

SCHOOL CO-CURRICULAR
MANAGEMENT SYSTEM

MUHAMMAD FYRUZ ISMAT BIN AZMI

BACHELOR OF COMPUTER SCIENCE
(COMPUTER SYSTEM &
NETWORKING) WITH HONOURS

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : MUHAMMAD FYRUZ ISMAT BIN AZMI

Date of Birth :

Title : SCHOOL CO-CURRICULAR MANAGEMENT SYSTEM

Academic Session : 2022 / 2023

I declare that this thesis is classified as:

- CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)*
- RESTRICTED (Contains restricted information as specified by the organization where research was done)*
- OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

(Student's Signature)

Muhammad Fyruz Ismat bin Azmi

Date: 02 DECEMBER 2022

(Supervisor's Signature)

Danakorn Nincarean A/L Eh Phone

Date: 23 JUNE 2023

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Perpustakaan Universiti Malaysia Pahang,
Universiti Malaysia Pahang,
Lebuhraya Tun Razak,
26300, Gambang, Kuantan.

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name
Thesis Title

Reasons (i)

 (ii)

 (iii)

Thank you.

Yours faithfully,

(Supervisor's Signature)

Date:

Stamp:

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.



SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and project* and in my opinion, this thesis and project is adequate in terms of scope and quality for the award of the Bachelor's degree in Computer Science (Computer System and Networking)

(Supervisor's Signature)

Full Name : DR Danakorn Nincarean A/L Eh Phon

Position : Senior Lecturer

Date : 23 JUNE 2023



STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

(Student's Signature)

Full Name : MUHAMMAD FYRUZ ISMAT BIN AZMI

ID Number : CA20149

Date : 02 DECEMBER 2022

SCHOOL CO-CURRICULAR MANAGEMENT SYSTEM

MUHAMMAD FYRUZ ISMAT BIN AZMI

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Bachelor Of Computer Science
(Computer System & Networking) With Honors

Faculty of Computing

UNIVERSITI MALAYSIA PAHANG

02 DECEMBER 2022

ACKNOWLEDGEMENTS

First, I would like to express my special gratitude to my supervisor, Ts Dr Danakorn Nincarean A/L Eh Phon for his guidance and support for me to complete my project. The supervision and suggestion guide me to the correct direction on idea generation. It has also enabled me to expand my knowledge and experiences on doing my final year project.

Next, I would like to thank my parent and friends who help me to generate idea on the project. In addition, my friends also contribute to knowledge sharing which help me a lot in completing the proposal.

Finally, I would like to show my gratitude to the project coordinator Dr Danakorn for coordinate my project by giving guidance and encouragement for me to complete the proposal. Moreover, the guidance in proposal writing enables me to understand the correct format in writing the proposal.

ABSTRAK

Sistem Pengurusan Kokurikulum Sekolah ialah penyelesaian termaju yang direka untuk mengautomasikan proses mengesan kehadiran pelajar dalam aktiviti kokurikulum. Kaedah tradisional penyimpanan rekod dan pelaporan sering membawa kepada kesilapan dan ketidaktepatan, menyukarkan sekolah untuk menilai penyertaan pelajar dalam aktiviti ini. Matlamat sistem ini adalah untuk menyediakan penyelesaian yang lebih cekap dan tepat bagi mengesan kehadiran dalam aktiviti kokurikulum. Ia dibangunkan menggunakan teknologi web dan sistem pengurusan pangkalan data, menawarkan antara muka berasaskan web untuk pengguna mengakses dan mengurus maklumat kehadiran. Sistem ini juga menggunakan pangkalan data untuk menyimpan dan mendapatkan semula rekod kehadiran, memastikan data selamat dan mudah diakses. Sistem Pengurusan Kehadiran Kokurikulum Sekolah telah dilaksanakan di Sekolah Kebangsaan Mat Kilau dan mendapat maklum balas yang positif daripada guru dan pelajar. Sistem ini telah mengurangkan dengan ketara masa yang diperlukan untuk mengesan kehadiran dan meningkatkan ketepatan rekod kehadiran. Penyelesaian ini mesra pengguna, selamat dan berskala, menjadikannya pilihan ideal untuk sekolah yang ingin mengautomasikan proses penjejakan kehadiran dan menyokong pembangunan pelajar.

ABSTRACT

The School Co-Curricular Management System is a cutting-edge solution designed to automate tracking student attendance in co-curricular activities. The traditional manual methods of record keeping and reporting often lead to errors and inaccuracies, making it difficult for schools to assess student participation in these activities. This system aims to provide a more efficient and accurate solution for managing co-curricular activities. It is developed using web technologies and database management systems, offering a web-based interface for users to access and manage information. The system also utilises a database to store and retrieve all the data records, ensuring the data is secure and easily accessible. The School Co-Curricular Management System has been implemented in Sekolah Kebangsaan Mat Kilau schools and has received positive feedback from teachers and students. The system has significantly reduced the time required to manage co-curricular data and improved the accuracy of co-curricular data records. This solution is user-friendly, secure, and scalable, making it an ideal choice for schools looking to automate the co-curricular process and support student development.

TABLE OF CONTENT

DECLARATION	
TITLE PAGE	
ACKNOWLEDGEMENTS	ii
ABSTRAK	iii
ABSTRACT	iv
TABLE OF CONTENT	v
LIST OF TABLES	ix
LIST OF FIGURES	x
LIST OF ABBREVIATIONS	xiii
CHAPTER 1 INTRODUCTION	1
1.1 Introduction	1
1.2 Problem Statement	2
1.3 Objective	5
1.4 Scope	5
1.5 Thesis Organization	6
CHAPTER 2 LITERATURE REVIEW	8
2.1 Introduction	8
2.2 Existing Systems/Works	9
2.2.1 GoSchooler	9
2.2.2 QuickSchools	11
2.2.3 Sahar Hassan Web-based Attendance Management System	13

2.3	Analysis/ Comparison of Existing System	15
2.3.1	Analysis of comparison on existing systems.	15
2.3.2	Relevance of comparison with project title	17
2.4	Summary	17
CHAPTER 3 METHODOLOGY		18
3.1	Introduction	18
3.2	Waterfall Model	18
3.2.1	Requirement Analysis Phase	20
3.2.2	Design Phase	20
3.2.3	Implementation Phase	21
3.2.4	Testing Phase	21
3.2.5	Maintenance Phase	22
3.3	Project Requirement	22
3.3.1	Functional Requirements	22
3.3.2	Non-Functional Requirements	22
3.3.3	Limitations and Constraints	23
3.4	Propose Design	24
3.4.1	Flowchart System Design	24
3.4.2	CONTEXT DIAGRAM	26
3.4.3	USE CASE DIAGRAM	27
3.4.4	ACTIVITY DIAGRAM	30
3.4.5	ERD DIAGRAM	32
3.5	Data Design	33
3.6	Proof of Initial Concept	38
3.6.1	Login Interface	38

3.6.2	Admin Dashboard	39
3.6.3	Add Staff Interface	40
3.6.4	Edit Staff Interface	41
3.6.5	Manage Staff Interface	42
3.6.6	Add Student Interface	43
3.6.7	Edit Student Interface	44
3.6.8	Manage Student Interface	45
3.6.9	Pop Up Notification for Add and Edit button	46
3.6.10	Pop Up Notification for Delete button	47
3.6.11	Staff Dashboard Interface	48
3.6.12	Cocurricular Attendance Interface	49
3.6.13	Add Activity Interface	51
3.6.14	Update Activty Interface	52
3.6.15	Manage Activity Interface	53
3.6.16	Add Student Achievement Interface	54
3.6.17	Manage Student Achievement Interface	55
3.6.18	Change Password Interface	56
3.7	Testing/Validation Plan	57
3.7.1	User Acceptance Test (UAT) form for Admin:	57
3.7.2	User Acceptance Test (UAT) form for User:	58
3.8	Potential Use of Proposed Solution	59
3.9	Gantt Chart	60
CHAPTER 4 RESULTS AND DISCUSSION		61
4.1	Introduction	61
4.2	Implementation Process	61

4.2.1	Interface	67
4.2.2	Database	89
4.2.3	Interface Coded	90
4.3	Testing and Result Discussion	99
CHAPTER 5 CONCLUSION		100
5.1	Introduction	100
5.2	Limitation and Constraint	100
5.3	Future Work	101
REFERENCES		102
APPENDIX A		103

LIST OF TABLES

Table 2.1	Comparison of existing systems	15
Table 3.1	Use Case Description of Admin menu	28
Table 3.2	Use Case Description of Instructor menu	29
Table 3.3	The database for the admin.	33
Table 3.4	The database for the Staff	33
Table 3.5	Data dictionary for Student table	34
Table 3.6	Data dictionary for Activity table	35
Table 3.7	Data dictionary for Attendance details table	35
Table 3.8	Data dictionary for Achievement table	36
Table 3.9	Data dictionary for Club table	36
Table 3.10	Data dictionary for Uniform table	37
Table 3.11	Data dictionary for Sport table	37
Table 3.12	User Acceptance Test (UAT) form for Admin	57
Table 3.13	User Acceptance Test (UAT) form for User	58
Table 5.1	User Acceptance Test (UAT) form for User 1	103
Table 5.2	User Acceptance Test (UAT) form for User 2	104
Table 5.3	User Acceptance Test (UAT) form for User 3	105
Table 5.4	User Acceptance Test (UAT) form for User 4	106
Table 5.5	User Acceptance Test (UAT) form for User 5	107
Table 5.6	User Acceptance Test (UAT) form for User 6	108
Table 5.7	User Acceptance Test (UAT) form for User 7	109
Table 5.8	User Acceptance Test (UAT) form for User 8	110
Table 5.9	User Acceptance Test (UAT) form for User 9	111
Table 5.10	User Acceptance Test (UAT) form for User 10	112

LIST OF FIGURES

Figure 1.1	Manual Attendance	3
Figure 1.2	Manual Activity	3
Figure 1.3	Manual Activity	4
Figure 1.4	Manual Attendance and Report	4
Figure 1.5	Thesis Organization	6
Figure 2.1	GoSchooler Dashboard	9
Figure 3.1	Waterfall Model	19
Figure 3.2	Meeting to collect requirements with client	20
Figure 3.3	Admin Flowchart	24
Figure 3.4	Instructor Flowchart	25
Figure 3.5	Context Diagram	26
Figure 3.6	Use Case Diagram for Proposed System	27
Figure 3.7	Admin Activity Diagram	30
Figure 3.8	Instructor Activity Diagram	31
Figure 3.9	ERD Diagram	32
Figure 3.10	Login Page Interface	38
Figure 3.11	Admin Dashboard Interface	39
Figure 3.12	Adding Staff Interface	40
Figure 3.13	Edit Staff Interface	41
Figure 3.14	Manage Staff Interface	42
Figure 3.15	Adding Student Interface	43
Figure 3.16	Edit Student Interface	44
Figure 3.17	Manage Student Interface	45
Figure 3.18	Example Popup Notification for Add User	46
Figure 3.19	Example Pop Up Notification for Delete button	47
Figure 3.20	Staff Dashboard Interface	48
Figure 3.21	Attendance Interface for Cocu	49
Figure 3.22	Adding Activity Interface	51
Figure 3.23	Edit Activity Interface	52
Figure 3.24	Manage Activity Interface	53
Figure 3.25	Student Achievement Details Interface	54
Figure 3.26	List of Student Achievement	55
Figure 3.27	Change Password for Instructor	56

Figure 3.28	Gantt Chart System Development	60
Figure 4.1	XAMPP Control panel	62
Figure 4.2	phpMyAdmin interfaces	62
Figure 4.3	PHP version	63
Figure 4.4	Composer	64
Figure 4.5	Laravel Project command created	64
Figure 4.6	Laravel File	65
Figure 4.7	CMS Laravel Models	66
Figure 4.8	CMS Laravel Controllers	66
Figure 4.9	CMS Laravel Resources	66
Figure 4.10	Login Interface	67
Figure 4.11	Dashboard Interface	67
Figure 4.12	Clubs Interface	68
Figure 4.13	Create Clubs Interface	69
Figure 4.14	Edit and delete Club Interface	69
Figure 4.15	Create Club Activities Interface	70
Figure 4.16	Club List Activities Interface	70
Figure 4.17	Sports Interface	71
Figure 4.18	Create Sports Interface	72
Figure 4.19	Edit and delete Sport Interface	72
Figure 4.20	Create Sport Activities Interface	73
Figure 4.21	Sport List Activities Interface	73
Figure 4.22	Uniforms Interface	74
Figure 4.23	Create Uniforms Interface	75
Figure 4.24	Edit and Delete Uniform Interface	75
Figure 4.25	Create Uniform Activities Interface	76
Figure 4.26	Uniform List Activities Interface	77
Figure 4.27	Attendance Function	77
Figure 4.28	View Attendees Student	78
Figure 4.29	Achievement Interface	79
Figure 4.30	Create Achievement Interface	79
Figure 4.31	Edit and delete Achievement Interface	80
Figure 4.32	Generate Certificate Function	81
Figure 4.33	Achievement Certificate	82
Figure 4.34	Achievement Record Interface	83

Figure 4.35	Create Achievement Record Interface	84
Figure 4.36	User Staff Interface	85
Figure 4.37	Create Staff Interface	85
Figure 4.38	Edit and Delete Staff Interface	86
Figure 4.39	Students Interface	87
Figure 4.40	Create Students Interface	87
Figure 4.41	Edit and Delete Student Interface	88
Figure 4.42	Dark Mode Features	88
Figure 4.43	CMS Database	89
Figure 4.44	Coding for User Resource	90
Figure 4.45	Coding for Student Resource	91
Figure 4.46	Coding for Club Resource	92
Figure 4.47	Coding for Sport Resource	93
Figure 4.48	Coding for Uniform Resource	94
Figure 4.49	Coding for Achievement Resource	95
Figure 4.50	Coding for Achievement Record Resource	96
Figure 4.51	Coding for Activities Resource	97
Figure 4.52	Coding for Attendance Function	98
Figure 4.53	Meeting with Client	99

LIST OF ABBREVIATIONS

CMS	Co-curricular Management System
UAT	User Acceptance Test

CHAPTER 1

INTRODUCTION

1.1 Introduction

There is no debate that education plays an integral part in the growth of a nation, and most advanced economies have implemented educational development to keep up with advances in many sectors. Co curricular education has promoted involvement in economics, culture, and social roles in society, as well as comprehension of a person through enhancing people's awareness of their surroundings.

In education, technology plays numerous functions, including access to educational resources, sharing academic materials, enhancing learning quality, and improving educational management efficiency. Hence, the institution needs to use the latest technology, such as the student co-curricular management system for co-curricular activity.(Fedena, 2021)

Nowadays, co-curricular management systems are essential for every institution; they can determine whether an institution, such as educational institutions, public or private sectors, will be successful in the future. A school co-curricular management system is a system that is used to track attendance and manage activity for extracurricular activities such as sports teams, clubs, and unit-beruniform. The domain background of this subject is rooted in the need for schools to have an effective way of managing and monitoring the participation of students in these activities. With the increasing number of extracurricular programs being offered by schools, it has become increasingly important for schools to have a system in place that allows them to easily track attendance and participation.(Mekari Talenta, 2020)

Based on the student's interest in school, and their union is the largest in the study environment of a university or institution, tracking absence and manage co-curricular data at a department with a significant number of students in a classroom is a challenging and time-consuming process. Furthermore, the process is time-consuming, and the department's personnel spend considerable work to finish the attendance rates for each student. As a result, co-curricular management system is an essential criterion employed for various objectives at many institutions and academic organizations.

1.2 Problem Statement

Co-curricular activities are an important part of a student's overall educational experience, providing opportunities for personal and social development, as well as for pursuing interests and hobbies outside of the classroom. However, tracking attendance for these activities can be challenging, particularly if they take place outside of normal school hours or in different locations. As a result, it can be difficult for teachers and coordinators to accurately track who is participating and to ensure that all students are able to fully benefit from these opportunities. The Teacher typically checks it at the beginning and end of the session. However, it is possible that a teacher will miss someone or that sure pupils will respond many times. Most management attendance is done manually(IceHrm, 2020), with all data gathered on paper and recorded in a register book. This method makes it difficult for teachers to manage student data manually. Furthermore, with a manual process, it will take time for the teachers to write back all the students' names in their register books. When done manually, the data has some duplication.

Furthermore, teachers face difficulties checking and manage co-curricular data based on particular criteria, such as finding the student's name on a given date. This technique needs to be more successful and systematic in dealing with the process of managing activity and attendance records. This method will make it easier and more routine for teachers to use technology.

1. Lack of an efficient and reliable system for tracking attendance and participation in extracurricular activities.

2. Difficulty in keeping accurate records of student participation and progress in extracurricular activities using traditional methods such as manual sign-in sheets that can be time-consuming and prone to errors.
3. Inability to easily generate reports on student participation and progress for teachers and other staff members.
4. Difficulty in monitoring and managing remote or online extracurricular activities.
5. Schools that have a large number of extracurricular activities may find it difficult to keep track of attendance and participation using traditional methods (National Center for Education Statistic, 2017).

No	DAFTAR DIRUMAH	KELOMPOK	WAKTU	DATE
1	AGILA BINTI AZHAR	4	10:00	10/10/2021
2	AGILA BINTI AZHAR	4	10:00	10/10/2021
3	AGILA BINTI AZHAR	4	10:00	10/10/2021
4	AGILA BINTI AZHAR	4	10:00	10/10/2021
5	AGILA BINTI AZHAR	4	10:00	10/10/2021
6	AGILA BINTI AZHAR	4	10:00	10/10/2021
7	AGILA BINTI AZHAR	4	10:00	10/10/2021
8	AGILA BINTI AZHAR	4	10:00	10/10/2021
9	AGILA BINTI AZHAR	4	10:00	10/10/2021
10	AGILA BINTI AZHAR	4	10:00	10/10/2021
11	AGILA BINTI AZHAR	4	10:00	10/10/2021
12	AGILA BINTI AZHAR	4	10:00	10/10/2021
13	AGILA BINTI AZHAR	4	10:00	10/10/2021
14	AGILA BINTI AZHAR	4	10:00	10/10/2021
15	AGILA BINTI AZHAR	4	10:00	10/10/2021
16	AGILA BINTI AZHAR	4	10:00	10/10/2021
17	AGILA BINTI AZHAR	4	10:00	10/10/2021
18	AGILA BINTI AZHAR	4	10:00	10/10/2021
19	AGILA BINTI AZHAR	4	10:00	10/10/2021
20	AGILA BINTI AZHAR	4	10:00	10/10/2021
21	AGILA BINTI AZHAR	4	10:00	10/10/2021
22	AGILA BINTI AZHAR	4	10:00	10/10/2021
23	AGILA BINTI AZHAR	4	10:00	10/10/2021
24	AGILA BINTI AZHAR	4	10:00	10/10/2021
25	AGILA BINTI AZHAR	4	10:00	10/10/2021
26	AGILA BINTI AZHAR	4	10:00	10/10/2021
27	AGILA BINTI AZHAR	4	10:00	10/10/2021
28	AGILA BINTI AZHAR	4	10:00	10/10/2021
29	AGILA BINTI AZHAR	4	10:00	10/10/2021
30	AGILA BINTI AZHAR	4	10:00	10/10/2021
31	AGILA BINTI AZHAR	4	10:00	10/10/2021
32	AGILA BINTI AZHAR	4	10:00	10/10/2021
33	AGILA BINTI AZHAR	4	10:00	10/10/2021
34	AGILA BINTI AZHAR	4	10:00	10/10/2021
35	AGILA BINTI AZHAR	4	10:00	10/10/2021
36	AGILA BINTI AZHAR	4	10:00	10/10/2021
37	AGILA BINTI AZHAR	4	10:00	10/10/2021
38	AGILA BINTI AZHAR	4	10:00	10/10/2021
39	AGILA BINTI AZHAR	4	10:00	10/10/2021
40	AGILA BINTI AZHAR	4	10:00	10/10/2021
41	AGILA BINTI AZHAR	4	10:00	10/10/2021
42	AGILA BINTI AZHAR	4	10:00	10/10/2021
43	AGILA BINTI AZHAR	4	10:00	10/10/2021
44	AGILA BINTI AZHAR	4	10:00	10/10/2021
45	AGILA BINTI AZHAR	4	10:00	10/10/2021
46	AGILA BINTI AZHAR	4	10:00	10/10/2021
47	AGILA BINTI AZHAR	4	10:00	10/10/2021
48	AGILA BINTI AZHAR	4	10:00	10/10/2021
49	AGILA BINTI AZHAR	4	10:00	10/10/2021
50	AGILA BINTI AZHAR	4	10:00	10/10/2021
51	AGILA BINTI AZHAR	4	10:00	10/10/2021
52	AGILA BINTI AZHAR	4	10:00	10/10/2021
53	AGILA BINTI AZHAR	4	10:00	10/10/2021
54	AGILA BINTI AZHAR	4	10:00	10/10/2021
55	AGILA BINTI AZHAR	4	10:00	10/10/2021
56	AGILA BINTI AZHAR	4	10:00	10/10/2021
57	AGILA BINTI AZHAR	4	10:00	10/10/2021
58	AGILA BINTI AZHAR	4	10:00	10/10/2021
59	AGILA BINTI AZHAR	4	10:00	10/10/2021
60	AGILA BINTI AZHAR	4	10:00	10/10/2021
61	AGILA BINTI AZHAR	4	10:00	10/10/2021
62	AGILA BINTI AZHAR	4	10:00	10/10/2021
63	AGILA BINTI AZHAR	4	10:00	10/10/2021
64	AGILA BINTI AZHAR	4	10:00	10/10/2021
65	AGILA BINTI AZHAR	4	10:00	10/10/2021
66	AGILA BINTI AZHAR	4	10:00	10/10/2021
67	AGILA BINTI AZHAR	4	10:00	10/10/2021
68	AGILA BINTI AZHAR	4	10:00	10/10/2021
69	AGILA BINTI AZHAR	4	10:00	10/10/2021
70	AGILA BINTI AZHAR	4	10:00	10/10/2021
71	AGILA BINTI AZHAR	4	10:00	10/10/2021
72	AGILA BINTI AZHAR	4	10:00	10/10/2021
73	AGILA BINTI AZHAR	4	10:00	10/10/2021
74	AGILA BINTI AZHAR	4	10:00	10/10/2021
75	AGILA BINTI AZHAR	4	10:00	10/10/2021
76	AGILA BINTI AZHAR	4	10:00	10/10/2021
77	AGILA BINTI AZHAR	4	10:00	10/10/2021
78	AGILA BINTI AZHAR	4	10:00	10/10/2021
79	AGILA BINTI AZHAR	4	10:00	10/10/2021
80	AGILA BINTI AZHAR	4	10:00	10/10/2021
81	AGILA BINTI AZHAR	4	10:00	10/10/2021
82	AGILA BINTI AZHAR	4	10:00	10/10/2021
83	AGILA BINTI AZHAR	4	10:00	10/10/2021
84	AGILA BINTI AZHAR	4	10:00	10/10/2021
85	AGILA BINTI AZHAR	4	10:00	10/10/2021
86	AGILA BINTI AZHAR	4	10:00	10/10/2021
87	AGILA BINTI AZHAR	4	10:00	10/10/2021
88	AGILA BINTI AZHAR	4	10:00	10/10/2021
89	AGILA BINTI AZHAR	4	10:00	10/10/2021
90	AGILA BINTI AZHAR	4	10:00	10/10/2021
91	AGILA BINTI AZHAR	4	10:00	10/10/2021
92	AGILA BINTI AZHAR	4	10:00	10/10/2021
93	AGILA BINTI AZHAR	4	10:00	10/10/2021
94	AGILA BINTI AZHAR	4	10:00	10/10/2021
95	AGILA BINTI AZHAR	4	10:00	10/10/2021
96	AGILA BINTI AZHAR	4	10:00	10/10/2021
97	AGILA BINTI AZHAR	4	10:00	10/10/2021
98	AGILA BINTI AZHAR	4	10:00	10/10/2021
99	AGILA BINTI AZHAR	4	10:00	10/10/2021
100	AGILA BINTI AZHAR	4	10:00	10/10/2021

Figure 1.1 Manual Attendance



Figure 1.2 Manual Activity



Figure 1.3 Manual Activity

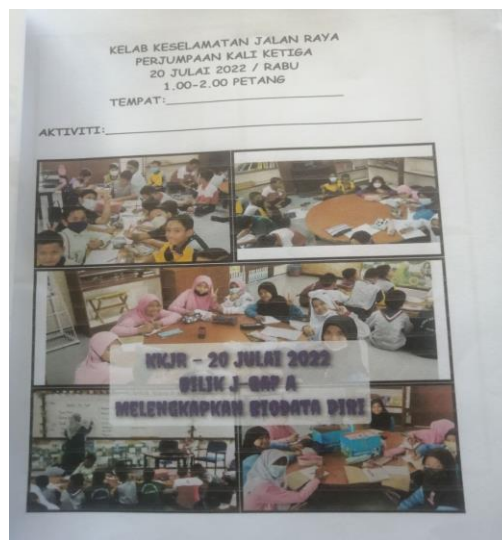


Figure 1.4 Manual Attendance and Report

Figures 1.1–1.4 depict manual attendance and reports provided by Client Sekolah Kebangsaan Mat Kilau. To tackle these issues, an effective Web-based attendance management system is designed to track student co-curricular activity in school. This application electronically records attendance and stores the records in a database.

As a result, Co-curricular activities are important for a student's personal and social development but tracking attendance can be challenging. Manual attendance tracking methods, such as using a register book, can be time-consuming and prone to errors. A web-based attendance management system can help to more accurately and efficiently

track attendance for co-curricular activities and can also make it easier to access and analyze attendance data. This system can help administrators and teachers more easily manage attendance data and reduce the loss of instructional time. It can also provide more accurate student statistical information.

1.3 Objective

The objectives of this project are:

1. To study the requirement needed from the existing student co-curricular management system.
2. To develop a school co-curricular management system for managing the co-curricular attendance, activities and achievement.
3. To evaluate the functionality of the developed student co-curricular management system.

1.4 Scope

The scope of this project:

- i. Platform
 - This project will be a web-based system that school teachers may access through any browser.
- ii. Development / Functionality
 - The project can manage student attendance and activities every week of cocurriular days.
 - The information of the student is viewable on mobile or web devices.
- iii. User
 - Instructors are the intended users of this web-based system.

iv. System

- This system provides a method for adjusting account settings such as passwords. The system will store the data of students and information. Provide the list of student attendance, student achievement, and club activities. Providing an interface to add or delete student data to Admin.

1.5 Thesis Organization

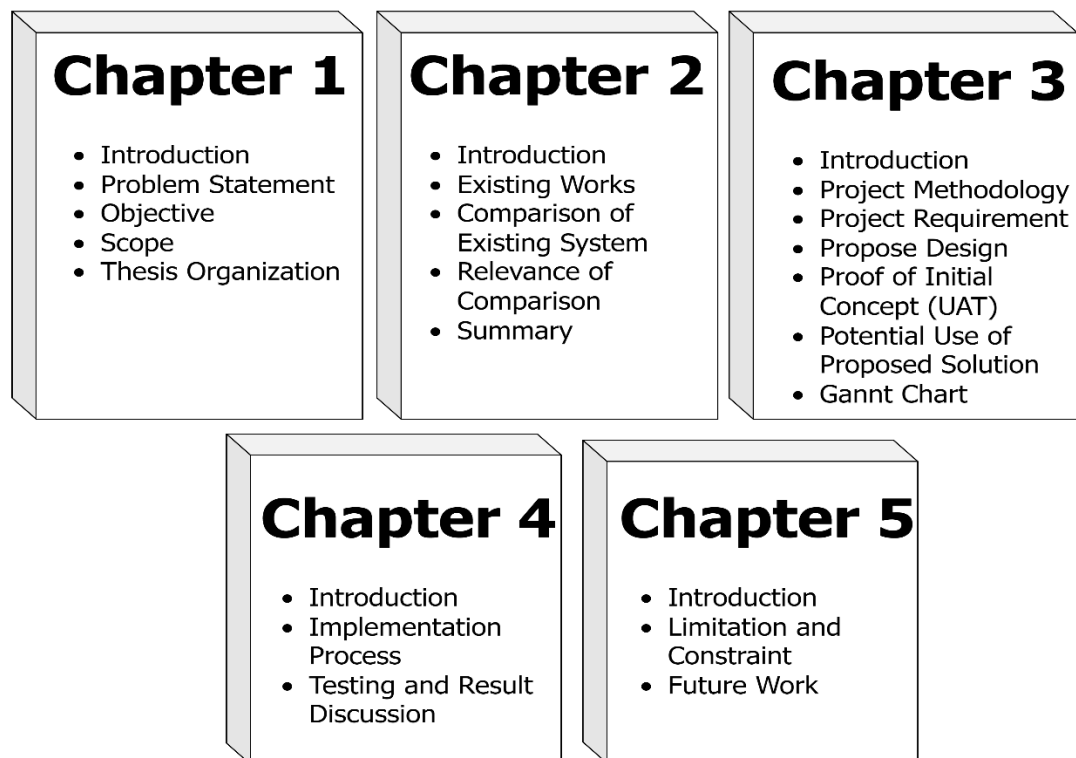


Figure 1.5 Thesis Organization

The Figure 1.5 shows the thesis organization for this paper that will be divided into three chapters: introduction, literature review, and methodology. The first chapter discusses the project by explaining the present state. Problems and troubles include a problem description, objectives, project scope, and importance.

The second chapter includes a discussion subject concerning the literature's opinion of an existing problem with an old type of irrigation system, as well as a comparison of the current project's answers to those of a similar past project.

Chapter 3 reviews the methods employed in this study. The research examines the project's need, data design, and client testing. It also reviews the system prototype before development. Additionally, a Gantt chart is included to illustrate the development timeline.

Chapter 4 presents the implementation of the Co-curricular Management System, covering aspects such as the interface, functionality, database, and system coding. The chapter also includes the testing results for the system and a discussion with the client from SK Mat Kilau.

Finally, Chapter 5 focuses on the limitations and constraints encountered during the development of the Co-curricular Management System. It discusses the challenges and constraints that may have impacted the system's development and implementation. Additionally, the chapter explores potential areas for future work and improvements to enhance the design.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter briefly describes the research analysis of current methods related to the “Student Co-curricular Management System” to be planned in the future. This chapter defines a comprehensive review of recent related systems. This chapter's most important thing to understand is how to use the system efficiently. Furthermore, the software and hardware methodology in constructing the database and the overall project must be clarified.

Student Co-Curricular Management Systems are sub-disciplines of institutional management that focus on specific issues faced by teachers in professional schools. Keeping track of student attendance has always been a problematic issue (Enggagge EH, 2020). It is typical to observe teachers taking students' attendance at the start of each school day, and sometimes even many times during the day, at schools, universities, and educational institutions worldwide. Several systems for recording student presence have been developed throughout the years.

The system review will include three existing systems: Quickschool, GoSchooler, and Sahar Hasan. This chapter reviews current systems and compares them to understand the present scenario better and identify the market's requirements.

