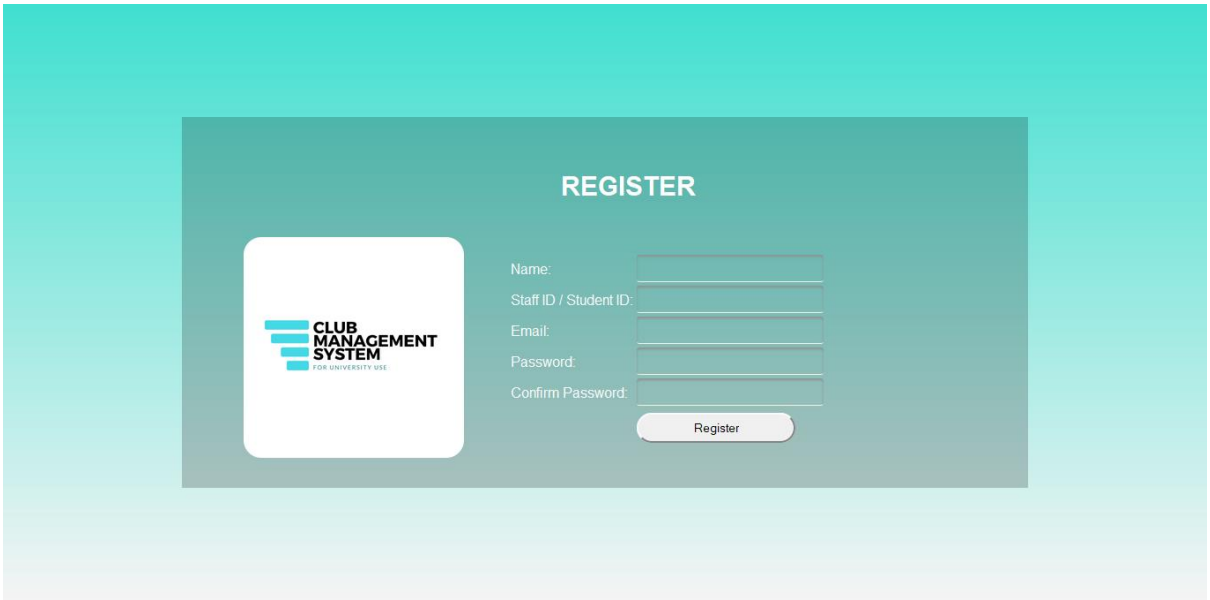

Figure 3.2.1 shows register User Interface for CMS. Register allows instructor and member to register into CMS before using the features in it.

3.2.2. Login (UC2)

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. Member and instructor 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