2.2 Existing Systems/Works

2.2.1 GoSchooler

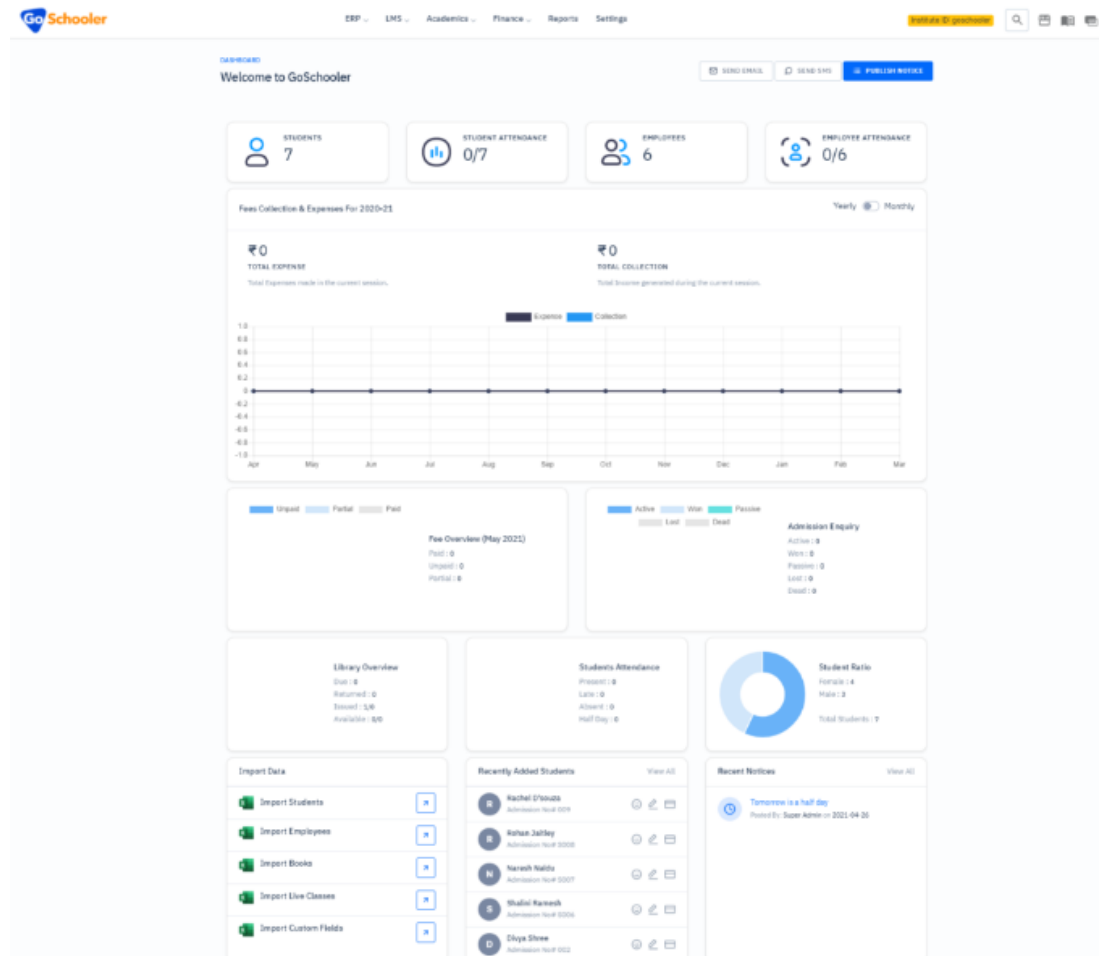


Figure 2.1 GoSchooler Dashboard

Figure 2.1 shows the GoSchooler Dashboard for GOSchool ERP. GoSchooler is the main product of KokSamLai Solutions, a school resource management software. It is an essential technological solution designed exclusively for schools to help with school management. It uses cutting-edge technology to provide the school's stakeholders with fresh managerial experience. GOSchool CMS directly administers a dynamic website, giving a mechanism for personalizing content for all stakeholders. With their CMS created exclusively for independent schools, the public website and internal portals will have a uniform appearance and aesthetic. eLearning powered by GOSchool is an online portal that solely provides information for educational purposes. This platform aggregates

material from several sources and allows students to quickly download and read online content relevant to their class and curriculum (Naresh Narra, 2020).

2.2.1.1 Features

GoSchooler gives the user a variety of functions, such as the ability to search for information by clicking the navigation bar at the top of the system that they desire based on categories. GoSchooler separated the users into two groups: students and teachers. The user can easily manage school data, such as student attendance, and view essential details. The user may safely save all student attendance information, statistics, and report cards while having simple access to data-driven reports, progress checks, and other features. Instead of tracking class attendance, the user may utilize GoSchooler's school ERP software to coordinate all courses, online tests, lectures, and scheduling.

2.2.1.2 Advantages and Disadvantages

The first advantage of the GoSchool system website is that it allows for quick and easy contact between parents, students, and teachers via the online or mobile app. The second benefit is that the user interface is user-friendly and easy to browse since it is similar to other websites, which causes the user to struggle to use the website. Users will also find it appealing due to its simplicity and ease of usage. Furthermore, this system cannot monitor consoles and other unsupported devices, such as Apple devices only compatible with Android devices and other devices that support it.

2.2.2 QuickSchools

STUDENTS	PRESENT	ABSENT	TARDY	ABSENT W/ EXCUSE	TARDY W/ EXCUSE	REMARKS
1ST GRADE (HOMEROOM: MS. DAVID)						
Alvin DeSilva	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Amber Phillips	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Felicity Bradshaw	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Gavin Parker	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	

Figure 2.2 QuickSchools Attendance Form

Sign in to your school

Enter your school's QuickSchools url.

.quickschools.com

[Continue](#)

[Don't remember your school's URL?](#)

Figure 2.3 QuickSchools Login Interface

Figure 2.2 shows QuickSchools Attendance Form. QuickSchools operates entirely differently than existing student attendance and information management systems. The user does not install hardware and software or maintain backups. QuickSchools handles everything on our end. Link to your school's QuickSchools.com website over the Internet. The Figure 2.3 shows login interface, After login to the system the user is ready to go. QuickSchools can grant users access to a sample account in minutes by filling out a simple form. QuickSchools is a clever tool for consistently and conveniently controlling student information and attendance. QuickSchools makes it

easier to manage student attendance at the organization. QuickSchools features both a smartphone app and an online version. Parents and teachers may use the smartphone application to track their daily attendance and other tasks (QuickSchools, 2020).

2.2.2.1 Features

The search feature is accessible on the QuickSchools software and application system, where the user may input the student name or student id, and the website will display the results depending on the user's search. Users are safe from threats as its security is proven the best for people's data. Parents who use QuickSchools can log in and access their children's information at any time via the Parent Portal. Parents may monitor their children's attendance, workload, and advancement. QuickSchools' feature report card is completely customizable. Users can customize their grading criteria, grading method, and report card formality. They may also print beautiful report cards and transcripts. Like other competitors, the website's user interface is incredibly user-friendly and consistent.

2.2.2.2 Advantages and Disadvantages

The first benefit of the QuickSchools website is that it is available on web-based and mobile platforms, making it easy to use from anywhere at any time over the internet. The website was created following the most recent design concept, minimalist design. Using the notion implies that the website's performance is satisfactory for a corporation new to using the web-based system. Users will also find it appealing due to its simplicity and ease of usage. Even though the website provides a wide range of features, QuickSchools needed new technology features for the user, such as facial recognition, RFID, or any biometric devices.

2.2.3 Sahar Hassan Web-based Attendance Management System

Sahar Hassan Attendance Management System is a dynamic and adaptable web-based solution. The figure 2.4 shows Dashboard Sahar Hassan Attendance Management System. It is accessible from any computer, regardless of location. The goal is to provide a web-based attendance program for the ICIT department to record student information such as topics, professors, and relevant fields. The attendance report will be created automatically, without human intervention, and will be error-free. The system's goal is to eliminate redundant data input and inaccuracies in time and attendance records, as well as to eliminate paperwork and save time. The system connects with a distant server's database. It computes the attendance percentage of pupils automatically, without any human paper-based effort. The solution assists end users with interactive design and automated attendance management processes (Hassan & Zubair Asghar, 2015).



Figure 2.4 Dashboard Sahar Hassan Attendance Management System

Student Name	DOB	Gender	Email	Phone	Address	Session	Program	Semester	Action
Jamil	1990-12-23	male	jamil123@yahoo.com	03458318654	house#12-street#6-Golshan Hameed colony-DIK-KBK-Pak	2011-2015	Bcs	2nd	Update Delete
Ali	1992-08-06	male	ali32@gmail.com	03458310978	House#21-Street#5-Behria Town-Isb	2011-2015	Mphil	1st	Update Delete
Farah	1992-12-03	female	farah43@gmail.com	03138575768	House#3-street#21-DIK-KFK-Pak	2013-2015	MCS	2nd	Update Delete

Figure 2.5 Sahar Hassan Attendance Management System Student Data

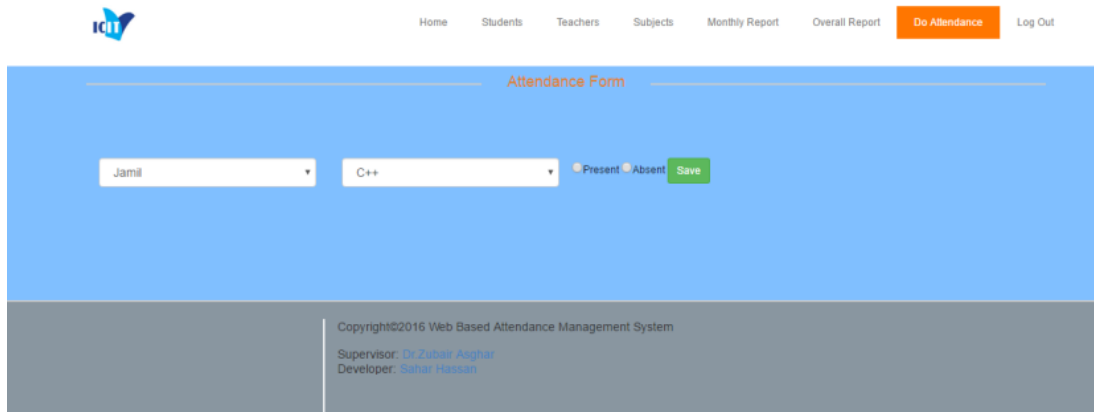


Figure 2.6 Sahar Hassan Attendance Management System Form

2.2.3.1 Features

This system may filter the results in rows and columns based on the Student's Name, DOB, Gender, Email, Phone, Address, Session, Program, and Semester. The user can access a list of table records for a registered student. The name and software are automatically obtained from the database. If the student was there, the present check box is also selected. If the student was absent, the absent check box is selected instead of the current check box. Information is saved by clicking the save button. The navigation at the top of the screen makes it easier for the user to operate and utilize the system. The website's user experience is straightforward and consistent.

2.2.3.2 Advantages and Disadvantages

The benefit of this system is that it is simple to use since the interface is similar to other websites, making the user struggle to use it. The drawbacks of this website are that the design is too plain and could be more to include something more intriguing to the webpage. The learner can also not only view formation data they have registered.

2.3 Analysis/ Comparison of Existing System

2.3.1 Analysis of comparison on existing systems.

This comparison was produced after researching each of the selected existing systems to differentiate the criteria of three current attendance management systems. Each strategy has advantages and disadvantages. Table 2.1 shows the comparison of the three current methods chosen for the literature study and the proposed system are summarised below.

Table 2.1 Comparison of existing systems

Systems	System 1	System 2	System 3	New System
Platform	-Mobile and web based	-Mobile and web based	-Web-based	-Web-based
Features	-Can filter the search result -User-friendly user interface -Store all student information, data, and report cards securely with easy access to data	-Can filter the search result -User-friendly user interface -Store all student information, data, and report cards securely with easy access to data	-Can filter the search result -Easy and straightforward to use -Store all student information, data, and report cards securely with easy access to data	-Can filter the Search result -User-friendly user interface -Store all student information, data, and report cards securely with easy access to data

	-Use RFID and Biometric systems			
Advantage	<p>-Uses minimalist and the latest design</p> <p>-Advanced performance monitoring feature.</p> <p>-User friendly</p> <p>-Provides both mobile and web versions.</p>	<p>-Capability to support various device types.</p> <p>-Uses minimalist and the latest design</p> <p>-Advanced performance monitoring feature.</p> <p>-User friendly</p> <p>-Provides both mobile and web version.</p>	<p>-Capability to support various device types.</p> <p>-User friendly</p>	<p>-Capability to support various device types.</p> <p>-Uses minimalist and the latest design</p> <p>-Advanced performance monitoring feature.</p> <p>-User friendly</p>
Disadvantage	-Cannot support various types of consoles and any other devices.	-There are no technological features such as facial recognition, RFID, or biometric devices.	-website design needs to be more complex.	-There are no technological features such as facial recognition, RFID, or biometric devices.

2.3.2 Relevance of comparison with project title

Compared to the three existing systems, the new suggested system will be developed primarily employing web-based. These systems may be accessed from any computer through the internet rather than having to be installed on each system from which you wish to use them. The method may also filter search results. It will make it easy for the user to track the student data by searching their name or anything related to the student, such as matric id. The user is likewise concerned about the user interface of the system they are using. A user-friendly user interface will entice users to return to the app and use it more frequently.

Furthermore, all current systems can securely store all student information, data, and report cards with quick access to data. RFID and biometric devices, which are new in this period, can be used in the GoSchool systems. The present system has an Advanced Performance monitoring function that will be employed for this proposed solution. The proposed plan would feature a simple, cutting-edge design that will entice users to use the system.

Because of each current system's many features and specialties, this comparison positively influences the title of School co-curricular management project. From the comparison, will help to improve the project/system in the future.

2.4 Summary

According to the literature review in Chapter 2, the user is mainly concerned with the filter feature while using the search function so that they may examine the results depending on their preferences, such as student name, staff id, and attendance update. The school cocurricular management system developed is suitable for future technology.

In a nutshell, this chapter shows the differences between the three existing systems reviewed. Each of the websites has its pros and cons and its attractive features. The future system will use advantage of three different systems and add more functions to build the system that can bring more efficiency than others.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter will explain the project methodology and define the project development procedure. The system development methodology is a framework that guides the developer to structure, plan, and methodically create the system. This chapter describes the system development process from beginning to end. There are five phases: requirement, design, implementation, testing, and maintenance. It will also discuss developing the Co-curricular Management System at each phase. This system will be transformed into a prototype during the development phase. The system prototype development process is divided into requirements and design.

3.2 Waterfall Model

The methodology is crucial in system development since it reduces the number of faults and delays during system deployment by providing a clear foundation for the developer to create a system. A well-chosen methodology allows the developer to create a solid plan, make better time and budget estimates, and offer explicit knowledge of the system to the client, developers, and end users.

Waterfall Model

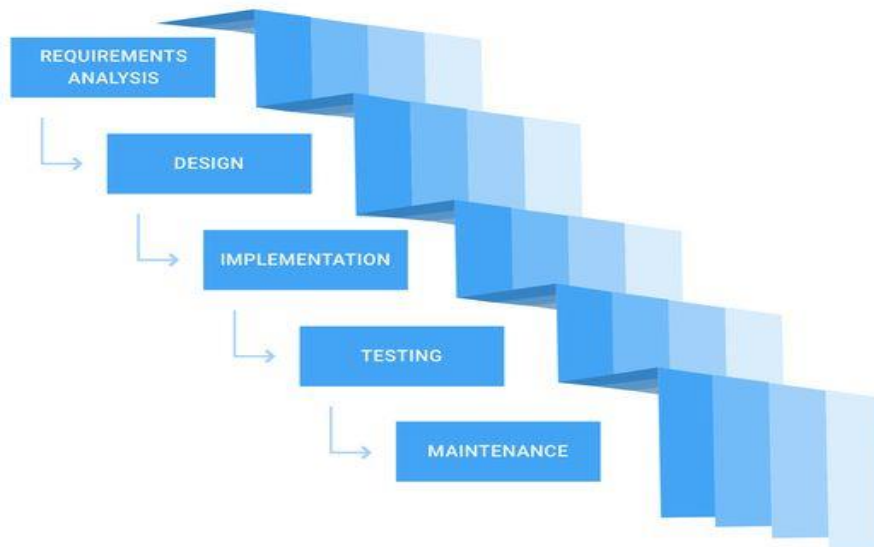


Figure 3.1 Waterfall Model

Figure 3.1 shows waterfall model phase. The Waterfall model was the very first SDLC approach used in software development. It is simple to grasp and use. The waterfall Model depicts software development as a sequential, linear process. There is no overlap between phases in a waterfall model, and each phase must be completed before the next step can begin. This means that each stage in the development process may start only after the one before it has been completed. In this waterfall style, the phases do not overlap (Rob Watts, 2022).

The Waterfall model is appropriate for this project since it has a short completion time. This technique is simple to administer since each phase has a specific product and a review procedure. Because this is a small project with well-defined requirements, the Waterfall methodology is appropriate for project development.

3.2.1 Requirement Analysis Phase

The project's goals, expectations, timelines, and possible challenges must all be stated and recognized during this requirement planning phase. Before beginning the design process, the co-curricular management system purpose and functional and non-functional requirements for this project will be determined. The requirements collecting step precedes the development phase and provides a thorough grasp of the system.

The school co-curricular management system is the proposed project. The system's responsibility is to track and record student co-curricular activities, attendance, every club, and achievements. The project requirements are collected at this phase. After determining the need, the strategy for the subsequent stages may be designed appropriately. The Software Requirement Specification will usually include all of the requirements. The requirement for the proposed system is getting from the potential user which is a Client from Sekolah Kebangsaan Mat Kilau.



Figure 3.2 Meeting to collect requirements with client

3.2.2 Design Phase

Following the collection of system requirements, the design phase of Sekolah Kebangsaan Mat Kilau school co-curricular management system will begin, during which

system models will be constructed. System design and interface design will be carried out. Flowcharts, use case diagrams, and context diagrams are used in system design to show how the system interacts with the end user. This system will be designed to meet the specific needs and requirements of Sekolah Kebangsaan Mat Kilau school. It will be tailored to the school's unique structure, policies, and procedures. It will be used by the school's administrators and teachers to track attendance in co-curricular activities, view reports, and manage databases. It will also use flowcharts, use case diagrams, and context diagrams to show how the system interacts with the end user, with the flowchart depicting the process of receiving attendance data, deciding based on the threshold level, issuing alerts, and storing data. The use case diagram will show the interaction between the instructor and the student, who can perform tasks such as viewing reports, managing databases, and assigning tasks. The interface design will be developed using storyboards to depict the navigation in the system.

3.2.3 Implementation Phase

Sekolah Kebangsaan Mat Kilau school co-curricular attendance management system system development, coding, and system testing are among the tasks to be completed. The outputs from the analysis and design phases are performed to develop the project during the implementation phase. Throughout the implementation phase, we also create the project's source code by referring to the needs and specifications specified during the previous steps. Furthermore, any updates or improvements to the system will be implemented during this time.

3.2.4 Testing Phase

The deployment phase is when the system's development is finished and ready for launch. The testing phase is when we do software verification and validation to ensure that the project meets all of the requirements agreed upon in earlier stages (Amna Zulqadar, 2022). The performance and usefulness of the Sekolah Kebangsaan Mat Kilau school co-curricular management system will be tested by the Sekolah Kebangsaan Mat Kilau teacher and staff member. Its purpose is to ensure that the system satisfies the criteria and conforms to the goals and scope of the school co-curricular management system. Actions such as code repair and augmentation will be carried out during the phase devoted to reducing the error rate of the proposed system. The system is ready for launch after testing is completed; however, in this project, testing will be done on the system prototype.

3.2.5 Maintenance Phase

The software maintenance phase of a school co-curricular management system is an ongoing process that involves updating, modifying, and upgrading the system to keep it current with user demands and to ensure its proper functioning (Amna Zulqadar, 2022). For Sekolah Kebangsaan Mat Kilau, this means regularly patching, upgrading and testing the system for errors and correcting them, as well as modifying the system to accommodate changes in user demands or requirements. Software maintenance is crucial for the effective operation of the system and for ensuring that the system remains functional and effective in tracking attendance and participation in extracurricular activities.

3.3 Project Requirement

3.3.1 Functional Requirements

- i. The system shall allow users to view the student status.
- ii. The system should allow the users to login and register.
- iii. The system shall allow users to search for sorting.
- iv. The system should display the list of students and student clubs.
- v. The server must back up the database automatically once a week.
- vi. The system shall authenticate a user when a user tries to log into the system.

3.3.2 Non-Functional Requirements

- i. Performance
The system should take at most 10 seconds to load the result.
- ii. Security
To ensure that sensitive user data is protected, the system must employ adequate security measures such as session and password hash functions.
- iii. Compatibility
The system may be used in any browser on any device, independent of the operating system.

3.3.3 Limitations and Constraints

The constraint of this project is that it will only be possible to collect attendance data if the network connection becomes reliable or is interrupted. If a component or device fails, it will not be notified and must be tested manually.

i. Time

Time is a significant factor in establishing a school co-curricular management system. The deadline is approaching quickly, and the project may need to be completed on time. Furthermore, the school management system has grown difficult due to the large amount of data required in the co-curricular system, which may take time to complete.

ii. Resources

The project needs more resources to delegate and must be completed one at a time rather than simultaneously, increasing the time required.

3.4 Propose Design

3.4.1 Flowchart System Design

3.4.1.1 Admin Flowchart

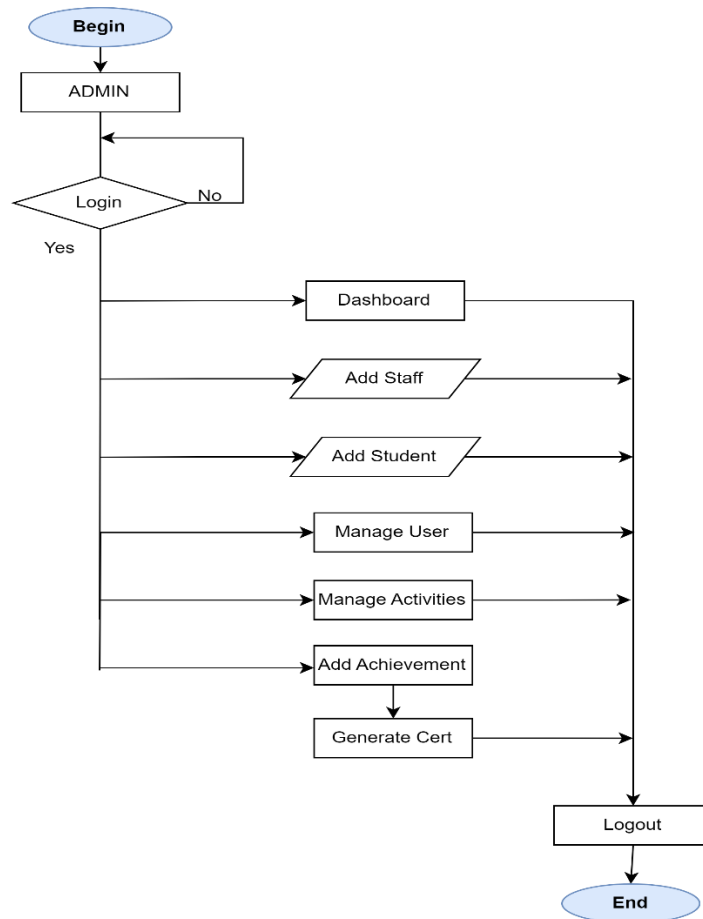


Figure 3.3 Admin Flowchart

Figure 3.3 indicates flowchart for the admin. The admin flowchart begins with the admin logging into the system. If the login is successful, the admin is directed to the dashboard where they have access to various features and options. However, if the login is not successful, the system prompts the admin to register an account or to login using the correct username and password. Once logged in, the admin has the ability to add staff members, students, and cocurricular clubs to the system. The admin can also view system statistics, such as attendance and participation, in the dashboard. Overall, the system provides a comprehensive and convenient way for the admin to manage and track co-curricular activities at their school.

3.4.1.2 Instructor Flowchart

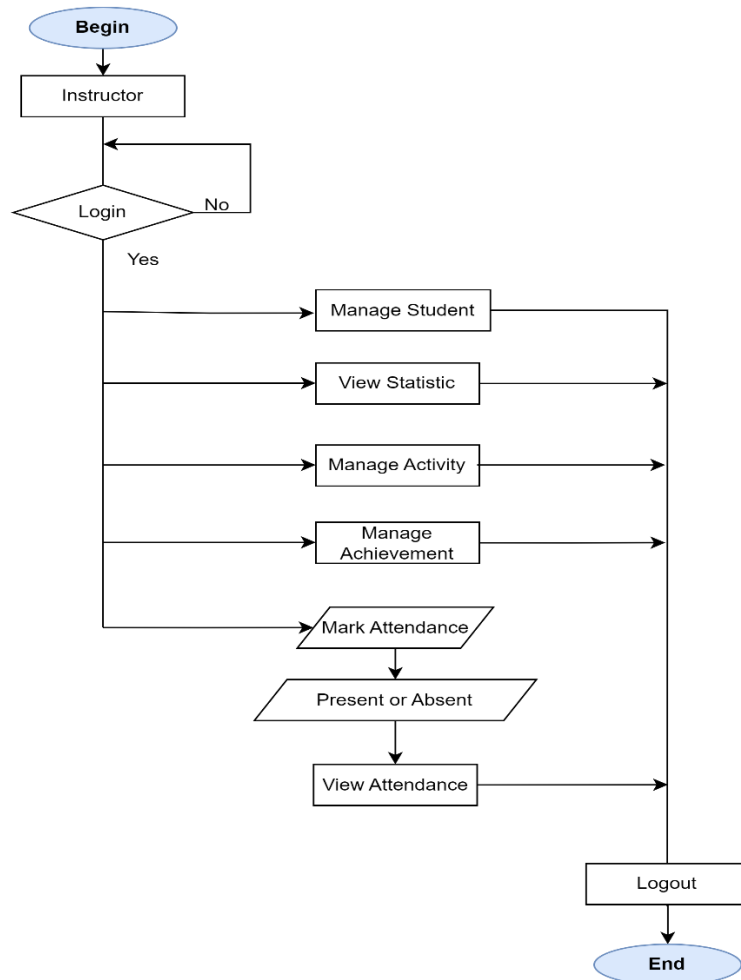


Figure 3.4 Instructor Flowchart

Figure 3.3 indicates flowchart for the staff. The staff flowchart of the cocurricular school co-curricular management system begins with staff members logging into the platform. If the login is successful, they are directed to a dashboard with various features. If the login is unsuccessful, they are prompted to either ask admin to register or enter the correct login credentials. Once logged in, staff members can change their password, manage student participation in activities and track their achievements. They can also mark attendance by selecting options such as present or absent. Staff members can also view statistics of the overall attendance, participation, and achievements in the dashboard. This system offers a comprehensive and user-friendly way for staff to manage and track co-curricular activities at their school

3.4.2 CONTEXT DIAGRAM

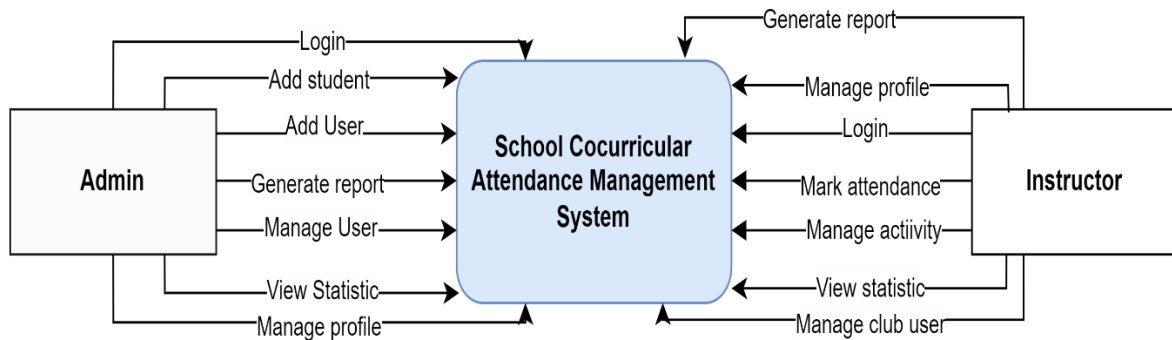


Figure 3.5 Context Diagram

The Figure 3.5 shows the context diagram for the proposed system includes two entities: administrator and instructor. From an administrative perspective, the system can provide an easy way to manage student and instructor accounts, as well as add and edit activities and events. Reports and statistics on attendance and participation can be generated, this can help administrators identify patterns of absenteeism, and help in monitoring the overall participation in extracurricular activities. An administrator can also assign roles and permissions to users, this helps in ensuring that only authorized users can access the system, providing an extra layer of security. Instructors, on the other hand, can use the system to mark attendance for their specific activities and manage their activities. They can mark attendance and monitor student's participation, this will help them in determining the students that are eligible to participate, ensuring that only students who are committed to participating are included. Reports and statistics on attendance and participation can also be viewed by instructors, which can provide useful information on the overall participation within the activities. They can also manage the club members by adding or removing students from the club or activity, this help instructors in keeping track of student co-curricular.

3.4.3 USE CASE DIAGRAM

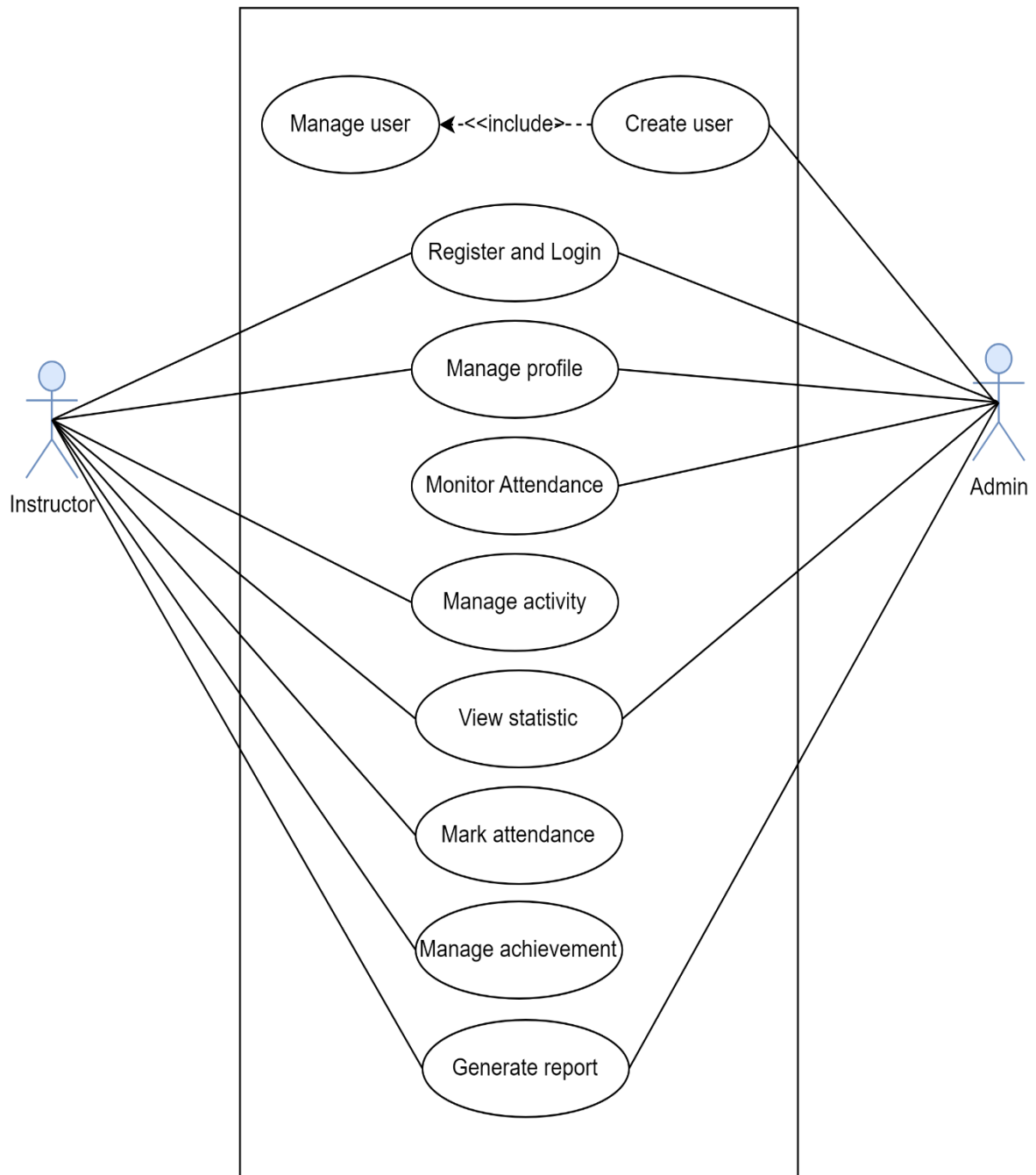


Figure 3.6 Use Case Diagram for Proposed System

3.4.3.1 Use Case Description of Admin Menu

Table 3.1 and 3.2 shows the use case diagram for the school co-curricular management system. The use case has two actors which are Admin and Staff. There are ten use cases that will be in the school co-curricular management system

Table 3.1 Use Case Description of Admin menu

Use Case Name	Admin menu
Brief Description	Admin can update, add or delete user details. Admin also can create a class and view overall statistic
Actor	Admin
Pre-condition	1. Must have an internet connection 2. Admin must access the website
Basic Flow	1. Admin can update, add or delete a users in the database 2. Admin can view statistic and manage user 3. Admin need to create student and staff 4. Admin can monitor attendance 5. Admin can manage activities and student achievement
The system	System gives the option to go back to the main menu

3.4.3.2 Use Case Description of Instructor Menu

Table 3.2 Use Case Description of Instructor menu

Use Case Name	Instructor menu
Brief Description	The instructor can mark attendance add activities and achievement
Actor	Instructor
Pre-condition	1. Must have an internet connection 2. User must access the website
Basic Flow	1. Instructor must login into the system 2. Manage profile, manage activity, and manage achievement 3. Create student achievement 4. Instructor can mark attendance
Post Condition	The system give the best option to select user or go back to the main menu

3.4.4 ACTIVITY DIAGRAM

3.4.4.1 Admin Activity Diagram

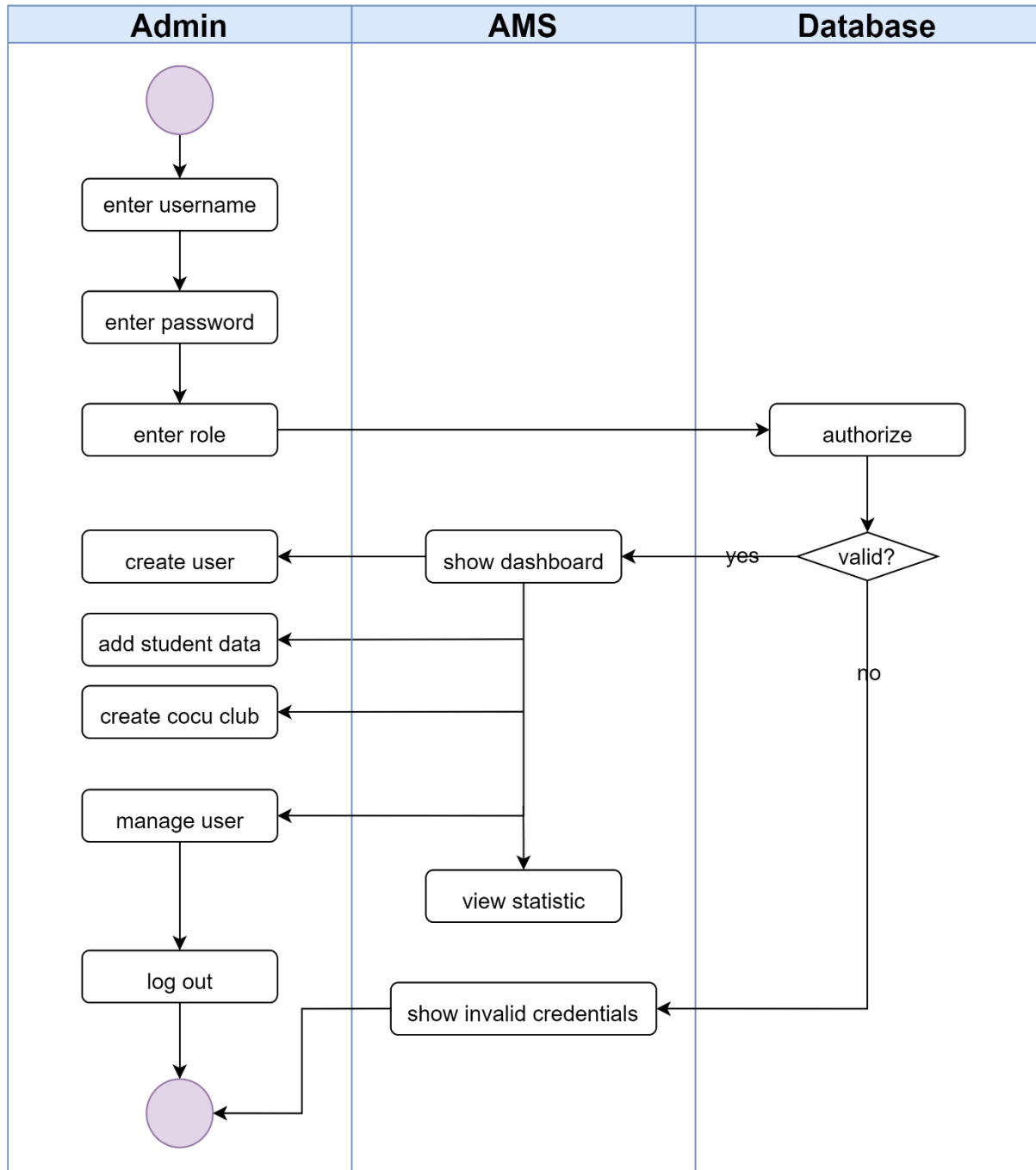


Figure 3.7 Admin Activity Diagram

3.4.4.2 Instructor Activity Diagram

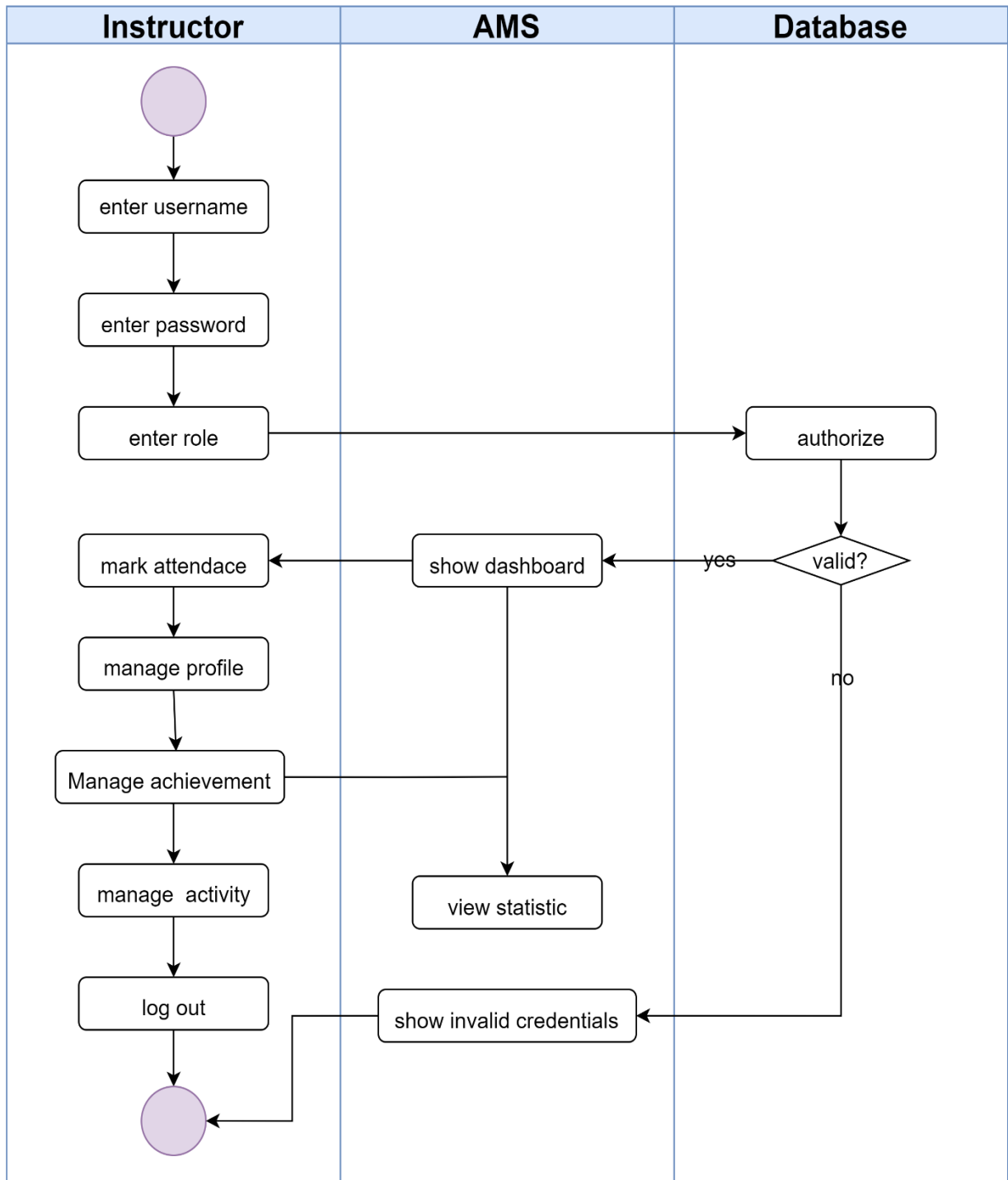


Figure 3.8 Instructor Activity Diagram

3.4.5 ERD DIAGRAM

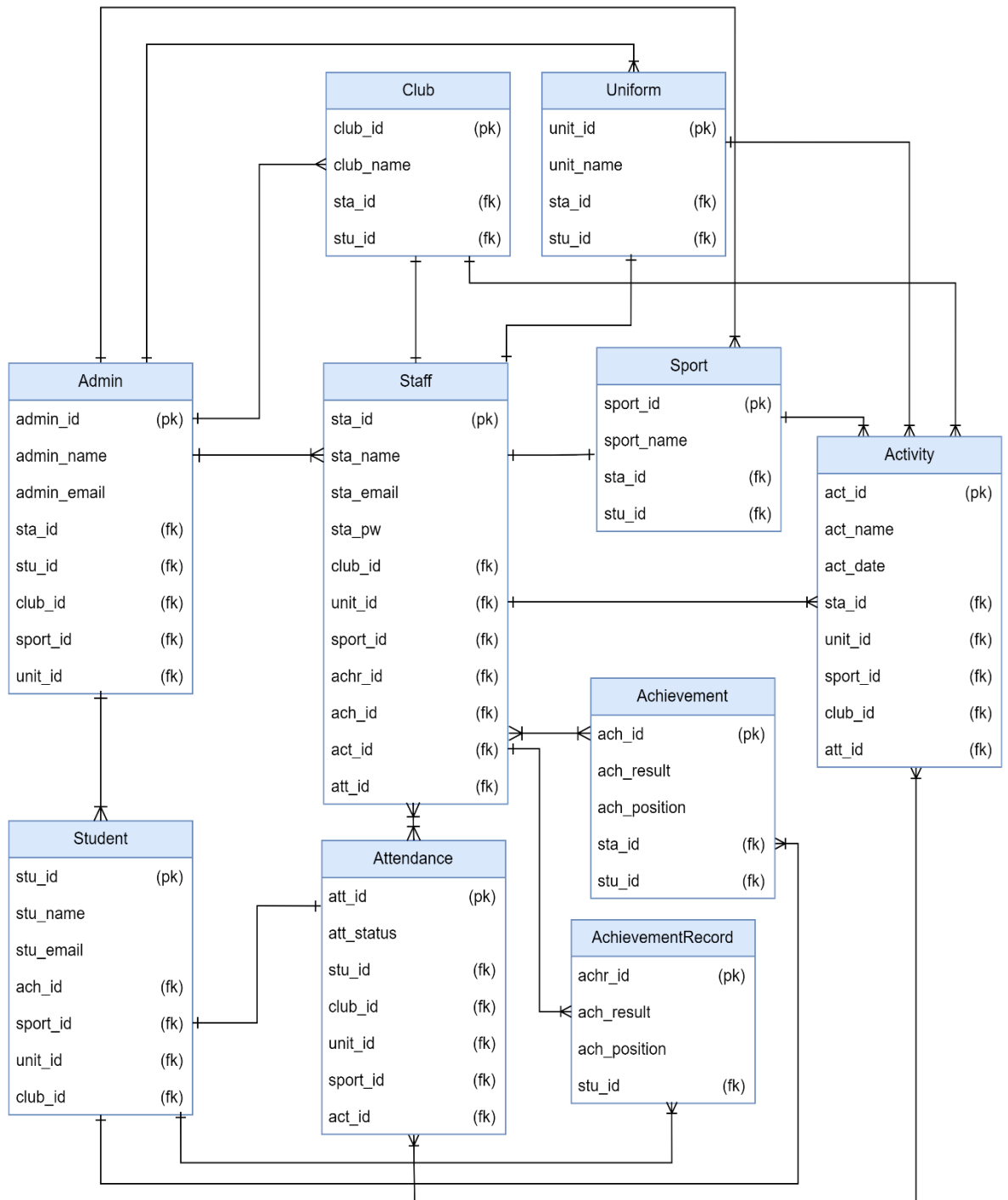


Figure 3.9 ERD Diagram

3.5 Data Design

Table 3.3 The database for the admin.

ATTRIBUTE	DESCRIPTION	TYPE	LENGTH	PK/FK
admin_id	Account ID for admin account	int	20	PK
admin_name	Name for admin account	varchar	20	
admin_pw	Password for admin account	varchar	30	
admin_email	Email address for admin account	email	50	
admin_pnum	Phone number for admin account	int	11	

Table 3.4 The database for the Staff

ATTRIBUTE	DESCRIPTION	TYPE	LENGTH	PK/FK
sta_id	Account ID for user account	id	20	PK
sta_pw	Password for user account	varchar	30	
sta_email	Email address for user account	varchar	40	

sta_name	Name for user account	varchar	100	
sta_num	Phone number for user account	int	30	
club_id	Foreign ID for club database	int	20	
unit_id	Foreign ID for Uniform database	int	20	
ach_id	Foreign ID for achievement database	int	20	
sport_id	Foreign ID for sport database	int	20	
act_id	Foreign ID for activity database	int	20	
att_id	Foreign ID for attendance database	int	20	

Table 3.5 Data dictionary for Student table

Field name	Data type	Field length	Constraint	Description
stu_id	int	20	Primary Key	The Student's ID
stu_name	varchar	100	-	The Student's name
stu_pw	varchar	30	-	The Student's password
stu_num	int	30	-	The Student's phone number

stu_email	varchar	30	-	The Student's email
ach_id	int	20	-	Foreign ID for achievement database
sport_id	int	20	-	Foreign ID for sport database
unit_id	int	20	-	Foreign ID for uniform database
club_id	int	20	-	Foreign ID for club database

Table 3.6 Data dictionary for Activity table

Field name	Data type	Field length	Constraint	Description
act_id	int	30	Primary Key	The Activity's ID
act_name	varchar	100	-	The Activity's Name
act_date	timestamp	-	-	The Activity's Date
sta_id	int	20	-	Foreign ID for staff database
unit_id	int	20	-	Foreign ID for unit database
sport_id	int	20	-	Foreign ID for sport database
club_id	int	20	-	Foreign ID for club database

Table 3.7 Data dictionary for Attendance details table

Field name	Data type	Field length	Constraint	Description
att_id	int	20	Primary Key	The Attendance ID

att_date	timestamp	-	-	The Attendance Date
att_status	varchar	10	-	The Attendance Status
stu_id	int	20	-	Foreign ID for student database
club_id	int	20	-	Foreign ID for club database
unit_id	int	20	-	Foreign ID for unit database
sport_id	int	20	-	Foreign ID for sport database

Table 3.8 Data dictionary for Achievement table

Field name	Data type	Field length	Constraint	Description
ach_id	int	20	Primary Key	The Achievement ID
ach_result	varchar	100	-	The Achievement Result
ach_position	varchar	20	-	The Achievement Position
stu_id	int	20	-	Foreign ID for student database

Table 3.9 Data dictionary for Club table

Field name	Data type	Field length	Constraint	Description
club_id	int	20	Primary Key	The Club ID
club_name	varchar	50	-	The Club name
sta_id	int	20	-	Foreign ID for student database
stu_id	int	20	-	Foreign ID for student database

Table 3.10 Data dictionary for Uniform table

Field name	Data type	Field length	Constraint	Description
unit_id	int	20	Primary Key	The Uniform ID
unit_name	varchar	50	-	The Uniform name
sta_id	int	20	-	Foreign ID for student database
stu_id	int	20	-	Foreign ID for student database

Table 3.11 Data dictionary for Sport table

Field name	Data type	Field length	Constraint	Description
sport_id	int	20	Primary Key	The Sport ID
sport_name	varchar	50	-	The Sport name
sta_id	int	20	-	Foreign ID for student database
stu_id	int	20	-	Foreign ID for student database

3.6 Proof of Initial Concept

3.6.1 Login Interface



Figure 3.10 Login Page Interface

Figure 3.10 shows the design of login interface for the school co-curricular management system. Login interface is a user interface that enables users to input their login details, such as a username and password, to gain access to a system or application. This type of interface is frequently utilized to access websites, online services, and other programs that need authentication to function. In certain cases, the login interface may also include an option to choose the user's role. The illustration provided in Figure 16 depicts the login interface for the system under consideration.

3.6.2 Admin Dashboard

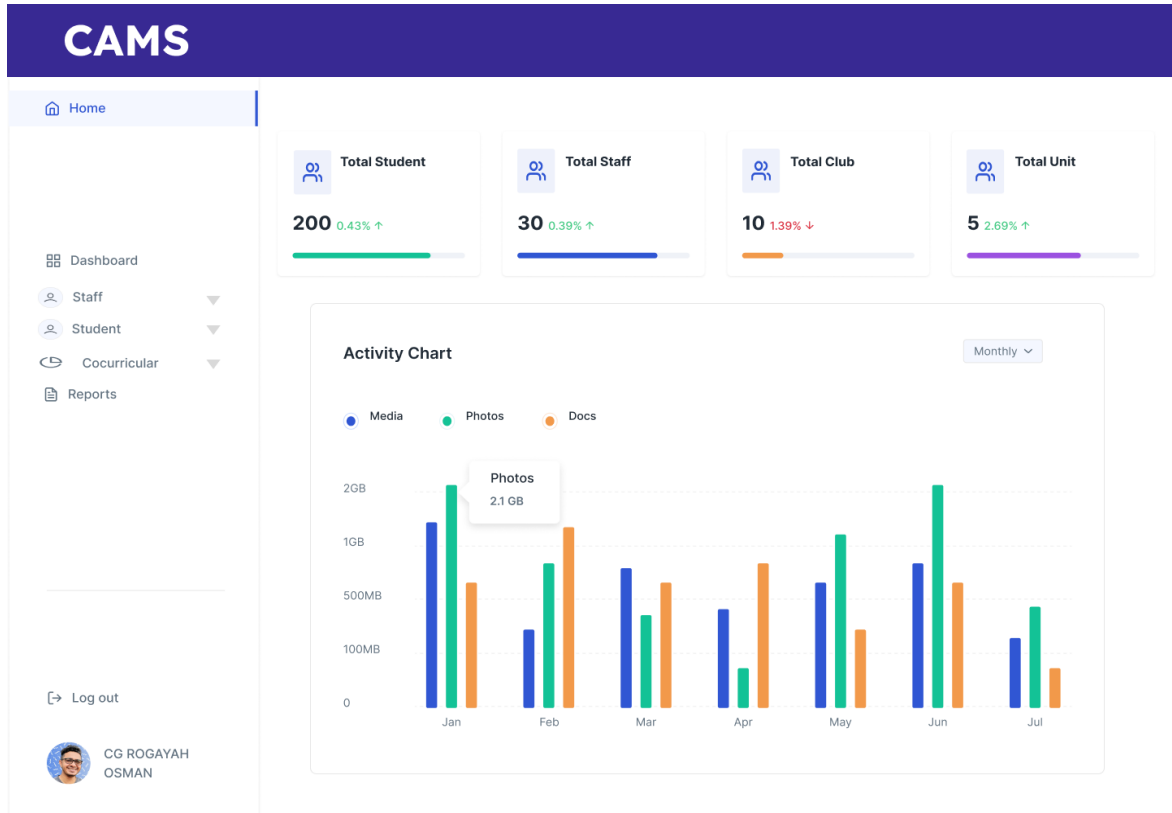


Figure 3.11 Admin Dashboard Interface

Figure 3.11 shows the design of Admin Dashboard Interface for the school co-curricular management system . A dashboard or home interface is a starting point or central hub within an application or system that allows users to access different features and functions of the application. It typically provides an overview of the user's current activity or status within the application, and may include links, buttons, or other controls for accessing the various features and options available. The design of the interface can vary, but it is usually user-friendly and easy to navigate. Based on Figure 17, it illustrates a dashboard interface that comprises of attendance statistics and graph which can be viewed easily.

3.6.3 Add Staff Interface

The screenshot shows the 'Add Staff' interface in the CAMS system. The interface is divided into a header, a sidebar, and a main content area. The header is dark blue with the 'CAMS' logo. The sidebar on the left contains navigation links: Home, Dashboard, Staff, Student, Cocurricular, and Reports. The main content area is titled 'Staff' and contains a form with the following fields: NAME (with 'Input' text), ID, CONTACT, EMAIL, USERNAME, PASSWORD (with masked characters), ADDRESS, CLUB (with a dropdown arrow), SPORT (with a dropdown arrow), and UNIT BERUNIFORM (with a dropdown menu open). The dropdown menu for UNIT BERUNIFORM shows four options: PENGAKAP, PUTERI ISLAM, PANDU PUTERI, and KADET REMAJA SEKOLAH. At the bottom right of the form, there are 'Cancel' and 'Save' buttons. The user profile in the sidebar shows 'CG ROGAYAH OSMAN'.

Figure 3.12 Adding Staff Interface

Figure 3.12 shows the design of adding staff interface for the school co-curricular management system. Adding staff interface is a feature within an application or system that allows users to manage and access information related to staff members. The admin need to put all the staff data and club for each co-curricular The design and layout of the staff interface can vary depending on the specific application or system, but it is commonly designed to be easy to use and navigate.

3.6.4 Edit Staff Interface

The screenshot displays the 'Edit Staff' interface within the CAMS system. On the left is a sidebar with a 'Home' button and a menu containing 'Dashboard', 'Staff', 'Student', 'Cocurricular', and 'Reports'. The main area is titled 'Staff' and contains a form for editing a staff member's details. The form fields are arranged in two columns:

- NAME:** MUHAMMAD HARITH HAZRIQ BIN ANUAR
- ID:** PHG41162
- CONTACT:** 017 721 4455
- EMAIL:** hh@gmail.com
- USERNAME:** HARITHCAMS
- PASSWORD:** [masked]
- ADDRESS:** UMP PARK, PERAMU PERMAI
- CLUB:** SAINS MATEMATIK
- SPORT:** BOLA SEPAK
- UNIT BERUNIFORM:** PUTERI ISLAM (selected from a dropdown menu that also includes PENGAKAP, PANDU PUTERI, and KADET REMAJA SEKOLAH)

At the bottom right of the form are two buttons: 'Cancel' and 'Update'.

Figure 3.13 Edit Staff Interface

Figure 3.13 shows the design of edit staff interface for the school co-curricular management system. An edit or update staff interface is a feature within an application or system that allows users to modify the information of existing staff members. It typically provides a form or set of fields where users can update the details of the staff member, such as name, contact information, and role. It may also include options for updating their schedules, attendance and performance. The design and layout of the update staff interface is commonly designed to be easy to use and navigate. It is usually accessible from the staff interface and is a crucial feature for managing the staff member's information efficiently.

3.6.5 Manage Staff Interface

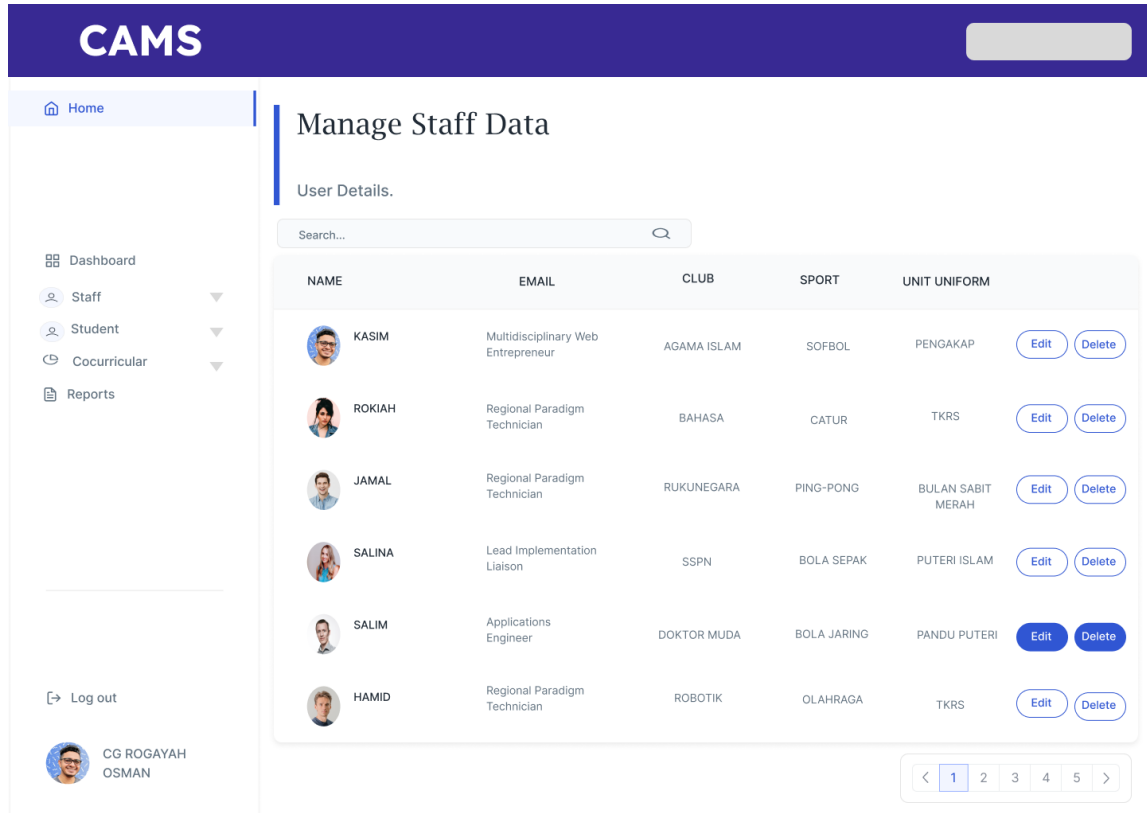


Figure 3.14 Manage Staff Interface

Figure 3.14 indicates the design of manage staff interface for the school co-curricular attendance management system. A manage staff interface is a feature within an application or system that allows users to manage the information of staff members. It typically provides an overview of the staff members and their details, and may include options for adding, editing, deleting and searching staff members. The edit feature allows users to modify the information of existing staff members, such as name, contact information, and co curricular information. The delete feature allows users to remove staff members from the system. The search feature allows users to find specific staff members based on certain criteria, such as name, role, or ID. These functionalities are often integrated into the manage staff interface, making it a powerful tool for managing and maintaining the staff information efficiently.

3.6.6 Add Student Interface

The screenshot shows the 'Add Student' interface in the CAMS system. The interface is divided into a header, a sidebar, and a main content area. The header is dark blue with the 'CAMS' logo. The sidebar on the left contains navigation links for 'Home', 'Dashboard', 'Staff', 'Student', 'Cocurricular', and 'Reports', along with a 'Log out' button and a user profile for 'CG ROGAYAH OSMAN'. The main content area is titled 'Student' and contains the following fields:

- NAME: Input field with the text 'Input'.
- ID: Input field.
- CONTACT: Input field.
- EMAIL: Input field.
- CLASS: Dropdown menu.
- ADDRESS: Input field.
- CLUB: Dropdown menu.
- UNIT BERUNIFORM: Dropdown menu with a list of options: PENGAKAP, PUTERI ISLAM, PANDU PUTERI, and KADET REMAJA SEKOLAH.
- SPORT: Dropdown menu.

At the bottom right of the main content area, there are two buttons: 'Cancel' and 'Save'.

Figure 3.15 Adding Student Interface

Figure 3.15 indicates the design of adding student interface for the school co-curricular management system. Adding student interface is a feature within an application or system that allows users to manage and access information related to student information data. The admin need top put all the student data and club for each co-curricular The design and layout of the staff interface can vary depending on the specific application or system, but it is commonly designed to be easy to use and navigate. There also have a cancel button to cancel add student or submit student information.

3.6.7 Edit Student Interface

The screenshot displays the 'Edit Student' interface within the CAMS system. The interface is organized into a sidebar on the left and a main content area on the right. The sidebar includes a 'Home' button and a menu with options: Dashboard, Staff, Student, Cocurricular, and Reports. At the bottom of the sidebar, there is a 'Log out' button and a user profile for CG ROGAYAH OSMAN. The main content area is titled 'Student' and contains a form with the following fields:

- NAME:** FIRDAUS IQMAL BIN SALMAN
- ID:** CA20149
- CONTACT:** 011 4567111
- EMAIL:** firdausiqmal13@gmail.com
- CLASS:** 6 LAKSAMANA
- ADDRESS:** NO 64, TAMAN BUKIT PERDANA, 41000, GOMBAK, MELAKA
- CLUB:** PEMBIMBING RAKAN SEBAYA
- SPORT:** BOLA BALING
- UNIT BERUNIFORM:** A dropdown menu is open, showing options: PENGAKAP, PUTERI ISLAM, PANDU PUTERI, and KADET REMAJA SEKOLAH.

At the bottom right of the form, there are two buttons: 'Cancel' and 'Update'.

Figure 3.16 Edit Student Interface

Figure 3.16 indicates the design of edit student interface for the school co-curricular management system. An edit or update student interface is a feature in school co-curricular management system that allows users to modify the information of existing student members. It typically provides a form or set of fields where users can update the details of the student member, such as name, contact information, and role. It may also include options for updating their student information and co-curricular information data. The design and layout of the update staff interface is commonly designed to be easy to use and navigate. It is usually accessible from the staff interface and is a crucial feature for managing the staff member's information efficiently.

3.6.8 Manage Student Interface

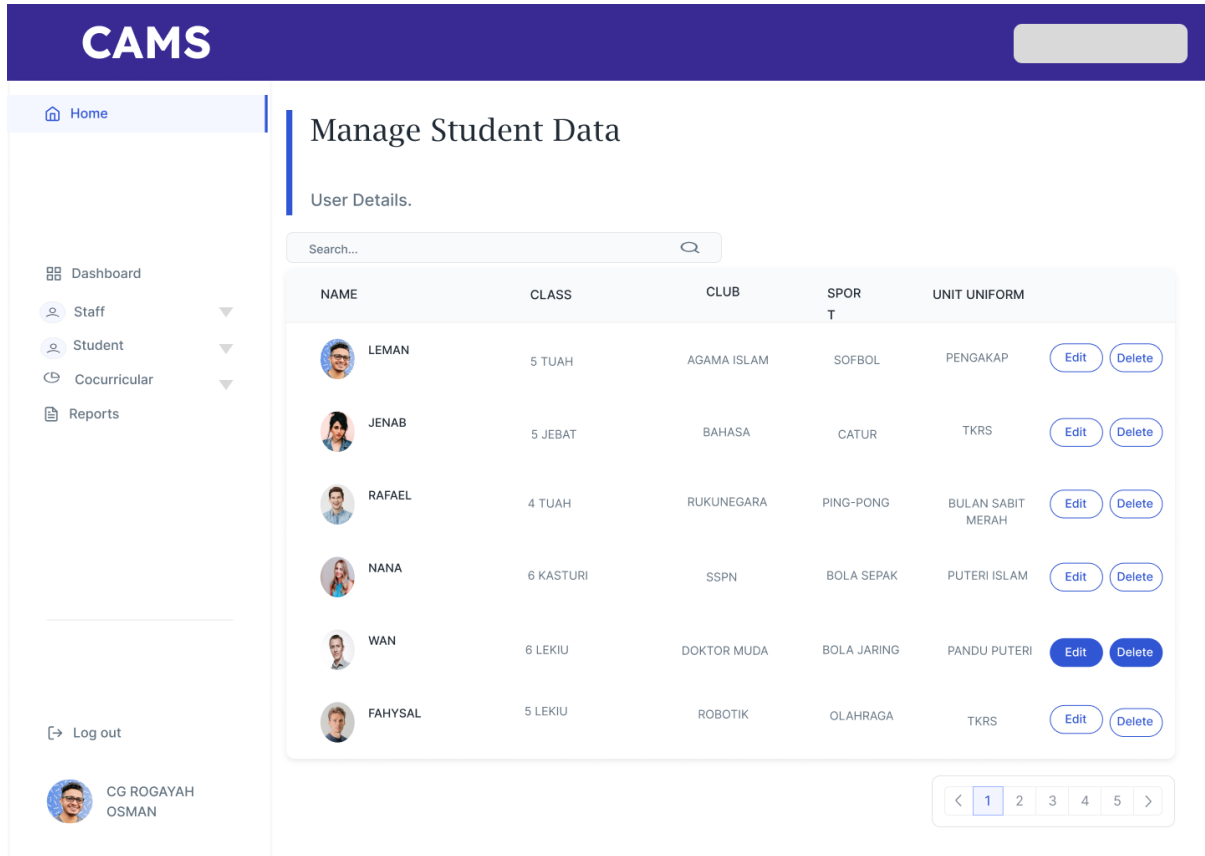


Figure 3.17 Manage Student Interface

Figure 3.17 depicts the design of manage student interface for the school co-curricular management system. A manage student interface is a feature in school co-curricular management system that allows users to manage the information of student data. It typically provides an overview of the student list and their details, and may include options for adding, editing, deleting and searching staff members. The edit feature allows users to modify the information of existing student information, such as name, contact information, and co-curricular information. The delete feature allows users to remove staff members from the system. The search feature allows users to find specific student members based on certain criteria, such as name, role, or ID. These functionalities are often integrated into the manage staff interface, making it a powerful tool for managing and maintaining the student information efficiently.