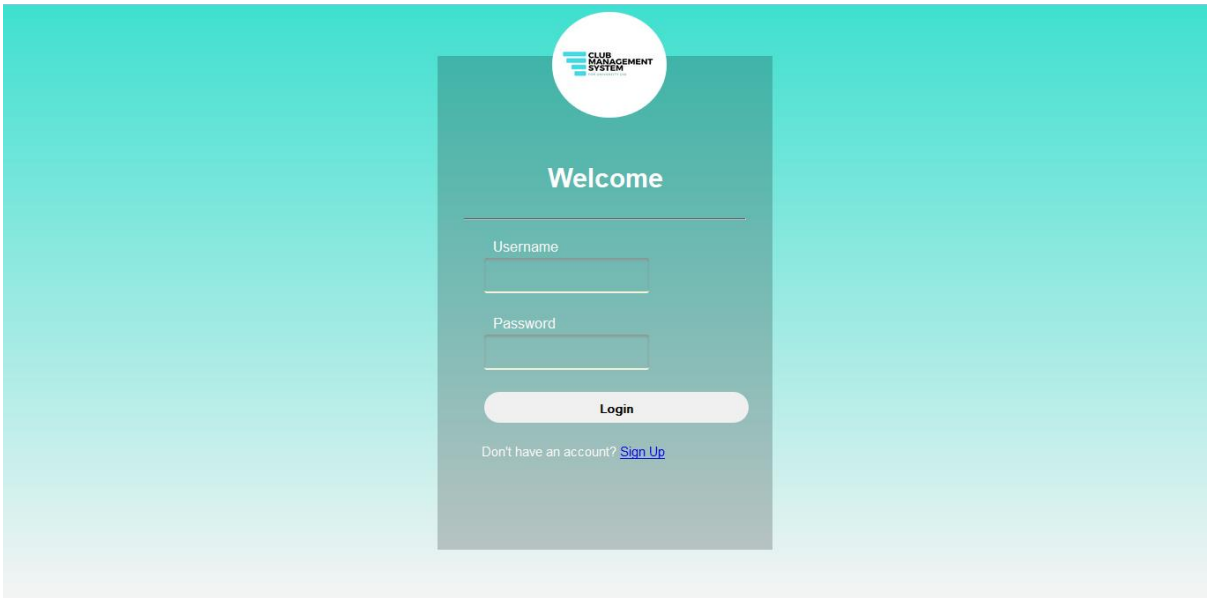

Figure 3.2.2 shows login User Interface for CMS. Member or instructor need to enter their username and password in order to login into the system.

3.2.3. Update Activity (UC2)

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. Instructor update the activities into the system.
5. **Pre-conditions –** Instructor need to login into the system and click update activity button.
6. **Post conditions –** The system will show new activities on the main page
7. **Notes –** None

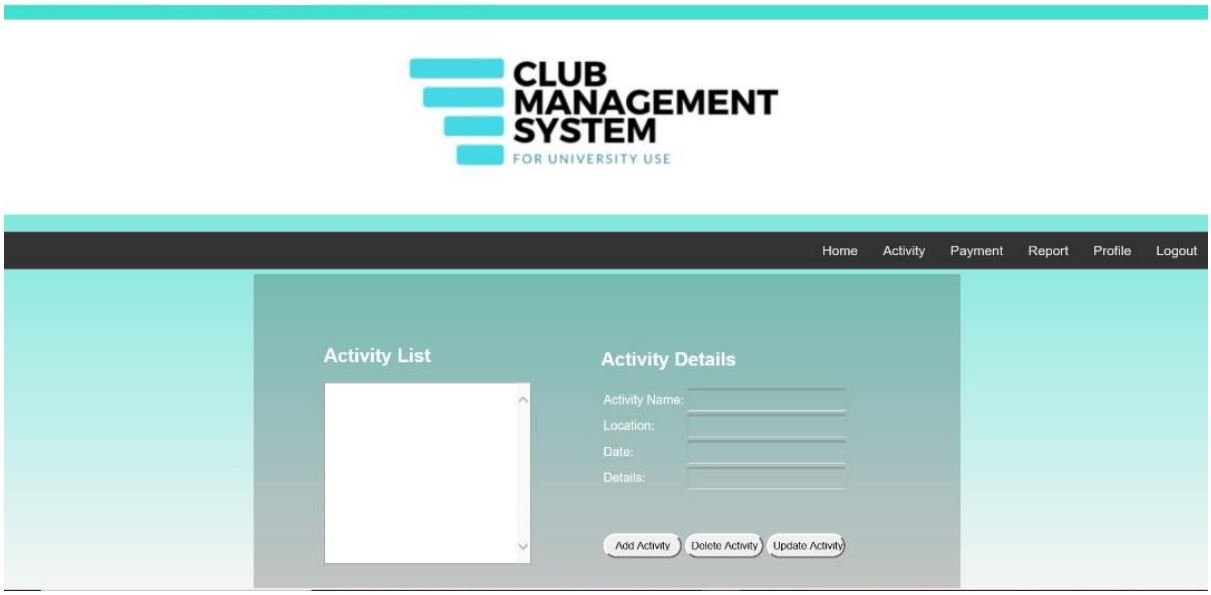

Figure 3.2.3 shows update activity User Interface. Update activity allows instructor to add and update about the activity for the club.