3.6.9 Pop Up Notification for Add and Edit button

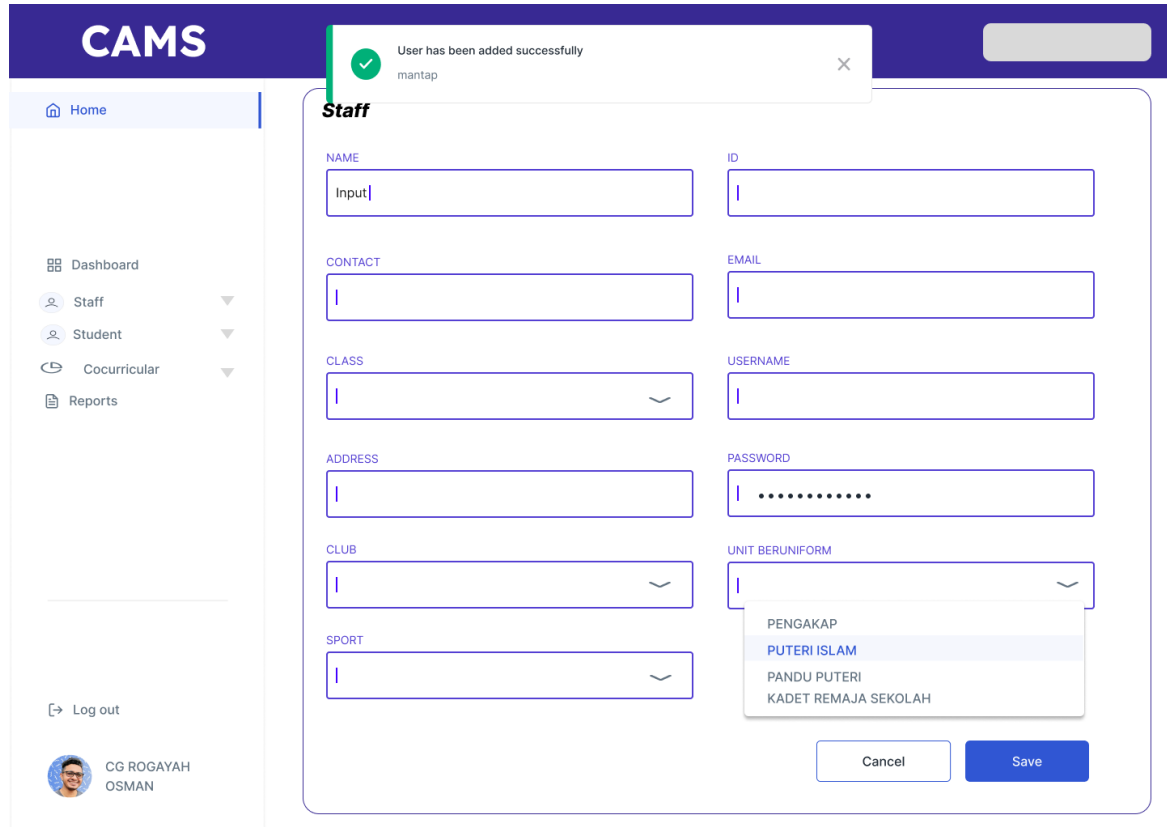


Figure 3.18 Example Popup Notification for Add User

Figure 3.18 depicts the example pop-up notification for add user for the school co-curricular management system. A pop-up notification for the add and edit button is a feature that displays a small window or message to confirm or notify the user of the action they are taking. When a user clicks the add button, a pop-up notification can appear to confirm that a new staff member has been added to the system, and when a user clicks the edit button, a pop-up notification can appear to confirm that the staff member's information has been updated. Pop-up notifications can also provide additional information or options, such as the option to undo an action or view the updated information. This feature can help to prevent accidental actions and provide users with feedback on the actions they have taken, which can improve the overall user experience and make the application more user-friendly.

3.6.10 Pop Up Notification for Delete button

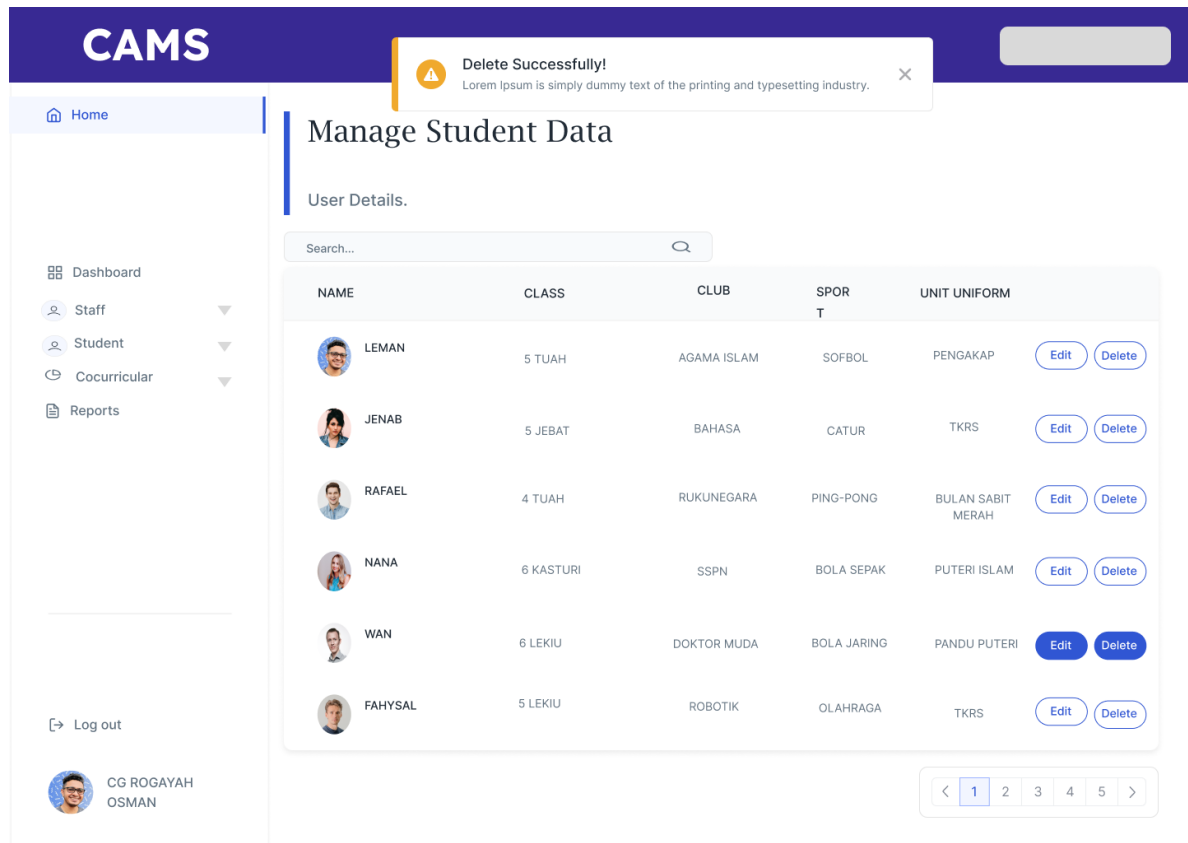


Figure 3.19 Example Pop Up Notification for Delete button

Figure 3.19 depicts the design example pop-up notification for delete button for the school co-curricular management system. A pop-up notification for the delete button is a feature that displays a small window or message to confirm or notify the user of the action they are taking when they click the delete button. The pop-up notification will typically appear to confirm the deletion of a staff member from the system and may include options such as cancel or confirm. It's common to put a warning message to confirm the deletion, as it's an irreversible action. The design and layout of the pop-up notification are commonly designed to be clear and easy to understand. This feature can help to prevent accidental deletion and provide users with feedback on the actions they have taken, which can improve the overall user experience and make the application more user-friendly.

3.6.11 Staff Dashboard Interface

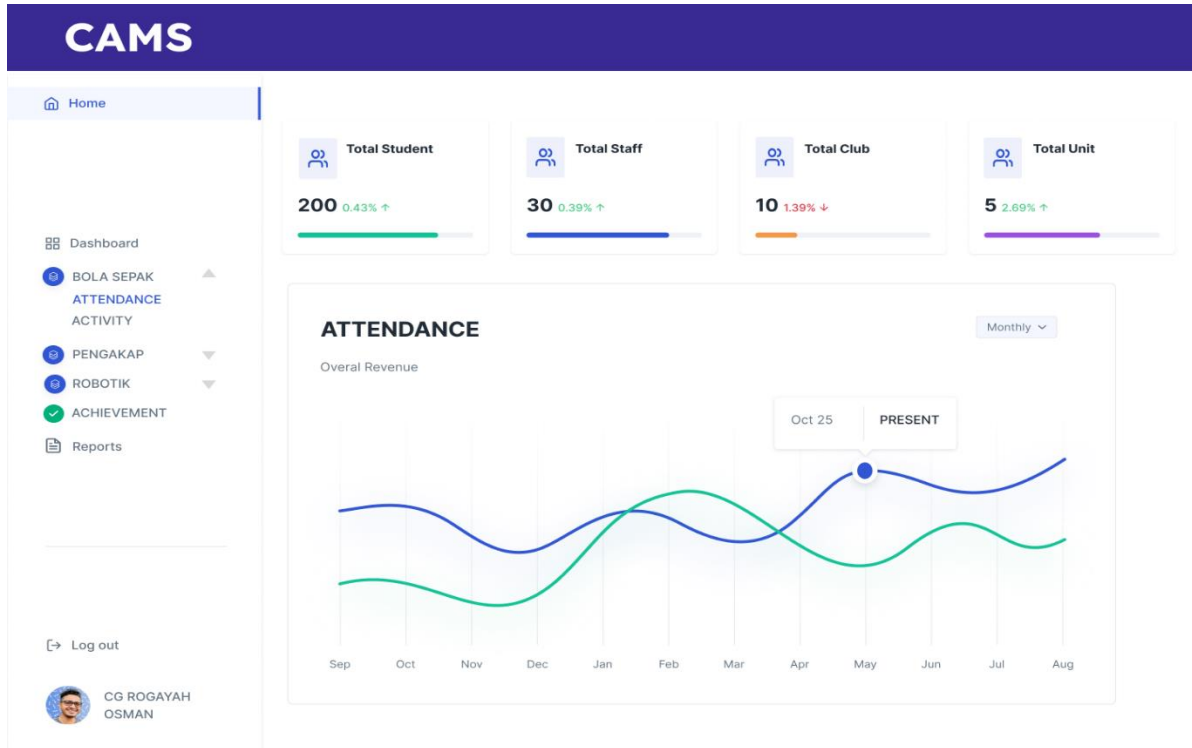


Figure 3.20 Staff Dashboard Interface

Figure 3.20 represents the design of staff dashboard interface for the school co-curricular management system. A staff dashboard interface for a co-curricular management system that includes statistics and data is a powerful tool for staff members to manage and track attendance for co-curricular activities. The interface provides an overview of the staff member's current activity or status within the system, and includes links, buttons, or other controls for accessing the various features and options available. The staff dashboard interface includes functionalities such as managing for different co-curricular activities, viewing, and managing student attendance records, and generating reports. One of the key features of this staff dashboard interface is its ability to display and analyse attendance data through various graphical representations such as bar charts or line graphs, which allows staff members to quickly visualize overall attendance trends. The design and layout of the staff dashboard interface is user-friendly and easy to navigate and can be customized based on the needs of the school.

3.6.12 Cocurricular Attendance Interface

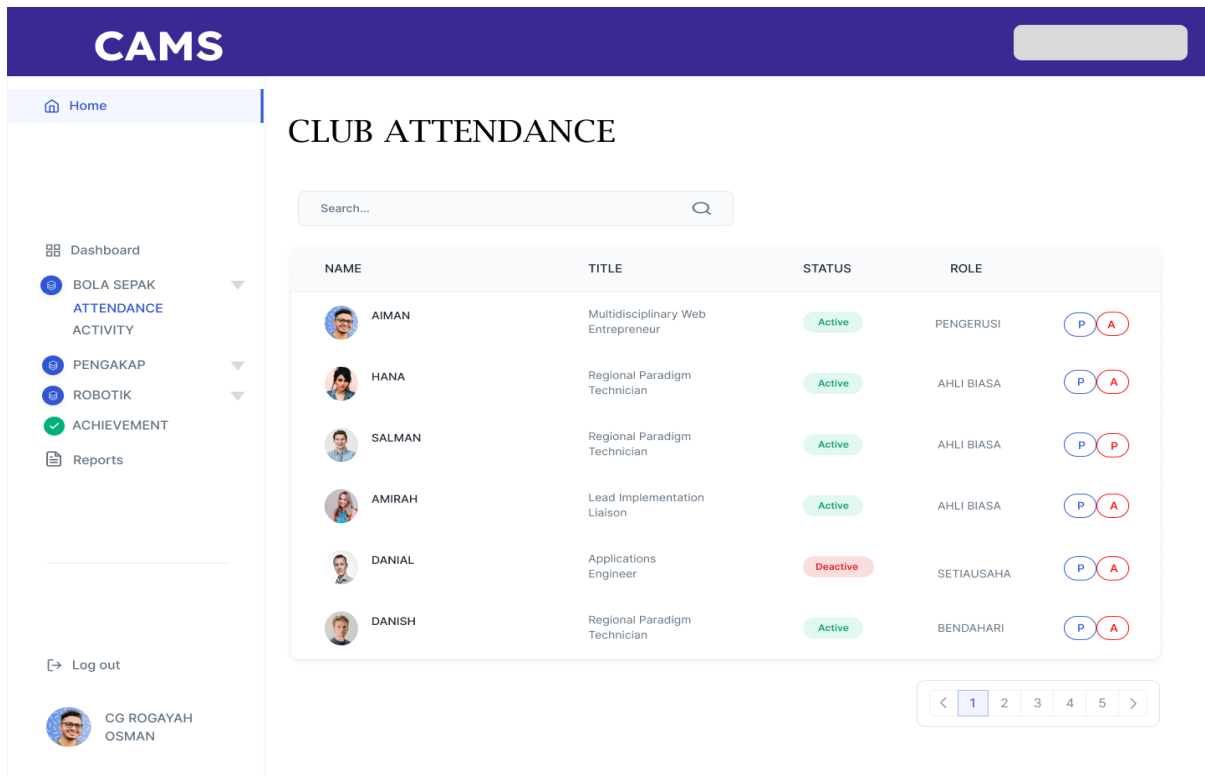


Figure 3.21 Attendance Interface for Cocu

Figure 3.21 represents the design of staff Co-curricular Attendance interface for the school co-curricular management system. A co-curricular attendance interface designed to mark student attendance using radio buttons and search feature is a useful tool for staff members to easily track and manage student attendance. The interface provides a table that displays a list of students and their attendance status, with radio buttons next to each student's name that staff members can use to mark attendance quickly and easily. The interface also includes a search bar that allows staff members to search for specific students by name or ID number, making it easy to find and mark attendance for individual students. This makes it easy for staff members to focus on the relevant data and makes the attendance recording process more efficient. This can help staff members to keep track of attendance and make informed decisions about student participation in co-curricular activities. The design of the interface is

user-friendly and easy to navigate, making it simple for staff members to mark attendance and update student records.

3.6.13 Add Activity Interface

The screenshot shows the 'New Activity' form in the CAMS system. The form is divided into several sections: a calendar for selecting the activity date (January 25, 2023, is selected), an 'Activity Name' input field, an 'Upload a File' section with a 'Choose file' button and 'No file chosen' text, a 'Message' text area with a 'Type your message' placeholder, and an 'Attachments' section with a dashed border and a 'Drag & drop or browse' instruction. At the bottom right, there are 'Cancel' and 'Save' buttons. The user profile 'CG ROGAYAH OSMAN' is visible in the bottom left corner.

Figure 3.22 Adding Activity Interface

Figure 3.22 represents the design of adding activity interface for the school co-curricular management system. Adding co-curricular activity interface is a feature in school co-curricular management system that allows users to add information related to co-curricular activities. The interface would typically include fields for entering the name of the activity, the date it takes place, and a description. One important feature of this interface is the ability to easily add files or images related to the activity, such as flyers, posters, or photos. This feature allows staff members to provide visual aids to help students understand the activity and it is also useful for promoting the activity to students. The interface should also include a field for the date of the activity, which will be used to schedule and track attendance.

3.6.14 Update Activity Interface

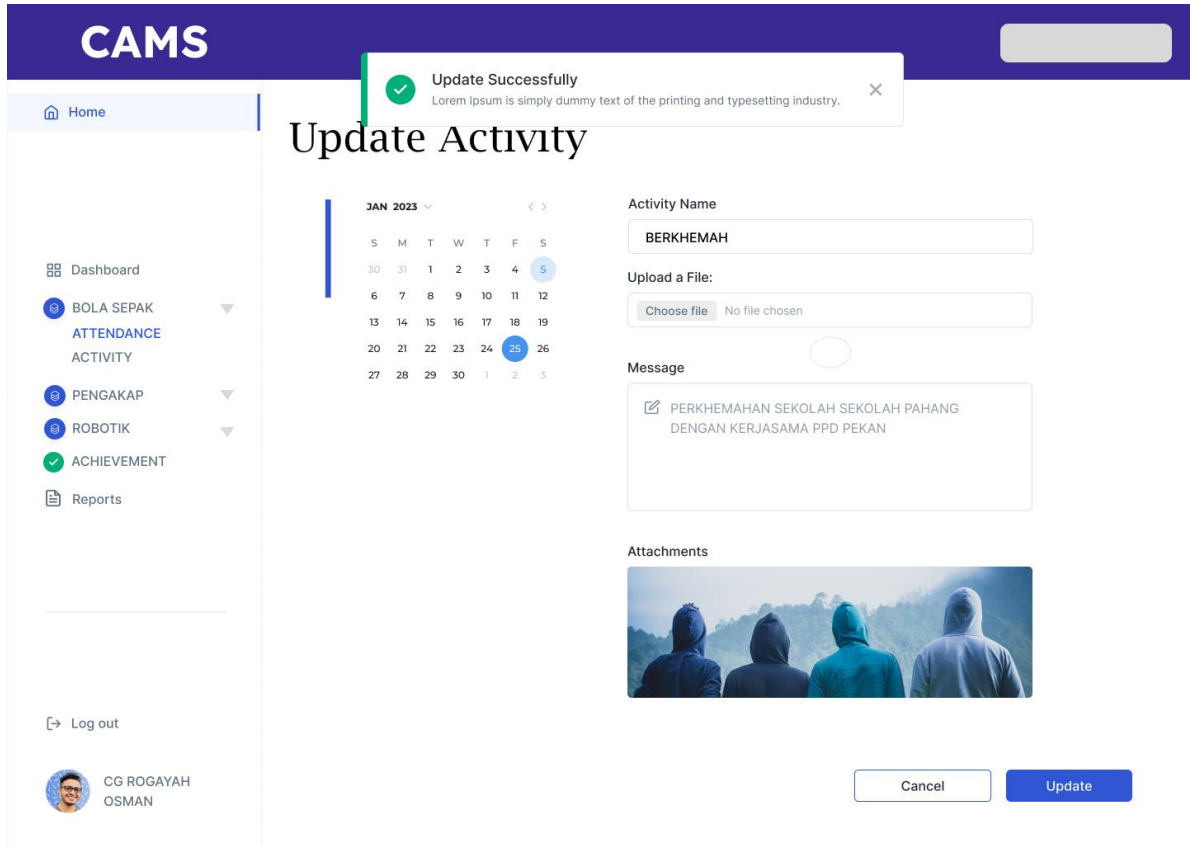


Figure 3.23 Edit Activity Interface

Figure 3.23 displays the design of edit activity interface for the school co-curricular management system. An editing co-curricular activity interface is a feature in school co-curricular management system that allows users to modify the information of existing co-curricular activities. It typically provides a form or set of fields where users can update the details of the activity, such as name, date, and a description. It may also include options for updating the files or images related to the activity. One of the key features of this interface is the ability to easily edit the date of the activity, which will be used to schedule and track attendance. This feature allows staff members to make changes to the activity schedule in case of any changes. It should also include the ability to edit or delete the files or images related to the activity. This interface will help staff members to make changes to the activities with ease, ensuring that the information related to the activities is up to date and accurate.

3.6.15 Manage Activity Interface

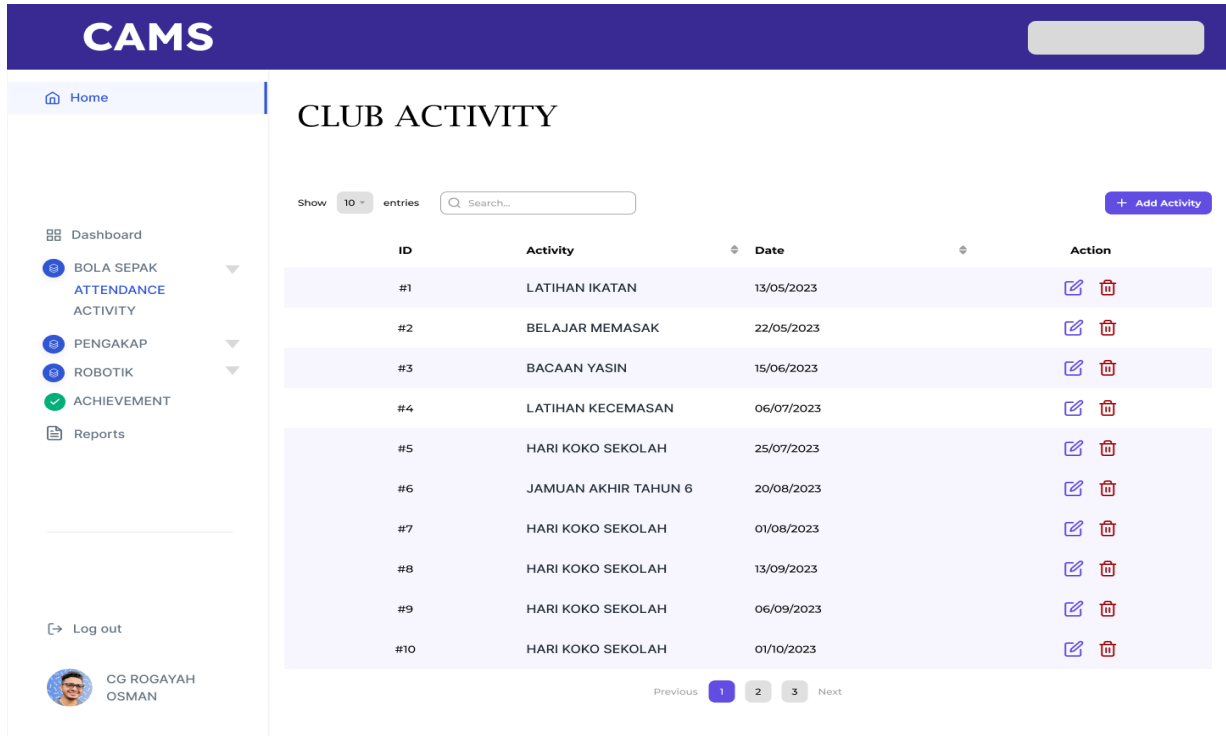


Figure 3.24 Manage Activity Interface

Figure 3.24 displays the design of manage activity interface for the school co-curricular management system. A manage activity interface is a feature in school co-curricular management system that allows users to manage and access information related to co-curricular activities. It typically provides an overview of the co-curricular activities and their details, and may include options for adding new activities, editing or deleting existing ones, and searching for specific activities. The interface would display a list of activities, which would be organized in a clear and easy to read format. The list would include the name of the activity, date, and a brief description. It would also include buttons such as "add," "edit," "delete," and "search" that allow users to manage the activities quickly and easily. The add button allows users to add new activities to the list, the edit button allows users to modify the information of existing activities, the delete button allows users to remove activities from the list, and the search button allows users to find specific activities based on certain criteria, such as name, date, or description.

3.6.16 Add Student Achievement Interface

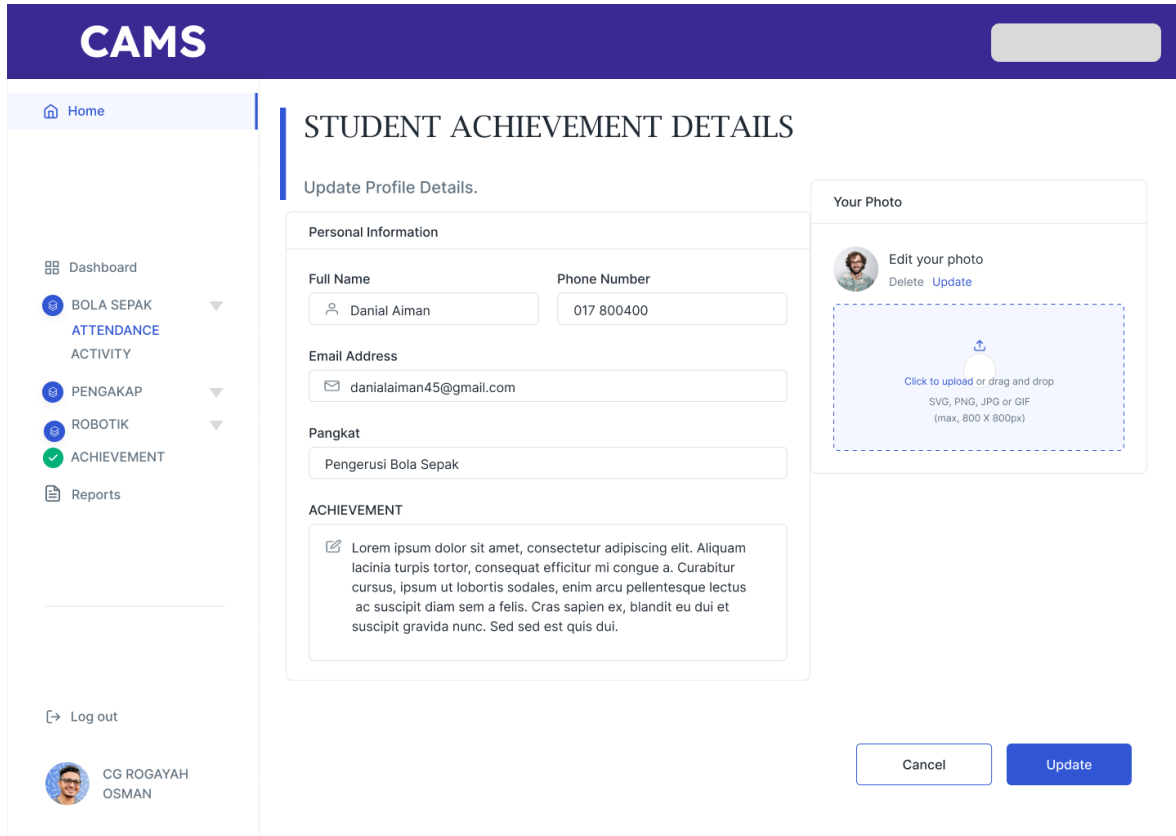


Figure 3.25 Student Achievement Details Interface

Figure 3.25 displays the design of student achievement details interface for the school co-curricular management system. An add student achievement interface in school co-curricular management system that allows users to add information related to students' achievements, such as certificates or awards. It typically provides an overview of the students' achievements and their details. The interface would likely include fields for entering the name of the achievement, the student's name, the date of the achievement and a description. It may also include the ability to upload a file or image of the certificate or award.

3.6.17 Manage Student Achievement Interface

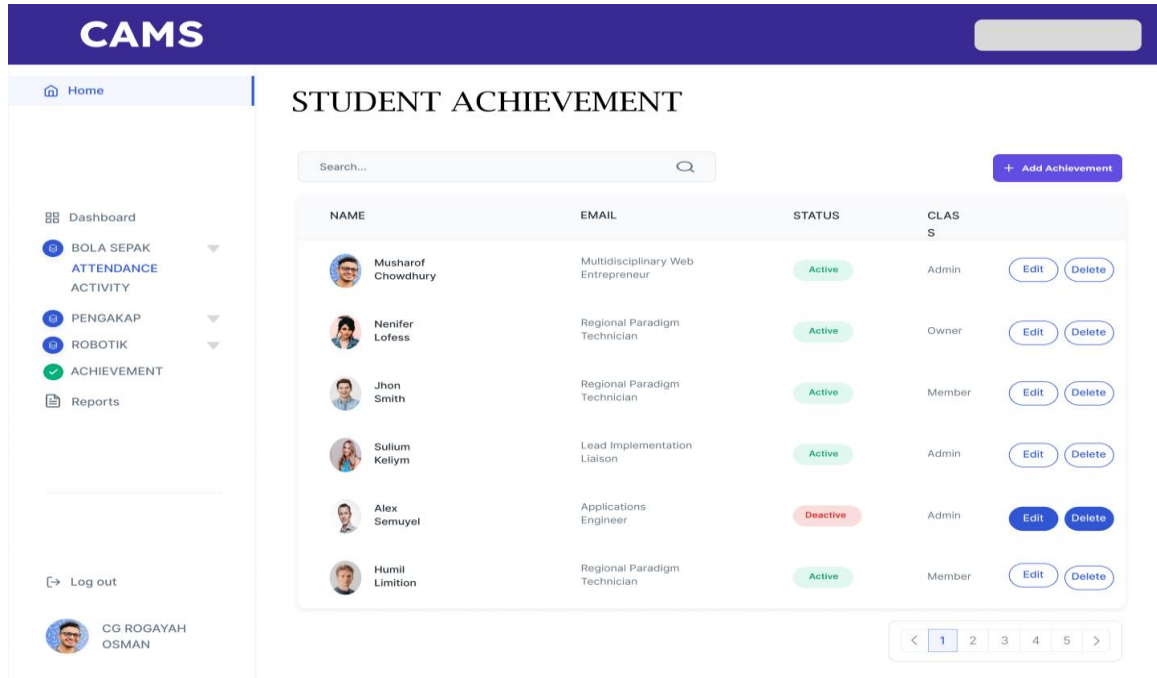


Figure 3.26 List of Student Achievement

Figure 3.26 shows the list of student achievement interface for the school co-curricular management system. A manage student achievement interface is a feature within an application or system that allows users to manage and access information related to students' achievements, such as certificates or awards. It typically provides an overview of the students' achievements and their details, and may include options for adding new achievements, editing or deleting existing ones, and searching for specific achievements. The interface would display a list of achievements, which would be organized in a clear and easy to read format. The list would include the name of the achievement, student's name, date of the achievement, and a brief description. It would also include buttons such as "add," "edit," "delete," and "search" that allow users to manage the achievements quickly and easily. The add button allows users to add new achievements to the list, the edit button allows users to modify the information of existing achievements, the delete button allows users to remove achievements from the list, and the search button allows users to find specific achievements based on certain criteria, such as student's name, date, or achievement name.

3.6.18 Change Password Interface

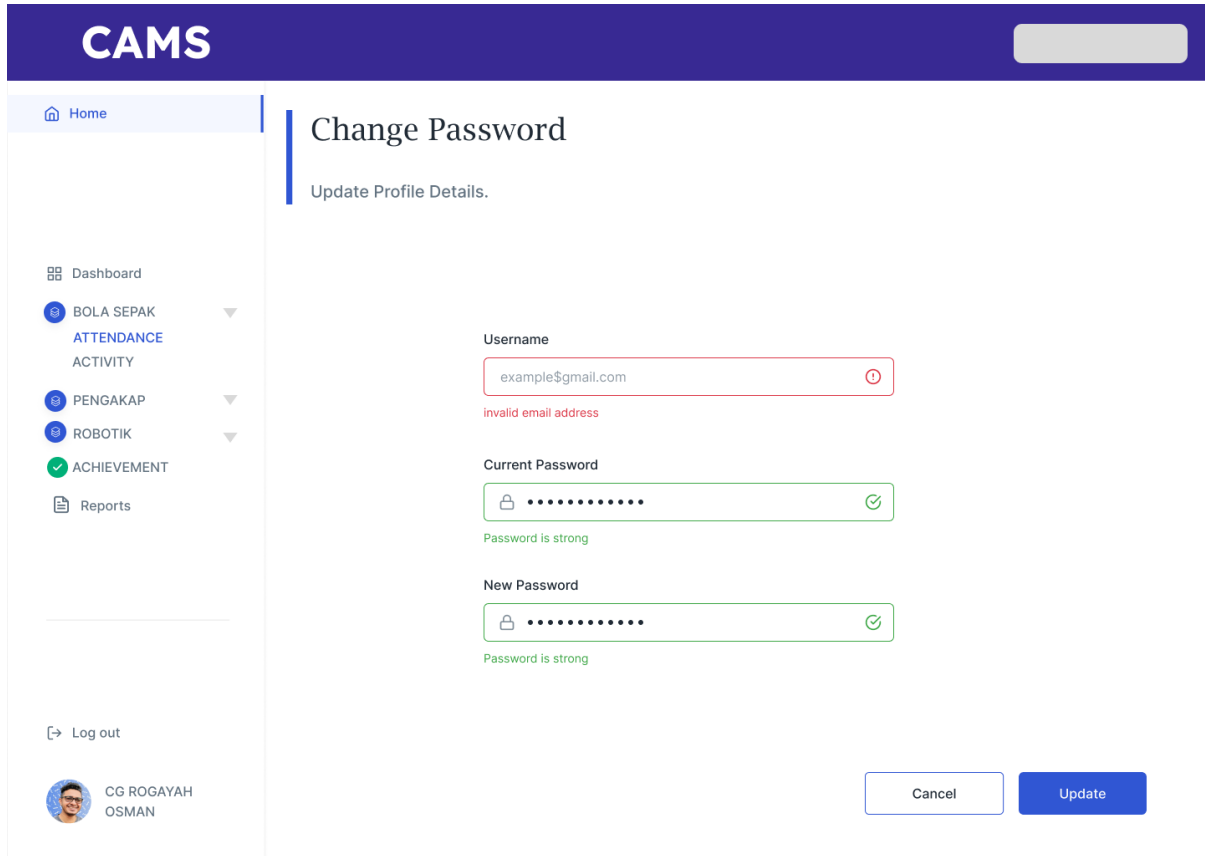


Figure 3.27 Change Password for Instructor

Figure 3.27 shows design of change password interface for the school co-curricular management system. A Instructor change password interface is a screen or set of screens within an application or system that allows a user to change their password. This interface is typically accessed through the login page or user settings, and may include fields for entering the current password, a new password, and a confirmation of the new password. It should also include clear instructions and error messages to guide the user through the process of resetting or changing their password. This interface is an important feature for ensuring the security of user's account and personal information.

3.7 Testing/Validation Plan

3.7.1 User Acceptance Test (UAT) form for Admin:

Table 3.12 User Acceptance Test (UAT) form for Admin

No.	Module	Status		Comment
1.	Login to the Sekolah Kebangsaan Mat Kilau CAMS	Pass	Fail	
2.	Registration Users for system	Pass	Fail	
3.	Main Interface or Dashboard features and function	Pass	Fail	
4.	Calculation data student in table	Pass	Fail	
5.	Update data users	Pass	Fail	
6.	Insert data users	Pass	Fail	
7.	Delete data users	Pass	Fail	

This test performed by:

Name: _____

Signature: _____

Date: _____

3.7.2 User Acceptance Test (UAT) form for User:

Table 3.13 User Acceptance Test (UAT) form for User

No.	Module	Status		Comment
1.	Dashboard features and function for staff	Pass	Fail	
2.	Login to the Sekolah Kebangsaan Mat Kilau CAMS	Pass	Fail	
3.	Insert and edit co-curricular Activity and Achievement	Pass	Fail	
4.	System Interface Design	Pass	Fail	
5.	Delete data activity and achievement	Pass	Fail	
6.	Search table function	Pass	Fail	
7.	Mark Attendance student function	Pass	Fail	

This test performed by:

Name: _____

Signature: _____

Date: _____

3.8 Potential Use of Proposed Solution

A co-curricular management system at a school can be a valuable tool for both administrators and teachers in managing and tracking attendance for extracurricular activities such as sports teams, clubs, and other after-school programs.

From an administrative perspective, the system can automate attendance tracking and reduce the workload associated with manual record-keeping. This can save time and improve efficiency, allowing administrators to focus on other important tasks. The system can also be set up to send reminders to students and parents about upcoming events and practices, as well as notifications when a student is absent. This can help improve communication and ensure that students and parents are aware of the importance of attendance and participation in these activities.

Additionally, the system can generate reports on attendance for individual students, teams, or clubs, which can be useful for tracking participation and identifying patterns of absenteeism. This can help administrators monitor the overall participation in extracurricular activities and identify areas that may require further attention.

From a teacher's perspective, the system can be used to monitor attendance and participation for eligibility, for example for a student to participate or represent the school in sports depending on the number of practices attended. By automating the attendance tracking process, teachers can easily access attendance records and determine the students that are eligible to participate. This can help ensure that teams and clubs are composed of students who are committed to participating and can help improve the overall experience for all involved.

Furthermore, the system could be implemented with access controls, ensuring that only authorized users, such as administrators and teachers, can access the data. This will ensure the privacy and security of student information and prevent unauthorized access or misuse of data. Overall, a co-curricular management system at school can be a valuable tool for both administrators and teachers in managing and tracking attendance for extracurricular activities. It can automate the attendance tracking process, improve communication, and enhance accountability, while reducing administrative burden and enhancing participation monitoring.

3.9 Gantt Chart

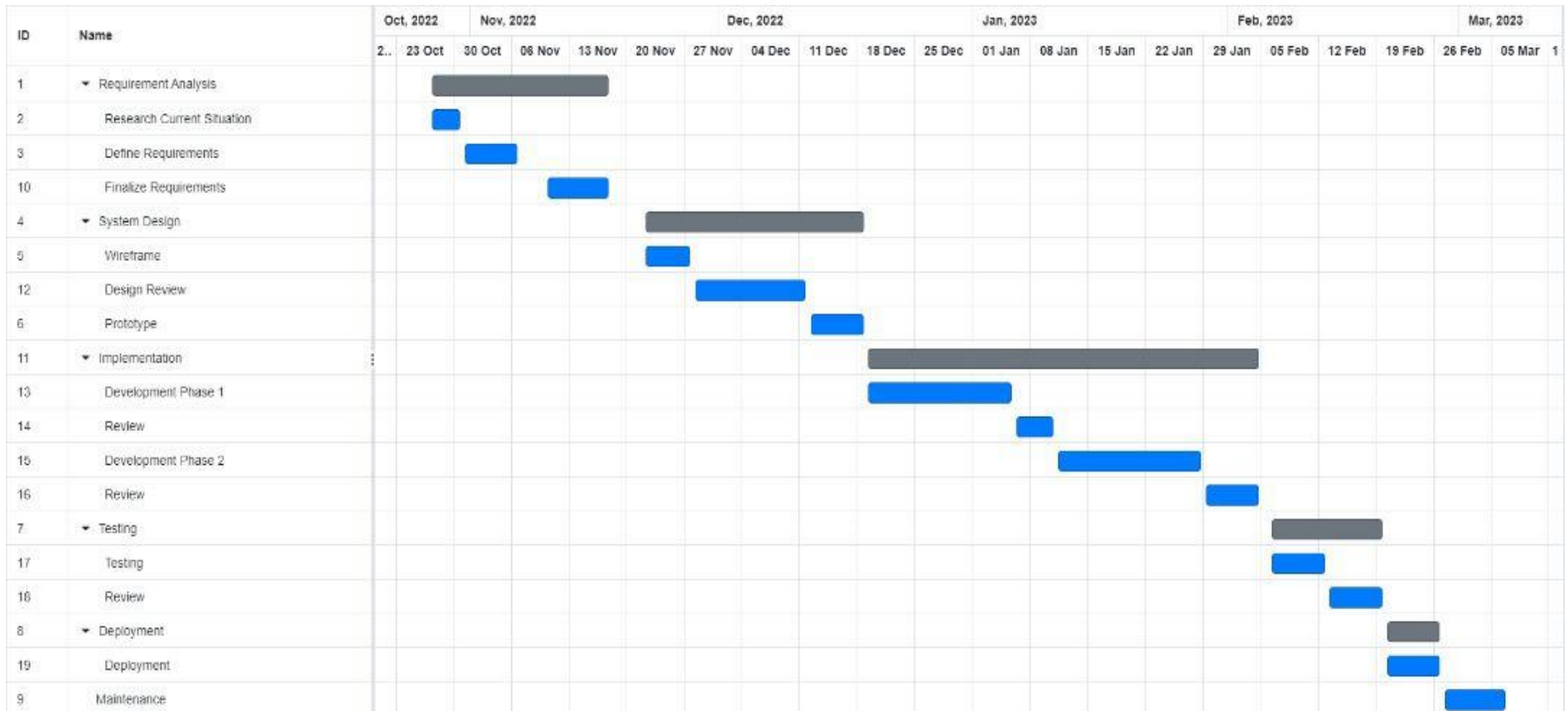


Figure 3.28 Gantt Chart System Development

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

Chapter 4 will cover the development, implement and testing of the School Cocurricular Management System, which required several essential components including MySql, visual studio code, Figma, and Laravel Framework. The system was designed for primary school teachers and staff at Sekolah Kebangsaan Mat Kilau. To ensure optimal functionality, thorough testing was conducted to identify and promptly resolve any potential errors.

4.2 Implementation Process

The process of developing the Cocurricular Management System (CMS) began with outlining its database. To accomplish this, an Entity Relationship Diagram (ERD) was created to determine the various entities that were related to the system. This diagram was instrumental in defining what data would be stored, specifying the attributes of each entity, and establishing the relationships between them.

After the ERD was finalized, it was implemented in phpMyAdmin by creating the necessary database and tables for the CMS. This allowed for the efficient storage and retrieval of information that would be used by the system to manage various cocurricular activities. By utilizing the ERD and phpMyAdmin, the development team was able to create a robust and efficient database that met the needs of the CMS.

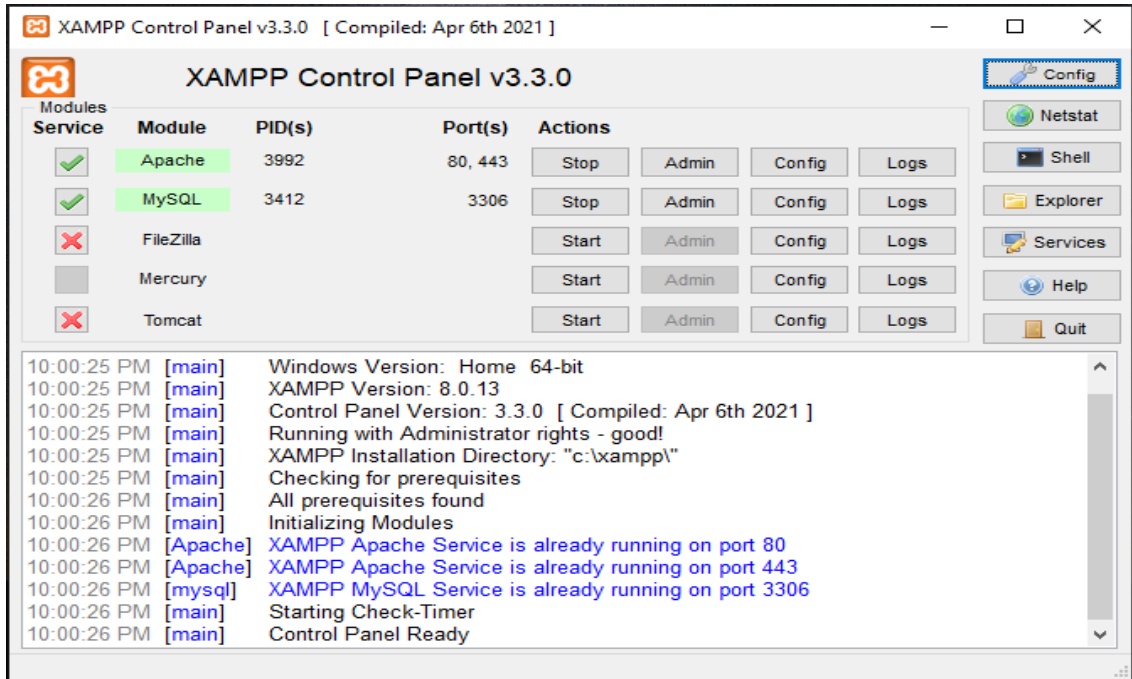


Figure 4.1 XAMPP Control panel

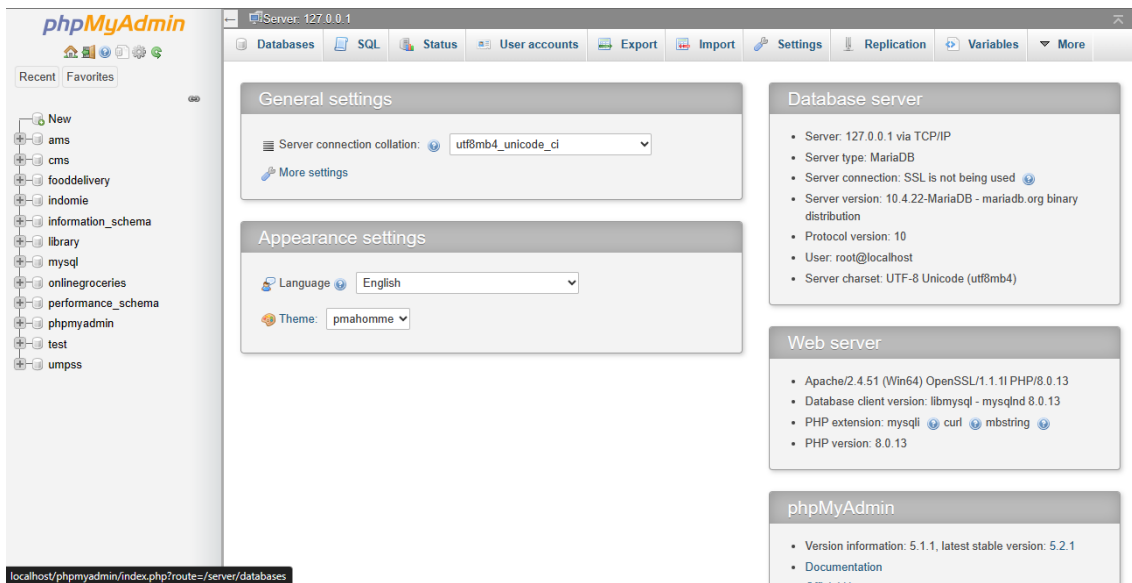
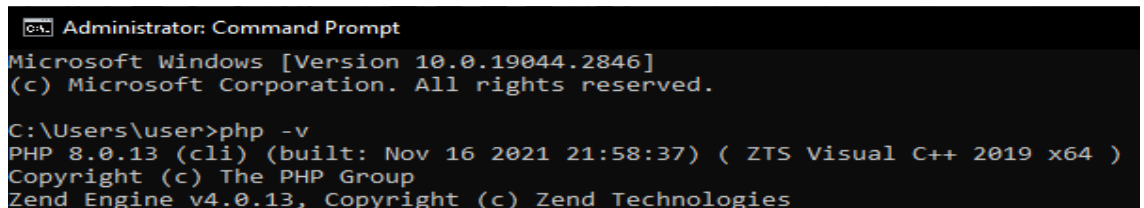


Figure 4.2 phpMyAdmin interfaces

To create the Cocurricular Management System (CMS), the development team utilized the Laravel Framework. Laravel is a popular web application framework that boasts an elegant, expressive syntax and a range of powerful features. One of its most notable strengths is its thorough dependency injection, which allows for easy and flexible management of application dependencies. Additionally, Laravel provides an impressive database abstraction layer that simplifies database management and makes it easier to work with databases in an object-oriented manner. Another valuable feature of Laravel is its ability to support queues and scheduled jobs, which can be utilized to automate and streamline various tasks within the CMS. Furthermore, Laravel's support for unit and integration testing makes it easier to ensure that the CMS is running smoothly and functioning as intended.

Creating Laravel project :

i. PHP installed on computer



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19044.2846]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>php -v
PHP 8.0.13 (cli) (built: Nov 16 2021 21:58:37) ( ZTS Visual C++ 2019 x64 )
Copyright (c) The PHP Group
Zend Engine v4.0.13, Copyright (c) Zend Technologies
```

Figure 4.3 PHP version

Figure 4.3 shows that PHP version 8.0.13 has been installed on the computer.

ii. Composer is installed.

```
C:\Users\user>composer -v

Composer version 2.5.4 2023-02-15 13:10:06

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display help for the given command. When no command is given display help for the list
  -q, --quiet                Do not output any message
  -V, --version              Display this application version
  --ansi|--no-ansi          Force (or disable --no-ansi) ANSI output
  -n, --no-interaction      Do not ask any interactive question
  --profile                  Display timing and memory usage information
  --no-plugins               Whether to disable plugins.
  --no-scripts               Skips the execution of all scripts defined in composer.json file.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache                 Prevent use of the cache
  -v|vv|vvv, --verbose      Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and
  3 for debug

Available commands:
  about             Shows a short information about Composer
  archive           Creates an archive of this composer package
  audit             Checks for security vulnerability advisories for installed packages
  browse            Opens the package's repository URL or homepage in your browser
  bump             Increases the lower limit of your composer.json requirements to the currently installed versions
  check-platform-reqs Check that platform requirements are satisfied
  clear-cache       [clearcache|cc] Cleans composer's internal package cache
  completion        Dump the shell completion script
  config            Sets config options
  create-project    Creates new project from a package into given directory
  depends           [why] Shows which packages cause the given package to be installed
  diagnose          Diagnoses the system to identify common errors
  dump-autoload     [dumpautoload] Dumps the autoloader
```

Figure 4.4 Composer

Based on Figure 4.4, it appears that the Composer software has been successfully installed on the computer. By installing Composer on the development machine, the development team was able to take advantage of its powerful features and streamline the process of managing dependencies within the Cocurricular Management System (CMS) project.

Create Laravel project:

```
PS C:\xampp\htdocs\koko> composer create-project --prefer-dist laravel/laravel koko
```

Figure 4.5 Laravel Project command created

Figure 4.5 displays the specific command that must be executed in the terminal window of Visual Studio Code in order to create the Cocurricular Management System (CMS) using the Laravel Framework. By running this command, the development team was able to quickly and efficiently generate the basic structure of the CMS, including its directories, files, and essential components.

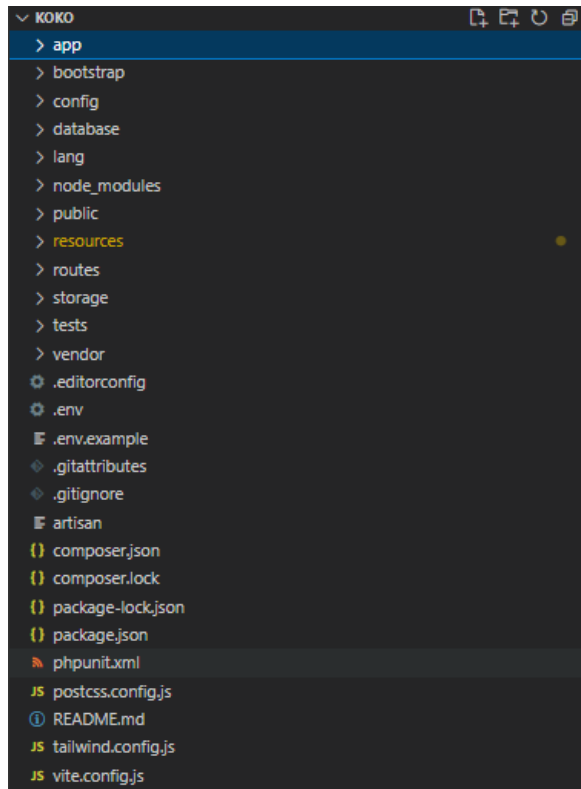


Figure 4.6 Laravel File

Figure 4.6 displays the file structure that was generated in the Cocurricular Management System (CMS) project prior to applying CSS and HTML code commands. This file structure represents the basic framework of the CMS and includes directories and files for essential components such as controllers, models, views, and routes.

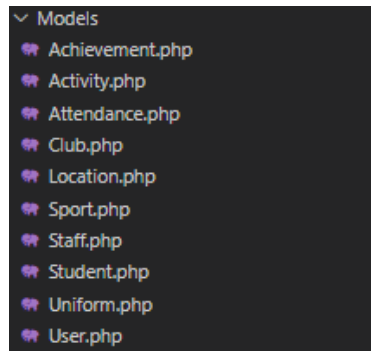


Figure 4.7 CMS Laravel Models

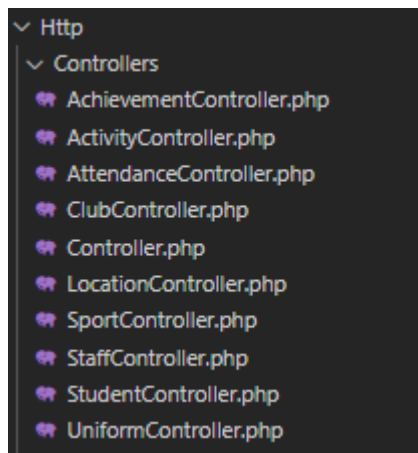


Figure 4.8 CMS Laravel Controllers

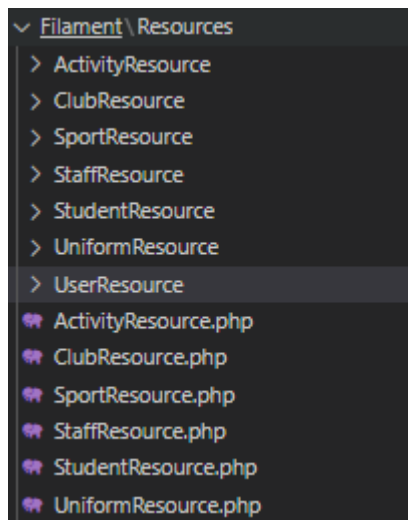


Figure 4.9 CMS Laravel Resources

4.2.1 Interface

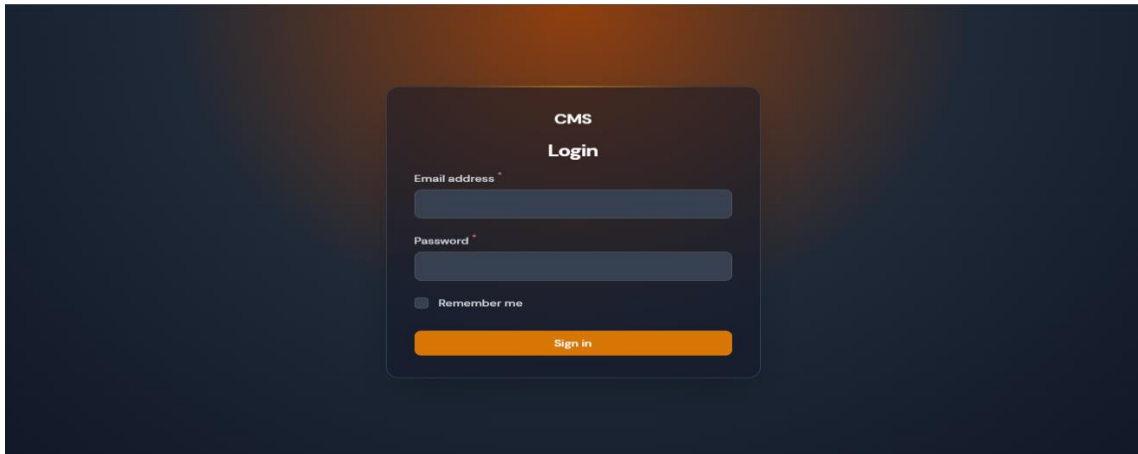


Figure 4.10 Login Interface

Figure 4.10 displays the login interface for the Cocurricular Management System (CMS), which allows users to enter their username and password in order to access the system. This interface serves as the gateway to the CMS and ensures that only authorized users are able to access its features and functionality.

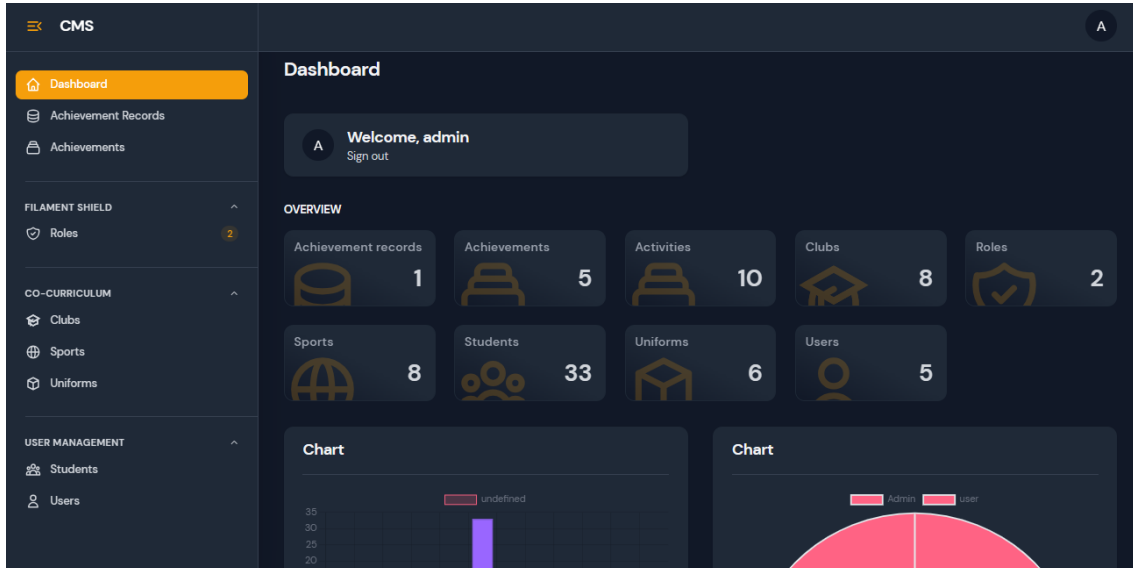


Figure 4.11 Dashboard Interface

Figure 4.11 displays the main interface of the Cocurricular Management System (CMS), which is the primary screen that users will see after they have successfully logged into the system. This interface provides an overview of the CMS and allows users to access various reports and tools related to cocurricular activities.

The screenshot shows the 'Clubs' interface in a CMS. The left sidebar includes 'Dashboard', 'Achievements', 'CO-CURRICULUM' (with 'Clubs' selected), 'Sports', 'Uniforms', and 'USER MANAGEMENT' (with 'Students' selected). The main content area is titled 'Clubs' and features a 'New club' button. Below the title is a filter for 'All Club' with a count of '8'. A search bar is present above a table with the following data:

Name	Person In Charge	Created at	Updated at	View	Edit
Sains dan Matematik	Khairin Chan	May 11, 2023 09:56:32	Jun 8, 2023 06:04:40	View	Edit
Alam Sekitar	Fyruz Ismat	Jun 8, 2023 06:03:51	Jun 8, 2023 06:05:00	View	Edit
Robotik	Fyruz Ismat	Jun 8, 2023 06:05:21	Jun 8, 2023 06:05:21	View	Edit
Usahawan Muda	Khairin Chan	Jun 8, 2023 06:05:40	Jun 8, 2023 06:05:40	View	Edit
Reka Cipta dan Inovasi	Fyruz Ismat	Jun 8, 2023 06:06:12	Jun 8, 2023 06:06:12	View	Edit
Pencegahan Jenayah	Fyruz Ismat	Jun 8, 2023 06:06:41	Jun 8, 2023 06:06:41	View	Edit
Kesenian, Kebudayaan dan Kraftangan	Fyruz Ismat	Jun 8, 2023 06:07:14	Jun 8, 2023 06:07:14	View	Edit
Keselamatan Jalan Raya	Fyruz Ismat	Jun 8, 2023 06:07:42	Jun 8, 2023 06:07:42	View	Edit

At the bottom of the table, it says 'Showing 1 to 8 of 8 results' and '10 per page'.

Figure 4.12 Clubs Interface

Figure 4.12 displays the clubs interface of the Cocurricular Management System (CMS), which is a key component of the system that allows users to manage and track cocurricular clubs in the school. This interface provides a comprehensive overview of all the cocurricular activities that have been registered within the system, and allows users to view, edit, delete, or create these clubs. There is a button to create a new uniforms club in the top right website, and a search function is in the table to make the users easier to use and search.

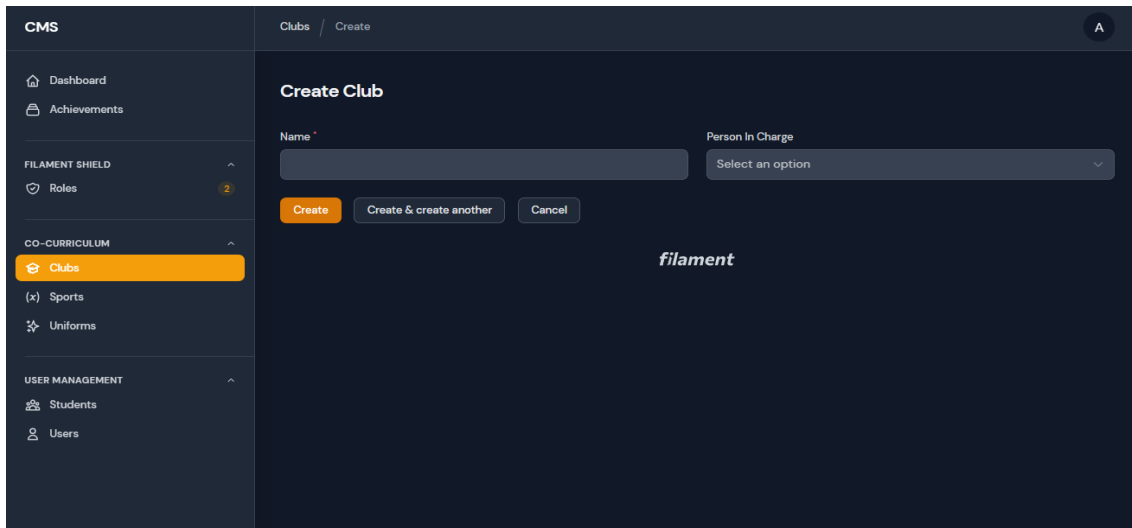


Figure 4.13 Create Clubs Interface

Figure 4.13 displays creates the clubs interface of the Cocurricular Management System (CMS), which is a key component of the system that allows users to create cocurricular clubs in the school. The user must enter the club's name and the staff id who will be in charge of the club.

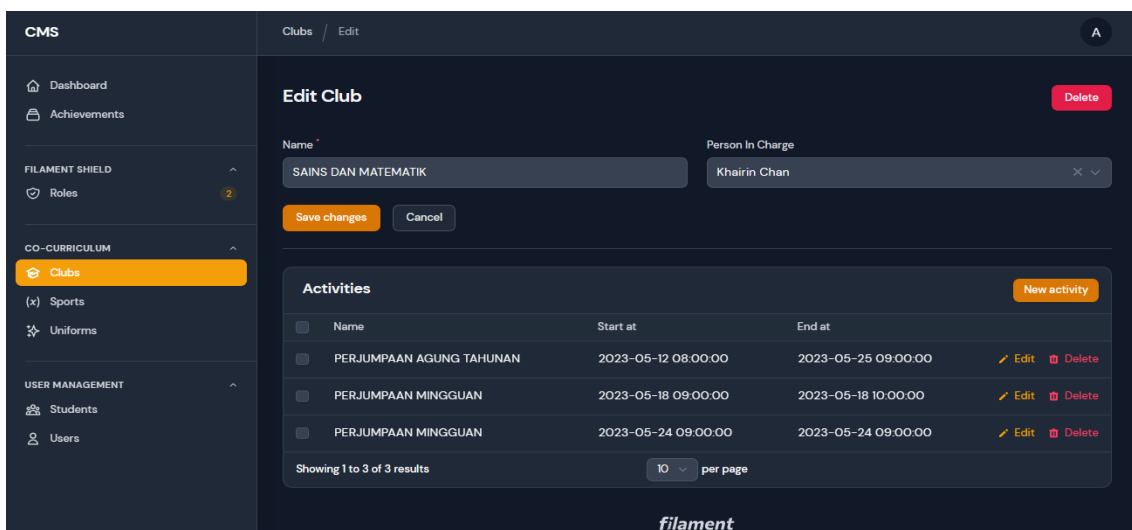


Figure 4.14 Edit and delete Club Interface

Figure 4.14 displays the view, edit, and delete club interface function. Admin can edit the name, and the person in charge of the club. Then after finishing the edit admin can save the change. There is also a button for deleting on the top right side red colour button, if admin want to delete the club, the delete message will be pop up on the web site.

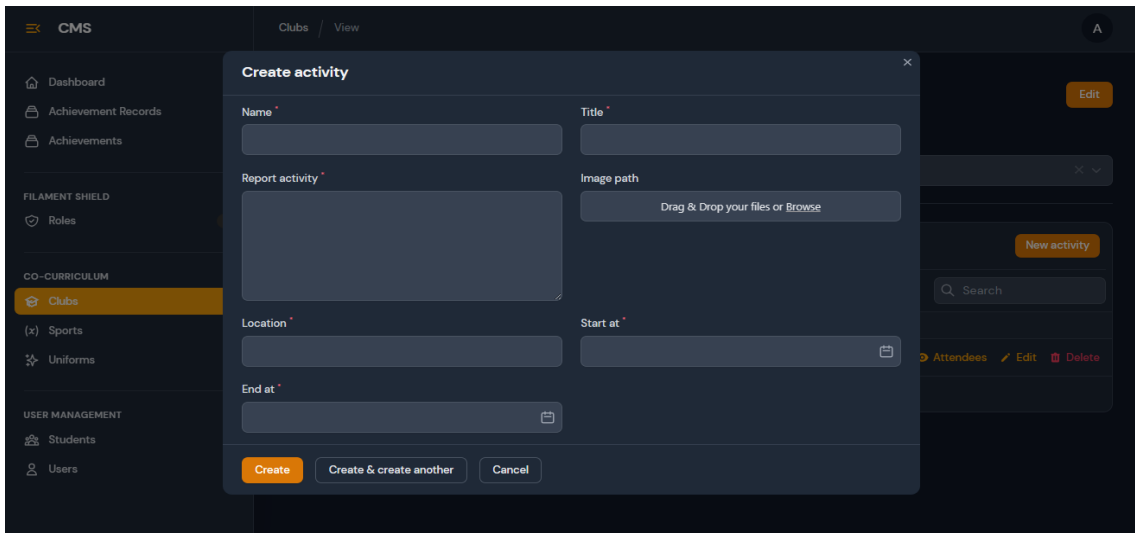


Figure 4.15 Create Club Activities Interface

Figure 4.15 displays create the club activities interface of the Cocurricular Management System (CMS), a vital component of the system that allows users to develop co-curricular club activities in the school. The user must enter the activity name, title, report activity, image for proven, location and the start and end of activities in the club. After input has been filled, the user or admin needs to enter the create button; the input data will show in the table.

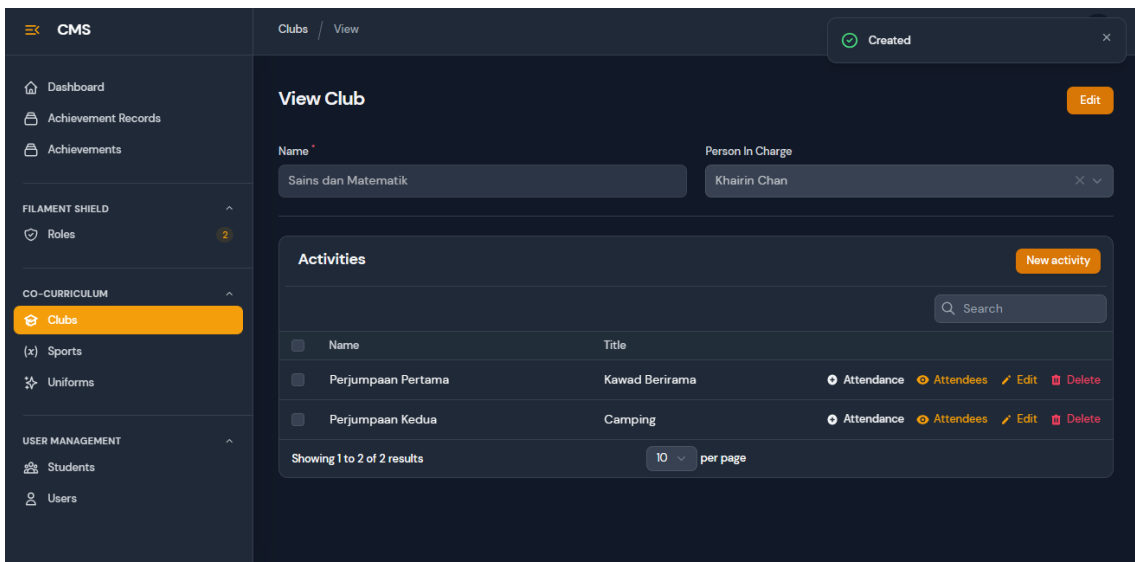


Figure 4.16 Club List Activities Interface

Figure 4.16 displays the list of activities interface for the club of the Cocurricular Management System (CMS), a key component of the system that allows users to manage

and track co-curricular activities within the school. This interface provides a comprehensive overview of all the co-curricular activities registered within the system, enabling users to edit or delete these activities as necessary. The activity module is inside the co-curricular module, where Admin will create the club, sports, and uniform, and then the staff will make the activities for each club in their days. This module is in the club relation manager resources.

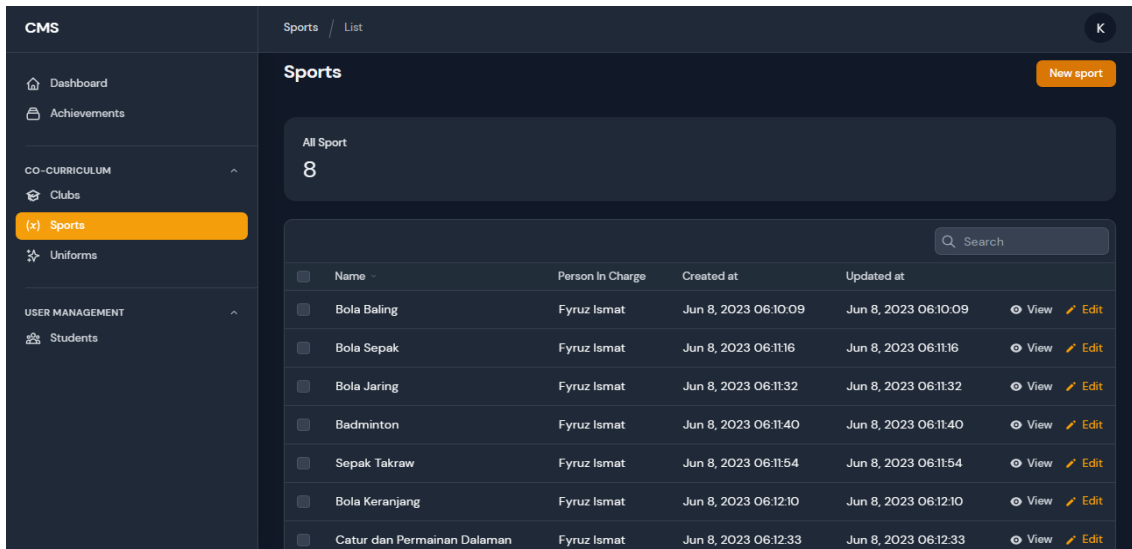


Figure 4.17 Sports Interface

Figure 4.15 displays the sports interface of the Cocurricular Management System (CMS), which is a key component of the system that allows users to manage and track cocurricular sports in the school. This interface provides a comprehensive overview of all the sports cocurricular that have been registered within the system, and allows users to view, edit, delete, or create these sports club. There is a button to create a new uniforms club in the top right website, and a search function is in the table to make the users easier to use and search.

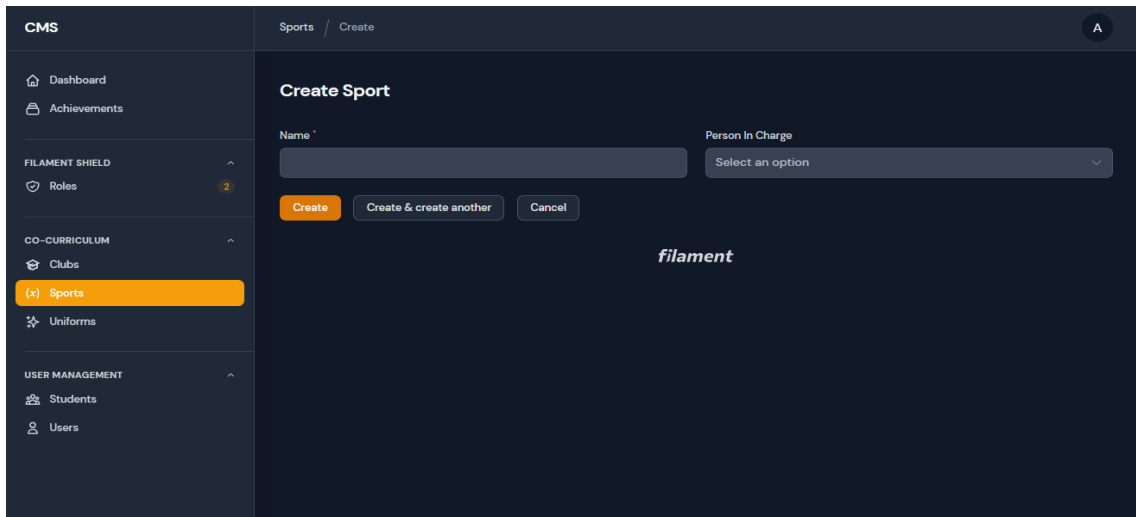


Figure 4.18 Create Sports Interface

Figure 4.18 displays creates the sports interface of the Cocurricular Management System (CMS), which is a key component of the system that allows users to create sports cocurricular in the school. The user must enter the sport's name and the staff id who will be in charge of the sport club.

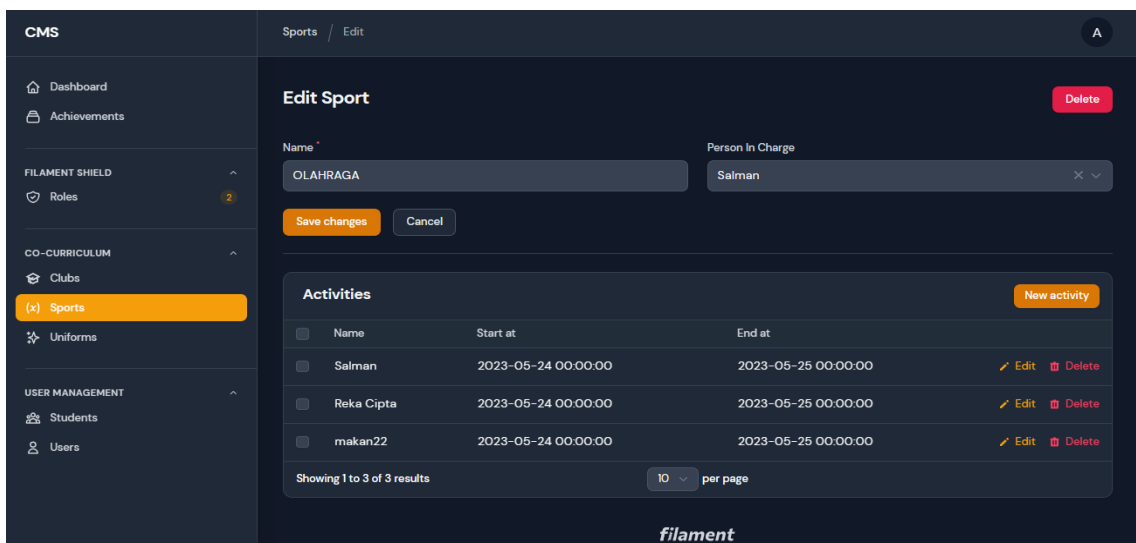


Figure 4.19 Edit and delete Sport Interface

Figure 4.19 displays the view, edit, and delete sport interface function. Admin can edit the name, and the person in charge of the sport club. Then after finishing the edit admin can save the change. There is also a button for deleting on the top right side red colour button, if admin want to delete the sport club, the delete message will be pop up on the web site.

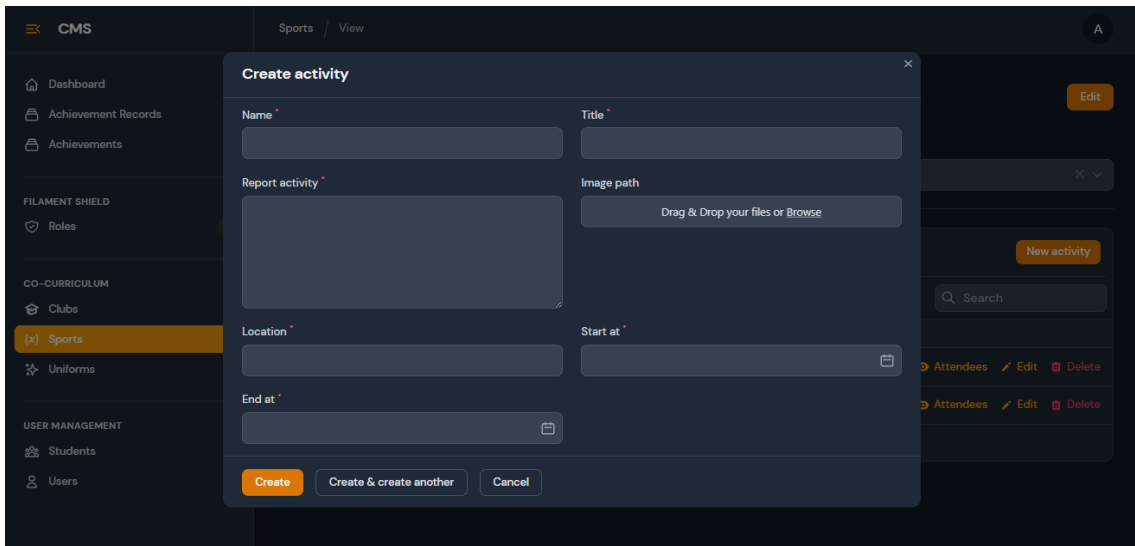


Figure 4.20 Create Sport Activities Interface

Figure 4.20 displays create the sport activities interface of the Cocurricular Management System (CMS), a vital component of the system that allows users to develop co-curricular uniform activities in the school. The user must enter the activity name, title, report activity, image for proven, location and the start and end of activities in the sport club. After input has been filled, the user or admin needs to enter the create button; the input data will show in the table.

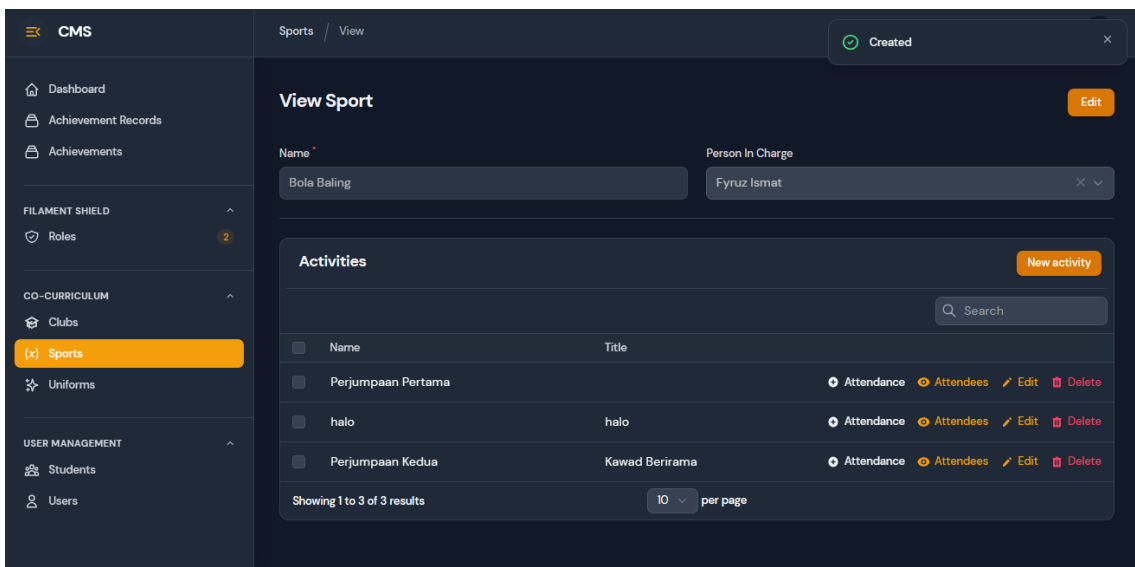


Figure 4.21 Sport List Activities Interface

Figure 4.21 displays the list of activities interface for the club of the Cocurricular Management System (CMS), a key component of the system that allows users to manage and track co-curricular activities within the school. This interface provides a comprehensive overview of all the co-curricular activities registered within the system, enabling users to edit or delete these activities as necessary. The activity module is inside the co-curricular module, where Admin will create the club, sports, and uniform, and then the staff will make the activities for each club in their days. This module is in the sport relation manager resources.

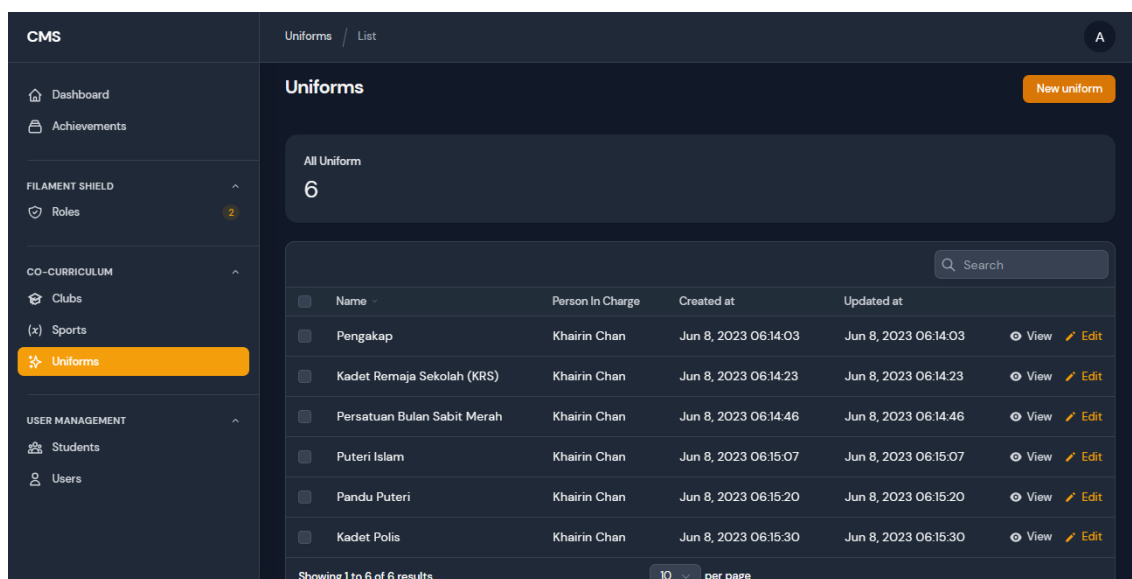


Figure 4.22 Uniformal Interface

Figure 4.22 displays the uniforms interface of the Cocurricular Management System (CMS), which is a key component of the system that allows users to manage and track cocurricular uniforms in the school. This interface provides a comprehensive overview of all the sports cocurricular that have been registered within the system, and allows users to view, edit, delete, or create these cocurricular uniforms club. There is a button to create a new uniforms club in the top right website, and a search function is in the table to make the users easier to use and search.

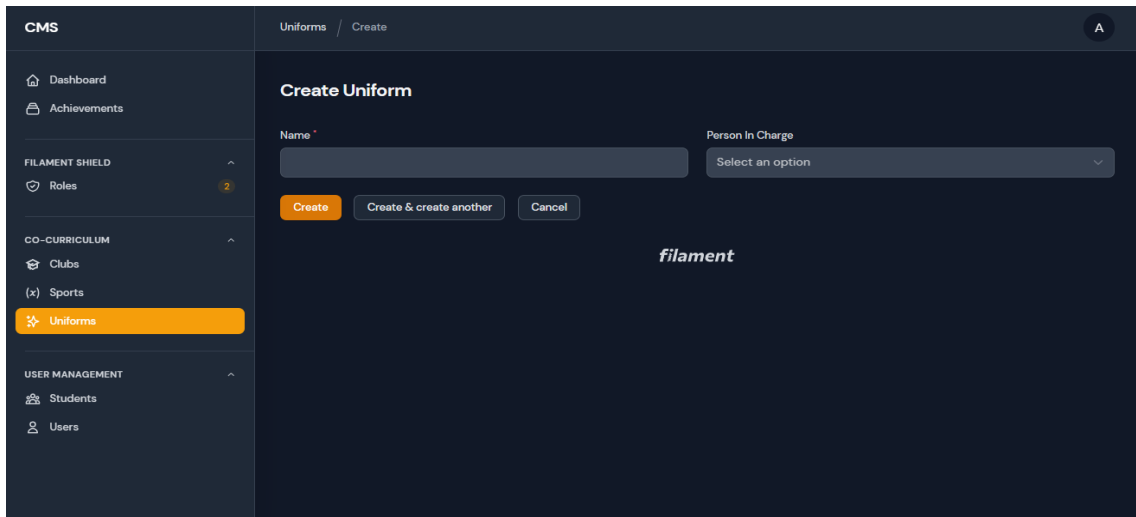


Figure 4.23 Create Uniforms Interface

Figure 4.23 displays creates the uniforms interface of the Cocurricular Management System (CMS), which is a key component of the system that allows users to create cocurricular uniforms in the school. The user must enter the uniform's name and the staff id who will be in charge of the uniform club.

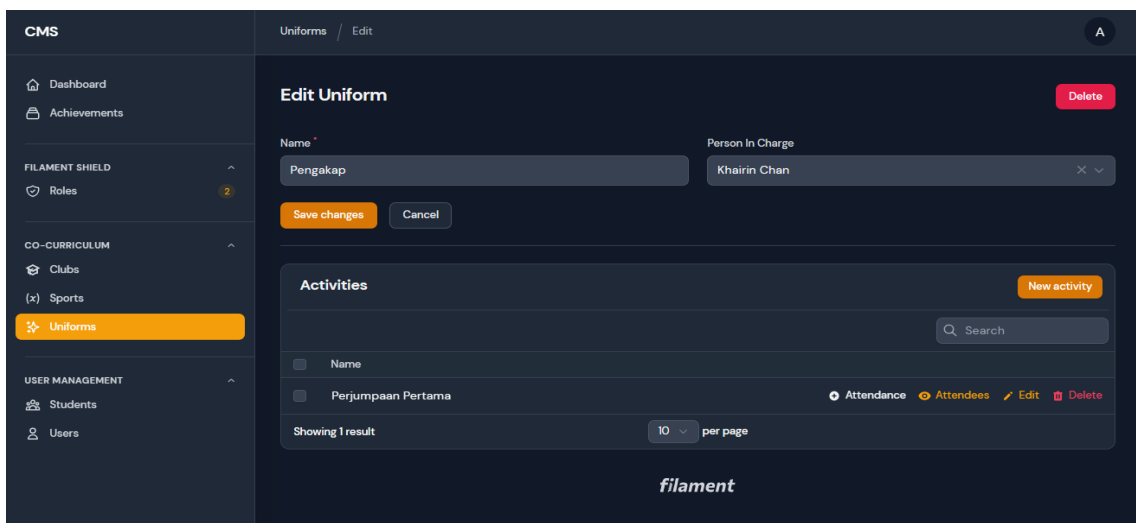


Figure 4.24 Edit and Delete Uniform Interface

Figure 4.24 displays the view, edit, and delete uniform interface function. Admin can edit the name, and the person in charge of the sport club. Then after finishing the edit admin can save the change. There is also a button for deleting on the top right side red colour button, if admin want to delete the sport club, the delete message will be pop up on the web site.

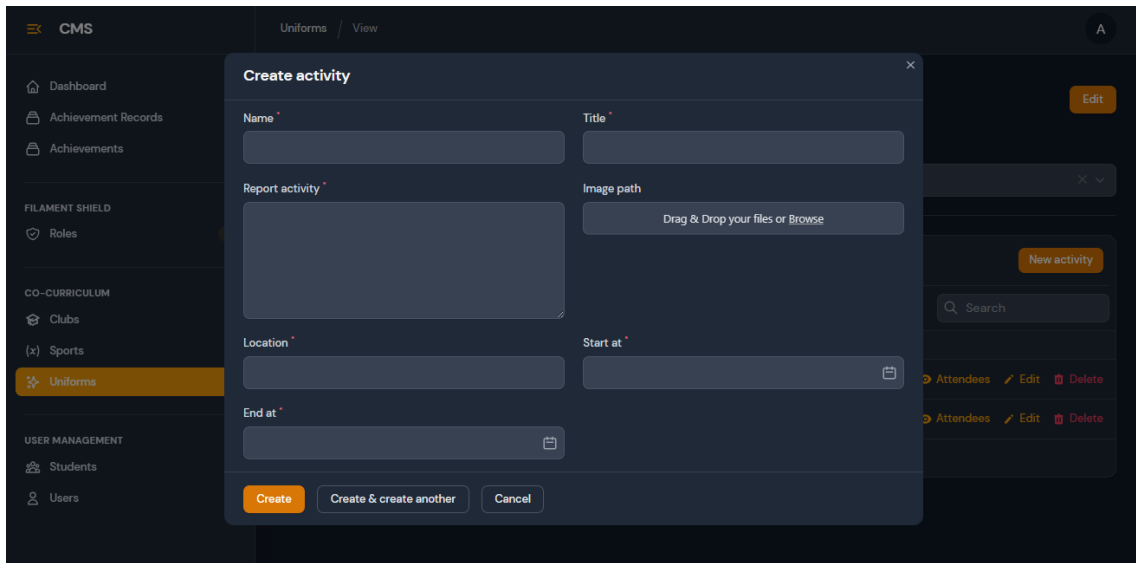


Figure 4.25 Create Uniform Activities Interface

Figure 4.25 displays create the uniform activities interface of the Cocurricular Management System (CMS), a vital component of the system that allows users to develop co-curricular uniform activities in the school. The user must enter the activity name, title, report activity, image for proven, location and the start and end of activities in the uniform club. After input has been filled, the user or admin needs to enter the create button; the input data will show in the table.

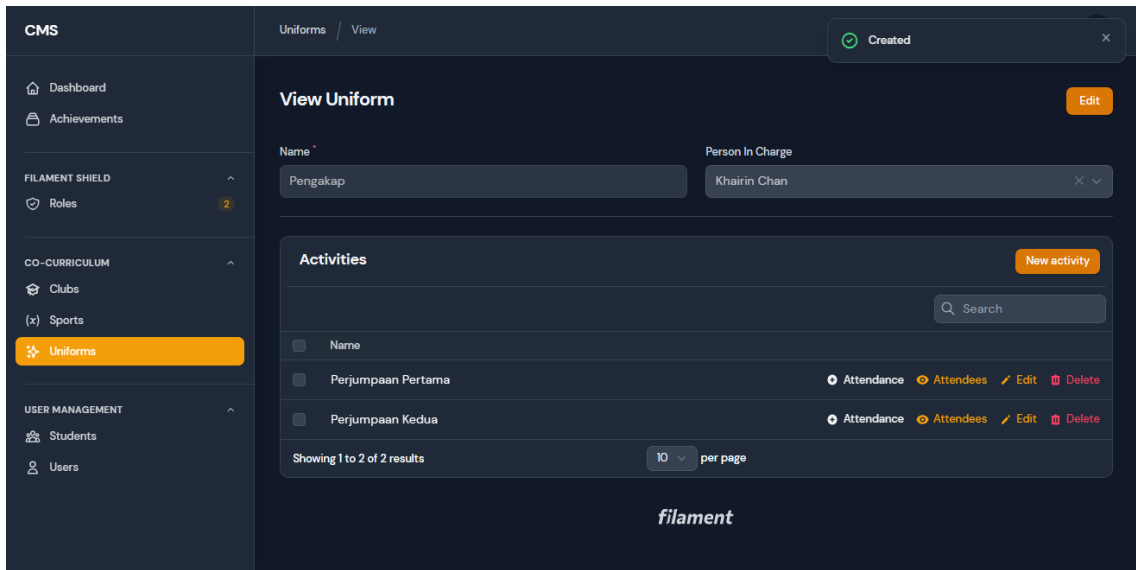


Figure 4.26 Uniform List Activities Interface

Figure 4.26 displays the list of activities interface for the uniform club of the Cocurricular Management System (CMS), a key component of the system that allows users to manage and track co-curricular activities within the school. This interface provides a comprehensive overview of all the co-curricular activities registered within the system, enabling users to edit or delete these activities as necessary. The activity module is inside the co-curricular module, where Admin will create the club, sports, and uniform, and then the staff will make the activities for each club in their days. This module is in the uniform relation manager resources.

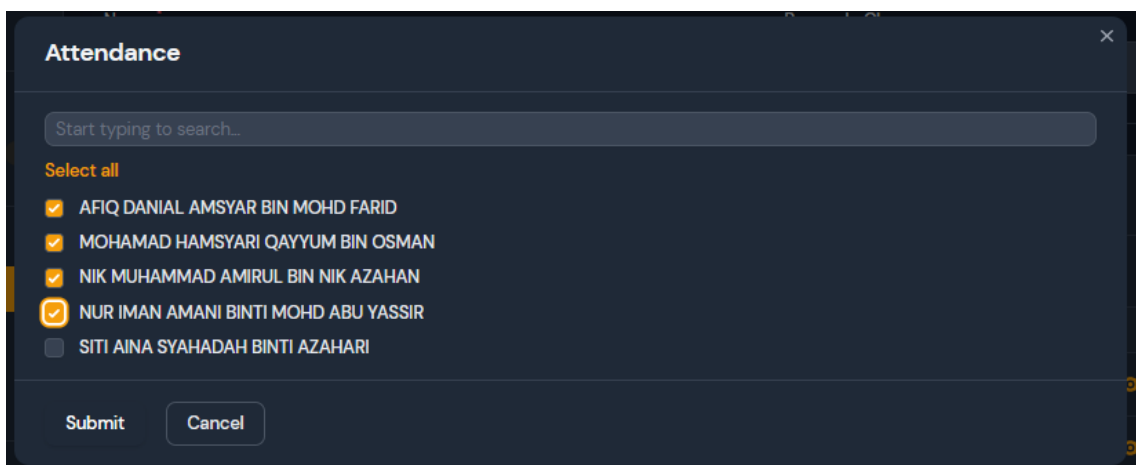


Figure 4.27 Attendance Function

Figure 4.27 shows the attendance Interface Function. The attendance interface allows teachers to mark the attendance of students participating in any club activity. The interface provides a user-friendly table where teachers can easily navigate and mark attendance. To mark attendance, the teacher must tick the radio button corresponding to each student who attends the activity and then click the submit button to save the attendance record. If the teacher needs to make changes to the attendance, they can click the back button in the table, allowing them to edit the attendance for any student by following the same steps. The interface efficiently captures and records students' attendance, ensuring accurate tracking of their participation in club activities. The attendance information will be displayed in the attendees function beside the attendance button, providing a comprehensive overview of which students attended the particular activity.

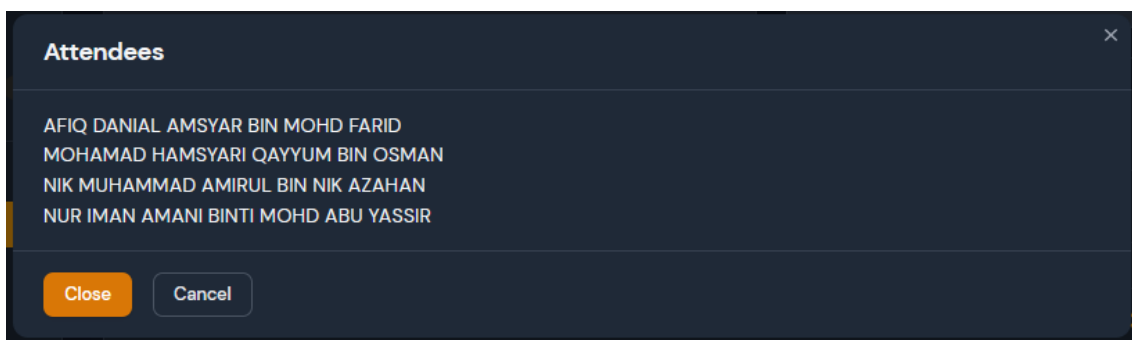


Figure 4.28 View Attendees Student

Figure 4.28 shows the view attendees student Function. The attendee's function in any club activity is designed to allow teachers to view the students who attended the activity. This function retrieves and displays the output of the attendance that has been marked in the attendance interface. By accessing the attendee's function, teachers can quickly and conveniently access a comprehensive list of students at the club activity. This information provides valuable insights into student participation and helps evaluate club events' effectiveness and engagement. The attendee's function is a convenient tool for teachers to monitor and analyze student attendance, enabling them to make informed decisions and adjustments for future activities.

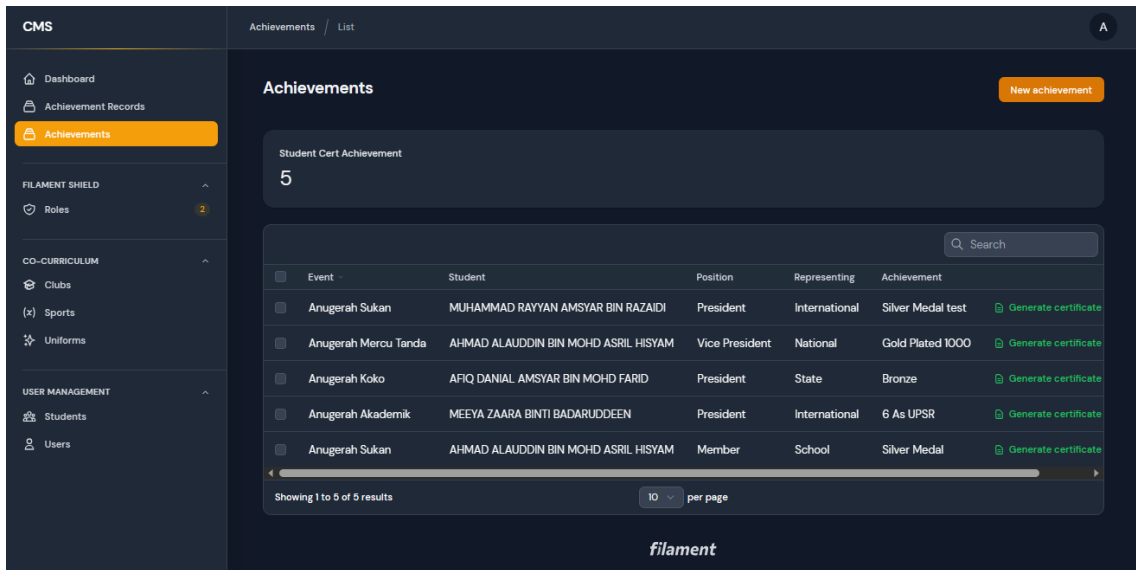


Figure 4.29 Achievement Interface

Figure 4.29 displays the achievements interface of the Cocurricular Management System (CMS), which is a key component of the system that allows users to manage and track cocurricular achievement for the students in the school. This interface provides a comprehensive overview of all the achievements cocurricular that have been create by the user for the student within the system, and allows users to view, edit, delete, or create these club achievements.

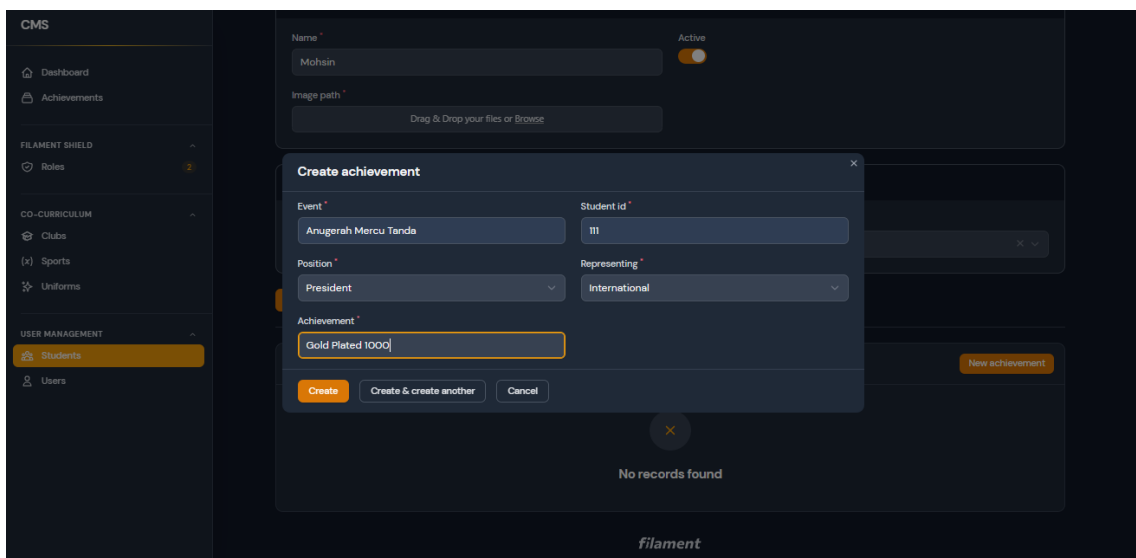


Figure 4.30 Create Achievement Interface

Figure 4.30 displays creates achievement interface of the Cocurricular Management System (CMS), which is a key component of the system that allows admin to create cocurricular achievement for the student in the school. The user or admin need to fill the required form such as event, position, representing, and achievement to create the achievement. After the admin creates the achievement, the created message will pop up on the website.

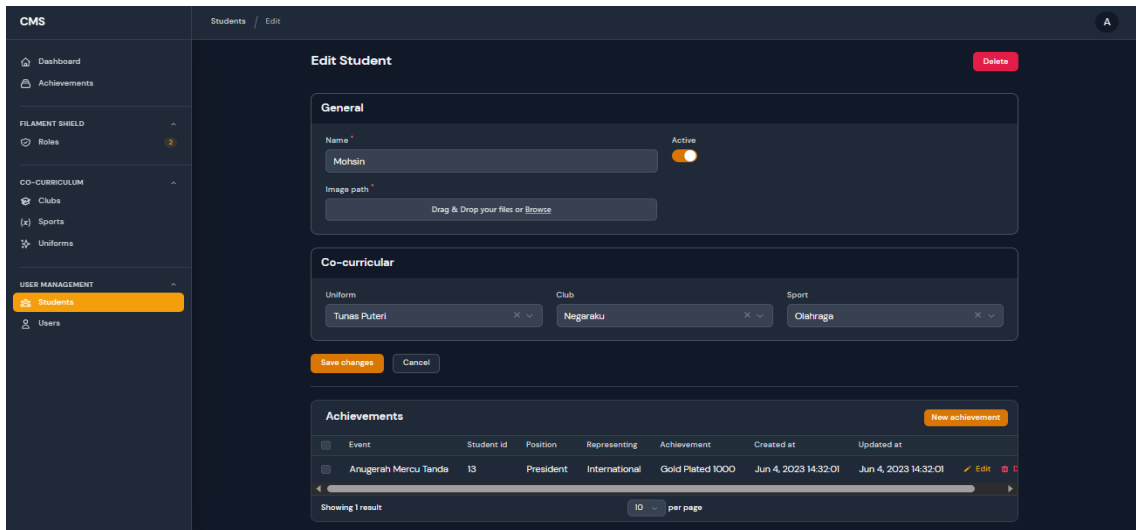


Figure 4.31 Edit and delete Achievement Interface

Figure 4.31 depicts the view, edit, and delete achievement interface module. Admin and users can edit the event, position, representing, and achievement of the students achievement. Then after finishing the edit admin can save the change. There is also a button for deleting on the top right side red colour button, if admin want to delete the sport club, the delete message will be pop up on the web site.

Student	Position	Representing	Achievement	
MUHAMMAD RAYYAN AMSYAR BIN RAZAIDI	President	International	Silver Medal test	Generate certificate Edit
AHMAD ALAUDDIN BIN MOHD ASRIL HISYAM	Vice President	National	Gold Plated 1000	Generate certificate Edit
AFIQ DANIAL AMSYAR BIN MOHD FARID	President	State	Bronze	Generate certificate Edit
MEEYA ZAARA BINTI BADARUDEEN	President	International	6 As UPSR	Generate certificate Edit

Showing 1 to 4 of 4 results 10 per page

Figure 4.32 Generate Certificate Function

Figure 4.32 shows the generate certificate function is crucial in generating a co-curricular certificate for a student who has participated in an extracurricular event organized by a school. This function encompasses several essential steps that ensure the certificate is personalized and accurately reflects the student's involvement in the event. The admin can click the button, and then it will bring to the certificate page that was auto-generated after the admin insert the input form in the create achievement for the student.

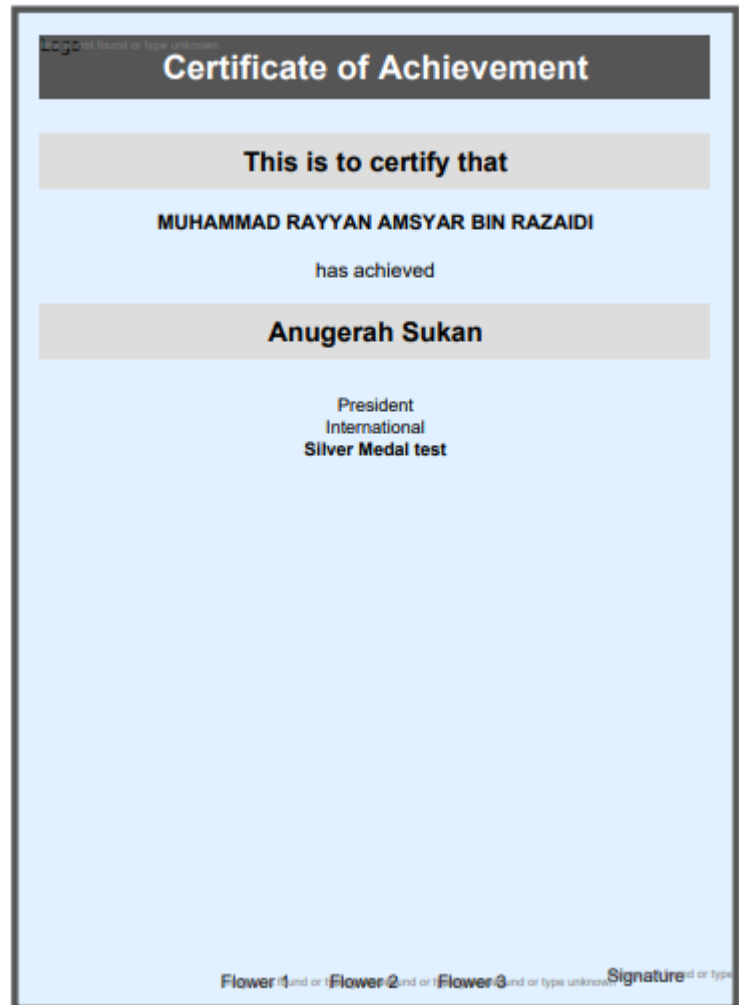


Figure 4.33 Achievement Certificate

Figure 4.33 depicts the Achievement Certificate interface. The certificate interface is designed to generate certificates after the administrator inputs the required information in the create achievement form. This interface allows the administrator to provide the necessary details, such as the student's name, event or activity name, and any additional relevant information. Once the form is submitted, the certificate generation process is triggered, creating a personalized certificate for the student who participated in the co-curricular event or activity. This interface streamlines the certificate generation process and ensures that students receive their certificates efficiently, recognizing their achievements in co-curricular pursuits.

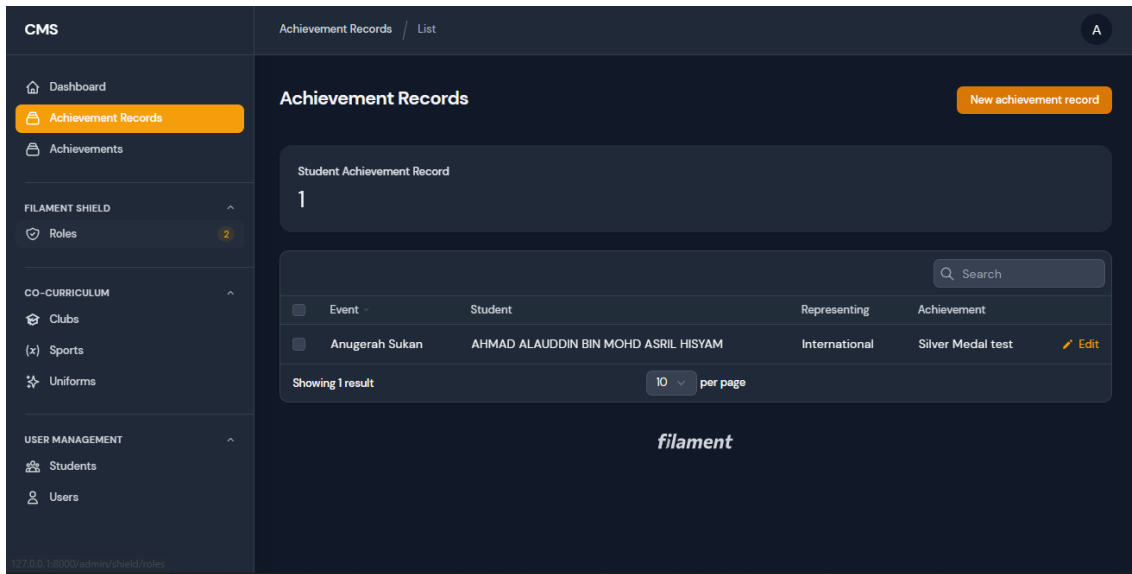


Figure 4.34 Achievement Record Interface

Figure 4.34 displays the achievements record interface of the Cocurricular Management System (CMS), a critical component of the system that allows users to manage and track the co-curricular achievement record for the students in the school. This interface differs from the achievements interface because it's only accessed by the user and doesn't have a generating function. This interface provides a comprehensive overview of all the achievements recorded co-curricular outside the school created by the user for the student within the system and allows users to view, edit, delete, or make these club achievements.

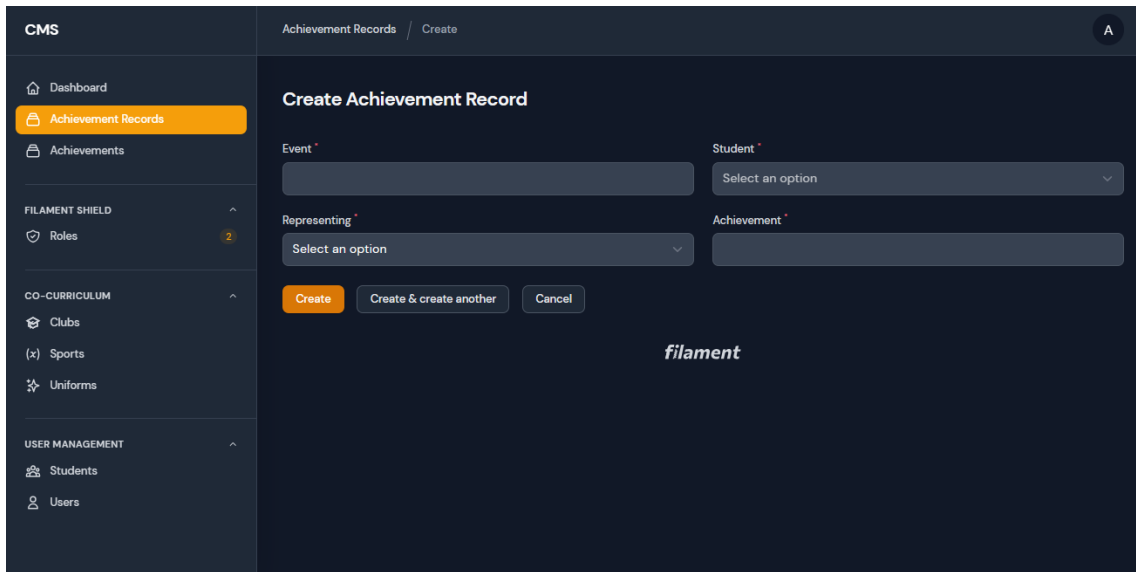


Figure 4.35 Create Achievement Record Interface

Figure 4.35 displays creates achievement record interface of the Cocurricular Management System (CMS), which is a key component of the system that allows admin to create cocurricular achievement for the student in the school. The user or admin need to fill the required form such as event, position, representing, and achievement to create the achievement record for extra curricular outside the school event. After the admin creates the achievement record, the created message will pop up on the website.

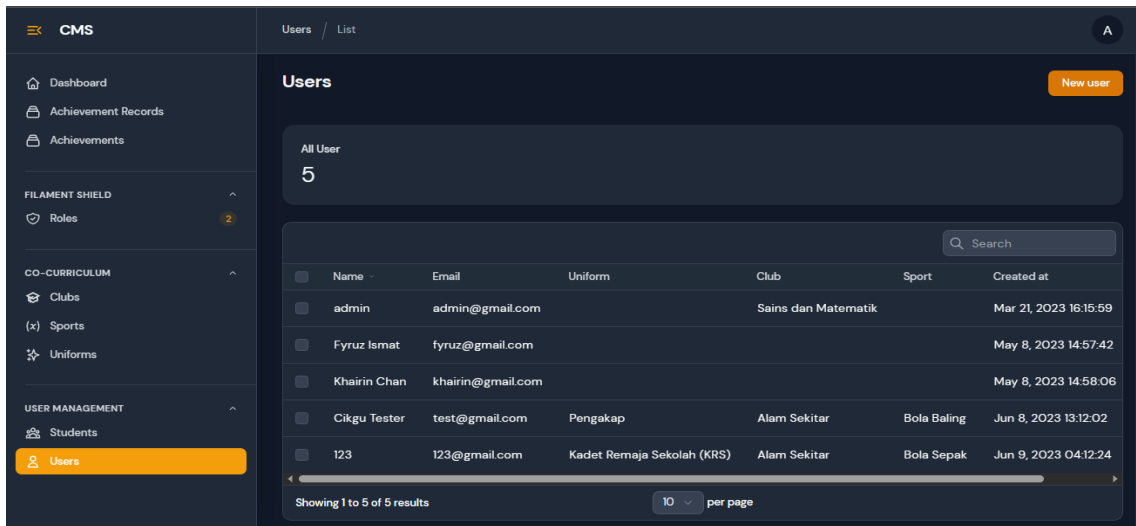


Figure 4.36 User Staff Interface

Figure 4.36 displays the staff interface and information of the Cocurricular Management System (CMS), which is a key component of the system that allows users to manage and track cocurricular staff in the school. This interface provides a comprehensive overview of all the staffs information that have been registered within the system, and allows users to edit, delete, or create these staffs information.

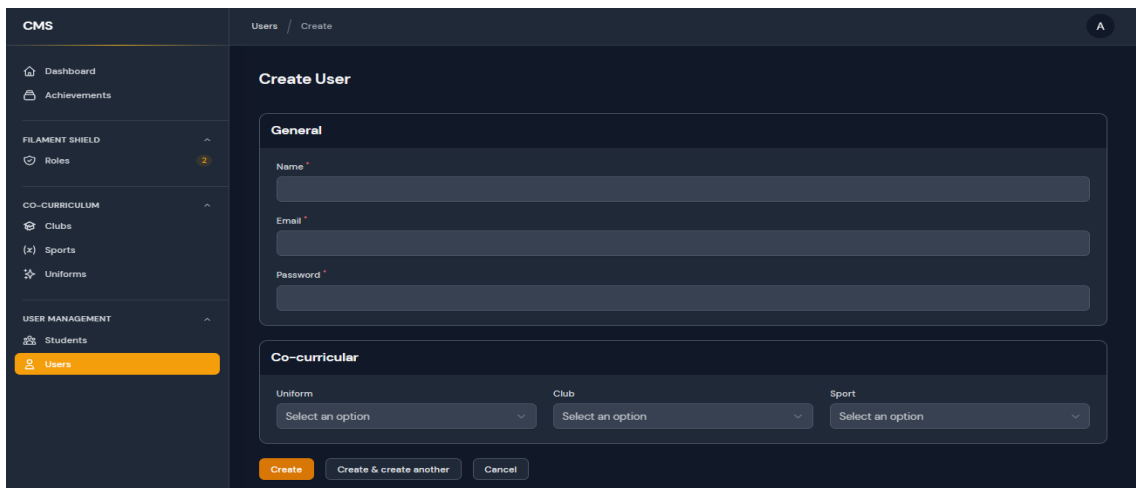


Figure 4.37 Create Staff Interface

Figure 4.37 displays creates staff interface of the Cocurricular Management System (CMS), which is a key component of the system that allows admin to create cocurricular staffs in the school. After the admin creates the staff, the created message will pop up on the website. Admin need to fill in all the input and choose the relation manager co-curricular function.

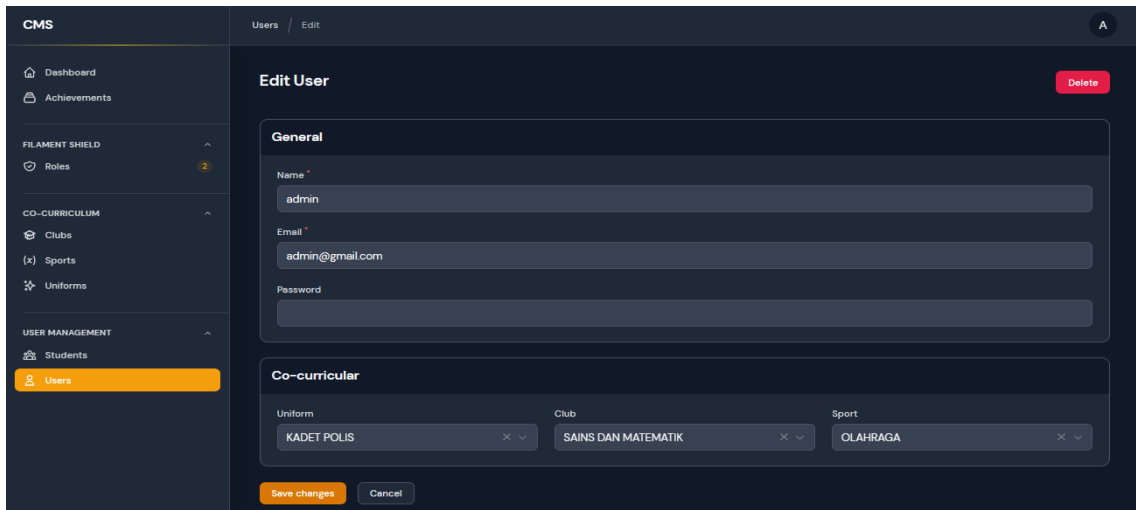


Figure 4.38 Edit and Delete Staff Interface

Figure 4.38 displays the view, edit, and delete staff interface function. Admin can edit the name, email, and password of the staff. Admin also can correct the staff cocurricular if wrong or wanna change the cocurricular club. Then after finishing the edit admin can save the change. There is also a button for deleting on the top right side red colour button, if admin want to delete the student the delete message will be pop up on the web site.

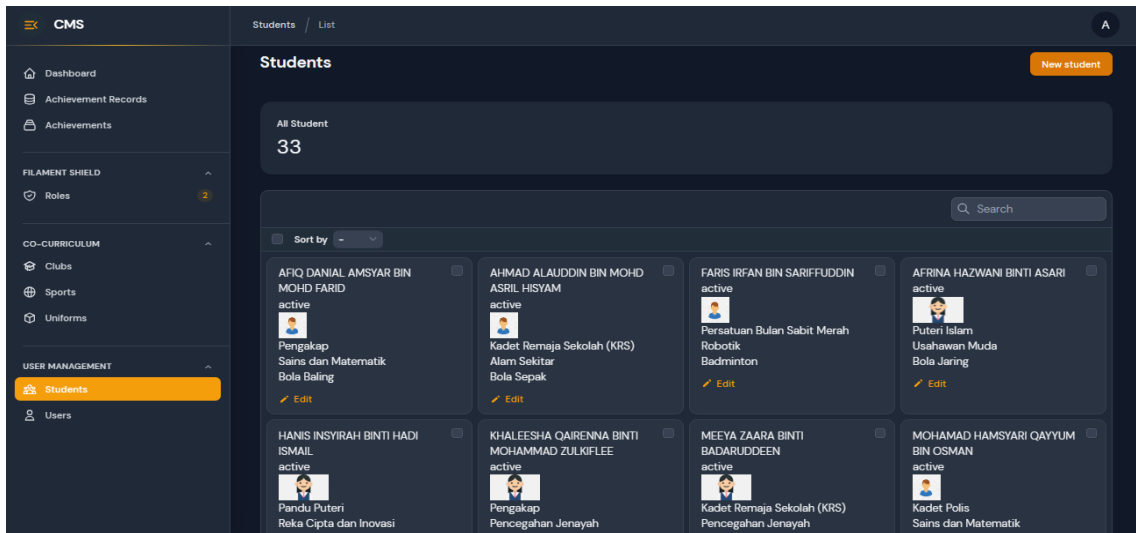


Figure 4.39 Students Interface

Figure 4.39 displays the staff interface and information of the Cocurricular Management System (CMS), which is a key component of the system that allows users to manage and track cocurricular students in the school. This interface provides a comprehensive overview of all the student information that have been registered within the system, and allows users to edit, delete, or create these students information.

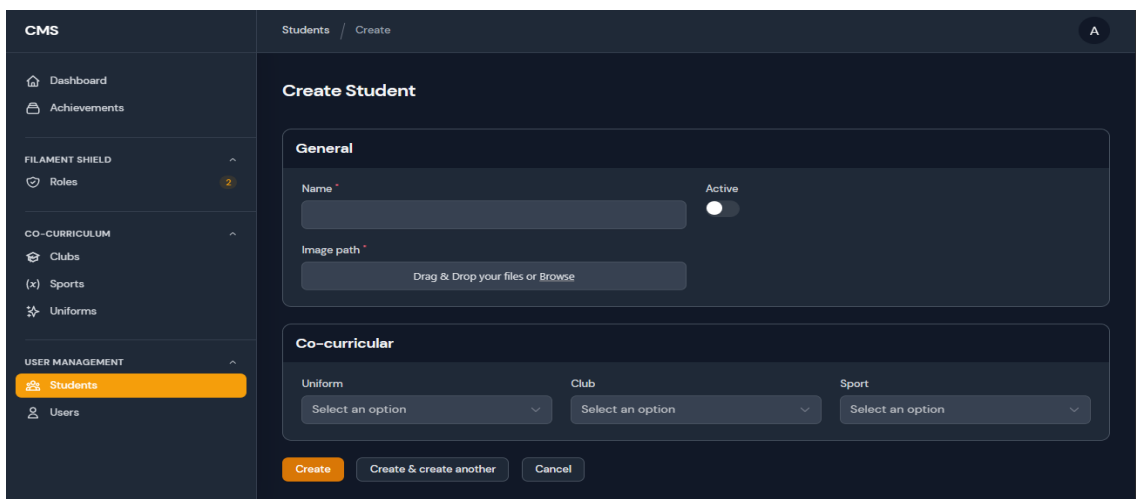


Figure 4.40 Create Students Interface

Figure 4.40 displays creates students interface of the Cocurricular Management System (CMS), which is a key component of the system that allows admin to create cocurricular students in the school. Admin must enter the name, the student status and the image path for student information. The student cocurricular admin will choose the club using the drop-down input function. After the admin creates the student, the created message will pop up on the website.

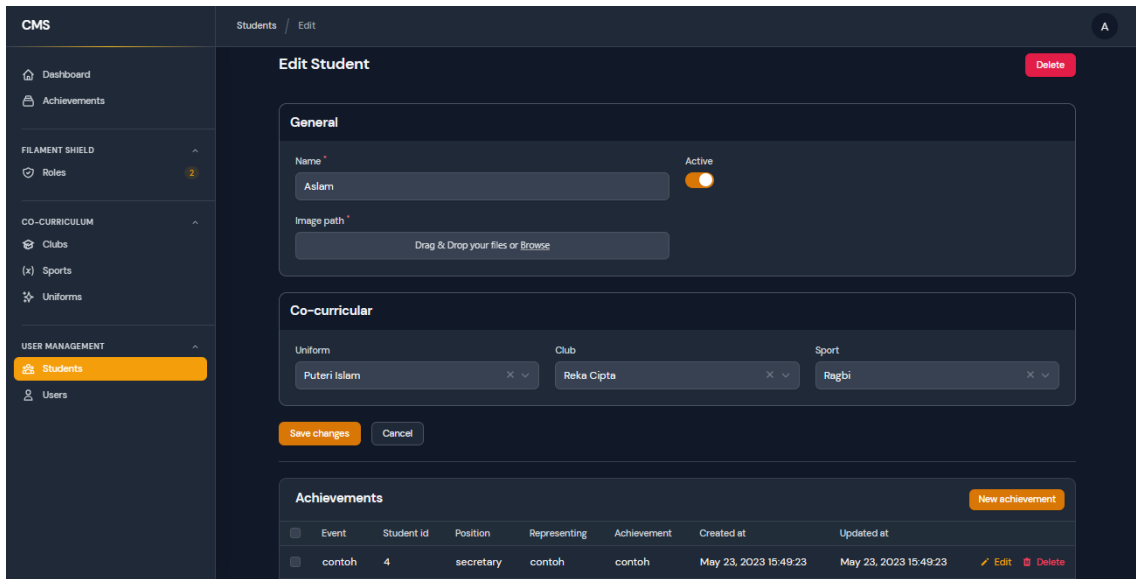


Figure 4.41 Edit and Delete Student Interface

Figure 4.41 shows the view, edit, and delete student interface function. Admin and staff can edit the name, status, and image of the student. Admin and staff also can correct the student cocurricular if wrong or wanna change the cocurricular club. Then after finishing the edit admin can save the change. There is also a button for deleting on the top right side red colour button, if admin want to delete the student the delete message will be pop up on the web site.

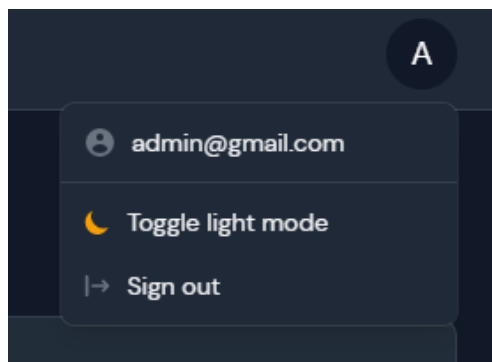
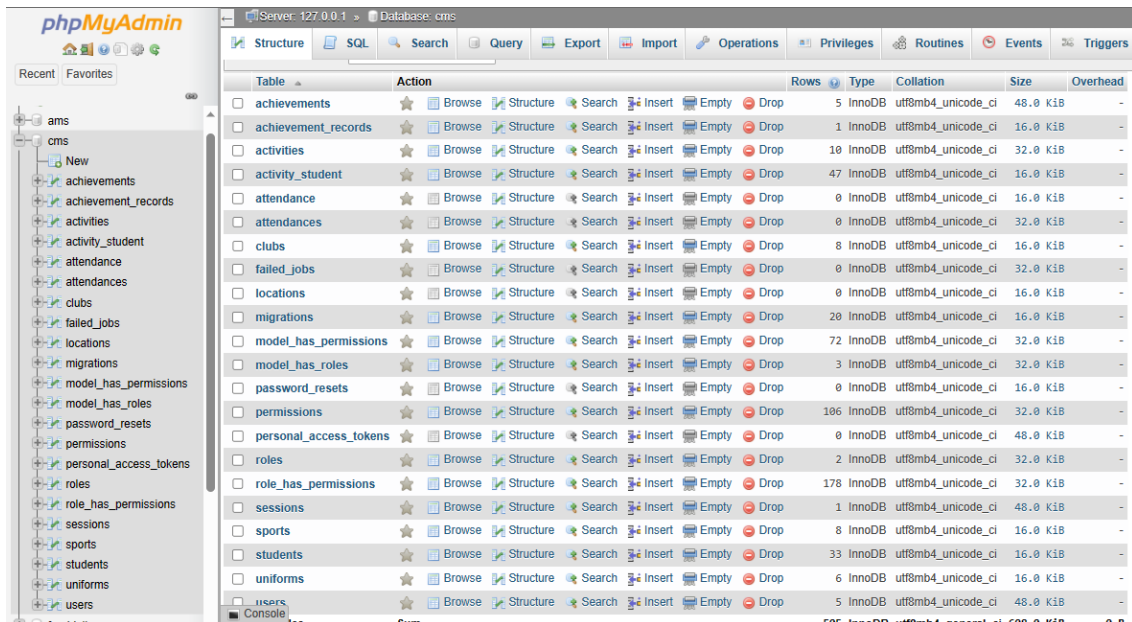


Figure 4.42 Dark Mode Features

Figure 4.42 depict the Cocurricular Management System (CMS) includes a dark mode feature that allows users to switch between the default light mode and a dark mode option. This feature is designed to provide users with a more comfortable and visually appealing experience, particularly when using the CMS for extended periods of time.

4.2.2 Database



The screenshot displays the phpMyAdmin interface for a database named 'cms'. The left sidebar shows a tree view of the database structure, including tables like 'achievements', 'attendance', 'clubs', 'migrations', 'permissions', 'roles', 'sessions', 'sports', 'students', and 'uniforms'. The main panel shows a table structure view with columns: Table, Action, Rows, Type, Collation, Size, and Overhead. The table list includes:

Table	Action	Rows	Type	Collation	Size	Overhead
achievements	5	InnoDB	utf8mb4_unicode_ci	48.0 K	KiB	-
achievement_records	1	InnoDB	utf8mb4_unicode_ci	16.0 K	KiB	-
activities	10	InnoDB	utf8mb4_unicode_ci	32.0 K	KiB	-
activity_student	47	InnoDB	utf8mb4_unicode_ci	16.0 K	KiB	-
attendance	0	InnoDB	utf8mb4_unicode_ci	16.0 K	KiB	-
attendances	0	InnoDB	utf8mb4_unicode_ci	32.0 K	KiB	-
clubs	8	InnoDB	utf8mb4_unicode_ci	16.0 K	KiB	-
failed_jobs	0	InnoDB	utf8mb4_unicode_ci	32.0 K	KiB	-
locations	0	InnoDB	utf8mb4_unicode_ci	16.0 K	KiB	-
migrations	20	InnoDB	utf8mb4_unicode_ci	16.0 K	KiB	-
model_has_permissions	72	InnoDB	utf8mb4_unicode_ci	32.0 K	KiB	-
model_has_roles	3	InnoDB	utf8mb4_unicode_ci	32.0 K	KiB	-
password_resets	0	InnoDB	utf8mb4_unicode_ci	16.0 K	KiB	-
permissions	106	InnoDB	utf8mb4_unicode_ci	32.0 K	KiB	-
personal_access_tokens	0	InnoDB	utf8mb4_unicode_ci	48.0 K	KiB	-
roles	2	InnoDB	utf8mb4_unicode_ci	32.0 K	KiB	-
role_has_permissions	178	InnoDB	utf8mb4_unicode_ci	32.0 K	KiB	-
sessions	1	InnoDB	utf8mb4_unicode_ci	48.0 K	KiB	-
sports	8	InnoDB	utf8mb4_unicode_ci	16.0 K	KiB	-
students	33	InnoDB	utf8mb4_unicode_ci	16.0 K	KiB	-
uniforms	6	InnoDB	utf8mb4_unicode_ci	16.0 K	KiB	-
users	5	InnoDB	utf8mb4_unicode_ci	48.0 K	KiB	-

Figure 4.43 CMS Database

Figure 4.43 displays the comprehensive database utilized by the Co-Curricular Management System. This diagram encompasses all the databases employed within the system. It represents the various data sources and structures used to store and manage the information related to co-curricular activities, events, student records, achievements, attendance, and other pertinent details. The database serves as a central repository, ensuring efficient storage and retrieval of data for the smooth functioning of the Co-Curricular Management System. It enables administrators, teachers, and other authorized users to access and manipulate the relevant information necessary for managing co-curricular activities effectively. By utilizing this database, the system can maintain accurate and up-to-date records, facilitate data analysis, and support decision-making processes related to co-curricular engagements within the educational institution.

4.2.3 Interface Coded