3.2.4. Update Profile (UC4)

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

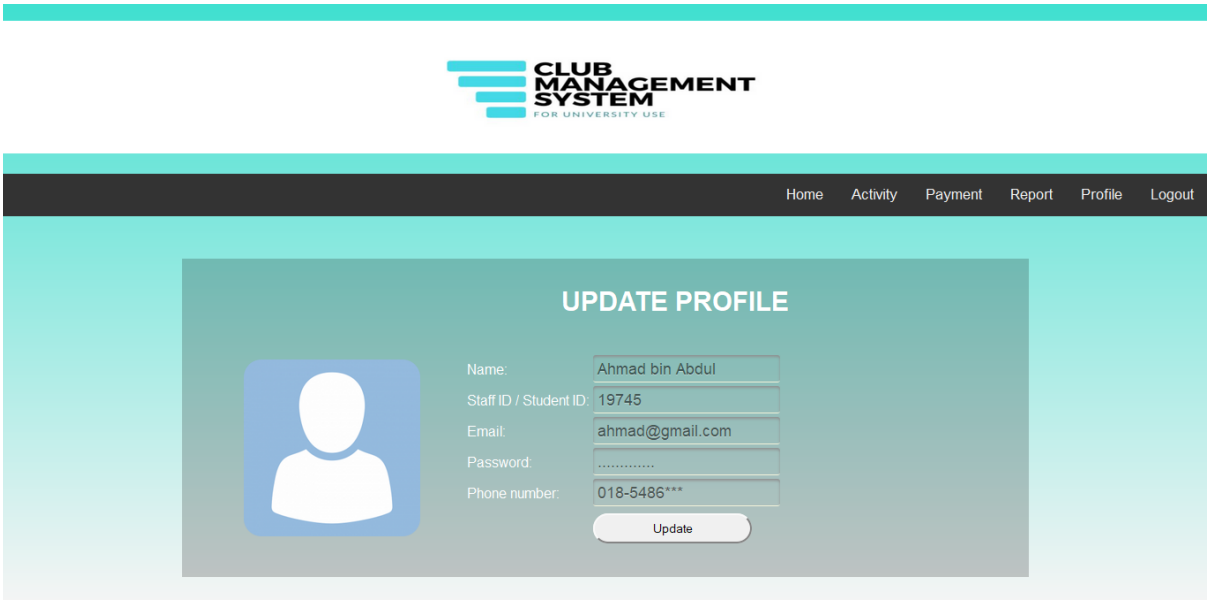

Figure 3.2.3 shows update profile User Interface for CMS. Update profile allows the instructor and member to update their email, password and phone number when it is necessary.

3.2.5. Access payment records (UC5)

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

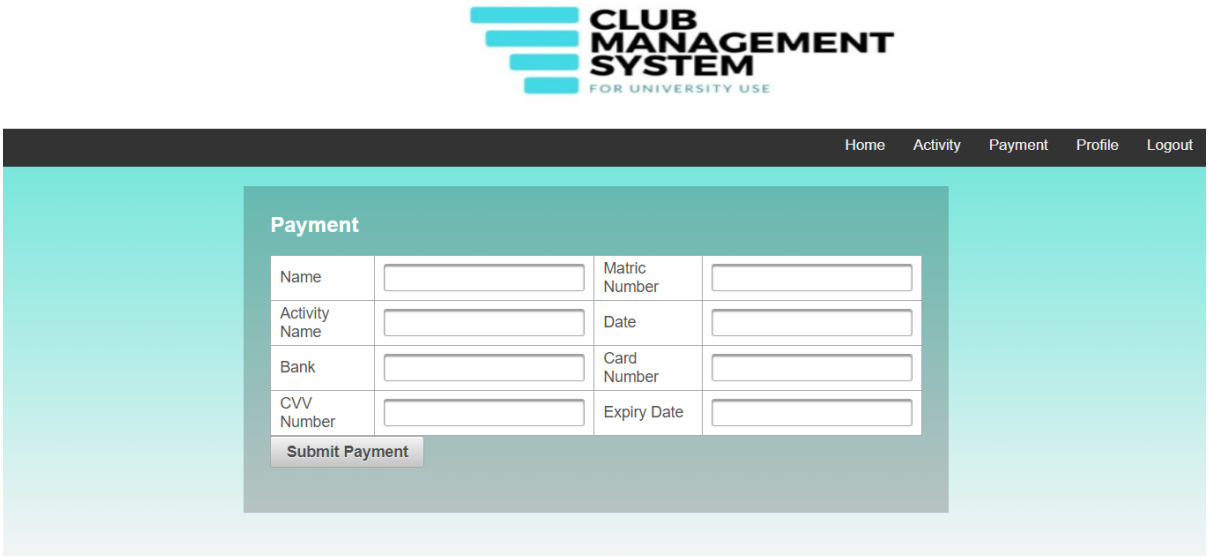

Figure 3.2.5 (1) shows payment User Interface for member. This payment interface allows member to make any payment needed for each activity.