```
public static function form(Form $form): Form
{
    return $form
        ->schema([
            Section::make('General')
                ->schema([
                    Forms\Components\TextInput::make('name')
                        ->required()
                        ->maxLength(255),
                    Forms\Components\TextInput::make('email')
                        ->email()
                        ->required()
                        ->maxLength(255),
                    Forms\Components\TextInput::make('password')
                        ->password()
                        ->required(fn (Page $livewire): bool => $livewire instanceof CreateRecord)
                        ->minLength(8)
                        ->same('passwordConfirmation')
                        ->dehydrated(fn ($state) => filled($state))
                        ->dehydrateStateUsing(fn ($state) => Hash::make($state)),
                    Forms\Components\TextInput::make('passwordConfirmation')
                        ->password()
                        ->required()
                        ->dehydrated(false),
                ]),
            Section::make('Co-curricular')
                ->schema([
                    Select::make('uniform_id')
                        ->label('Uniform')
                        ->options(Uniform::all()->pluck('name', 'id'))
                        ->searchable()
                        ->required(),
                    Select::make('club_id')
                        ->label('Club')
                        ->options(Club::all()->pluck('name', 'id'))
                        ->searchable()
                        ->required(),
                    Select::make('sport_id')
                        ->label('Sport')
                        ->options(Sport::all()->pluck('name', 'id'))
                        ->searchable()
                        ->required(),
                ])->columns(3),
        ]);
}
0 references | 0 overrides
public static function table(Table $table): Table
{
    return $table
        ->columns([
            Tables\Columns\TextColumn::make('name')->searchable()->sortable(),
            Tables\Columns\TextColumn::make('email'),
            Tables\Columns\TextColumn::make('uniform.name')
                ->label('Uniform'),
            Tables\Columns\TextColumn::make('club.name')
                ->label('Club'),
            Tables\Columns\TextColumn::make('sport.name')
                ->label('Sport'),
            Tables\Columns\TextColumn::make('created_at')
                ->dateTime(),
        ])
        ->filters([
            //
        ])
        ->actions([
            Tables\Actions\EditAction::make(),
        ])
        ->bulkActions([
            Tables\Actions\DeleteBulkAction::make(),
        ])
};
```

Figure 4.44 Coding for User Resource