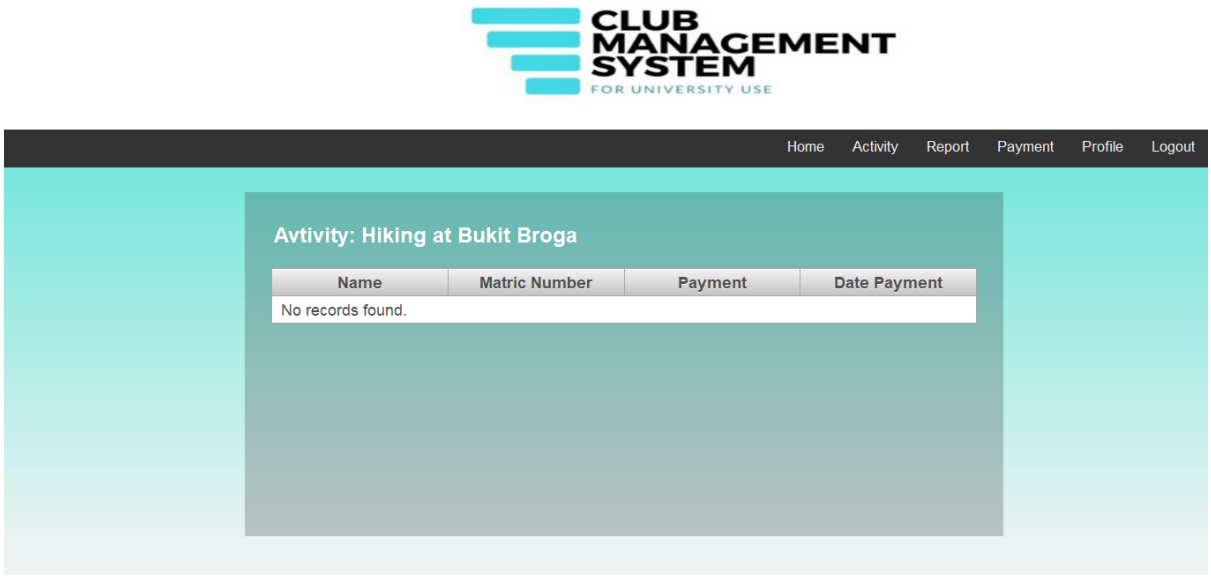

Figure 3.2.5 (2) shows payment record User Interface. Payment record allows instructor to monitor member that make payment for certain activity.

3.2.6. Generate report (UC6)

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

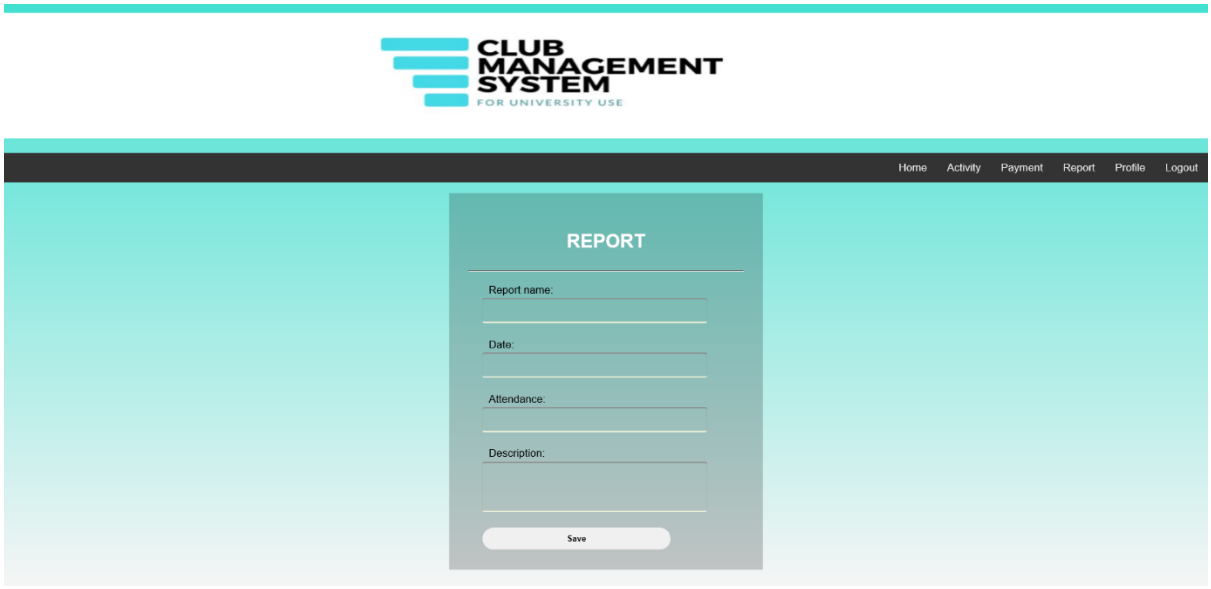

Figure 3.2.6 shows generate report User Interface. Generate report allows instructor to create a report for certain activity based on the activity name, date, attendance and description.

3.2.7. View activity (UC7)

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

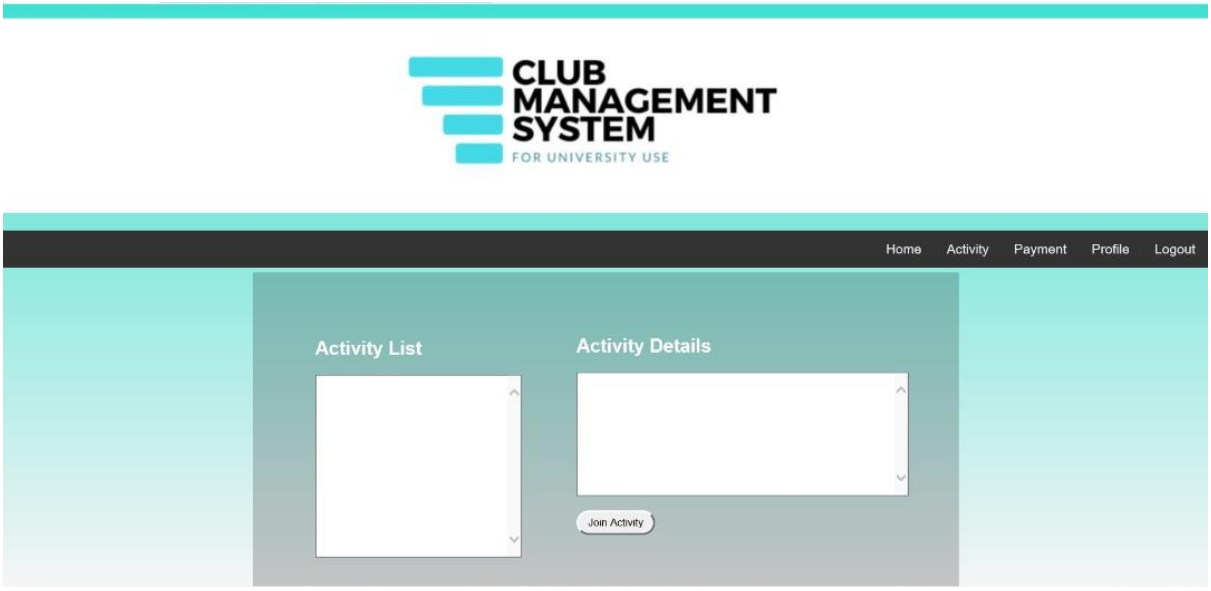

Figure 3.2.7 shows view activity User Interface. View activity interface allows member to view activity of their club including the details of the activity.

4. Other Non-functional Requirements

4.1. Performance Requirements

The application should be portable, and user should be possible to key in the data without facing any technical problem. Since the application will hold large of members, the database should be scalable. In order to use the software for long time, the capacity of user that can be enter 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 to 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 the future enhancements such as the additional form for the user to enter.

4.2. Safety Requirements

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

4.3. 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 charge 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 entered the username and password.

4.4. Software Quality Attributes

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