```

public static function form(Form $form): Form
{
    return $form
        ->schema([
            Section::make('General')
                ->schema([
                    Forms\Components\TextInput::make('name')
                        ->required()
                        ->maxLength(255),

                    Toggle::make('status')
                        ->label('Active')
                        ->inline(false),

                    FileUpload::make('image_path')
                        ->required()
                        ->disk('logo')
                        ->required()
                        ->imagePreviewHeight('250')
                        ->loadingIndicatorPosition('left')
                        ->panelAspectRatio('2:1')
                        ->panelLayout('integrated')
                        ->removeUploadedFileButtonPosition('right')
                        ->uploadButtonPosition('left')
                        ->uploadProgressIndicatorPosition('left')
                ])
                ->columns(2),

            Section::make('Co-curricular')
                ->schema([
                    Select::make('uniform_id')
                        ->required()
                        ->label('Uniform')
                        ->options(Uniform::all()->pluck('name', 'id'))
                        ->searchable(),

                    Select::make('club_id')
                        ->required()
                        ->label('Club')
                        ->options(Club::all()->pluck('name', 'id'))
                        ->searchable(),

                    Select::make('sport_id')
                        ->required()
                        ->label('Sport')
                        ->options(Sport::all()->pluck('name', 'id'))
                        ->searchable(),
                ])
                ->columns(3),
        ]);
}

0 references | 0 overrides
public static function table(Table $table): Table
{
    return $table
        ->columns([
            Tables\Columns\TextColumn::make('name')->searchable()->sortable(),
            Tables\Columns\TextColumn::make('status')->getStateUsing(function ($record) {
                return $record->status ? ('active') : ('not active');
            }),
            Tables\Columns\ImageColumn::make('image_path')->disk('logo'),
            Tables\Columns\TextColumn::make('uniform.name'),
            Tables\Columns\TextColumn::make('club.name'),
            Tables\Columns\TextColumn::make('sport.name'),
        ])
        ->filters([
            //
        ])
        ->actions([
            Tables\Actions\EditAction::make(),
        ])
        ->bulkActions([
            Tables\Actions\DeleteBulkAction::make(),
        ])
}

```

Figure 4.45 Coding for Student Resource

```

public static function form(Form $form): Form
{
    return $form
        ->schema([
            Forms\Components\TextInput::make('name')
                ->translateLabel()
                ->required()
                ->maxLength(255),
            Select::make('user_id')
                ->label('Person In Charge')
                ->options(User::all()->pluck('name', 'id'))
                ->searchable(),
        ]);
}

0 references | 0 overrides
public static function table(Table $table): Table
{
    return $table
        ->columns([
            Tables\Columns\TextColumn::make('name')->searchable()->sortable(),
            Tables\Columns\TextColumn::make('user_id')
                ->label('Person In Charge')
                ->formatStateUsing(fn ($state): ?string => User::find($state)?->name ?? null),
            Tables\Columns\TextColumn::make('created_at')
                ->dateTime(),
            Tables\Columns\TextColumn::make('updated_at')
                ->dateTime(),
        ])
        ->filters([
            //
        ])
        ->actions([
            Tables\Actions\ViewAction::make(),
            Tables\Actions>EditAction::make(),
        ])
        ->bulkActions([
            Tables\Actions>DeleteBulkAction::make(),
        ]);
}

```

Figure 4.46 Coding for Club Resource

```

public static function form(Form $form): Form
{
    return $form
        ->schema([
            Forms\Components\TextInput::make('name')
                ->translateLabel()
                ->required()
                ->maxLength(255),
            Select::make('user_id')
                ->label('Person In Charge')
                ->options(User::all()->pluck('name', 'id'))
                ->searchable(),
        ]);
}

0 references | 0 overrides
public static function table(Table $table): Table
{
    return $table
        ->columns([
            Tables\Columns\TextColumn::make('name')->searchable()->sortable(),
            Tables\Columns\TextColumn::make('user_id')
                ->label('Person In Charge')
                ->formatStateUsing(fn ($state): ?string => User::find($state)?->name ?? null),
            Tables\Columns\TextColumn::make('created_at')
                ->dateTime(),
            Tables\Columns\TextColumn::make('updated_at')
                ->dateTime(),
        ])
        ->filters([
            //
        ])
        ->actions([
            Tables\Actions\ViewAction::make(),
            Tables\Actions>EditAction::make(),
        ])
        ->bulkActions([
            Tables\Actions>DeleteBulkAction::make(),
        ]);
}
}

```

Figure 4.47 Coding for Sport Resource

```

public static function form(Form $form): Form
{
    return $form
        ->schema([
            Forms\Components\TextInput::make('name')
                ->translateLabel()
                ->required()
                ->maxLength(255),
            Select::make('user_id')
                ->label('Person In Charge')
                ->options(User::all()->pluck('name', 'id'))
                ->searchable(),
        ]);
}
0 references | 0 overrides
public static function table(Table $table): Table
{
    return $table
        ->columns([
            Tables\Columns\TextColumn::make('name')->searchable()->sortable(),
            Tables\Columns\TextColumn::make('user_id')
                ->label('Person In Charge')
                ->formatStateUsing(fn ($state): ?string => User::find($state)?->name ?? null),
            Tables\Columns\TextColumn::make('created_at')
                ->dateTime(),
            Tables\Columns\TextColumn::make('updated_at')
                ->dateTime(),
        ])
        ->filters([])
        //
        ->actions([
            Tables\Actions\ViewAction::make(),
            Tables\Actions>EditAction::make(),
        ])
        ->bulkActions([
            Tables\Actions>DeleteBulkAction::make(),
        ]);
}

```

Figure 4.48 Coding for Uniform Resource


```

public static function form(Form $Form): Form
{
    return $form
        ->schema([
            Forms\Components\TextInput::make('event')
                ->required()
                ->maxLength(255),
            Forms\Components\Select::make('student_id')
                ->required()
                ->relationship('student', 'name')
                ->searchable(),
            Select::make('position')
                ->options([
                    'President' => 'President',
                    'Vice President' => 'Vice President',
                    'Secretary' => 'Secretary',
                    'Treasurer' => 'Treasurer',
                    'Member' => 'Member',
                ])
                ->required(),
            Select::make('representing')
                ->options([
                    'International' => 'International',
                    'National' => 'National',
                    'State' => 'State',
                    'District' => 'District',
                    'School' => 'School',
                ])
                ->required(),
            Forms\Components\TextInput::make('achievement')
                ->required()
                ->maxLength(255),
        ]);
}

```

```

public static function table(Table $table): Table
{
    return $table
        ->columns([
            Tables\Columns\TextColumn::make('event')->searchable()->sortable(),
            Tables\Columns\TextColumn::make('student.name'),
            Tables\Columns\TextColumn::make('position'),
            Tables\Columns\TextColumn::make('representing'),
            Tables\Columns\TextColumn::make('achievement'),
        ])
        ->filters([
            //
        ])
        ->actions([
            Tables\Actions\Action::make('Generate certificate')
                ->action(function (Achievement $record) {
                    return Redirect::to(route('certificate.show', ['id' => $record->id]));
                })
                ->icon('heroicon-o-document-text')
                ->color('success'),
            Tables\Actions\EditAction::make(),
        ])
        ->bulkActions([
            Tables\Actions\DeleteBulkAction::make(),
        ]);
}

```

Figure 4.49 Coding for Achievement Resource

```

public static function form(Form $form): Form
{
    return $form
        ->schema([
            Forms\Components\TextInput::make('event')
                ->required()
                ->maxLength(255),
            Forms\Components\Select::make('student_id')
                ->required()
                ->relationship('student', 'name')
                ->searchable(),
            Select::make('representing')
                ->options([
                    'International' => 'International',
                    'National' => 'National',
                    'State' => 'State',
                    'District' => 'District',
                    'School' => 'School',
                ])
                ->required(),
            Forms\Components\TextInput::make('achievement')
                ->required()
                ->maxLength(255),
        ]);
}

```

```

public static function table(Table $table): Table
{
    return $table
        ->columns([
            Tables\Columns\TextColumn::make('event')->searchable()->sortable(),
            Tables\Columns\TextColumn::make('student.name'),
            Tables\Columns\TextColumn::make('representing'),
            Tables\Columns\TextColumn::make('achievement'),
        ])
        ->filters([
            //
        ])
        ->actions([
            Tables\Actions\EditAction::make(),
        ])
        ->bulkActions([
            Tables\Actions\DeleteBulkAction::make(),
        ]);
}

```

Figure 4.50 Coding for Achievement Record Resource

```

public static function form(Form $form): Form
{
    return $form
        ->schema([
            Forms\Components\TextInput::make('name')
                ->required()
                ->maxLength(255),
            Forms\Components\TextInput::make('title')
                ->required()
                ->maxLength(255),
            Forms\Components\Textarea::make('report_activity')
                ->required()
                ->maxLength(500),
            Forms\Components\FileUpload::make('image_path')
                ->multiple(),
            Forms\Components\TextInput::make('location')
                ->required()
                ->maxLength(255),
            Forms\Components\DateTimePicker::make('start_at')
                ->required(),
            Forms\Components\DateTimePicker::make('end_at')
                ->required(),
        ]);
}

```

```

public static function table(Table $table): Table
{
    return $table
        ->columns([
            Tables\Columns\TextColumn::make('name')->searchable(),
            Tables\Columns\TextColumn::make('title'),
        ])
        ->filters([
            //
        ])
        ->headerActions([
            Tables\Actions\CreateAction::make(),
        ])
}

```

Figure 4.51 Coding for Activities Resource

```

->actions([
  Tables\Actions\Action::make('attendance')
    ->label('Attendance')
    ->color('blue')
    ->icon('heroicon-s-plus-circle')
    ->form([
      Forms\Components\CheckboxList::make('students')
        ->disableLabel()
        ->searchable()
        ->options(fn ($record) => Student::where('club_id', $record->club_id)->pluck('name', 'id'))
        ->bulkToggleable(),
    ])
    ->action(function ($data, $record) {
      $record->students()->sync($data['students']);
    }),

  Tables\Actions\Action::make('attendees')
    ->label('Attendees')
    ->icon('heroicon-s-eye')
    ->modalContent(function ($record) {
      $students = $record->students;

      $html = '';

      if (count($students)) {
        foreach ($students as $student) {
          $html .= $student->name . '</br>';
        }
      }
      return new HtmlString($html);
    })
    ->modalButton('Close')
    ->action(fn () => null),
  Tables\Actions\EditAction::make(),
  Tables\Actions>DeleteAction::make(),
]);

```

Figure 4.52 Coding for Attendance Function

4.3 Testing and Result Discussion

After the development of the Cocurricular Management System (CMS) is completed, a thorough testing process is carried out to ensure that the system functions as intended and is user-friendly.

The UAT is crucial for identifying any issues that may have been missed during the development process and ensuring that the system meets the needs and expectations of its intended users. In addition to the UAT, a usability test is also conducted to evaluate the ease of use and user satisfaction of the CMS. For this purpose, a UAT Form is created to gather feedback from staff regarding their experience using the web-based system. The questionnaire in the form is designed to assess various aspects of usability, such as the clarity of the interface, ease of navigation, and overall satisfaction with the system.

The feedback gathered from the testers reveals both pass and fail results that need improvement before delivering the fully functional Co-curricular Management System to the client. The analysis shows that 85% of the system passed the testing, while 15% failed, according to the testers. However, after finalization and further testing, the system has been improved based on the requirements provided by the testers and the client. By incorporating user feedback into the development process, the CMS can be continuously refined and enhanced to meet user needs better. The testing results can be referenced in Appendix A, which includes the testing form.

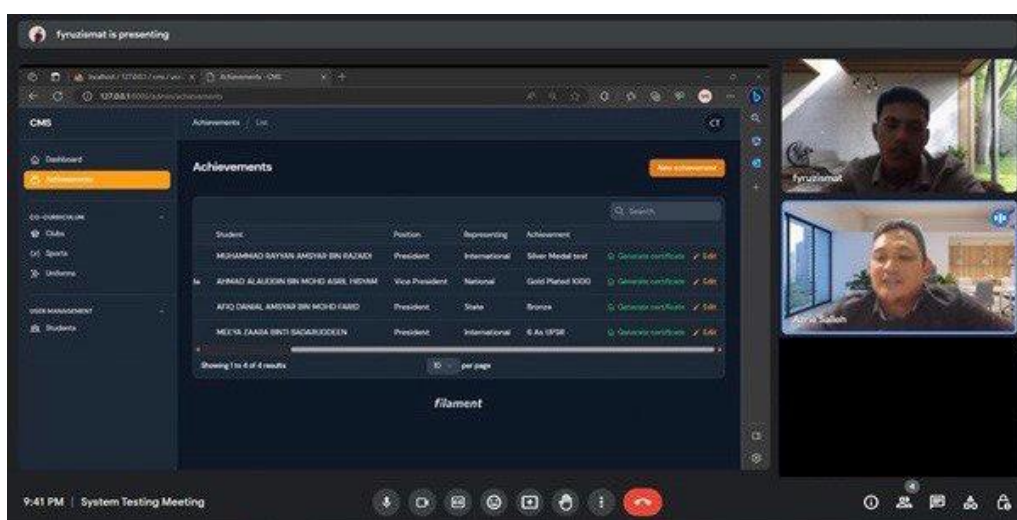


Figure 4.53 Meeting with Client

CHAPTER 5

CONCLUSION

5.1 Introduction

Chapter 5 provides an overview of the development process of the Cocurricular Management System (CMS), which is designed to help teachers and staff manage their cocurricular data more efficiently. The system is built using modern technologies like Laravel Framework, Figma, and Visual Studio Code to ensure it is scalable, user-friendly, and effective in achieving the objectives identified in Chapter 1.

The Waterfall methodology is employed in the development process, which is known for its highly structured and organized approach to project management. The chapter also summarizes the key features of the system, such as login and registration, user interfaces, adding and editing activities, and generating reports. Additionally, the testing process is highlighted as a critical aspect of the development process, including the User Acceptance Test (UAT) and usability testing, which helps to ensure the system meets the needs and expectations of its intended users.

5.2 Limitation and Constraint

i. Limitation of Time

The time constraints for the development of the CMS may force the team to focus on completing certain features while leaving others behind. As a result, critical functions or features may not be fully developed, which could compromise the system's effectiveness. The limited time may also restrict the development team's capacity to consider input from stakeholders or implement changes based on user testing.

Consequently, the project team must carefully manage their time to ensure the essential features receive the necessary attention and are developed optimally despite the time limitations.

ii. Limitation of resources

A lack of resources, including personnel, hardware, or software, can significantly affect the development of the Cocurricular Management System. This can result in delays, incomplete features, and inadequate testing, ultimately leading to a final product that may not meet user expectations. However, the development team can address these limitations by prioritizing critical features, managing time effectively, and utilizing available resources efficiently. In addition, alternative solutions such as open-source software or volunteer contributors can be considered to overcome resource constraints.

5.3 Future Work

There are few enhancements that can be applied for future improvement of the Cocurricular Management System (CMS).

i. Develop the CMS with mobile application for IOS and Android platform

ii. Collaborate with any organization to make sure that the system is known and trusted such as Ministry of High Education.

REFERENCES

- Amna Zulqadar. (2022, September 29). *SDLC Waterfall Model: The 6 phases you need to know about*. <https://rezaid.co.uk/sdlc-waterfall-model/>
- Engage EH. (2020). *Our School Management System | Engage School MIS*.
https://engagesmis.com/engage-mis/?utm_source=google&utm_medium=cpc&utm_term=student%20attendance%20management&gclid=Cj0KCCQiA8aOeBhCWARIsANRFrQF2xa_inbhRF5W8sRbDoe_Djw-GSuAfQ15on1nsqrZSFz2BsU3_KAAaAjvfEALw_wcB
- Fedena. (2021). *Importance of Student Attendance Management System - Fedena Blog*.
<https://fedena.com/blog/2021/09/importance-of-student-attendance-management-system.html>
- Hassan, S., & Zubair Asghar, M. (2015). *Web-based Attendance Management System CFP: BESC 2018 and Special Session on Social Computing for Business Intelligence and Analytics (SC-BI&A 2018) View project Hierarchical Temporal Memory model for sequence learning. View project*. <https://www.researchgate.net/publication/303023739>
- IceHrm. (2020, January 22). *Difficulties in Manual Attendance Tracking*.
<https://icehrm.com/blog/difficulties-in-manual-attendance-tracking/>
- Mekari Talenta. (2020). *Attendance Management System, Types, and Benefits*.
<https://www.talenta.co/blog/insight-talenta/attendance-management-system-types-and-benefits/>
- Naresh Narra. (2020). *Attendance – GoSchooler*. <https://goschooler.com/attendance/>
- National Center for Education Statistic. (2017, June). *Extracurricular Participation And Student Engagement*. <https://nces.ed.gov/pubs95/web/95741.asp>
- QuickSchools. (2020). *School Management Software Demo in 90-second*.
<https://www.quickschools.com/quickschools/quick-tour>
- Rob Watts. (2022). *What is Waterfall Methodology? – Forbes Advisor*.
<https://www.forbes.com/advisor/business/what-is-waterfall-methodology/>

APPENDIX A

Table 5.1 User Acceptance Test (UAT) form for User 1

No.	Module	Status		Comment
1.	Dashboard features and function for staff	Pass	Fail	Succes, good
2.	Login to the Sekolah Kebangsaan Mat Kilau CAMS	Pass	Fail	Succes, good
3.	Insert and edit co-curricular Activity and Achievement	Pass	Fail	Need more input data
4.	System Interface Design	Pass	Fail	Need some improvement and put background
5.	Delete data activity and achievement	Pass	Fail	Succes, good
6.	Search table function	Pass	Fail	Succes, good
7.	Mark Attendance student function	Pass	Fail	Succes, good

This test was performed by:

Name: Azrie bin Salleh

Date: 08/06/2023



AZRIE BIN SALLEH
PENOLONG KANAN PETANG
SK PENGKALAN TENTERA
25150 KUANTAN.

Signature:

Table 5.2 User Acceptance Test (UAT) form for User 2

No.	Module	Status		Comment
1.	Dashboard features and function for staff	Pass	Fail	Success, good
2.	Login to the Sekolah Kebangsaan Mat Kilau CAMS	Pass	Fail	Success, good
3.	Insert and edit co-curricular Activity and Achievement	Pass	Fail	Success, good
4.	System Interface Design	Pass	Fail	Success, good
5.	Delete data activity and achievement	Pass	Fail	Success, good
6.	Search table function	Pass	Fail	Success, good
7.	Mark Attendance student function	Pass	Fail	Success, good

This test was performed by:

Name: Hajah Marina binti Yaakob

Date: 08/06/2023



HJH MARINA BT HAJI YAAKOB
Guru Penolong Kanan HEM
SK Pengkalan Tentara
25150 Kuantan

Signature:

Table 5.3 User Acceptance Test (UAT) form for User 3

No.	Module	Status		Comment
1.	Dashboard features and function for staff	Pass	Fail	Success, good
2.	Login to the Sekolah Kebangsaan Mat Kilau CAMS	Pass	Fail	Success, good
3.	Insert and edit co-curricular Activity and Achievement	Pass	Fail	Success, good
4.	System Interface Design	Pass	Fail	Improvement the design
5.	Delete data activity and achievement	Pass	Fail	Success, good
6.	Search table function	Pass	Fail	Success, good
7.	Mark Attendance student function	Pass	Fail	Success, good

This test was performed by:

Name: Nor Azwani binti Mohd Azalli

Date: 08/06/2023




Signature:

Table 5.4 User Acceptance Test (UAT) form for User 4

No.	Module	Status		Comment
1.	Dashboard features and function for staff	Pass	Fail	Success, good
2.	Login to the Sekolah Kebangsaan Mat Kilau CAMS	Pass	Fail	Success, good
3.	Insert and edit co-curricular Activity and Achievement	Pass	Fail	Success, good
4.	System Interface Design	Pass	Fail	Success, good
5.	Delete data activity and achievement	Pass	Fail	Success, good
6.	Search table function	Pass	Fail	Success, good
7.	Mark Attendance student function	Pass	Fail	Success, good

This test was performed by:

Name: Hafizah binti Ilham Hussin

Date: 08/06/2023

Signature:




Table 5.5 User Acceptance Test (UAT) form for User 5

No.	Module	Status		Comment
1.	Dashboard features and function for staff	Pass	Fail	Succes, good
2.	Login to the Sekolah Kebangsaan Mat Kilau CAMS	Pass	Fail	Success, good
3.	Insert and edit co-curricular Activity and Achievement	Pass	Fail	Success, good
4.	System Interface Design	Pass	Fail	Put background
5.	Delete data activity and achievement	Pass	Fail	Success, good
6.	Search table function	Pass	Fail	Success, good
7.	Mark Attendance student function	Pass	Fail	Success, good

This test was performed by:

Name: Noor Syakina binti Ahmad

Date: 08/06/2023




Signature:

Table 5.6 User Acceptance Test (UAT) form for User 6

No.	Module	Status		Comment
1.	Dashboard features and function for staff	Pass	Fail	Success, good
2.	Login to the Sekolah Kebangsaan Mat Kilau CAMS	Pass	Fail	Success, good
3.	Insert and edit co-curricular Activity and Achievement	Pass	Fail	Success, good
4.	System Interface Design	Pass	Fail	Success, good
5.	Delete data activity and achievement	Pass	Fail	Success, good
6.	Search table function	Pass	Fail	Success, good
7.	Mark Attendance student function	Pass	Fail	Success, good

This test was performed by:

Name: Habshim bin Mahmud

Date: 08/06/2023

Signature:




Table 5.7 User Acceptance Test (UAT) form for User 7

No.	Module	Status		Comment
1.	Dashboard features and function for staff	Pass	Fail	Success, good
2.	Login to the Sekolah Kebangsaan Mat Kilau CAMS	Pass	Fail	Success, good
3.	Insert and edit co-curricular Activity and Achievement	Pass	Fail	Success, good
4.	System Interface Design	Pass	Fail	Improve interface design
5.	Delete data activity and achievement	Pass	Fail	Success, good
6.	Search table function	Pass	Fail	Success, good
7.	Mark Attendance student function	Pass	Fail	Success, good

This test was performed by:

Name: Suhaiza binti Sapiin

Date: 08/06/2023

Signature:




Table 5.8 User Acceptance Test (UAT) form for User 8

No.	Module	Status		Comment
1.	Dashboard features and function for staff	Pass	Fail	Success, good
2.	Login to the Sekolah Kebangsaan Mat Kilau CAMS	Pass	Fail	Success, good
3.	Insert and edit co-curricular Activity and Achievement	Pass	Fail	Success, good
4.	System Interface Design	Pass	Fail	Improve design
5.	Delete data activity and achievement	Pass	Fail	Success, good
6.	Search table function	Pass	Fail	Success, good
7.	Mark Attendance student function	Pass	Fail	Success, good

This test was performed by:

Name: Ruziah binti Abdullah

Date: 08/06/2023

uf



Signature:

Table 5.9 User Acceptance Test (UAT) form for User 9

No.	Module	Status		Comment
1.	Dashboard features and function for staff	Pass	Fail	Success, good
2.	Login to the Sekolah Kebangsaan Mat Kilau CAMS	Pass	Fail	Success, good
3.	Insert and edit co-curricular Activity and Achievement	Pass	Fail	Success, good
4.	System Interface Design	Pass	Fail	Success, good
5.	Delete data activity and achievement	Pass	Fail	Success, good
6.	Search table function	Pass	Fail	Success, good
7.	Mark Attendance student function	Pass	Fail	Success, good

This test was performed by:

Name: Marjam binti Mat Saad

Date: 08/06/2023



Signature:



Table 5.10 User Acceptance Test (UAT) form for User 10

No.	Module	Status		Comment
1.	Dashboard features and function for staff	Pass	Fail	Success, good
2.	Login to the Sekolah Kebangsaan Mat Kilau CAMS	Pass	Fail	Success, good
3.	Insert and edit co-curricular Activity and Achievement	Pass	Fail	Success, good
4.	System Interface Design	Pass	Fail	Put a better color
5.	Delete data activity and achievement	Pass	Fail	Success, good
6.	Search table function	Pass	Fail	Success, good
7.	Mark Attendance student function	Pass	Fail	Success, good

This test was performed by:

Name: Mohd Amran bin Mustafa

Date: 08/06/2023

Signature: