

**STUDENT ACADEMIC MANAGEMENT SYSTEM FOR SMAS
MUHAMMADIYAH 1 BANDA ACEH**

MUHAMMAD DARLEN SAVA

UNIVERSITI TEKNOLOGI MALAYSIA

UNIVERSITI TEKNOLOGI MALAYSIA**DECLARATION OF THESIS / UNDERGRADUATE PROJECT REPORT AND
COPYRIGHT**

Author's full name : MUHAMMAD DARLEN SAVA

Date of Birth : 07/05/2001

Title : STUDENT ACADEMIC MANAGEMENT SYSTEM FOR SMAS
MUHAMMADIYAH 1 BANDA ACEH

Academic Session : 2022/2023

I declare that this thesis is classified as:

CONFIDENTIAL(Contains confidential information under the
Official Secret Act 1972)***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)

1. I acknowledged that Universiti Teknologi Malaysia reserves the right as follows:
2. The thesis is the property of Universiti Teknologi Malaysia
3. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.
4. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

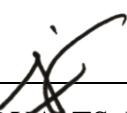
**SIGNATURE OF STUDENT**

A19EC0258

**SIGNATURE OF SUPERVISOR**PROF. MADYA. TS. DR. MOHD
SHAHIZAN BIN OTHMAN**MATRIX NUMBER****NAME OF SUPERVISOR**

NOTES : If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction

“I hereby declare that we have read this thesis and in my opinion this thesis is sufficient in term of scope and quality for the award of the degree of Bachelor of Computer Science (Software Engineering)”

Signature : 

Name of Supervisor : PROF. MADYA. TS. DR. MOHD SHAHIZAN BIN
OTHMAN

Date : 1 JULY 2023

STUDENT ACADEMIC MANAGEMENT SYSTEM FOR SMAS
MUHAMMADIYAH 1 BANDA ACEH

MUHAMMAD DARLEN SAVA

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Bachelor of Computer Science (Software Engineering)

School of Computing
Faculty of Engineering
Universiti Teknologi Malaysia

JULY 2022

DECLARATION

I declare that this thesis entitled “*Student Academic Management System for SMAS Muhammadiyah 1 Banda Aceh*” is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.



Signature :

Name : MUHAMMAD DARLEN SAVA

Date : 1 JULY 2023

DEDICATION

This thesis is dedicated to my late father. He taught me to be an independent and strong man who can support the family when he eventually is not there to do it. This thesis is also dedicated to my mother and brother, which have always helped me until now and forever.

ACKNOWLEDGEMENT

I would like to thank you to my supervisor, which has been very helpful when helping preparing this thesis and being patience at all time. His guidance has left a powerful impact for me during this project. I would also like to thank you to my fellow colleagues that I met along the way to be super helpful by sharing information and such. Special thanks to my cousin which provided me with the required data for this project. Last but not least I can't thank you enough for my mother and brother that help kept my spirit up until this moment.

ABSTRACT

In this day and age, modern technologies have started to change how things work. Many institutions have begun implementing a sophisticated curriculum and teaching and learning system in the academic field. This system helps immensely reduce teacher workload and simplifies primary operations such as marking student reports. It is also beneficial to have this online system in situations such as the pandemic. While the scope of this sophisticated system is more significant than ever, some still don't have the luxury to experience it. These institutions that still do some of their operations manually did not have the resources to upgrade to the modernized system. Moreover, private institutions like SMAS MUHAMMADIYAH 1, located in Banda Aceh, Indonesia, have more challenges since the government will not be accommodating in this case. In this case, they will more likely fall behind in technological advancement. Thus, a web app system that has functions to convert all of those manual operations to be done more efficiently is needed. This web app will provide some of the most critical tasks in managing students' academic data, such as managing academic reports, marking students, checking attendance, creating and viewing timetables, managing students' tuition records, and managing teacher and student information. In this project, the methodology used to develop the system is the Agile methodology. The project requirements are also provided using Unified Modelling Language (UML), such as use case diagrams and sequence diagrams. These requirements will be available in the Software Requirements Specification. The design of the system is also provided in the Software Design Document. At the same time, the future testing plan will be documented in the Software Testing Documentation.

ABSTRAK

Pada zaman ini, teknologi moden telah mula mengubah cara kerja. Banyak institusi telah mula melaksanakan kurikulum dan sistem pengajaran dan pembelajaran yang canggih dalam bidang akademik. Sistem ini sangat membantu mengurangkan beban kerja guru dan memudahkan operasi utama seperti menanda laporan pelajar. Ia juga berfaedah untuk mempunyai sistem dalam talian ini dalam situasi seperti wabak. Walaupun skop sistem canggih ini lebih penting berbanding sebelum ini, ada yang masih tidak mempunyai kemewahan untuk mengalaminya. Institusi ini yang masih melakukan beberapa operasi mereka secara manual tidak mempunyai sumber untuk menaik taraf kepada sistem moden. Lebih-lebih lagi, institusi swasta seperti SMA MUHAMMADIYAH 1, yang terletak di Banda Aceh, Indonesia, mempunyai lebih banyak cabaran kerana kerajaan tidak akan bertolak ansur dalam kes ini. Dalam kes ini, mereka lebih berkemungkinan akan ketinggalan dalam kemajuan teknologi. Oleh itu, sistem aplikasi web yang mempunyai fungsi untuk menukar semua operasi manual tersebut untuk dilakukan dengan lebih cekap diperlukan. Apl web ini akan menyediakan beberapa tugas paling kritikal dalam mengurus data akademik pelajar, seperti mengurus laporan akademik, menanda pelajar, menyemak kehadiran, mencipta dan melihat jadual waktu, mengurus rekod tuisyen pelajar dan mengurus maklumat guru dan pelajar. Dalam projek ini, metodologi yang digunakan untuk membangunkan sistem ialah metodologi Agile. Keperluan projek juga disediakan menggunakan Bahasa Pemodelan Bersepadu (UML), seperti rajah kes penggunaan dan rajah jujukan. Keperluan ini akan tersedia dalam Spesifikasi Keperluan Perisian. Reka bentuk sistem juga disediakan dalam Dokumen Reka Bentuk Perisian. Pada masa yang sama, rancangan ujian masa hadapan akan didokumenkan dalam Dokumentasi Pengujian Perisian.

TABLE OF CONTENTS

	TITLE	PAGE
DECLARATION		ii
DEDICATION		iii
ACKNOWLEDGEMENT		iv
ABSTRACT		v
ABSTRAK		vi
TABLE OF CONTENTS		vii
LIST OF TABLES		x
LIST OF FIGURES		xi
LIST OF ABBREVIATIONS		xiii
LIST OF APPENDICES		xiv
CHAPTER 1 INTRODUCTION		1
1.1 Introduction	1	
1.2 Problem Background	2	
1.3 Project Aim	2	
1.4 Project Objectives	2	
1.5 Project Scope	3	
1.6 Project Importance	3	
1.7 Report Organization	3	
CHAPTER 2 LITERATURE REVIEW		5
2.1 Introduction	5	
2.2 Case Study (If any)	5	
2.2.1 Company Organization Structure	5	
2.2.2 Manual Operation	6	
2.3 Current System Analysis	10	
2.4 Similar Existing System Analysis	10	
2.4.1 UTM Academic Management System	10	

2.4.2	Evaluation	12
2.4.3	Sekawan Media Aplikasi Rapor Online	13
2.5	Comparison Between Existing Systems	14
2.6	Literature Review of Technology Used	15
2.6.1	PHP	15
2.6.2	Vue.js + Tailwind	15
2.6.3	Laravel	16
2.6.4	Visual Studio Code	16
2.6.5	PostgreSQL	16
2.6.6	pgAdmin	16
2.6.7	Amazon S3	17
2.6.8	Amazon Elastic Compute Cloud (Amazon EC2)	17
2.7	Chapter Summary	17
CHAPTER 3	SYSTEM DEVELOPMENT METHODOLOGY	19
3.1	Introduction	19
3.2	Methodology Choice and Justification	19
3.3	Phases of Chosen Methodology	20
3.3.1	Requirements	21
3.3.2	Design	21
3.3.3	Development	21
3.3.4	Testing	21
3.3.5	Deployment	22
3.3.6	Feedback	22
3.4	Technology Used Description	22
3.5	System Requirement Analysis	23
3.5.1	Software Requirements	23
3.5.2	Hardware Requirements	23
3.6	Chapter Summary	24
CHAPTER 4	REQUIREMENT ANALYSIS AND DESIGN	25
4.1	Introduction	25

4.2	Requirement Analysis	25
4.2.1	Functional Requirements	25
4.2.1.1	Use Case Diagram	25
4.2.1.2	Actors Description	27
4.2.1.3	Use Case Description	27
4.2.1.4	Use Case Specification	28
4.2.1.5	Sequence Diagram	31
4.2.1.6	Activity Diagram	33
4.2.2	Non-functional Requirements	34
4.3	Project Design	35
4.4	Database Design	36
4.4.1	Data Dictionary	37
4.5	Interface Design	38
4.5.1	Page Navigation Design	38
4.5.2	User Interface Design	40
4.6	Chapter Summary	43
CHAPTER 5	IMPLEMENTATION AND TESTING	44
5.1	Introduction	44
5.2	Coding of System Main Functions	44
5.3	Interface of System Main Functions	44
5.4	Testing	
5.4.1	Black box Testing	
5.4.1.1	System Flow	
5.4.1.2	Input Output Verification	
5.4.1.3	Error Messages	
5.4.2	White box Testing	
5.4.3	User Testing	
5.5	Chapter Summary	
CHAPTER 6	CONCLUSION	44
5.1	Introduction	44
5.2	Achievement of Project Objectives	44

5.3	Suggestions for Future Improvement	44
-----	------------------------------------	----

REFERENCES	46
-------------------	-----------

LIST OF TABLES

TABLE NO.	TITLE	PAGE
Table 2.1	Comparison Between Existing System and Proposed System	14
Table 3.1	Technology Used and Description	22
Table 4.1	Use Case Specification for Uc002 - Manage Academic Report	30
Table 4.2	Data Dictionary of Academic report Table	37
Table 4.3	Data Dictionary of Knowledge mark Table	38

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
Figure 2.1	Organization Chart of SMAS MUHAMMADIYAH 1 BANDA ACEH	6
Figure 2.2	Manual Marking of Students	7
Figure 2.3	First Page of The Report Book of a Student	8
Figure 2.4	Second Page of The Report Book of a Student	8
Figure 2.5	Attendance Book	9
Figure 2.6	Tuition Card	9
Figure 2.7	Myutm, Portal to UTM ACAD	11
Figure 2.8	UTM ACAD	11
Figure 2.9	Semester Data UI Of Evaluation	12
Figure 2.10	Students Data UI Of Evaluation	13
Figure 2.11	School Final Exam Marking UI Of Sekawan Media Aplikasi Rapor Online	13
Figure 2.12	Generated Report Book UI of Sekawan Media Aplikasi Rapor Online	14
Figure 3.1	Agile Methodology Model	20
Figure 4.1	Use Case Diagram of Student Academic Management System	26
Figure 4.2	Sequence Diagram of UC002 – Manage Academic Report	32
Figure 4.3	Activity Diagram of UC0015-Manage User	33
Figure 4.4	Architecture Model Diagram for Sams	35
Figure 4.5	Erd Of Student Academic Management System	36
Figure 4.6	Page Navigation for Teacher in Sams	39
Figure 4.7	Page Navigation for Student in Sams	39

Figure 4.8	Page Navigation for Admin in Sams	40
Figure 4.9	Login Page for Sams	41
Figure 4.10	Marking Page for Teacher in Sams	41
Figure 4.11	Academic Report Page for Teacher in Sams	42
Figure 4.12	Account Management for Admin in Sams	42

LIST OF ABBREVIATIONS

SRS	-	Software Requirement Specification
SDD	-	Software Design Document
STD	-	Software Testing Documentation
UML	-	Unified Modelling Language
SAMS	-	Student Academic Management System
MVC	-	Model-View-Controller
UTM	-	Universiti Teknologi Malaysia

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
Appendix A	Software Requirement Specification	47
Appendix B	Software Design Document	112
Appendix C	Software Testing Documentation	182
Appendix D	Gantt Chart	201

CHAPTER 1

INTRODUCTION

1.1 Introduction

In the current day of modern technologies, many tasks and jobs have started to be done automatically throughout many fields of work. In the academic fields, the system to manage anything related to it has been significantly improved. We can lay out several reasons why this is happening. The first one is the usual suspect, which is globalization. More and more institutions introduce other cultures into their curriculum, such as English language subjects. Globalization results in a more culturally and linguistically diversified population (Miller et al., 2009). Other reasons, such as technological advancement and the research put into better teaching and learning environments, led to many sophisticated systems used daily to manage students, teachers, and the institution's data.

However, there are still some areas that still do crucial and time-consuming tasks manually. One of them is managing the academics of a student. While many institutions have already made sophisticated systems that manage all of a student's data, here in Aceh, Indonesia is still a step behind.

Thus, a web app system for managing students' academics is needed to combat this. This Student Academic Management System will help teachers ease their work, save time, and make sure that students and their parents have access to their data. It is not without a reason why the system will be developed as a web app because the most significant benefit of a website is that it can be accessed by almost every device without the need to install an actual app.

1.2 Problem Background

One institution that still does not have a website to manage students' academic data is SMAS Muhammadiyah, a private high school in Banda Aceh City, Indonesia. For many years now, teachers in SMAS Muhammadiyah have been dealing with report cards that they update each semester manually. While the teachers changing the report cards means that the students or their parents do not have access to their academic data for a while until the teachers completely update it. The report card collection usually happens when a final exam occurs; the students will hand in their report cards before the exam starts. After the final exam finishes, they will get their report cards back on "Report cards distribution day." Nevertheless, it is time-consuming, but there is also a risk of losing the report card and being more vulnerable to human error.

Another problem that they are facing is the tuition management of the students. It is still done manually without a system by confirming to the student's parents/guardians. The parents/guardians will send the teacher the receipt of payment, usually through a messaging app such as WhatsApp. Not only is it cumbersome, but also time-consuming for both parties. The same can be said for attendance checking. SMAS Muhammadiyah still uses an attendance book that is provided for each classroom. This attendance record will be filled in manually by the teacher that teaches that day in that particular classroom. The data of all students is also being kept in multiple files and books, making it decentralized. The timetable that the students use is also manually printed out, and the student needs to write down the schedule/timetable themselves.

1.3 Project Aim

This project aims to create an effective solution to enable the institution to convert its manual student management system into a computerized system.

1.4 Project Objectives

The objectives of the project are:

- a) To identify the requirements of the student academic management system
- b) To design the student academic management system based on the Stakeholder's requirement
- c) To develop the student academic management system based on the Stakeholder's requirement
- d) To test the functionalities of the student academic management system according to stakeholder's requirement

1.5 Project Scope

The scopes of the project are:

- a) This project only focused on the conversion of manually managing student data to a website that stores all of the student's data, which can be accessed by teachers, and students.
- b) SMAS Muhammadiyah 1 will provide the data that is needed for the system
- c) Users in this system are teachers, students, and admins of SMAS Muhammadiyah.

1.6 Project Importance

The project will significantly increase the efficiency of the institution's student management and increase the teacher's productivity. It also provides students and their parent's ease of access to academic data. The proposed automated system will also minimize the risk of human error from all parties involved.

1.7 Report Organization

This PSM 1 report will consist of 5 chapters: Introduction, Literature Review, Methodology, Requirement Analysis and Design, and Conclusion. The introduction chapter consists of an introduction to the project and the problem background, aim, objectives, scope, and importance of the project. In chapter 2, the literature review will mainly discuss the case study with its organizational structure and manual operation. There will also be an analysis of the current system and comparing existing systems.

The technology used will also be included in this chapter. In the 3rd chapter, the methodology was chosen for the project, and its justification will be explained. The chosen methodology's phases then will be briefly dissected one by one. In chapter 4, the requirement analysis and design will be laid out. This requirement analysis and design are required before taking the system into the development phase. The 5th chapter, which is the final chapter of this report, will consist of the report's conclusion. This chapter will also include achievements and suggestions so far during the project.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, the literature review involved with the proposed system will be described in detail. It will become the source of information about existing systems or identical systems from the current market. By comparing them with the proposed system the enhancement that can be made will become clear. The technology that will be used in the development will also be stated.

2.2 Case Study (If any)

The proposed system is based on a case study from one of the institutions in Banda Aceh, Aceh, Indonesia. The name of the institution is SMAS MUHAMMADIYAH 1 BANDA ACEH which is a private high school that has been existing for around 62 years. They are one of few high schools in Banda Aceh that have accreditation of A.

2.2.1 Company Organization Structure

SMAS Muhammadiyah 1 is a private high school in Aceh, Indonesia. Just like other institutions they also have the same organizational structure at their level. They consist of teachers from across Aceh that are qualified with quality and in-depth knowledge of their subject while also maintaining their responsibilities as a teacher. The picture below is the organizational structure for SMAS Muhammadiyah 1.



Figure 2.1: Organization Chart of SMAS MUHAMMADIYAH 1 BANDA ACEH

2.2.2 Manual Operation

The current system used by SMAS Muhammadiyah 1 to manage academic data is still involved using traditional and manual ways. This is ranging from marking student reports, and managing attendance records, to even keeping track of tuition records. For the marking student report process, it takes multiple teachers to recap all of the marks a student got. First, the teacher that is responsible for the course will collect their student's marks from a table that they have to fill in each time they give a mark for an assignment or in-class exercise. Then this excel table will be sent to the homeroom teacher. In the end, the homeroom teacher will have multiple excel files that they will put into the report book. The report book is also made using word/excel. From this point, we can conclude that the data is still not centralized because it moved from teacher to teacher.

As for managing the attendance record, they are still using an attendance book which will be filled by teachers that will teach in the class that day. This attendance book then will be used as a reference for the teacher by the end of the semester to be inputted into students' report books on how much they are absent because of sickness, and with/without permission. In managing the tuition record, the teachers at SMAS Muhammadiyah also still use what we called tuition book/card. This tuition book/card works the same way as the others mentioned above. The homeroom teacher will receive payment from the student's parents and when they confirm the payment then they will approve it in the tuition book that the student has paid for the semester. The students also need to manually write down the schedule/timetable because it is usually only printed once and put in the front of the class. All of these manual operations can be seen in the pictures below,

DAFTAR NILAI
SEMESTER : GANJIL TAHUN PELAJARAN : 2020/2021

Mata Pelajaran Kelas	: Fisika	Kelompok Mapel Guru Mapel	: IIS : Rahmawati, S.Pd				
No Urut	NIS NISN	Nama Siswa	L/P	Nilai Positif (NP) Angka Huruf	Kompetensi	Deskripsi Kompetensi	
1	9391 0022235491	Putri Afifia Riesity	P	78	C	Pengertian	Cakup matang memahami materi pelajaran yang diajukan namun perlu ditanyakan lagi terkait pada materi Monera puya dan Moner nercia.
				78	C	Keterampilan	Sudah terampil dan mandiri dalam mengerjakan tugas yang di berikan namun perlu diberikan penjelasan sedikit-sedikit yang diberikan dan masih perlu ditanyakan juga kesiapan berlatungnya.
				-	B	Sikap	Baik pertimbangkan sikap sopan dan senang untuk guru dan silaber. Lebih disiplin dan lebih bertanggung jawab terhadap semua tugas yang diberikan.
2		Alfarisi	L	72	C	Pengertian	Cakup matang memahami materi pelajaran yang diajukan, namun perlu ditanyakan lagi terkait pada materi Hukum pascal dan Hukum archimedes
				75	C	Keterampilan	Cakup terampil dan mandiri dalam mengerjakan tugas yang di berikan namun perlu keterangan dalam mengerjakan soal-soal yang diberikan dan masih perlu ditanyakan juga kesiapan berlatungnya.
						Sikap	Baik pertimbangkan sikap sopan dan senang terhadap guru dan diberapkan lebih disiplin dan lebih bertanggung jawab terhadap semua tugas yang diberikan.

Figure 2.2: Manual marking of students

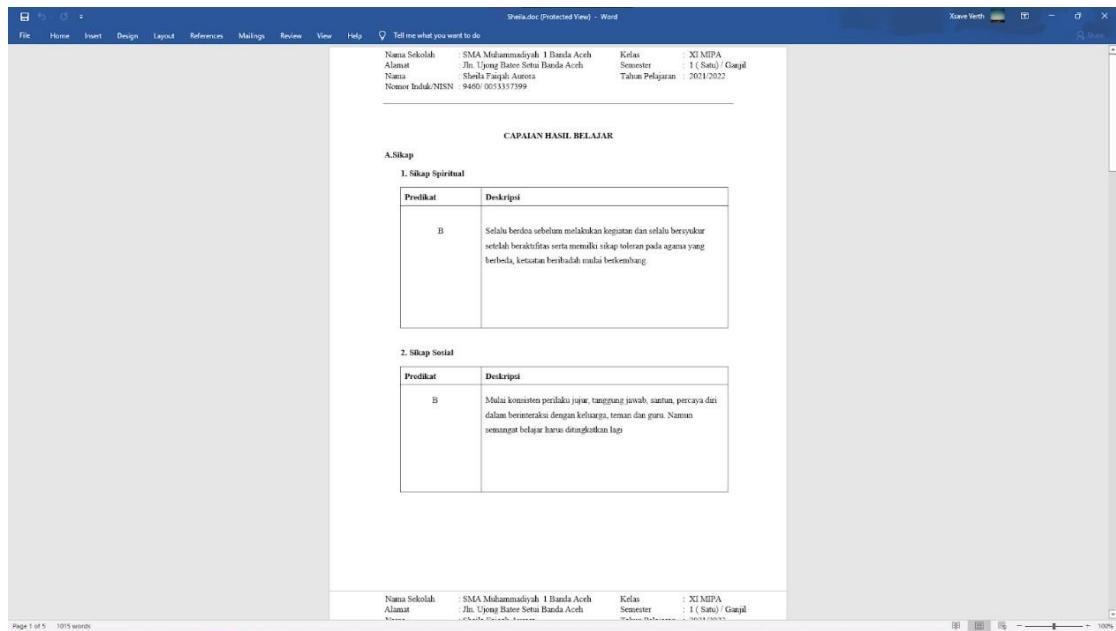


Figure 2.3: First page of the report book of a student

Nama Sekolah : SMA Muhammadiyah 1 Banda Aceh		Kelas : XI MIPA				
Alamat : Jln. Ujung Batas Setia Banda Aceh		Semester : 1 (Satu) / Ganjil				
Nama : Shela Faqih Aisyah		Tahun Pelajaran : 2021/2022				
Nomor Induk/NISN : 9460 0053357399						
B. Pengetahuan dan Keterampilan						
Kode	Mata Pelajaran	KKM	Pengetahuan		Keterampilan	
			Nilai	Predikat	Nilai	Predikat
Kelompok A (Human)						
1	Pendidikan Agama dan Budi Pekerti	70	85	B	85	B
2	Pendidikan Pancasila dan Kewarganegaraan	70	85	B	85	B
3	Bahasa Indonesia	70	85	B	85	B
4	Matematika	70	83	B	84	B
5	Sejarah Indonesia	75	83	B	80	B
6	Bahasa Inggris	75	90	B	90	B
Kelompok B (Uman)						
1	Seni Rupa	75	85	B	83	B
2	Pendidikan Jasmani, Olah Raga, dan Kesehatan	75	85	B	83	B
3	Prakarya dan Kreativitas	80	85	B	85	B
4	---					
Kelompok C (Peminatan)						
1	Biologi	75	84	B	82	B
2	Kimia	70	82	B	82	B
3	Fisika	70	85	B	84	B
4	Matematika	70	82	B	80	B
5	Ekonomi	70	83	B	83	B
6	---					

* : Bila ada

Tabel inferensi predikat berdasarkan KKM

KKM	Predikat			
	D-Kurang	C-Cukup	B-Baik	A-Sangat Baik
70	< 70	70 - 79	80 - 89	90 - 100

Figure 2.4: Second page of the report book of a student

Figure 2.5: Attendance Book

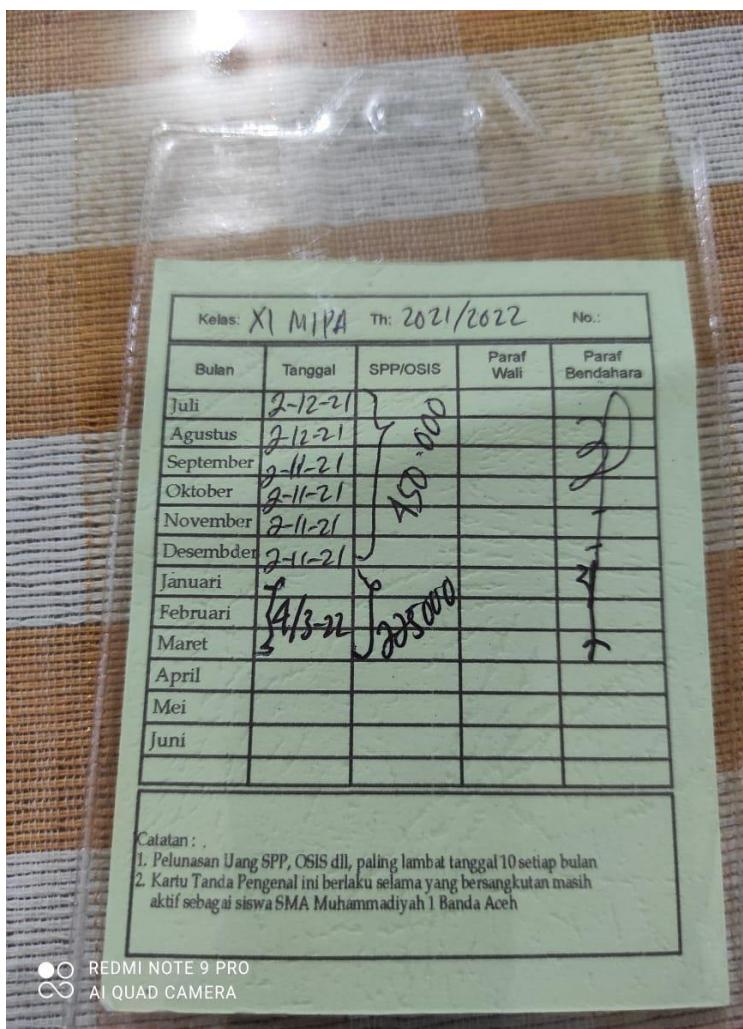


Figure 2.6: Tuition Card

2.3 Current System Analysis

In the current system several platforms are used causing the data to be decentralized. Some of the examples of platforms used are Microsoft Word, Microsoft Excel, Whatsapp, and also a manual operation using handwriting. The proposed system will fix this issue by having a centralized database, this way the data will not be scattered across different platforms. The current system also involved using handwriting in some instances such as filling in the attendance book or tuition book/card. Microsoft Word and Excel are used for filling in the report book and keeping track of students' marks respectively. There are also some parts of the report book in which the teacher has to handwrite them. As for Whatsapp, it is usually used to confirm the payment of students' tuition through their parents or even the students themselves. The current task of taking students' attendance is also done manually. The teachers that are teaching in that particular classroom will take the student's attendance based on how many hours they teach on that day with a total of 8-hour slots per day. When the next teacher comes, they will also take attendance once again and all of this is recorded in a book called attendance book.

2.4 Similar Existing System Analysis

2.4.1 UTM Academic Management System

UTM Academic Management System (ACAD) is a web-based system that helps to manage student academic data. This system allows the management of students' data such as grades, personal information, and finances among other things. This system is personalized for students and lecturers of University Teknologi Malaysia. Although it is a bit convoluted to access because it is hidden behind another website called My UTM which then will redirect users to this system, it still provides its intended functions. Pictures below show the UI of my UTM and UTM ACAD

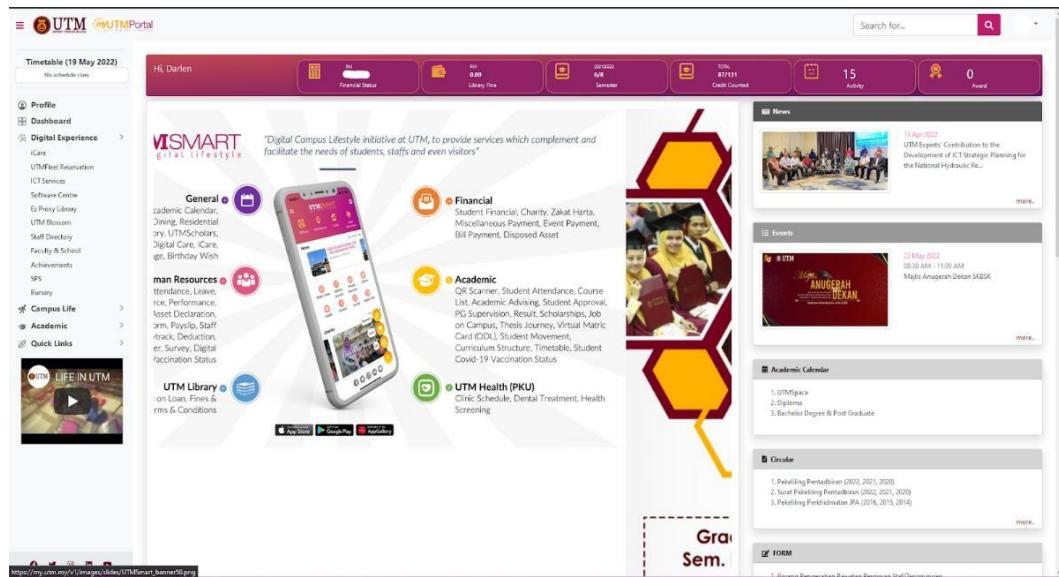


Figure 2.7: myUTM, Portal to UTM ACAD

Figure 2.8: UTM ACAD

2.4.2 Evaluation

The evaluation application is used to assess student learning results. This application evaluates the productivity of students based on exam scores, tests, and other activities (evaluation, 2016). Based on the brief explanation, this application mainly functions to help teachers monitor their student's growth and progress online.

Below is the UI of evaluation:

The screenshot shows the 'Semester' page of the 'evaluation' application. The left sidebar has a dark theme with categories like NAVIGASI UTAMA, Dashboard, Master Data (Tahun Ajaran, Semester, Sekolah, Ekstrakurikuler, Mata Pelajaran, Guru, Siswa, Rombel, Anggota Rombel), KBM, Penilaian (selected), Laporan, and Kesiswaan. The main content area has a red header with 'evaluation', 'SMA Negeri 4 Semarang', 'Login Sebagai Admin Sekolah', 'Admin SMA N 4 SMG <', 'Dashboard', and 'Semester'. It includes a 'Tambah Data' button, a search bar, and a table titled 'Periode Semester Aktif'. The table shows four rows of data:

	Tahun Ajaran	Nama	Semester	Tanggal Mulai	Tanggal Selesai	Act.
<input type="checkbox"/>	2015	2015/2016 Ganjil	1	2015-07-01	2015-12-31	<input checked="" type="checkbox"/> Edit <input checked="" type="checkbox"/> Hapus
<input type="checkbox"/>	2015	2015/2016 Genap	2	2016-01-01	2016-06-30	<input checked="" type="checkbox"/> Edit <input checked="" type="checkbox"/> Hapus
<input type="checkbox"/>	2016	2016/2017 Ganjil	1	2016-07-01	2016-12-31	<input checked="" type="checkbox"/> Edit <input checked="" type="checkbox"/> Hapus
<input type="checkbox"/>	2016	2016/2017 Genap	2	2017-01-01	2017-06-30	<input checked="" type="checkbox"/> Edit <input checked="" type="checkbox"/> Hapus

At the bottom, it says '©2016 evaluation SMA Negeri 4 Semarang. Dikembangkan oleh dinustek.', 'Kami sedang offline, ... Chatra.', and a small icon.

Figure 2.9: Semester Data UI of evaluation

	nama	NIPD	NISN	Jenis Kelamin	Tempat, Tanggal Lahir	Tanggal Diterima	Act.
<input type="checkbox"/>	ANITA PUTRI MULYASARI	0000000000000000	0000000000000000	Perempuan	Semarang , 06 September 2000	07-09-2015	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Hapus
<input type="checkbox"/>	ABDULRAHMAN RIZKI	0000000000000000	0000000000000000	Laki-Laki	Semarang , -	07-09-2015	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Hapus
<input type="checkbox"/>	ANITA SUDITIAWANI	0000000000000000	0000000000000000	Perempuan	Jepara , 07 Desember 2000	09-07-2015	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Hapus
<input type="checkbox"/>	ANITA WIDYA SENIWI	0000000000000000	0000000000000000	Laki-Laki	Ungaran , 14 November 1999	09-07-2015	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Hapus
<input type="checkbox"/>	ANITA YULIA MULYATI APRESIA	0000000000000000	0000000000000000	Laki-Laki	Klaten , 17 April 2015	09-07-2015	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Hapus
<input type="checkbox"/>	ANITA YESTRI TITIAPUTRI	0000000000000000	0000000000000000	Perempuan	Semarang , 28 Juli 2000	09-07-2015	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Hapus
<input type="checkbox"/>	ANITA ZAHRA TRAIBO	0000000000000000	0000000000000000	Perempuan	Semarang , 19 Desember 1999	09-07-2015	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Hapus
<input type="checkbox"/>	ANITA YULIA APRESIA	0000000000000000	0000000000000000	Perempuan	Semarang , 26 November 2000	09-07-2015	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Hapus
<input type="checkbox"/>	ANITA YULIA HERLAMANING	0000000000000000	0000000000000000	Laki-Laki	Semarang , 22 Agustus 2000	09-07-2015	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Hapus
<input type="checkbox"/>	ANITA YULIA HERLAMANING	0000000000000000	0000000000000000	Perempuan	Mojokerto , 19 September 1999	09-07-2015	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Hapus
<input type="checkbox"/>	ANITA YULIA HERLAMANING	0000000000000000	0000000000000000	Laki-Laki	Medan , 25 Mei 2000	09-07-2015	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Hapus
<input type="checkbox"/>	ANITA YULIA HERLAMANING	0000000000000000	0000000000000000	Laki-Laki	Semarang , 05 Agustus 2000	09-07-2015	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Hapus
<input type="checkbox"/>	ANITA YULIA HERLAMANING	0000000000000000	0000000000000000	Laki-Laki	Semarang , 13 April 2000	09-07-2015	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Hapus
<input type="checkbox"/>	ANITA YULIA HERLAMANING	0000000000000000	0000000000000000	Laki-Laki	Pontianak , 06 Mei 2000	09-07-2015	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Hapus
<input type="checkbox"/>	ANITA YULIA HERLAMANING	0000000000000000	0000000000000000	Perempuan	Semarang , 27 April 2000	09-07-2015	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Hapus
<input type="checkbox"/>	ANITA YULIA HERLAMANING	0000000000000000	0000000000000000	Perempuan	Kab. Semarang , 15 September 1999	09-07-2015	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Hapus
<input type="checkbox"/>	ANITA YULIA HERLAMANING	0000000000000000	0000000000000000	Perempuan	Banjarmasin , 29 Desember 1999	09-07-2015	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Hapus
<input type="checkbox"/>	ANITA YULIA HERLAMANING	0000000000000000	0000000000000000	Perempuan	Malang , 01 Juli 2000	09-07-2015	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Hapus
<input type="checkbox"/>	ANITA YULIA HERLAMANING	0000000000000000	0000000000000000	Perempuan	Semarang , 09 Oktober 2000	09-07-2015	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Hapus

Figure 2.10: Students data UI of evaluation

2.4.3 Sekawan Media Aplikasi Rapor Online

Rapor Online is a web-based application that makes it easy to report student learning outcomes. The Rapor Online application helps in reporting student grades from elementary to upper levels (Sekawan Media). Some of the highlights of this system are the functions such as counseling which the other existing system did not have.

Nilai UAS

TH 2019/2020 Genap

UNDUH PANDUAN

FORM PENCARIAN

Tahun Ajaran: 2019 - 2020

Semester: Ganjil

Kelas | Lintas Minat: XIA3-XI MIPA 3

Pelajaran: -Pilih-

CARI DATA

REFRESH

TABEL NILAI UAS

Tampilan: 10 data per halaman

No Mata Pelajaran Kelas | Lintas Minat Nama Guru

No	Mata Pelajaran	Kelas Lintas Minat	Nama Guru
1.			
2.			
3.			
4.			
5.			
6.			
7.			
8.			
9.			
10.			

UPLOAD NILAI UAS INPUT NILAI ESSAI

Figure 2.11: School final exam marking UI of Sekawan Media Aplikasi Rapor Online

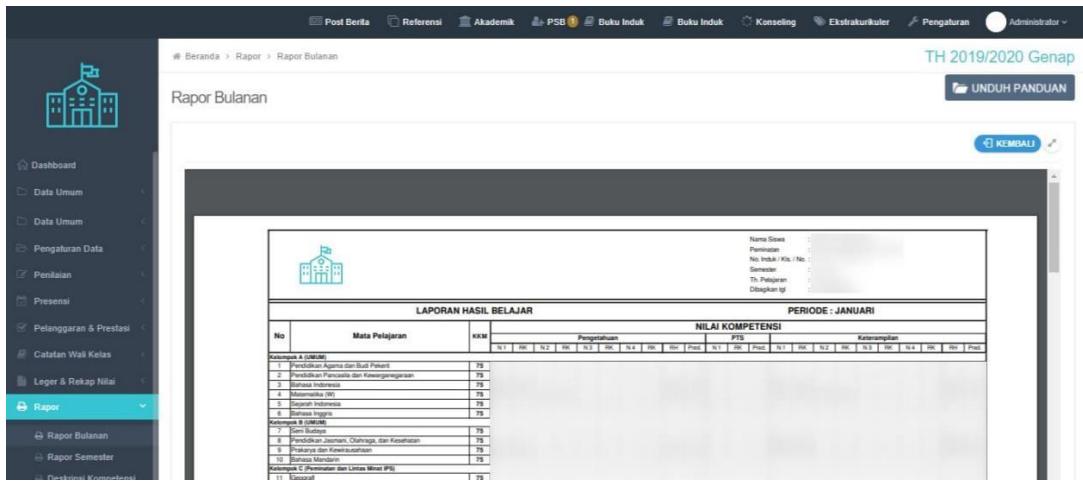


Figure 2.12: Generated report book UI of Sekawan Media Aplikasi Rapor Online

2.5 Comparison Between Existing Systems

In table 2.1 the comparison of functions/features between existing systems is shown. The proposed system is not included in the table because it is to be expected to include the functions that the existing system did not have or have. The main similarity between them is that all of them have the function to mark and generate reports. The existing system also has similarities with them not having a function to generate a timetable. Other than that they are pretty diverse in terms of having the functions or not.

Table 2.1: Comparison between existing system and proposed system

Function/Feature	UTM Academic Management System	evaluation	Sekawan Media Aplikasi Rapor Online	Student Academic Management System
Report Marking	No	Yes	Yes	Yes
Manage Academic Report	No	Yes	Yes	Yes
Generate Report	Yes	Yes	Yes	Yes

Student's Profile Management	Yes	No	No	Yes
Teacher's Profile Management	Yes	No	No	Yes
Tuition Record Management	Yes	No	No	Yes
Admin Dashboard	No	Yes	Yes	Yes
Generating Timetable	No	No	No	Yes
View Timetable	No	No	No	Yes
Centralized Database	Yes	Yes	Yes	Yes
Attendance Record Management	No	No	Yes	Yes
Multiple User role type Access	Yes	Yes	No	Yes

2.6 Literature Review of Technology Used

In this part development tools and technology used for the proposed system will be examined.

2.6.1 PHP

The proposed system will be developed using PHP which is a server scripting language that is fast, flexible, pragmatic, and suitable for web applications such as the proposed system. This language will also be used in conjunction with its framework which will be explained below.

2.6.2 Vue.js + Tailwind

Vue.js and Tailwind will be used as the front-end of the system. Vue.js is a JavaScript framework that builds on top of vanilla HTML, CSS, and JavaScript. It

covers most of the basic or common features that are needed for front-end development. It is very flexible and can be used in many ways. As for Tailwind, it is a CSS framework that can be used for rapidly developing user interfaces. With these two combined it should be sufficient to develop the front-end.

2.6.3 Laravel

Laravel is one of the most popular frameworks for PHP. It will be mainly focused on the backend side of the system such as database, and authentication among other things. By using Laravel it means that the development will follow along with the MVC (Model View Controller) architecture. Using MVC as its architecture and design pattern will have some benefits such as faster development and easier to organize a large-size web application. Its large library of packages/API will help the development immensely to cut down some of the basic functions.

2.6.4 Visual Studio Code

Visual Studio Code is the choice for a code editor in this development of the Student Academic Management System. This code editor must be not too unfamiliar among developers because it is one of if not the most popular code editor software. Not only because it supports multiple languages but also has a large collection of extensions that makes coding easier and manageable.

2.6.5 PostgreSQL

PostgreSQL is an open-source relational database that has many features and is highly extensible. It can handle small or bigger datasets while still maintaining security and performance. It is also the only relational database that meets the SQL:2016 Core conformance.

2.6.6 pgAdmin

Since the development will be using PostgreSQL as the database it is only logical to also use the most popular tools for this open-source relational database.

pgAdmin is the tool for managing PostgreSQL, it is included with a convenient UI that helps and simplifies creating, maintaining, and using the database.

2.6.7 Amazon S3

Amazon S3 is an object storage service from the tech giant Amazon that provides cloud storage with industry-leading in scalability, data availability, security, and performance. It is also cost-effective compared to other cloud storage and also configurable in such a way that meets the needs of stakeholder's requirements.

2.6.8 Amazon Elastic Compute Cloud (Amazon EC2)

Amazon EC2 is a web hosting provided by amazon. It is used by many big company names such as Netflix, Twitch, and LinkedIn. It is mainly for enterprise websites that usually have a lot of traffic to handle.

2.7 Chapter Summary

This Chapter mainly covers the case study which in this case is SMAS Muhammadiyah 1 and also their current system. Then there is the comparison between the existing system that has similar functions and the field of the proposed system. The technology that will be used in the development is also described in this chapter.

CHAPTER 3

SYSTEM DEVELOPMENT METHODOLOGY

3.1 Introduction

One of the most critical aspects of developing a system. There are a lot of methods that have their benefits and drawbacks. This chapter will discuss the chosen process and the justification for it. The phases of the selected methodology will also look in-depth, and a brief description of the technology used in the system's development and its system requirements will follow.

3.2 Methodology Choice and Justification

For developing any system, there are several methodologies or what is usually called The Software Development Life Cycle (SDLC) to choose from. Some popular ones are Agile, Lean, Waterfall, Iterative, Spiral, and DevOps. They all differ in one way but still have the same goal: to aid development quickly and cost-effectively. Agile, specifically Agile Scrum, will be chosen as the methodology for developing the Student Academic Management System.

Agile methodology was chosen for several reasons, but the methodology will be explained in short. Agile methodology is one of the most popular methodologies used in development nowadays. The Agile SDLC development approach emphasizes teamwork, customer satisfaction, and development across several short cycles, or sprints, instead of a top-down process with a single set of steps (mondayblog, 2022).

In agile methodology, the development of the system will be separated into different iterations, usually called “sprints.” In these sprints, a task or function of the system has to be done based on the time set. Because of this, the development will have a rapid delivery of the system since each sprint, there is something to show to the

stakeholders. It is also flexible because the stakeholders can add requirements anytime based on their feedback for each iteration/sprint. Thus, the satisfaction of stakeholders can also be seen for each iteration.

3.3 Phases of Chosen Methodology

In this part of the chapter, the phases of Agile methodology will be defined and explained one by one. The phases are as follows: Requirements, Design, Development, Testing, Deployment, and Feedback. The depiction of these phases is as in the picture below.

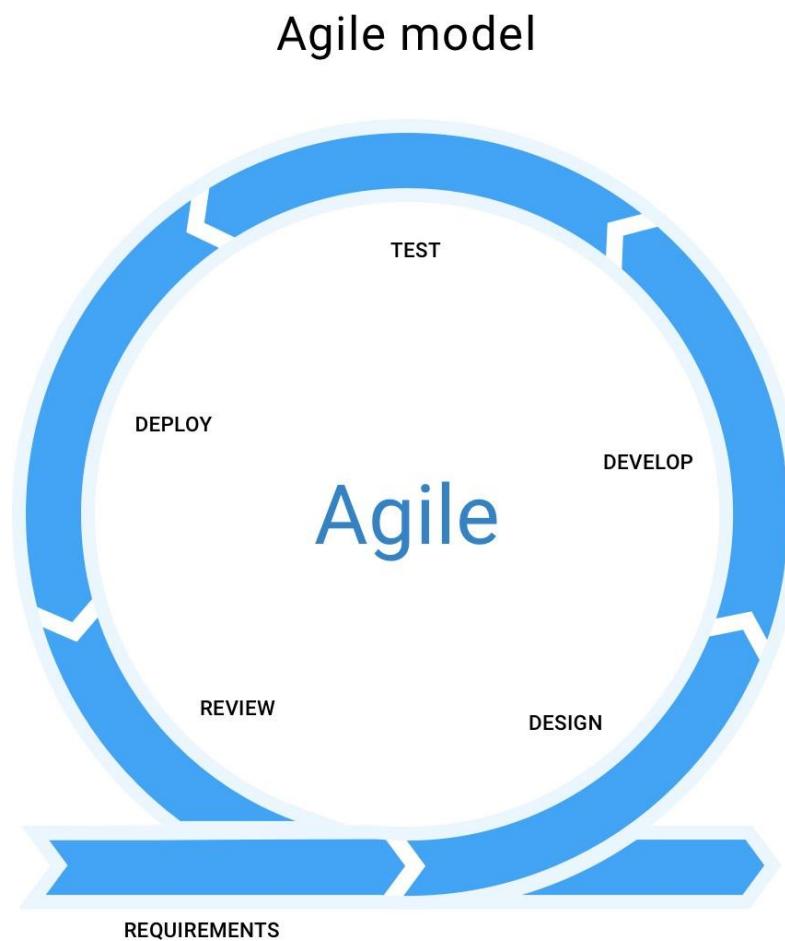


Figure 3.1: Agile methodology model

3.3.1 Requirements

In this phase, the developer must assemble a list of requirements from the stakeholders. This requirement can be acquired by interviewing or discussing it with the stakeholders. The stakeholders will then be responsible for the backlog and user stories; meanwhile, the developer must ensure they have enough information and continue to the next step. This phase is crucial because it can determine the future of the project cost in time or money.

3.3.2 Design

The focus in this phase is to define more in-depth details of the project from the requirements that have been set. These details include the project scope, objective, what tools will be used in the project, and the system requirement, just some examples. Discussing furthermore with the stakeholders is also considered reasonable at this phase.

3.3.3 Development

In the development phase, the developer must transform the requirements and design into a functional system from the previous two phases. This phase is the foundation of the Agile methodology. Thus, it is where most of the developer's attention is drawn.

3.3.4 Testing

As the name suggests, testing is the phase where the product/system is tested. During this phase, it is not possible to eliminate all bugs that reside in the system but still enough to make the system go into the deployment stage.

3.3.5 Deployment

After testing is done and the bugs that have been encountered patched, then the deployment of the system can be done. Even though the system has been deployed, changes or fixes regarding newly found bugs can still be done.

3.3.6 Feedback

After the product/system has been delivered to the stakeholders and is available to several users, collecting their feedback will be the last step of agile methodology. From the feedback gained, the developer can work to improve the system.

3.4 Technology Used Description

In this part, the development tools and technology used for the proposed system will be discussed briefly.

Table 3.1: Technology used and description

Technology	Description
Windows 10	Operating System
Google Drive	Documentation
Figma	Mockup interface
Draw.io	Design diagram
Visual Studio Code	Code Editor
Laravel	PHP framework
PHP	Programming Language
Amazon S3	Cloud Storage
Amazon EC2	Web Hosting
Vue.js	Javascript framework
Tailwind	CSS framework

PostgreSQL	Relational Database
pgAdmin	Database management tools

3.5 System Requirement Analysis

System requirements analysis will help specify the required software and hardware to run the proposed system with the user in mind. Subchapter 3.5.1 will show the software requirements, consisting of OS and web browser choices. Subchapter 3.5.2 shows the hardware requirements, consisting of each OS and its required processor, RAM and Hard Drive.

3.5.1 Software Requirements

1. Operating system: Windows 8 or later, macOS Sierra 10.12 or later, and 64-bit Ubuntu 14.04+(Linux).
2. Web Browser: Mozilla Firefox version 48 or later, Internet Explorer version 11 or later, Google Chrome version 54 or later

3.5.2 Hardware Requirements

1. Processor: Windows or Linux with Pentium 4 or newer processor that supports SSE2 or Mac computer with an Intel x86 or Apple silicon processor.
2. RAM: Windows or Linux with 512MB of RAM / 2GB of RAM for the 64-bit version or Mac computer with 512 MB of RAM.
3. Windows, Linux, or Mac with 200MB of hard drive space.

3.6 Chapter Summary

The chosen methodology and the justification for choosing it is the main key point in this chapter. But there is also an in-depth look at what the methodology has to offer in the form of its phases. The technology used is also described briefly with its description. Last but not least, the system requirement analysis is also stated, which includes both software and hardware requirements to run the system.

CHAPTER 4

REQUIREMENT ANALYSIS AND DESIGN

4.1 Introduction

This chapter delve into the requirements to develop the system and the design of the system. The requirements analysis will be divided into 2 part which are the functional requirements that consist of diagrams such as use case diagram, sequence diagrams and activity diagrams and the non-functional requirements. The design of the system is also included which are the project architecture design, database design and interface design. The complete diagrams and design can be found in the appendix section that separated into 3 parts, appendix A for SRS, appendix B for SDD and appendix C for STD.

4.2 Requirement Analysis

This section will discuss about functional and non-functional requirements of the system from the stakeholder.

4.2.1 Functional Requirements

In this part, the features and behaviors of the system will be discussed. It is crucial to have functional requirements since it define the main functions of the system.

4.2.1.1 Use Case Diagram

To represent the main functions and features of the system, use case diagram is used since it can describe the important component of the system along with its

relationship with the users of the system. The use case diagram of SAMS is depicted in figure 4.1.

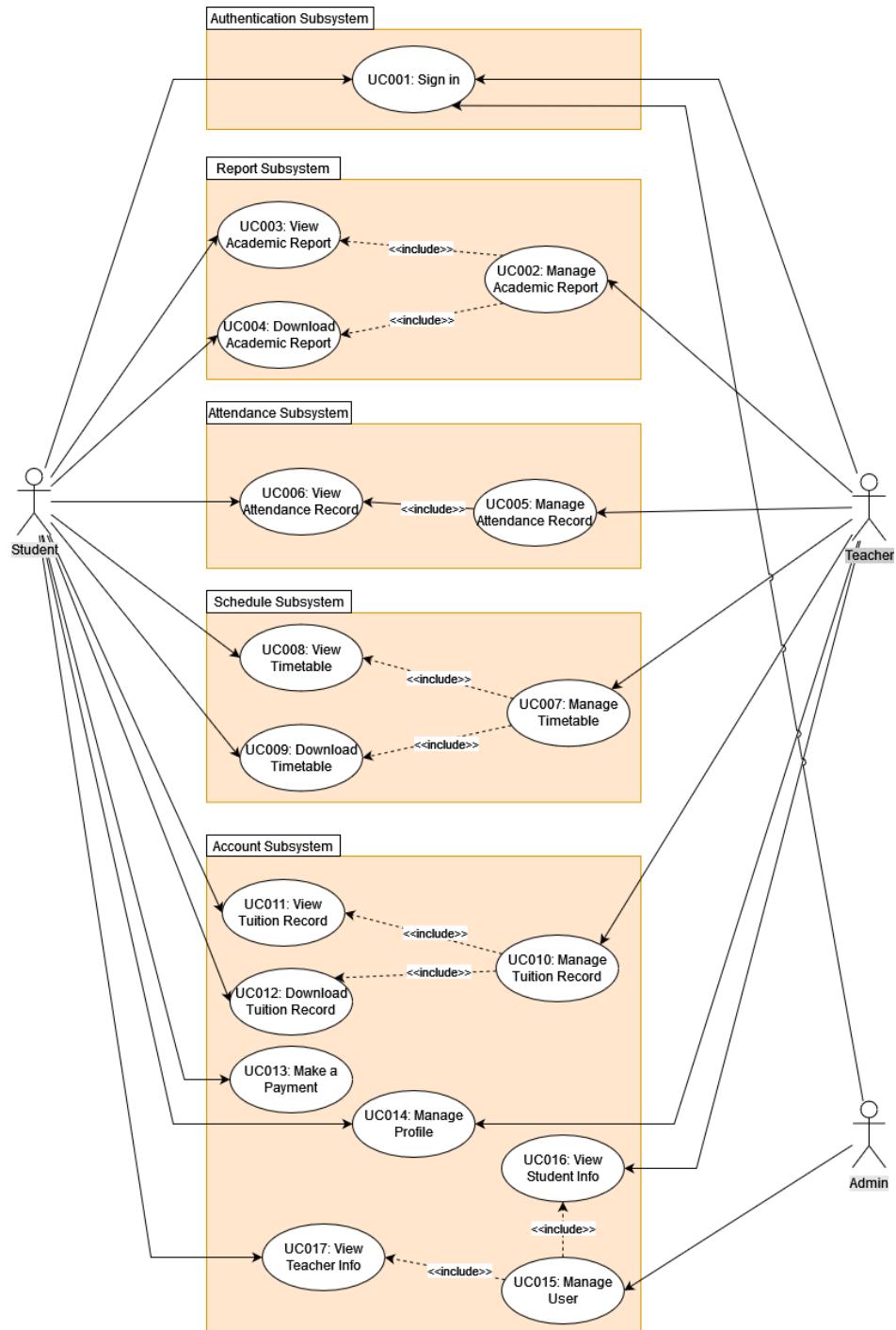


Figure 4.1 Use Case Diagram of Student Academic Management System

The figure above shows all of the functions that exist in the proposed system. Besides the use case and subsystems there are three users that directly involved in the system. The arrow shows users can only access certain functions. There are 17 use cases that divided into 5 subsystems. There are multiple use cases that included into

another use case, this is because certain user like student can only access some of the original functions. The description on what the actors and use case's role in the system will be discussed in the next subchapter.

4.2.1.2 Actors Description

Actors involved in the system are Student, Teacher and Admin. For student they will get access to most of their data by viewing or downloading it. As for teacher they mostly have access to managing use case means that they gain access to a lot of CRUD operations. Lastly, the admin has a role to manage all of the user account since creating users account is done by the admin. So, they have quite an important role in the system.

4.2.1.3 Use Case Description

Use Case Description describes how each of the use case works if an actors or user using them. All of 17 uses cases will be briefly explaining here.

- 1) UC001 – Sign In: This use case is used for every user in the system to enter the system
- 2) UC002 – Manage Academic Report: This use case is used for teacher to manage the academic report and marking of students
- 3) UC003 - View Academic Report: This use case is used for student to view their academic report
- 4) UC004 – Download Academic Report: This use case is used for student to download their academic report
- 5) UC005 – Manage Attendance Record: This use case is used for teacher to manage the attendance record including to take attendance during class
- 6) UC006 – View Attendance Record: This use case is used for student to views the attendance record
- 7) UC007 – Manage Timetable: This use case is used for student to manage the attendance record including to take attendance during class

- 8) UC008 – View Timetable: This use case is used for student to views the timetable
- 9) UC009 – Download Timetable: This use case is used for student to download the timetable
- 10) UC010 – Manage Tuition Record: This use case is used for teacher to manage the tuition record including approving or rejecting tuition payment
- 11) UC011 – View Tuition Record: This use case is used for student to view the tuition record
- 12) UC012 – Download Tuition Record: This use case is used for student to download the tuition record
- 13) UC013 – Make a Payment: This use case is used for student to download the tuition record
- 14) UC014 – Manage Profile: This use case is used for student and teacher to view and edit their profile
- 15) UC015 – Manage User: This use case is used for admin to manage user account both student and teacher
- 16) UC016 – View Student Info: This use case is used for teacher to view list of students and their detailed information
- 17) UC017 – View Teacher Info: This use case is used for student to view list of students and their detailed information

4.2.1.4 Use Case Specification

Use case specification will describes the use cases in a more detailed manner. In this case, one of the use cases will be used as an example on how use case specification works. The rest of the use case specification can be found in the SRS in the appendix A.

Use Case ID:	UC-002
Use Case Name:	Manage Academic Report
Actors:	Teacher
Description:	This use case is used for teacher to manage the academic report and marking of students
Pre-conditions:	<ol style="list-style-type: none"> Logged in to the system
Normal Flow:	<ol style="list-style-type: none"> Go to the “Rapor” page If the teacher clicks “Penilaian” performs AF1 If the teacher clicks “Edit Penilaian” performs AF2 Select academic year Select semester Select class Select student The system displays the academic report If the teacher wants to download the academic report performs AF3 If the teacher is a homeroom teacher and clicks “Buat Rapor” page performs AF4 If the teacher wants to delete the academic report performs AF5
Alternative Flow:	<ol style="list-style-type: none"> Marking <ol style="list-style-type: none"> System redirects teacher to “Penilaian” page System display the marking form Fill in the marking form System saves student mark Edit Marking <ol style="list-style-type: none"> System redirects teacher to “Edit Penilaian” page System displays the student that the teacher already marked Select student System displays the marking form Edit the mark Download Academic Report <ol style="list-style-type: none"> Click the “Unduh” button to download Create Academic Report <ol style="list-style-type: none"> System redirects teacher to “Buat Rapor” page Select Student Teacher fills in additional information Teacher clicks “Buat Rapor” button Delete Academic Report <ol style="list-style-type: none"> Click “Trash” icon to delete the academic report
Exception:	<p>E.1 Required field(s) is/are empty in marking form</p> <ol style="list-style-type: none"> an error message will be displayed by the system Performs NF2/AF1 <p>E.2 Required field(s) is/are empty when creating Academic Report</p>

Table 4.1: Use case specification for UC002 - Manage Academic Report

	<ol style="list-style-type: none"> 1. an error message will be displayed by the system 2. Performs NF9/AF4
Post-condition s:	<ol style="list-style-type: none"> 1. Teacher successfully views the academic report 2. Teacher successfully downloads the academic report 3. Teacher successfully marks the student 4. Teacher successfully edits the marks of the student 5. Teacher successfully creates the academic report
Related Requirement:	-

4.2.1.5 Sequence Diagram

Sequence diagram is designed to show the action between actors/users and the system's objects. In this case, UC002 will be used again to represent the sequence diagram. The rest of the diagram can be found in the appendix B.

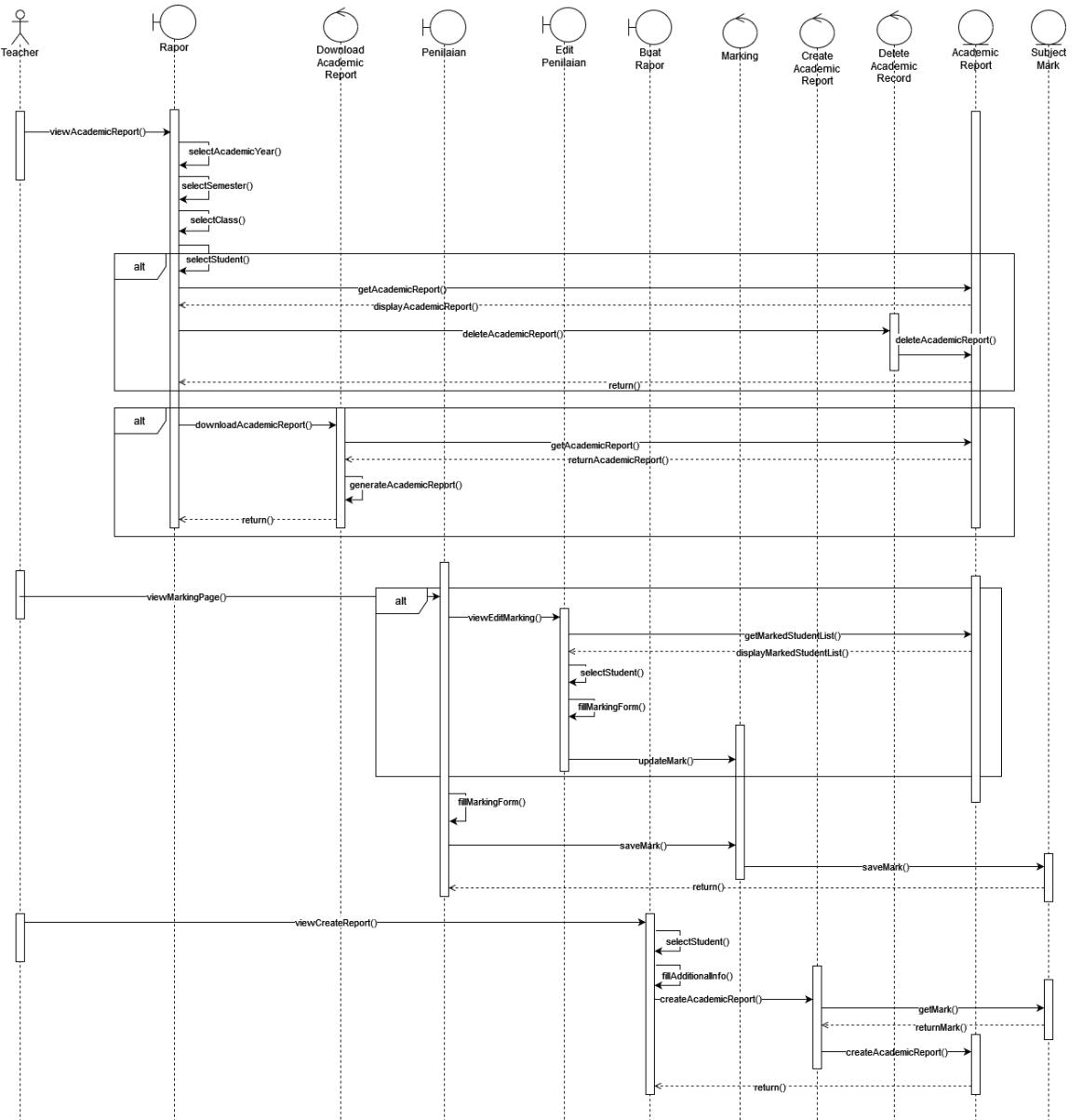


Figure 4.2: Sequence diagram of UC002 – Manage Academic Report

The figure above depicts how the users interact with object in the system through UC002. There are two alternatives that can be done by the teacher when viewing academic report which are deleting them or downloading the academic report. The Teacher can also go to the marking interface to mark student or they can edit the mark in the edit mark page. The last action that the teacher can do in this use case is to create the academic report. After all the action done indicated by the arrow going into the entity and return back to the boundary.

4.2.1.6 Activity Diagram

Activity diagram is used to demonstrate the flow of a certain functions. All of the involved actor will start at the same time and destination and also end at the same destination. UC015-Manage User will be used as the activity diagram in this part. Other activity diagram can be found in appendix A.

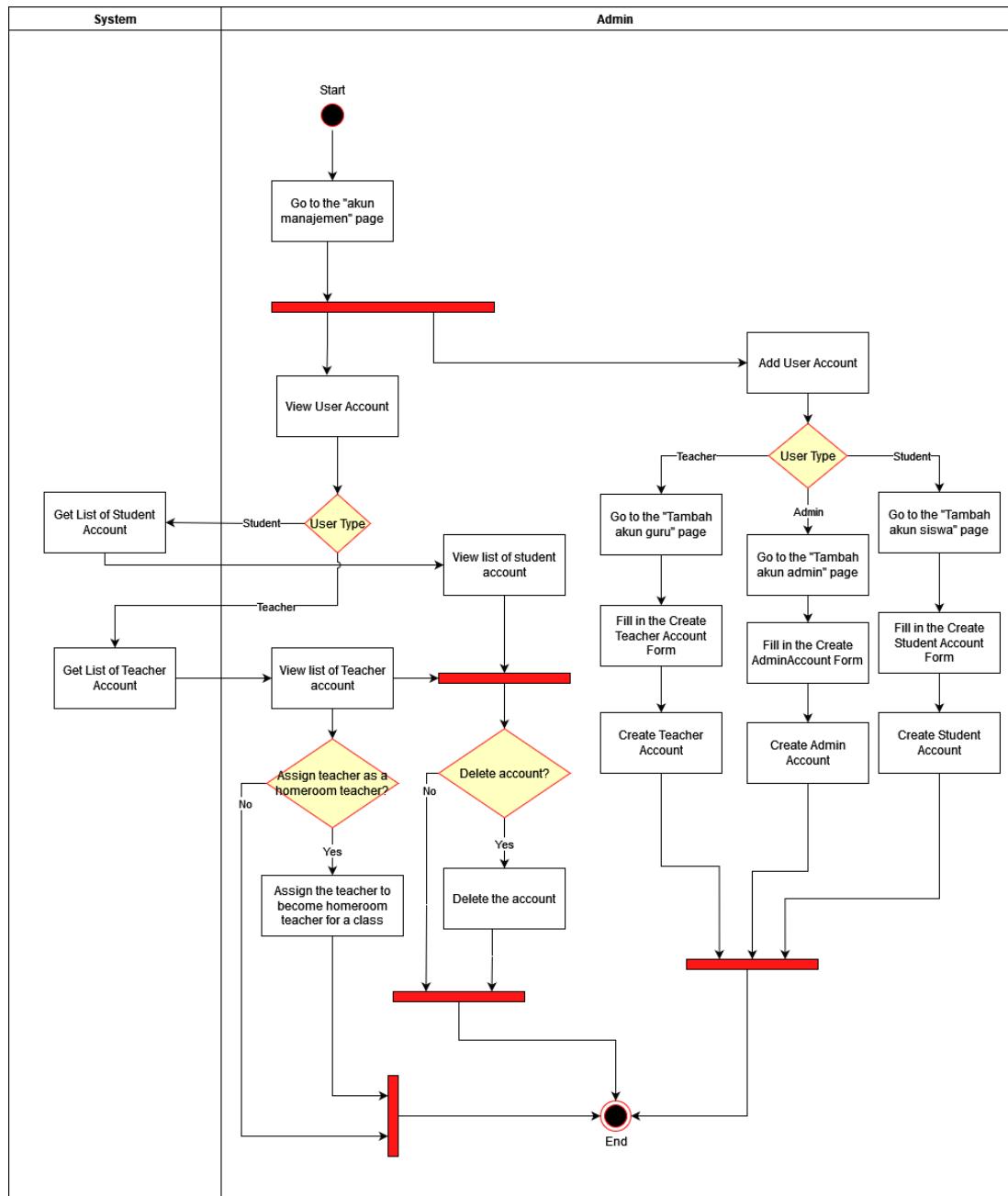


Figure 4.3: Activity diagram of UC0015-Manage User

In this UC, the admin can go to account management page to manage all user account. The admin has a few conditions from the start when viewing user account, they can choose to view students account or teacher account. They can also delete the user account when the system displays the user list. Additionally, when viewing the teacher account list, they can assign them to classes to act as a homeroom teacher. The admin can also add new account since this is the only way for user to be added to the system. Once again, the admin can choose what type of user account they want to create. When creating student account, they can assign the semester and academic year of that student.

4.2.2 Non-functional Requirements

Different from functional requirements, non-functional requirements act as the constraints of the system. Non-functional requirements more focused toward the system itself on what it can do to improve the functionality of the system. The system must achieve the requirements that are listed below.

- 1) Availability – The user should be able to access the system at any moment thus, the system must be available all day every day.
- 2) Security – The system should only allow authorized user from SMAS Muhammadiyah 1 to access the system. The system must provide protection to all of the data exist in the system.
- 3) Usability – The user should be able to explore the system with ease after inspecting the system a little bit. The system must be simple to use and comprehend.
- 4) Portability – The system should work with any web browser. The user shall be able to access the system from their preferred web browser

4.3 Project Design

The architecture used for this proposed system is MVC or Model-View-Controller. In this model the code structure separated into different part which are mode, view and controller. The model in MVC is used as a bridge to transfer data between view and controller. As the name suggest the view in the MVC is for the user interface. And the controller is used for calling the functions.

One of the most important benefits of using MVC in a web-based app is how flexible it is which led to faster process when developing. Since its flexible, it means that changes made will not entirely affect other part of the system. Additionally, it has the benefit of helping the developers repeating the same code because model which represents data is separated from the logic in the controller. All of these benefits will greatly help developing SAMS. Following is the MVC diagram of SAMS.

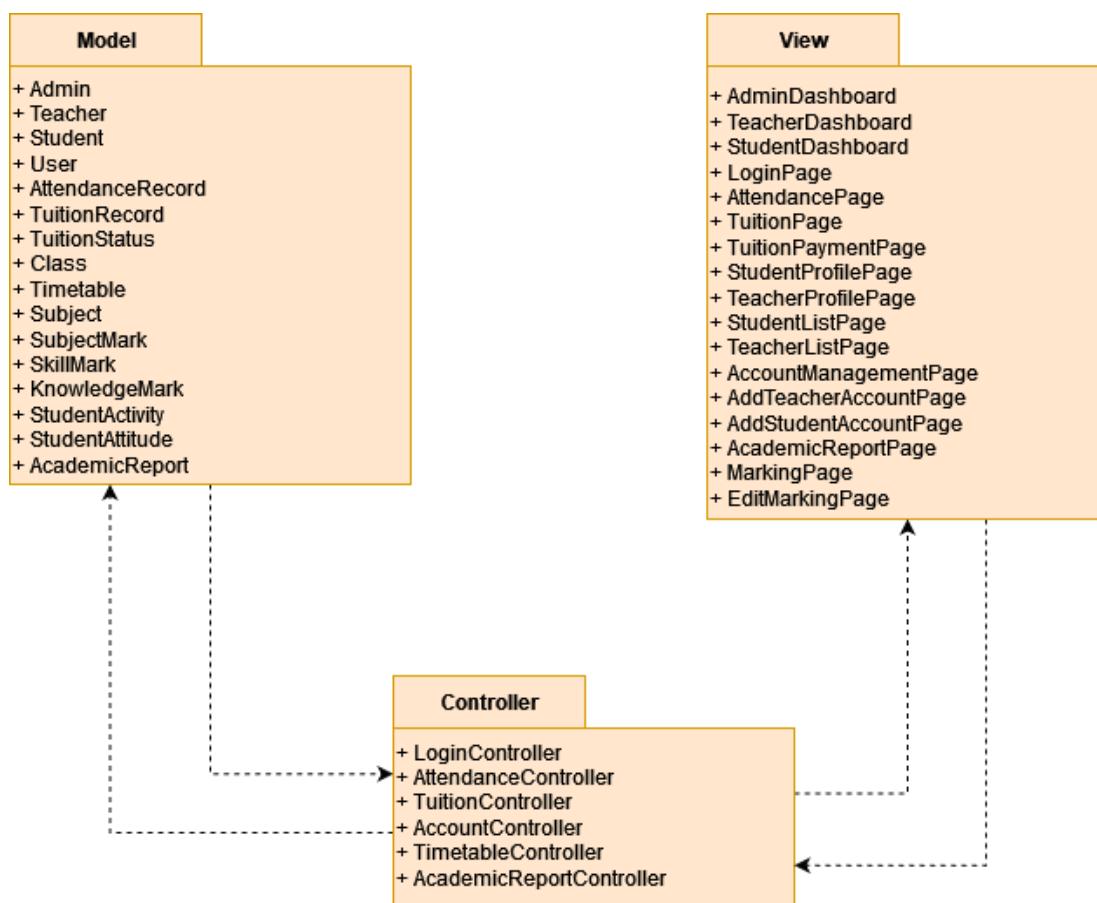


Figure 4.4: Architecture model diagram for SAMS

4.4 Database Design

In this part of the chapter, the design of the database for SAMS will be explored. Since database is a crucial part when developing the proposed system, an excellent database design is required. Following is the Entity Relationship Diagram (ERD) of Student Academic Management System.

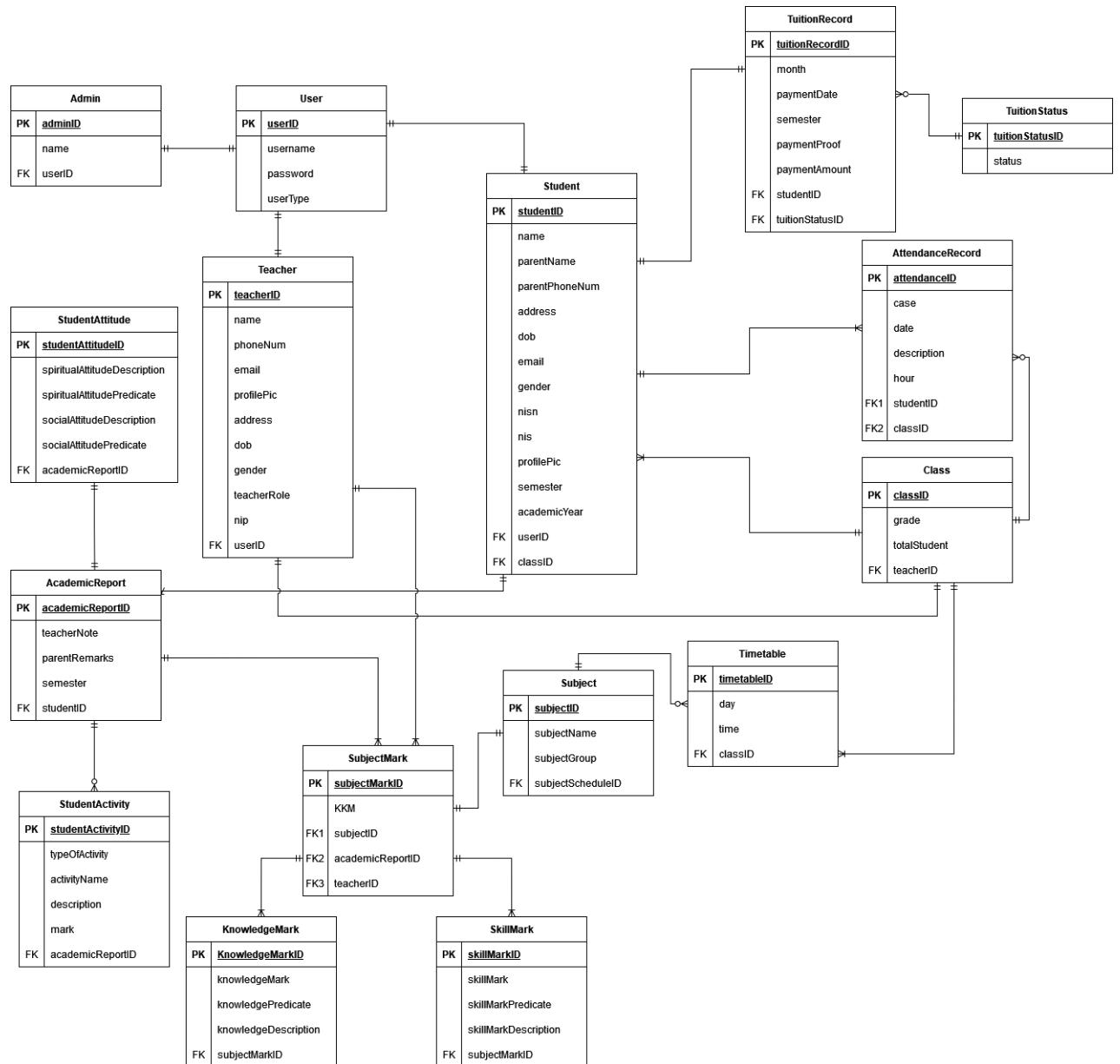


Figure 4.5: ERD of Student Academic Management System

From the ERD figure above, there are total of 16 entities. The purpose of the user entity is to store username and password from all of the different type of users. The academic report table will hold all of the marks of a student when all of the teacher finished marking and the creation of academic report can begin. The mark itself separated into two tables which are knowledge mark and skill mark. The subject mark table then will collect both type of mark and store it until the teacher create the academic report. The attendance table is used for the attendance checking function while the tuition record table is used for managing student's tuition payment. The timetable table will store the timetable created by teacher.

4.4.1 Data Dictionary

Data dictionary provide description about the attributes from the figure above. Here are the data dictionary for two entities which are academic report and knowledge mark.

Table 4.2: Data dictionary of AcademicReport Table

AcademicReport Table			
Field Name	Datatype	Constraint	Description
academicReportID	BIGINT	Primary Key	Unique ID of the AcademicReport table
teacherNote	TEXT	not null	Note from the teacher about the student
semester	VARCHAR	not null	The semester of the student
studentID	BIGINT	Foreign Key	Unique ID of the Student table

Table 4.3: Data dictionary of KnowledgeMark Table

KnowledgeMark table			
Field Name	Datatype	Constraint	Description
knowledgeMarkID	BIGINT	Primary Key	Unique ID of the KnowledgeMark table
knowledgeMark	INT	not null	Knowledge mark of a student for a subject
knowledgePredicate	CHAR	not null	The predicate of the knowledge mark that the student got for a subject
knowledgeDescription	TEXT	nullable	The Description of the knowledge mark that the student got for a subject
subjectMarkID	BIGINT	Foreign Key	Unique ID of the SubjectMark table

4.5 Interface Design

Interface design will be shown in this section. The section will be separated into two parts, Site map or page navigation diagram and the mock up user interface design.

4.5.1 Page Navigation Design

There are three users/actors in the system which mean there will be three different dashboards for each user. The following figures are the page navigation design for teacher, student and admin.

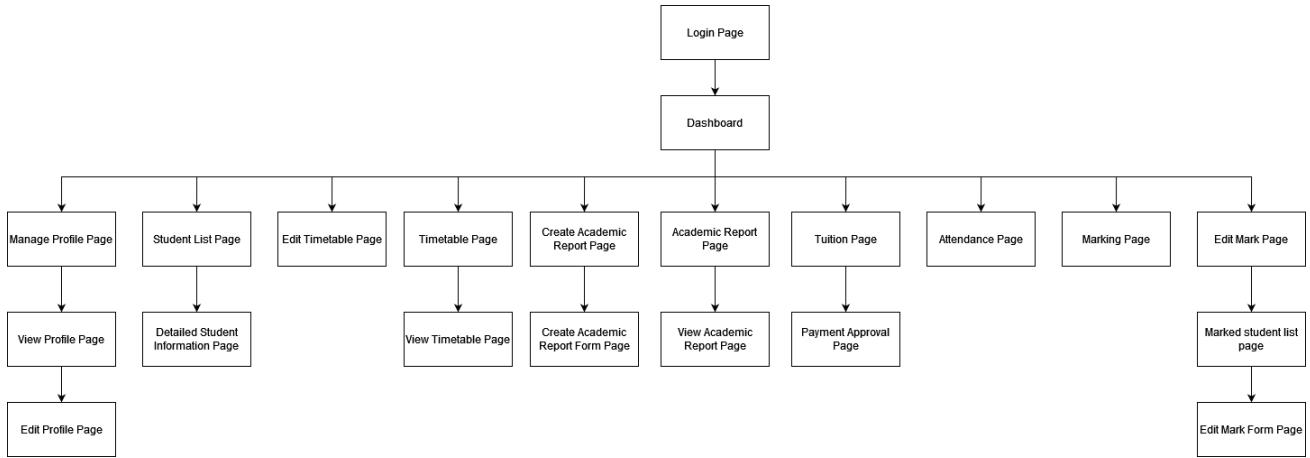


Figure 4.6: Page Navigation for Teacher in SAMS

After entering the system successfully through login, the teacher will be redirect to their own dashboard. From this dashboard the teacher can access all kind of different functions from creating academic report page to managing their profile.

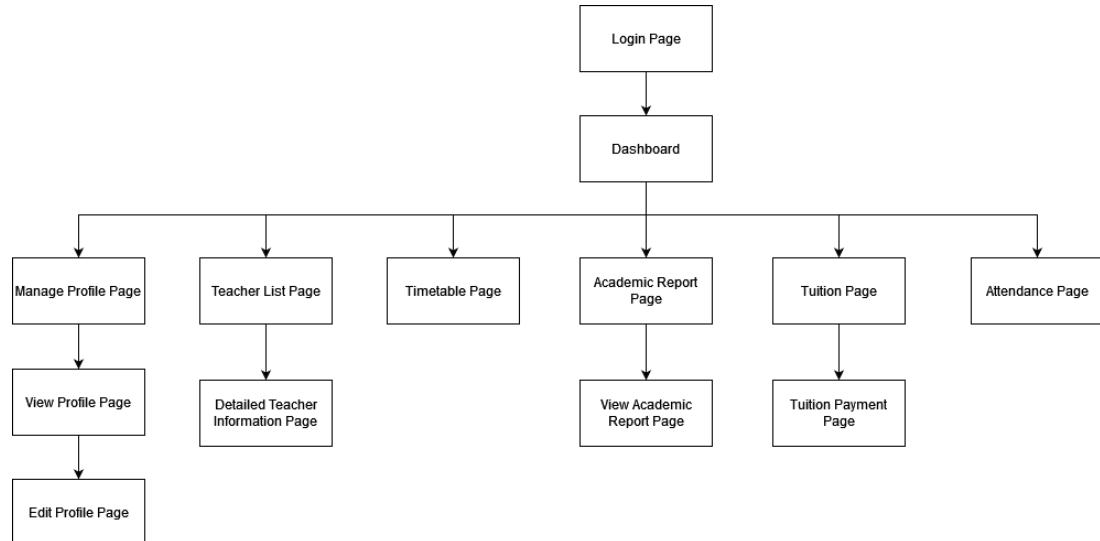


Figure 4.7: Page Navigation for Student in SAMS

After the student login into the system, they also welcomed with their own dashboard. From the dashboard the student can access manage profile page, teacher list page, timetable page, academic report page, tuition page and attendance page.

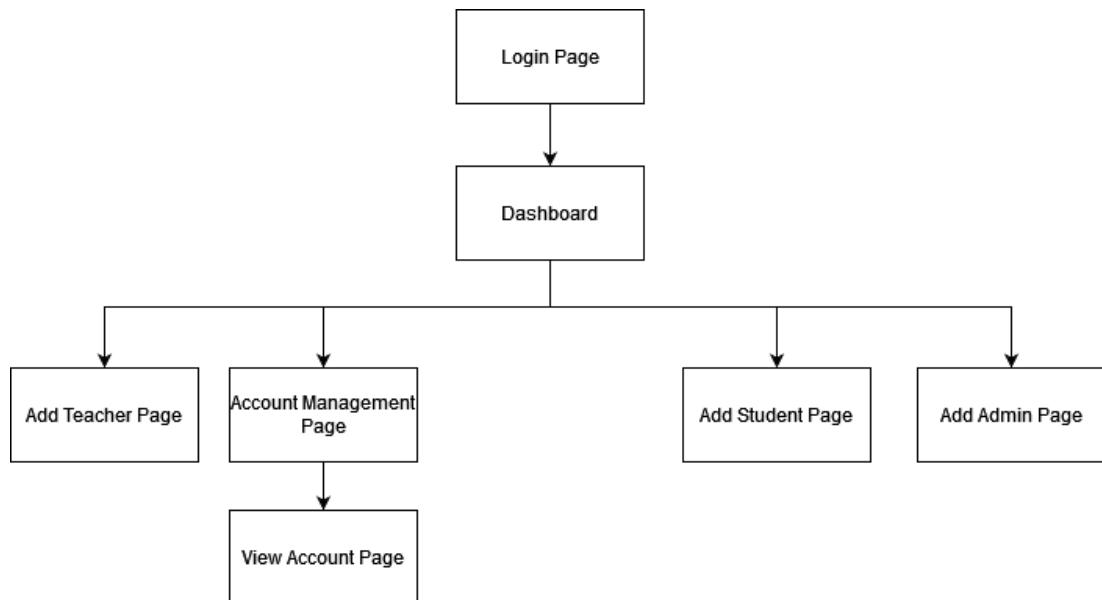


Figure 4.8: Page Navigation for Admin in SAMS

From the figure above, the admin can access to multiple pages after successfully enter the system. These pages are added teacher page, account management page, add student page and add admin page.

4.5.2 User Interface Design

In this part, some of the mockup UI design will be shown, the complete mockup design can be found in the SDD in appendix B.

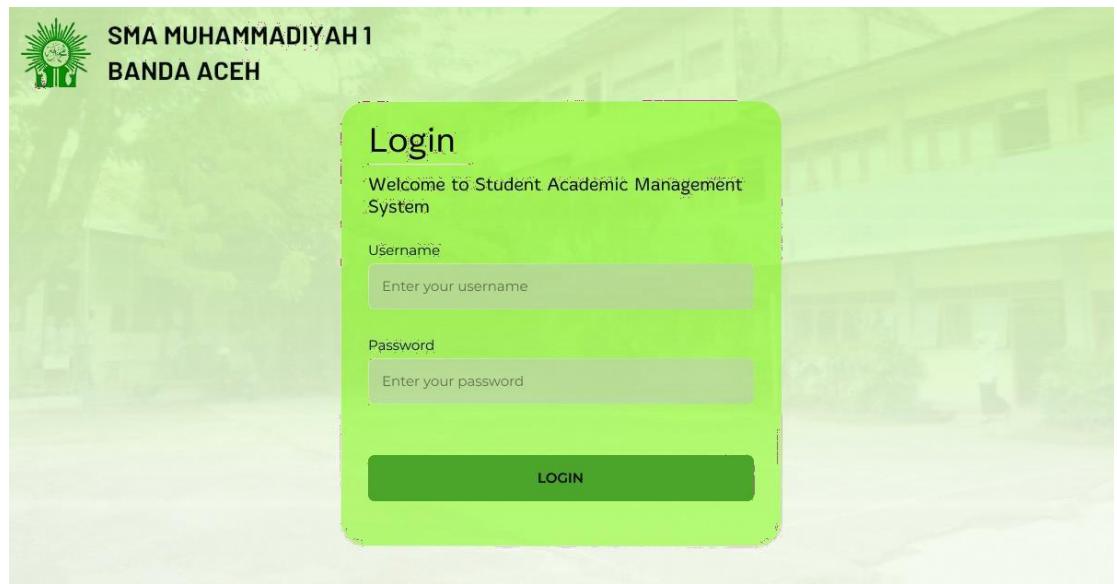


Figure 4.9: Login Page for SAMS

This page is used for the user to enter the system. The username and password will be provided by the admin since they are the only one who can make users account. After successfully login using valid credentials, each user will be redirected to their own dashboard.

The image shows the marking page for teachers in the SAMS system. On the left is a sidebar with navigation links: Dashboard, Rapor, Buat Rapor, Penilaian (which is highlighted in green), Edit Penilaian, Daftar Siswa, Jadwal Pelajaran, Ubah Jadwal Pelajaran, SPP, and Absensi. The main content area is titled "Penilaian". It includes dropdown menus for "Kelas" (XI-MIPA), "Semester" (Ganjil), "Siswa" (Sheila Faiqah Aurora), "Tahun Pelajaran" (2021/2022), "Mata Pelajaran" (Matematika), "KKM" (70), and "Aspek" (Keterampilan). There is also a numeric input field for "Nilai" (84) and a large text area for "Deskripsi". A "Simpan" button is located at the bottom right.

Figure 4.10: Marking Page for Teacher in SAMS

In this page, the teacher can begin marking students. They can select between two type of marks which are knowledge mark and skill mark. They can also set the KKM which mean passing grade.

The screenshot shows the 'Rapor' (Report) section of the SAMS application. On the left sidebar, under 'Rapor', there are links for 'Buat Rapor', 'Penilaian', 'Edit Penilaian', 'Daftar Siswa', 'Jadwal Pelajaran', 'Ubah Jadwal Pelajaran', 'SPP', and 'Absensi'. The main content area is titled 'Rapor' and includes a 'Sortir' (Sort) section with dropdown menus for 'Tahun Ajaran' (2021/2022), 'Semester' (Ganjil), and 'Kelas' (XI MIPA). Below this is a table with a single row for 'Sheila Faiqah Aurora'. The table columns are 'Nama' (Name), 'NIS/NISN', 'Peminatan' (Majors), and actions. The 'Actions' column contains a 'Tinjau' (View) button with a magnifying glass icon, a 'Download' button with a download icon, and a delete button with a red 'X' icon.

Figure 4.11: Academic Report Page for Teacher in SAMS

In this page the teacher can sort and search the student that they want to view or download their academic report. The view academic report page is available under appendix B.

The screenshot shows the 'Manajemen Akun' (Account Management) section of the SAMS application. On the left sidebar, under 'Manajemen Akun', there are links for 'Tambah Akun Guru', 'Tambah Akun Siswa', and 'Tambah Akun Admin'. The main content area is titled 'Manajemen Akun' and includes a 'Sortir' (Sort) section with a dropdown menu set to 'Guru'. Below this is a table listing five teacher accounts. The table columns are 'Nama' (Name), 'NIP', 'Wali Kelas' (Homeroom Teacher), and actions. The 'Actions' column contains a delete button with a red 'X' icon. At the top right of the table is a 'Tambah Akun Guru' (Add Teacher Account) button.

Figure 4.12: Account Management for Admin in SAMS

An admin can choose which user to view in this page. In this case the teacher account list is displayed. The admin then can set the teacher to be a homeroom teacher for a specific class or they can delete unused teacher account.

4.6 Chapter Summary

This chapter consist of requirement and designs for the proposed system. The requirement stated whether functional or non-functional will act as a standard for the system. The Model-View-Controller based system architecture design, database design and mockup interface design are also included in the chapter that will provide some basis for when developing the system later.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Introduction

In this chapter the implementation of the system along with the testing will be discussed. The chapter will provide the looks and explanation of the coding and interfaces of the system's main functions. The testing that are done to the system such as Black box testing, White box testing and User testing will also be discussed.

5.2 Coding of System Main Functions

Below are some of the main functions in the system

5.2.1 Login Function

```
const ADMIN = 1;
const TEACHER = 2;
const STUDENT = 3;

public function store(LoginRequest $request) {
    $request->authenticate();
    $request->session()->regenerate();

    switch (auth()->user()->userType) {
        case self::ADMIN:
            return redirect('/dashboard-admin');
        case self::TEACHER:
            return redirect('/dashboard-guru');
        case self::STUDENT:
            return redirect('/dashboard-siswa');
        default:
            return redirect('/');
    }
}
```

Figure 5.1: Login Function for Users

5.2.2 Logout Function

```
public function destroy(Request $request)
{
    $userType = auth()->user()->userType;
    Auth::guard('web')->logout();

    $request->session()->invalidate();
    $request->session()->regenerateToken();

    switch ($userType) {
        case self::ADMIN:
            return redirect('/');
        case self::TEACHER:
            return redirect('/');
        case self::STUDENT:
            return redirect('/');
        default:
            return redirect('/');
    }
}
```

Figure 5.2: Logout Function for Users

5.2.3 Add Teacher Account Function (Admin)

```
public function storeTeacher(Request $request)
{
    $data = $request->validate([
        'username' => 'required|string|max:255',
        'password' => 'required|string|confirmed|min:8',
        'userType' => ['required', Rule::in([1, 2, 3])],
    ]);

    $data['password'] = Hash::make($request->password);

    $user = User::create($data);

    if ($request->userType === 2) {
        $teacherData = [
            'user_id' => $user->id,
        ];

        $teacher = new Teacher($teacherData);
        $teacher->user_id = $user->id;
        $teacher->save();
    }

    return Inertia::render('Admin/TambahAkunGuru');
}
```

Figure 5.3: Add Teacher Account Function for Users

5.2.4 Add Admin Account (Admin)

```
public function storeAdmin(Request $request)
{
    $data = $request->validate([
        'name' => 'required|string|max:255',
        'username' => 'required|string|max:255',
        'password' => 'required|string|confirmed|min:8',
        'userType' => ['required', Rule::in([1, 2, 3])],
    ]);

    $data['password'] = Hash::make($request->password);

    $user = User::create($data);

    if ($request->userType === 1) {
        $admin = new Admin();
        $admin->name = $request->name;
        $user->admin()->save($admin);
    }

    return Inertia::render('Admin/TambahAkunAdmin');
}
```

Figure 5.4: Add Admin Account Function for Admin

5.2.5 Add Student Account (Admin)

```
public function storeStudent(Request $request)
{
    $data = $request->validate([
        'username' => 'required|string|max:255',
        'password' => 'required|string|confirmed|min:8',
        'userType' => ['required', Rule::in([1, 2, 3])],
        'selectedYear' => 'required',
        'selectedSemester' => 'required',
        'selectedPeminatan' => 'required',
        'selectedClassroom' => 'required',
    ]);

    $data['password'] = Hash::make($request->password);

    $user = User::create($data);

    if ($request->userType === 3) {
        $studentData = [
            'user_id' => $user->id,
            'semester' => $request->selectedSemester,
            'academicYear' => $request->selectedYear,
            'peminatan' => $request->selectedPeminatan,
            'classroom_id' => $request->selectedClassroom,
        ];
        $student = Student::create($studentData);
    }

    return Inertia::render('Admin/TambahAkunSiswa');
}
```

Figure 5.5: Add Student Account Function for Users

5.2.6 Display User Based on Roles Function (Admin)

```
public function getUser($type = null)
{
    $query = User::query();

    if ($type === 'admin') {
        $query->where('userType', 1)->with('admin');
    } elseif ($type === 'teacher') {
        $query->where('userType', 2)->with('teacher');
    } elseif ($type === 'student') {
        $query->where('userType', 3)->with('student');
    }

    $users = $query->get();

    foreach ($users as $user) {
        if ($user->student) {
            $user->name = $user->student->name;
            $user->nis_nism = $user->student->nis . '/' . $user->student->nism;
            $user->peminatan = $user->student->peminatan;
        } elseif ($user->teacher) {
            $user->name = $user->teacher->name;
            $user->nip = $user->teacher->nip;
            $user->teacherRole = $user->teacher->teacherRole;
        } elseif ($user->admin) {
            $user->name = $user->admin->name;
        }
    }

    return Inertia::render('Admin/ManajemenAkun', [
        'users' => $users,
        'filter' => $type
    ]);
}
```

Figure 5.6: Display User Function for Admin

5.2.7 Teacher List Function (Student)

```
public function getUser($type = null)
{
    $teachers = Teacher::with('user')->get();

    return Inertia::render('Student/DaftarGuru', ['teachers' => $teachers]);
}
```

Figure 5.7: Teacher List Function for Student

5.2.8 Update Profile Function (Student)

```
public function updateProfile(Request $request)
{
    $request->validate([
        'password' => ['nullable', 'string', 'min:8', 'confirmed'],
        'name' => ['nullable', 'string', 'max:255'],
        'parentName' => ['nullable', 'string', 'max:255'],
        'parentPhoneNum' => ['nullable', 'string', 'max:255'],
        'address' => ['nullable', 'string', 'max:255'],
        'dob' => ['nullable', 'date'],
        'gender' => ['nullable', 'string', 'max:255'],
        'nis' => ['nullable', 'integer'],
        'nism' => ['nullable', 'integer'],
    ]);

    $user = auth()->user();
    if ($request->filled('password')) {
        $user->password = Hash::make($request->password);
        $user->save();
    }

    $student = $user->student;
    if ($request->filled('name') || $request->filled('parentName') || $request->filled('parentPhoneNum') || $request->filled('address') || $request->filled('all()'))
        $student->update($request->all());

    return Inertia::render('Student/StudentProfile');
}
```

Figure 5.8: Update Profile Function for Student

5.2.9 View Profile (Student)

```
public function getProfile(Request $request)
{
    $user = auth()->user();
    $student = $user->student;

    return Inertia::render('Student/StudentProfile', [
        'student' => $student,
        'user' => $user
    ]);
}
```

Figure 5.9: View Profile Function for Student

5.2.10 View Tuition Record (Student)

```
public function getTuitionRecords()
{
    $user = auth()->user();
    $student = $user->student;
    $id = $student->id;

    $tuitionRecordsFromDatabase = TuitionRecord::where('student_id', $id)
        ->get();

    $months_string = ['Januari', 'Februari', 'Maret', 'April', 'Mei', 'Juni', 'Juli', 'Agustus', 'September', 'Oktober', 'November', 'Desember'];
    $months = range(1, 12);

    $tuitionRecords = collect($months)->map(function ($month) use ($tuitionRecordsFromDatabase, $months_string) {
        $record = $tuitionRecordsFromDatabase->firstWhere('month', (int) $month);

        if ($record) {
            return [
                'month' => $months_string[$record->month - 1],
                'paymentDate' => $record->paymentDate,
                'paymentAmount' => $record->paymentAmount,
                'paymentProof' => $record->paymentProof,
                'tuitionStatus' => [
                    'status' => $record->tuitionStatus
                ],
            ];
        } else {
            return [
                'month' => $months_string[$month - 1],
                'paymentDate' => null,
                'paymentAmount' => null,
                'paymentProof' => null,
                'tuitionStatus' => [
                    'status' => null
                ],
            ];
        }
    });

    return Inertia::render('Student/StudentSPP', [
        'tuitionRecords' => $tuitionRecords,
        'studentId' => $id,
        'academicYear' => $student->academicYear,
    ]);
}
```

Figure 5.10: View Tuition Function for Student

5.2.11 Save Tuition Record (Student)

```
public function saveTuitionRecord(Request $request)
{
    $user = auth()->user();
    $student = $user->student();
    $id = $student->id;

    $month = $request->input('month');

    $tuitionRecord = TuitionRecord::where('student_id', $id)
        ->where('month', $month)
        ->first();

    if (!$tuitionRecord) {
        $tuitionRecord = new TuitionRecord;
        $tuitionRecord->student_id = $id;
        $tuitionRecord->month = $month;
    }

    if ($request->hasFile('paymentProof')) {
        $file = $request->file('paymentProof');
        $filename = time() . '.' . $file->getClientOriginalExtension();
        $file->move(public_path('uploads'), $filename);
        $tuitionRecord->paymentProof = $filename;
    }

    $tuitionRecord->paymentDate = $request->input('paymentDate');
    $tuitionRecord->paymentAmount = $request->input('paymentAmount');

    $unpaidStatus = TuitionStatus::where('status', 'Unpaid')->first();
    if ($unpaidStatus) {
        $tuitionRecord->tuitionStatus_id = $unpaidStatus->id;
    } else {

        $unpaidStatus = new TuitionStatus;
        $unpaidStatus->status = 'Unpaid';
        $unpaidStatus->save();
        $tuitionRecord->tuitionStatus_id = $unpaidStatus->id;
    }

    $tuitionRecord->save();

    return response()->json(['message' => 'Tuition record saved successfully']);
}
```

Figure 5.11: Save Tuition Function for Student

5.2.12 View Academic Report (Student)

```
public function viewAcademicReport($id)
{
    $user = auth()->user();
    $student = $user->student();
    $id = $student->id;

    $academicReport = $id->academicReport;

    $subjectMarks = $academicReport->subjectMarks;

    $skillMarks = [];
    $knowledgeMarks = [];
    foreach ($subjectMarks as $subjectMark) {
        $skillMark = $subjectMark->skillMark;
        $knowledgeMark = $subjectMark->knowledgeMark;
        array_push($skillMarks, $skillMark);
        array_push($knowledgeMarks, $knowledgeMark);
    }

    return Inertia::render('Student/studentRapor', [
        'academicReport' => $academicReport,
        'subjectMarks' => $subjectMarks,
        'skillMarks' => $skillMarks,
        'knowledgeMarks' => $knowledgeMarks,
    ]);
}
```

Figure 5.7: View Academic Report Function for Student

5.2.13 View Student (Teacher)

```
public function getUserPenilaian($type = null)
{
    $students = Student::with('user')->get()->map(function ($student) {
        $student->nis_nism = $student->nis . '/' . $student->nism;
        $student->classroom_id = $student->classroom_id;
        return $student;
    });

    $classrooms = Classroom::all();
    return Inertia::render('Teacher/Penilaian', ['students' => $students, 'classrooms' => $classrooms]);
}

public function getUser($type = null)
{
    $students = Student::with('user')->get()->map(function ($student) {
        $student->nis_nism = $student->nis . '/' . $student->nism;
        $student->classroom_id = $student->classroom_id;
        return $student;
    });

    $classrooms = Classroom::all();
    return Inertia::render('Teacher/DaftarSiswa', ['students' => $students, 'classrooms' => $classrooms]);
}

public function getUserSPP($type = null)
{
    $students = Student::with('user')->get()->map(function ($student) {
        $student->nis_nism = $student->nis . '/' . $student->nism;
        $student->classroom_id = $student->classroom_id;
        return $student;
    });

    $classrooms = Classroom::all();
    return Inertia::render('Teacher/SPP', ['students' => $students, 'classrooms' => $classrooms]);
}

public function getUserAbsensi($type = null)
{
    $students = Student::with('user')->get()->map(function ($student) {
        $student->nis_nism = $student->nis . '/' . $student->nism;
        $student->classroom_id = $student->classroom_id;
        return $student;
    });

    $classrooms = Classroom::all();
    return Inertia::render('Teacher/Absensi', ['students' => $students, 'classrooms' => $classrooms]);
}

public function getUserRapor($type = null)
{
    $students = Student::with('user')->get()->map(function ($student) {
        $student->nis_nism = $student->nis . '/' . $student->nism;
        $student->classroom_id = $student->classroom_id;
        return $student;
    });
    You, 2 days ago * commit project's staging version

    $classrooms = Classroom::all();
    return Inertia::render('Teacher/TeacherRapor', ['students' => $students, 'classrooms' => $classrooms]);
}
```

Figure 5.13: View Student Function for Teacher in various pages

5.2.14 View Input Mark (Teacher)

```
public function getInputMarksPage($studentId)
{
    $student = Student::with('user')->find($studentId);
    $subjects = Subject::all(); // Fetch all subjects

    return Inertia::render('Teacher/InputPenilaian', ['student' => $student, 'subjects' => $subjects]);
}
```

Figure 5.14: View Input Mark Function for Teacher

5.2.15 View Tuition Record (Teacher)

```
public function getTuitionRecords($id)
{
    $student = Student::find($id);

    $tuitionRecordsFromDatabase = TuitionRecord::with('tuitionStatus')->where('student_id', $id)->get();

    $months_string = ['Januari', 'Februari', 'Maret', 'April', 'Mei', 'Juni', 'Juli', 'Agustus', 'September', 'Oktober', 'November', 'Desember'];
    $months = range(1, 12);
    $tuitionRecords = collect($months)->map(function ($month) use ($tuitionRecordsFromDatabase, $months_string) {
        $record = $tuitionRecordsFromDatabase->firstWhere('month', $month);

        if ($record) {
            return [
                'month' => $months_string[$record->month -1],
                'paymentDate' => $record->paymentDate,
                'paymentAmount' => $record->paymentAmount,
                'paymentProof' => $record->paymentProof,
                'tuitionStatus' => [
                    'status' => $record->tuitionStatus->status ?? null,
                ],
                'tuitionStatus_id' => $record->tuitionStatus->id ?? null,
            ];
        } else {
            return [
                'month' => $months_string[$month -1],
                'paymentDate' => null,
                'paymentAmount' => null,
                'paymentProof' => null,
                'tuitionStatus' => [
                    'status' => null
                ],
                'tuitionStatus_id' => null,
            ];
        }
    });
}

// get tuition status
$tuitionStatuses = TuitionStatus::all();

return Inertia::render('Teacher/ViewSPP', [
    'tuitionRecords' => $tuitionRecords,
    'studentId' => $id,
    'studentName' => $student->name,
    'tuitionStatuses' => $tuitionStatuses,
]);
}
```

Figure 5.15: View Tuition Record Function for Teacher

5.2.16 Update Tuition Status (Teacher)

```
public function updateTuitionStatus(Request $request, $id, $month)
{
    $request->validate([
        'tuitionStatus_id' => 'required|exists:tuition_statuses,id',
    ]);

    $tuitionRecord = TuitionRecord::where('student_id', $id) ->where('month', $month)->first();

    if (!$tuitionRecord) {
        return response()->json(['message' => 'Tuition record not found'], 404);
    }

    $tuitionRecord->tuitionStatus_id = $request->input('tuitionStatus_id');
    $tuitionRecord->save();

    return redirect()->back();
}
```

Figure 5.16: View Tuition Status Function for Teacher

5.2.17 View Profile Page (Teacher)

```
public function getProfile(Request $request)
{
    $user = auth()->user();
    $teacher = $user->teacher;

    return Inertia::render('Teacher/TeacherProfile', [
        'teacher' => $teacher,
        'user' => $user
    ]);
}
```

Figure 5.17: View Profile Function for Teacher

5.2.18 View Edit Profile Page (Teacher)

```
public function getProfileEdit(Request $request)
{
    $user = auth()->user();
    $teacher = $user->teacher;

    return Inertia::render('Teacher/TeacherProfileEdit', [
        'teacher' => $teacher,
        'user' => $user
    ]);
}
```

Figure 5.18: View Edit Profile Function for Teacher

5.2.19 Update Profile (Teacher)

```
public function updateProfile(Request $request)
{
    You, 2 minutes ago • Uncommitted changes
    $request->validate([
        'password' => ['nullable', 'string', 'min:8', 'confirmed'],
        'name' => ['nullable', 'string', 'max:255'],
        'phoneNum' => ['nullable', 'string', 'max:255'],
        'address' => ['nullable', 'string', 'max:255'],
        'dob' => ['nullable', 'date'],
        'gender' => ['nullable', 'string', 'max:255'],
        'teacherRole' => ['nullable', 'string', 'max:255'],
        'nip' => ['nullable', 'integer'],
    ]);

    $user = auth()->user();
    if ($request->filled('password')) {
        $user->password = Hash::make($request->password);
        $user->save();
    }

    $teacher = $user->teacher;
    if ($request->filled('name') || $request->filled('phoneNum') || $request->filled('address') || $request->filled('dob') ||
        $request->filled('gender') || $request->filled('teacherRole') || $request->filled('nip')) {
        $teacher->update($request->all());
    }

    return Inertia::render('Teacher/TeacherProfile');
}
```

Figure 5.19: Update Profile Function for Teacher

5.2.20 Save Mark (Teacher)

```
public function saveMark(Request $request)
{
    $validatedData = $request->validate([
        'kkm' => 'required|integer',
        'nilai' => 'required|integer',
        'deskripsi' => 'required|string',
        'aspek' => 'required|string',
        'subject_id' => 'required|integer',
        'student_id' => 'required|integer',
    ]);

    $subjectMark = new SubjectMark;
    $subjectMark->kkm = $validatedData['kkm'];
    $subjectMark->subject_id = $validatedData['subject_id'];
    $subjectMark->teacher_id = auth()->user()->id;
    $subjectMark->save();

    if ($validatedData['aspek'] == 'skill') {
        $mark = new SkillMark;
        $mark->skillMark = $validatedData['nilai'];
        $mark->skillMarkPredicate = $this->getPredicate($validatedData['nilai']);
        $mark->skillMarkDescription = $validatedData['deskripsi'];
    } else {
        $mark = new KnowledgeMark;
        $mark->knowledgeMark = $validatedData['nilai'];
        $mark->knowledgePredicate = $this->getPredicate($validatedData['nilai']);
        $mark->knowledgeDescription = $validatedData['deskripsi'];
    }

    $mark->subjectMark_id = $subjectMark->id;
    $mark->save();

    return response()->json(['message' => 'Mark saved successfully']);
}
```

Figure 5.20: Save Mark Function for Teacher

5.2.21 View Timetable (Teacher)

```
public function getTimetable(Request $request, $classId)
{
    $classrooms = Classroom::all();

    $timetable = Timetable::where('class_id', $classId)
        ->with([
            'hour1Subject', 'hour2Subject', 'hour3Subject', 'hour4Subject',
            'hour5Subject', 'hour6Subject', 'hour7Subject', 'hour8Subject'
        ])->get();

    $subjects = Subject::all();

    $subjects->push((object)[
        'id' => -1,
        'subjectName' => 'Break Time'
    ]);

    return Inertia::render('Teacher/JadwalPelajaran', ['classrooms' => $classrooms, 'timetable' => $timetable, 'subjects' => $subjects]);
}
```

Figure 5.21: View Timetable Function for Teacher

5.2.22 View Timetable (Student)

```
public function getTimetableStudent(Request $request, $classId)
{
    $classrooms = Classroom::all();

    $timetable = Timetable::where('class_id', $classId)
        ->with([
            'hour1Subject', 'hour2Subject', 'hour3Subject', 'hour4Subject',
            'hour5Subject', 'hour6Subject', 'hour7Subject', 'hour8Subject'
        ])->get();

    $subjects = Subject::all();

    $subjects->push((object)[
        'id' => -1,
        'subjectName' => 'Break Time'
    ]);

    return Inertia::render('Student/JadwalPelajaran', ['classrooms' => $classrooms, 'timetable' => $timetable, 'subjects' => $subjects]);
}
```

Figure 5.22: View Timetable Function for Student

5.2.23 Search Function (General)

The search function is implemented in the front end. Below is one of the example of the implementation.

```
const filters = ref({
    classroom_id: { value: null, matchMode: FilterMatchMode.EQUALS },
    semester: { value: null, matchMode: FilterMatchMode.CONTAINS },
    academicYear: { value: null, matchMode: FilterMatchMode.CONTAINS },
    global: { value: null, matchMode: FilterMatchMode.CONTAINS },
    name: { value: null, matchMode: FilterMatchMode.STARTS_WITH },
    status: { value: null, matchMode: FilterMatchMode.EQUALS },
    verified: { value: null, matchMode: FilterMatchMode.EQUALS }
});
```

Figure 5.23: Search Function in General

5.3 Interfaces of System Main Functions

Below are some of the main functions interfaces of the system.

5.3.1 Login Page

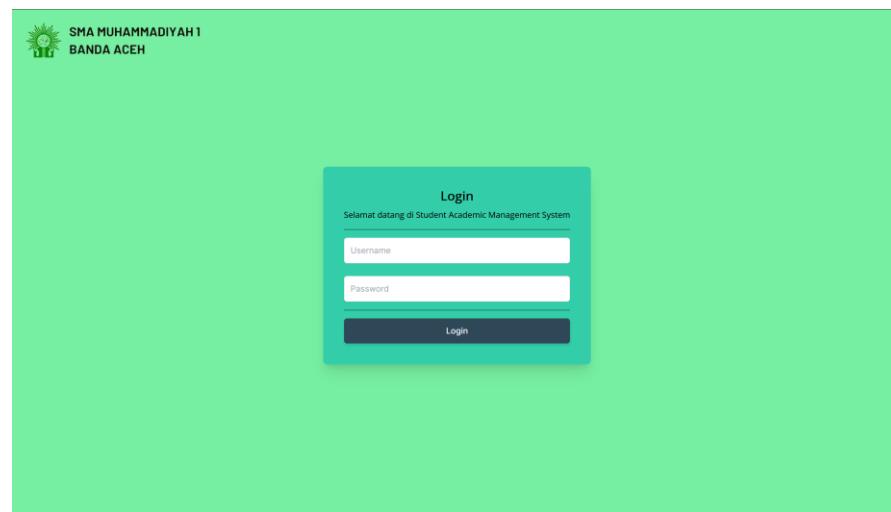


Figure 5.24: Login Page for All Users

5.3.2 Admin Dashboard Page

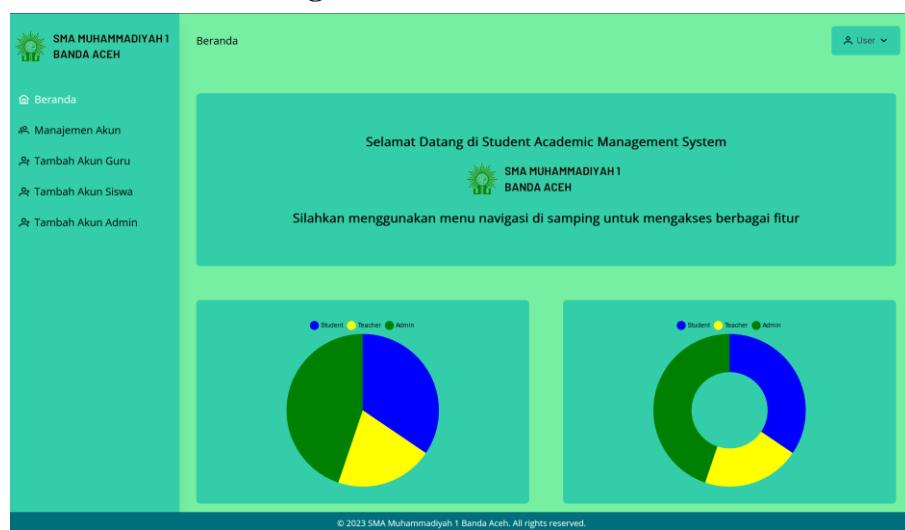


Figure 5.25: Dashboard Page for Admin

5.3.3 Account Management Page

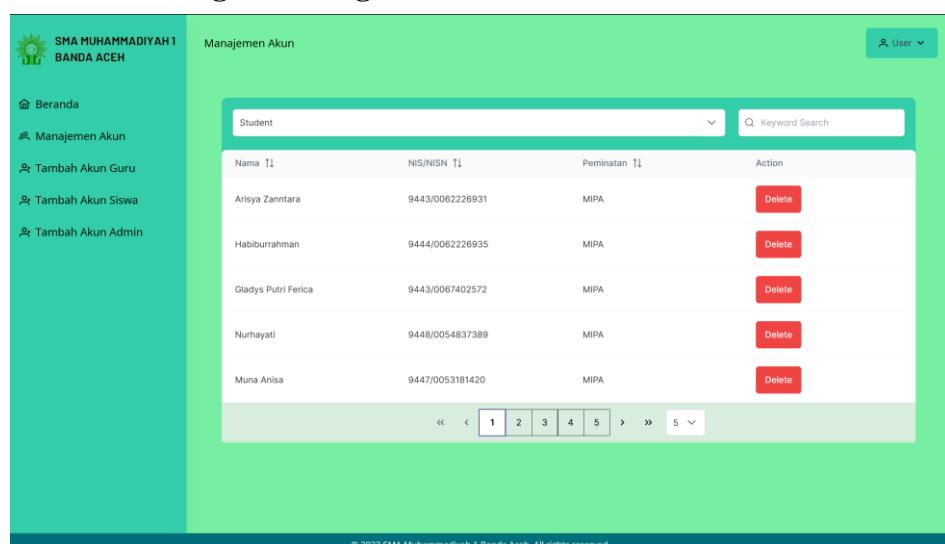


Figure 5.26: Account Management Page for Admin

5.3.4 Add Teacher Account Page

SMA MUHAMMADIYAH 1
BANDA ACEH

Beranda

Manajemen Akun

Tambah Akun Guru

Tambah Akun Siswa

Tambah Akun Admin

Tambah Akun Guru

Nama Pengguna

Kata Sandi

Masukkan Kembali Kata Sandi

Tambah

© 2023 SMA Muhammadiyah 1 Banda Aceh. All rights reserved.

Figure 5.27: Add Teacher Page for Admin

5.3.5 Add Student Account Page

SMA MUHAMMADIYAH 1
BANDA ACEH

Beranda

Manajemen Akun

Tambah Akun Guru

Tambah Akun Siswa

Tambah Akun Admin

Tambah Akun Siswa

Nama Pengguna

Kata Sandi

Masukkan Kembali Kata Sandi

Tahun Ajaran

Semester

Peminatan

Kelas

Tambah

© 2023 SMA Muhammadiyah 1 Banda Aceh. All rights reserved.

Figure 5.28: Add Student Page for Admin

5.3.6 Add Admin Account Page

Tambah Akun Admin

Nama Lengkap

Nama Pengguna

Kata Sandi

Masukkan Kembali Kata Sandi

Tambah

Figure 5.29: Add Admin Page for Admin

5.3.7 Student Dashboard Page

Beranda

	1	2	3	4	5	6	7	8
Senin	Matematika							
Selasa	Matematika							
Rabu	Matematika							
Kamis	Fisika							
Jumat	Fisika							
Sabtu	Fisika							
Minggu	Biologi							

Figure 5.30: Dashboard Page for Student

5.3.8 Student Academic Report Page

The screenshot shows a web-based application interface for students. On the left side, there is a vertical sidebar with a teal background containing the school logo and name "SMA MUHAMMADIYAH 1 BANDA ACEH". Below the logo, there are several navigation links: "Beranda", "Rapor", "SPP", "Absensi", and "Daftar Guru". The main content area has a light green background and contains the word "Rapor" at the top. Below it are three dropdown menus: "Tahun Ajaran", "Semester", and "Kelas", followed by a "Unduh" button. At the bottom of the page, there is a dark footer bar with the text "© 2023 SMA Muhammadiyah 1 Banda Aceh. All rights reserved."

Figure 5.31: Academic Report Page for Student

5.3.9 Student Tuition Page

The screenshot shows the tuition payment page. The sidebar on the left includes the school logo and name, and navigation links for "Beranda", "Rapor", "SPP" (selected), "Absensi", and "Daftar Guru". The main content area has a light green background and displays the title "SPP". Below it is a section titled "Input SPP" with a dropdown for "Tahun Ajaran 2021/2022". There are four input fields: "Pilih Bulan" (dropdown), "Pilih Tanggal" (date input), "Masukkan Jumlah" (text input), and "Upload Bukti Pembayaran" (file upload input with placeholder "Browse... No file selected"). Below these inputs is a table showing tuition records:

Bulan	Tanggal	Total	Bukti	Status
Januari	2023-01-03	1111111		
Februari	2023-02-08	1234561	1688058194.jpeg	
Maret				
April				
Mei				
Juni	2023-01-02	800076		
Juli				

At the bottom of the page, there is a dark footer bar with the text "© 2023 SMA Muhammadiyah 1 Banda Aceh. All rights reserved."

Figure 5.32: Tuition Page for Student

5.3.10 Student Attendance Page

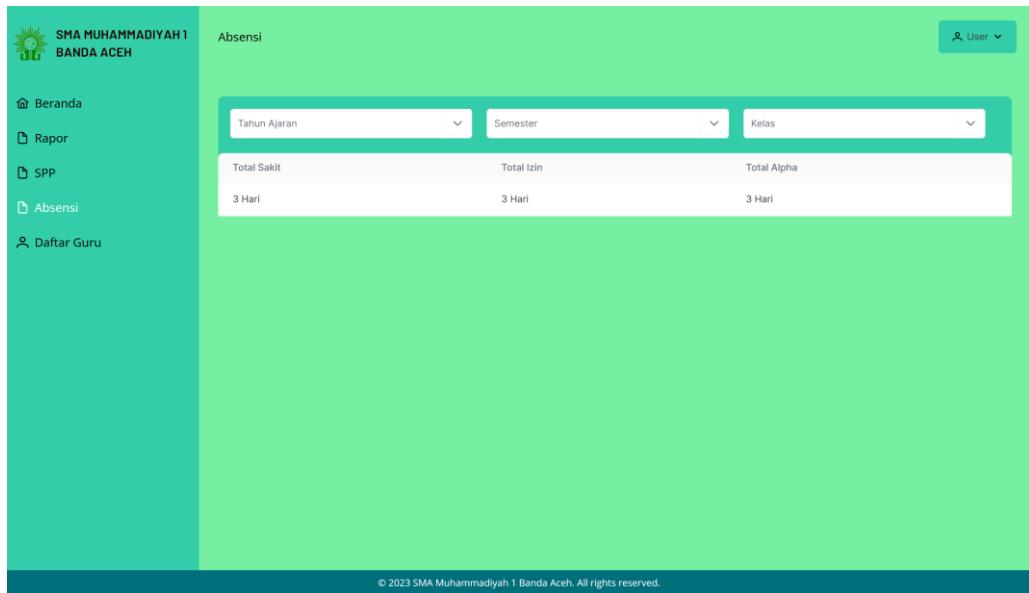


Figure 5.33: Attendance Page for Student

5.3.11 Teacher List Page

Daftar Guru	
	<input type="text" value="Q Keyword Search"/>
Nama ↗	NIP ↗
Suhartina, S.Pd., M.Pd.	196908311990112000
Juaairiah, S.Pd.	196212311985032030
Drs. A. Hamid	196006051987031000
Melly Hastuty, S.Pd.	197407272003122020
Mariana, S.Ag.	197602042005042020
« < 1 2 3 4 > » 5 ▾	

Figure 5.34: Teacher List Page for Student

5.3.12 Teacher Manage Academic Report Page

The screenshot shows a web application interface for managing academic records. The left sidebar has a teal background with the school logo and navigation links: Beranda, Rapor, Penilaian, Jadwal Pelajaran, Daftar Siswa, SPP, and Absensi. The main content area has a light green background and is titled 'Rapor'. It features a search bar with dropdowns for Tahun Ajaran, Semester, Kelas, and a Keyword Search field, along with a 'Reset Filters' button. Below the search bar is a table with columns: Nama, NIS/NISN, Peminatan, and Action. The table contains five rows of student data:

Nama	NIS/NISN	Peminatan	Action
Arisya Zanntara	9443/0062226931	MIPA	Lihat, Buat, Unduh
Habiburrahman	9444/0062226935	MIPA	Lihat, Buat, Unduh
Gladys Putri Ferica	9443/0067402572	MIPA	Lihat, Buat, Unduh
Nurhayati	9448/0054837389	MIPA	Lihat, Buat, Unduh
Muna Anisa	9447/0053181420	MIPA	Lihat, Buat, Unduh

At the bottom of the table is a navigation bar with links for page numbers (1-5) and arrows.

Figure 5.35: Manage Academic Record Page for Teacher

5.3.13 Teacher Manage Marks Page

This screenshot shows the 'Penilaian' (Assessment) page. The left sidebar is identical to Figure 5.35. The main content area has a light green background and is titled 'Penilaian'. It features a search bar with dropdowns for Tahun Ajaran, Semester, Kelas, and a Keyword Search field, along with a 'Reset Filters' button. Below the search bar is a table with columns: Nama, NIS/NISN, Peminatan, Aspek, and Action. The table contains five rows of student data, similar to Figure 5.35, but includes an additional 'Aspek' column with dropdown menus and 'Ubah' (Change) buttons.

Figure 5.36: Manage Marks Page for Teacher

5.3.14 Teacher Manage Timetable Page

The screenshot shows a 'Jadwal Pelajaran' (Class Timetable) page. On the left is a sidebar with the school logo and navigation links: Beranda, Rapor, Penilaian, Jadwal Pelajaran, Daftar Siswa, SPP, and Absensi. The main area is titled 'Jadwal Pelajaran' and shows a grid for five days (Monday to Friday). Each day has five time slots (1 to 5). Each slot contains a dropdown menu for selecting subjects. For example, Monday's slots show Kimia, Matematika, Ekonomi, Bahasa Indonesia, and Bahasa Inggris.

Figure 5.37: Manage Timetable Page for Teacher

5.3.15 Teacher Student List Page

The screenshot shows a 'Daftar Siswa' (Student List) page. The sidebar includes links for Beranda, Rapor, Penilaian, Jadwal Pelajaran, Daftar Siswa, SPP, and Absensi. The main content area is titled 'Daftar Siswa' and features a table of student information. At the top of the table are filters for Tahun Ajaran, Semester, Kelas, and a Keyword Search input field with a 'Reset Filters' button. The table columns are Nama, NIS/NISN, Peminatan, and Action. Each row contains a student's name, their NIS/NISN number, their academic level (MIPA), and a green 'Lihat' (View) button. Below the table is a navigation bar with page numbers from 1 to 5.

Figure 5.38: Student List Page for Teacher

5.3.16 Teacher Manage Tuition Page

 SMA MUHAMMADIYAH 1 BANDA ACEH		Daftar Siswa			
		Tahun Ajaran	Semester	Kelas	<input type="text" value="Q. Keyword Search"/> Reset Filters
		Nama ↑	NIS/NISN ↑↓	Peminatan ↑↓	Action
		Arisya Zanntara	9443/0062226931	MIPA	<button>Lihat SPP</button>
		Habiburrahman	9444/0062226935	MIPA	<button>Lihat SPP</button>
		Gladys Putri Ferica	9443/0067402572	MIPA	<button>Lihat SPP</button>
		Nurhayati	9448/0054837389	MIPA	<button>Lihat SPP</button>
		Muna Anisa	9447/0053181420	MIPA	<button>Lihat SPP</button>
<< < 1 2 3 4 5 > >> 5 ↴					

Figure 5.39: Manage Tuition Page for Teacher

5.3.17 Teacher Manage Attendance Page

 SMA MUHAMMADIYAH 1 BANDA ACEH		Absensi			
		Tanggal	Total Jam	Kelas	<input type="text" value="Q. Keyword Search"/>
		Nama ↑↓	NIS/NISN ↑↓	Sakit/Izin/Alpha/Hadir	Keterangan
		Arisya Zanntara	9443/0062226931	Sebab	<input type="text" value="Keterangan"/>
		Habiburrahman	9444/0062226935	Sebab	<input type="text" value="Keterangan"/>
		Gladys Putri Ferica	9443/0067402572	Sebab	<input type="text" value="Keterangan"/>
		Nurhayati	9448/0054837389	Sebab	<input type="text" value="Keterangan"/>
		Muna Anisa	9447/0053181420	Sebab	<input type="text" value="Keterangan"/>
<< < 1 2 3 4 5 > >> 5 ↴					

Figure 5.40: Manage Attendance Page for Teacher

5.4 Testing

There is a couple testing method that will be done to the system. They are black box testing, white box testing and user testing. Testing is an important process of the system development since it can identify errors and bugs. This establish the system to be as close as possible to the requirement.

5.4.1 Black box Testing

Black box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. In black box testing, the test done is more focused towards the input and output of the system without knowing how the application receive the input and produce the output. Black box testing also done from the perspective of the user to ensure that the functions works and follow the stated requirements

5.4.1.1 System Flow

The system flow of an application refers to the sequence of events or operations that a system undergoes when performing specific tasks. In black box testing, testers observe and evaluate whether the system behaves as expected when performing a particular sequence of actions.

5.4.1.2 Input Output Verification

Input Output Verification in black box testing focuses on testing the system's response to given inputs and whether the output is as expected. Testers provide various types of valid and invalid inputs and verify if the system produces the correct and expected output. This process helps in identifying any discrepancies in the system's functionality.

5.4.1.3 Error Message

Error message testing in black box testing focuses on the system's response to erroneous conditions or inputs. This involves deliberately inputting incorrect or invalid data and verifying the system's error messages. The objective is to ensure that the system correctly detects errors and handles them

5.4.2 White box Testing

In white box testing, the tester that conducts the testing has information on the system's internal structure, design, and implementation. The goal of white box testing is to verify a system's internal operations, focusing on internal testing of code structures, code flows, and internal system logic. Some of example of this type of

testing are provided below in the User Testing subchapter.

5.4.3 User Testing

User Testing will be performed by real users to evaluate the system. User Testing provides direct input on how real users use a system, thereby helping to understand where the system stands from the user's perspective. The user testing results show average user satisfaction with the system. The user testing has been done to the system, and the result of the test can be seen in the table 5.1.

Table 5.1 User Testing Result

User Acceptance Testing Result	
Questions	Average Satisfaction
1. Does the system effectively perform all the functions requirements?	8.5/10
2. Is the user interface of the system intuitive and easy to navigate?	9/10
3. Does the system handle errors effectively and provide meaningful error messages?	7.5/10
4. Is the system's performance satisfactory under various loads?	8/10
5. Are the system's security measures effective and reliable?	8/10

Table 5.2 Admin Add Student Account

Tester Name	RAHMAWATI
Date	23 June 2023
Module	Account Module
Instruction	
<ol style="list-style-type: none"> 1. Login with Admin Account 2. Click “Tambah Akun Siswa” 3. Enter the required input 4. Click “Tambah” 5. Logout by clicking the top right menu and click “Keluar” button 	
Expected Result	
<ol style="list-style-type: none"> 1. Should be able to logged in as Admin 2. Should be able to view the “Tambah Akun Siswa” Page 3. Should be able to input all the required fields 4. Should be able to create the student account with the provided input 5. Should be able to logged out from the system 	
Actual Result	
<ol style="list-style-type: none"> 1. Successfully logged in as Admin 2. Successfully view the “Tambah Akun Siswa” Page 3. Successfully input all the required fields 4. Successfully create the student account with the provided input 5. Successfully logged out from the system 	

Table 5.3 Admin Add Teacher Account

Tester Name	RAHMAWATI
Date	23 June 2023
Module	Account Module
Instruction	
<ol style="list-style-type: none"> 1. Login with Admin Account 2. Click “Tambah Akun Guru” 3. Enter the required input 4. Click “Tambah” 5. Logout by clicking the top right menu and click “Keluar” button 	
Expected Result	
<ol style="list-style-type: none"> 1. Should be able to logged in as Admin 2. Should be able to view the “Tambah Akun Guru” Page 3. Should be able to input all the required fields 4. Should be able to create the student account with the provided input 5. Should be able to logged out from the system 	
Actual Result	
<ol style="list-style-type: none"> 1. Successfully logged in as Admin 2. Successfully view the “Tambah Akun Guru” Page 3. Successfully input all the required fields 4. Successfully create the student account with the provided input 5. Successfully logged out from the system 	

Table 5.4 Teacher Marking Student

Tester Name	YULIANA
Date	23 June 2023
Module	Report Module
Instruction	
<ol style="list-style-type: none"> 1. Login with Teacher Account 2. Click “Penilaian” 3. Select the filter to view the desired student 4. Click the “Nilai” button 5. Input the required field 6. Click the “Save” button 7. Logout by clicking the top right menu and click “Keluar” button 	
Expected Result	
<ol style="list-style-type: none"> 1. Should be able to logged in as Teacher 2. Should be able to view the “Penilaian” Page 3. Should be able to view the desired student 4. Should be able to view the input mark page 5. Should be able to save the mark for the student 6. Should be able to logged out from the system 	
Actual Result	
<ol style="list-style-type: none"> 1. Successfully logged in as Teacher 2. Successfully view the “Penilaian” Page 3. Successfully view the desired student 4. Successfully view the input mark page 5. Successfully save the mark for the student 6. Successfully logged out from the system 	

Table 5.5 Student View Academic Report

Tester Name	ARISYA ZANNITARA
Date	23 June 2023
Module	Report Module
Instruction	
<ol style="list-style-type: none"> 1. Login with Student Account 	

<ol style="list-style-type: none"> 2. Click “Rapor” 3. Select the filter to preview the desired Academic Report 4. Click “Unduh” 5. Logout by clicking the top right menu and click “Keluar” button
Expected Result
<ol style="list-style-type: none"> 1. Should be able to logged in as Student 2. Should be able to view the “Rapor” Page 3. Should be able to view the desired Academic Report 4. Should be able to download the Academic Report 5. Should be able to logged out from the system
Actual Result
<ol style="list-style-type: none"> 1. Successfully logged in as Student 2. Successfully view the “Rapor” Page 3. Successfully view the desired Academic Report 4. Successfully download the Academic Report 5. Successfully logged out from the system

5.5 Chapter Summary

This chapter conclude the implementation and testing of the Student Academic Management System. All of the codes and interfaces main function are discussed and showed thoroughly. The testing done which are Black box testing, White box testing and User testing was also provided and explained.

CHAPTER 6

CONCLUSION

6.1 Introduction

The final chapter for this report will consist of recap from the discovery and findings from previous chapters. Overall, the requirement from the stakeholder for developing the system has been analyzed thoroughly and correct actions has been taken during the development phase. This chapter also provides the achievement during PSM 2 and the suggestions for improvement of the system.

6.2 Achievement of Project Objectives

First and foremost, all of the problems that the stakeholders faced have been identified. After identifying carefully, the problems, the proposed solution appears which is to make a web app for managing the students' academic data. With the proposed solution in mind, gathering all of the requirements from stakeholders then began. Before listing the requirements, conducting research on similar existing systems is also important. From gathering data from similar systems and analyzing the current system the methodology for developing the system is chosen. Along with the methodology came the required technology, software, and hardware for developing the system. Then, the stakeholders provide requirements for the proposed system which include mockup interfaces and STD that can be used as a reference in PSM 2.

In PSM 2, the system has been successfully developed. The development phase consisted of various stages such as system design, coding, and testing. Each stage was meticulously carried out to ensure that the system meets the requirements provided by the stakeholders. The system also implements all the required function stated in PSM 1. The system has been implemented in a test environment such as User Acceptance Testing to ensure its proper functioning and fixing bugs as much as possible.

6.3 Suggestions for Future Improvement

While the project has been successfully developed, there are still some improvement that can be made for it. Below are some of the things that needed to be improved for the system:

- a) Integrated payment system for tuition payment, so that the payment system is more efficient and secure
- b) Integration with third party LMS (Learning Management System)
- c) Integration with email services such as notification.

REFERENCES

- My.utm.my.* (n.d.). Retrieved May 16, 2022, from <https://my.utm.my/home>
- Applikasi Rapor online - solusi.* Sekawan Media. (2022, May 16). Retrieved May 16, 2022, from <https://www.sekawanmedia.co.id/solusi/aplikasi-rapor-online/>
- Miller, J., Kostogriz, A., & Gearon, M. (Eds.). (2009). Culturally and linguistically diverse classrooms: New dilemmas for teachers. Bristol, UK: Multilingual Matters.
- Dinustek. (n.d.). *Solusi Rapor Sekolah online.* evaluation. Retrieved May 16, 2022, from <https://evaluation.id/>

Appendix A Software Requirement Specification

Software Requirements Specification

Student Academic Management System

Version 1.0

23 June 2022

School of Computing, Faculty of Engineering

Revision Page

6.3.1 Overview

This is the first draft of the Software Requirements Specification (SRS) for Student Academic Management System (SAMS).

6.3.2 Target Audience

The target audience are students, teachers, and admin of SMAS Muhammadiyah 1 Banda Aceh

6.3.3 Project Team Members

Muhammad Darlen Sava

6.3.4 Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Muhammad Darlen Sava	First Draft	23 June 2022

Note:

This Software Requirements Specification (SRS) template is based on IEEE Std 830-1998, organized by modules according to system features (Appendix A.5 of the IEEE Std, 830-1998, Section 5) and customized to meet the need of SCSJ2203 course at Faculty of Computing, UTM. Compiled and checked by Shahida Sulaiman, PhD on 20 March 2016. Examples of models are from Satzinger (2011).

Table of Contents

1	Introduction	6
1.1	Purpose	6
1.2	Scope	6
1.3	Definitions, Acronyms and Abbreviations	6
1.4	References	6
1.5	Overview	7
2	Overall Description	8
2.1	Product Perspective	8
2.1.1	System Interfaces	8
2.1.2	User Interfaces	8
2.1.3	Hardware Interfaces	9
2.1.4	Software Interfaces	9
2.1.5	Communication Interfaces	9
2.1.6	Memory	9
2.1.7	Operations	10
2.1.8	Site Adaptations Requirements	10
2.2	Product Functions	10
2.3	User Characteristic	12
2.4	Constraints	12

2.5	Assumption and Dependencies	13	
2.6	Apportioning of Requirements	13	
3	Specific Requirements	14	
3.1	External Interface Requirements	14	
3.1.1	User Interfaces	15	
3.1.2	Hardware Interfaces	16	
3.1.3	Software Interfaces	16	
3.1.4	Communication Interfaces	16	
3.2	System Features		
3.2.1	Module Authentication	17	
	3.2.1.1	UC001: Use Case Sign In	17
3.2.2	Module Report		20
	3.2.2.1	UC002: Use Case Manage Academic Report	20
	3.2.2.2	UC003: Use Case View Academic Report	24
	3.2.2.3	UC004: Use Case Download Academic Report	26
3.2.3	Module Attendance		28
	3.2.3.1	UC005: Use Case Manage Attendance Record	29

	3.2.3.2	UC006: Use Case View Attendance Record	32
3.2.4		Module Schedule	35
	3.2.4.1	UC007: Use Case Manage Timetable	35
	3.2.4.2	UC008: Use Case View Timetable	38
	3.2.4.3	UC009: Use Case Download Timetable	40
3.2.5		Module Account	42
	3.2.5.1	UC010: Use Case Manage Tuition Record	42
	3.2.5.2	UC011: Use Case View Tuition Record	45
	3.2.5.3	UC012: Use Case Download Tuition Record	47
	3.2.5.4	UC013: Use Case Make a Payment	49
	3.2.5.5	UC014: Use Case Manage Profile	51
	3.2.5.6	UC015: Use Case Manage User	54
	3.2.5.7	UC016: Use Case View Student Info	58
	3.2.5.8	UC017: Use Case View Teacher Info	60

3.3	Performance Requirements	111
3.4	Design Constraints	111
3.5	Software System Attributes	111
3.6	Other Requirements	111

Appendices (if any)

1. Introduction

1.1 Purpose

This SRS describes the requirement for the Student Academic Management System which is a system dedicated to SMAS Muhammadiyah 1 Banda Aceh. This SRS is required to determine the requirements of the stakeholder, it also acts as a reference point for the development of the project.

1.2 Scope

The current system of SMAS Muhammadiyah is still involving manual task execution with a decentralized database. So this project only focused on the conversion of manually managing student data to a website that stores all of the student's data, which can be accessed by teachers, students, and their parents. As such, SMAS Muhammadiyah will only provide the student's data throughout the development of the project. The users in this system will be the teachers, students/parents, and admins of SMAS Muhammadiyah. The goal of this project is to create an effective solution to enable the institution to convert its manual student management system into a computerized system.

1.3 Definitions, Acronyms, and Abbreviation

- | | |
|---------|---------------------------------------|
| 1. SRS | - Software Requirements Specification |
| 2. SAMS | - Student Academic Management System |

1.4 References

- My.utm.my.* (n.d.). Retrieved May 16, 2022, from <https://my.utm.my/home>
- Applikasi Rapor online - solusi.* Sekawan Media. (2022, May 16). Retrieved May 16, 2022, from <https://www.sekawanmedia.co.id/solusi/aplikasi-rapor-online/>
- Miller, J., Kostogriz, A., & Gearon, M. (Eds.). (2009). Culturally and linguistically diverse classrooms: New dilemmas for teachers. Bristol, UK: Multilingual Matters.

Dinustek. (n.d.). *Solusi Rapor Sekolah online*. evaluation. Retrieved May 16, 2022, from <https://evaluation.id/>

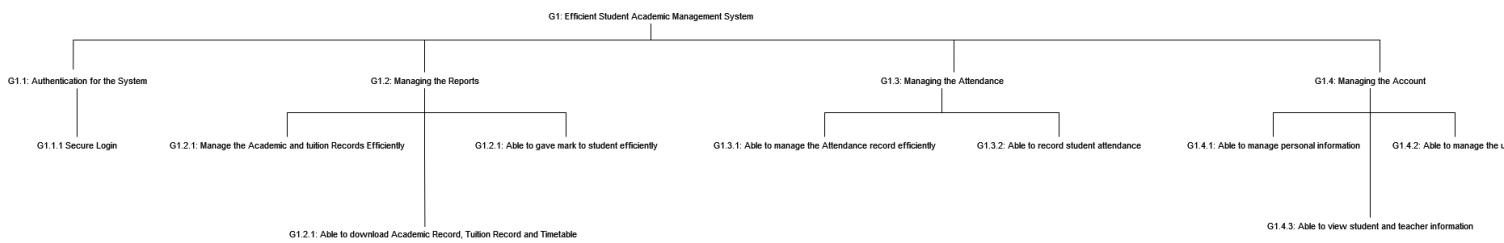
1.5 Overview

This SRS is divided into three chapters. The first chapter is the introduction which consists of the purpose and scope of the project. It also includes all the acronyms/abbreviation descriptions and references that exist in this SRS and a short overview of the whole project. For the second chapter, The SRS focused on the overall description of the project. This chapter delves into the general requirement for the project/product as it only provides some background requirements for the overall product. In the final chapter, the specific requirements for the project are stated. It is the most important part of this SRS since it contains the domain model, state machine diagram, and modules for each of the use cases.

2. Overall Description

Figure 2.1: Use Case Diagram of Student Academic Management System

2.1 Product Perspective



Attached is the goal model for the system. It described the perspective from the stakeholders on how they want the system to be

2.1.1 System Interfaces

SAMS will be developed as a web-based. The development will be implemented using Laravel and Tailwind. As for the database, PostgreSQL is used along with PgAdmin as its interface for managing the database

2.1.2 User Interfaces

SAMS is designed to be as intuitive as possible so that it will be user-friendly. The system also uses the Indonesian language because the users are Indonesian so it can make navigating the system easier

2.1.2.1 Login Interface

All of the users will have to go through this interface to enter the system. After the user logs in using valid credentials they will be redirected to each respective main interface

2.1.2.2 Main Interface for Student

This interface will be displayed after the student successfully logs in to the system. Here they can access to multiple function such as view academic reports, timetables and more.

2.1.2.3 Main Interface for Teacher

This interface will be displayed after the teacher successfully logs in to the system. Here they can access multiple functions such as creating academic reports, editing timetables, and more.

2.1.2.4 Main Interface for Admin

This interface will be displayed after the teacher successfully logs in to the system. Here they can access multiple functions such as creating a student account, creating a teacher account, creating an admin account, and more.

2.1.3 Hardware Interfaces

The system will work on any personal computer or laptop that can run the web browser required to access the system.

2.1.4 Software Interfaces

The software that is required to access the system can be any modern web browser. Options such as google chrome, Mozilla Firefox, and Opera are recommended

2.1.5 Communication Interfaces

Reliable TCP/IP protocols such as HTTP and FTP are used for communication over the internet to maintain maximum compatibility and reliability.

2.1.6 Memory

RAM: Windows or Linux with 512MB of RAM / 2GB of RAM for the 64-bit version or Mac computer with 512 MB of RAM.

2.1.7 Operations

The user's account can only be created by the admin thus making the system isolated from the public. After creating the account the admin will give the login information to the users. The same goes for the link, it will be a special link made for SMAS Muhammadiyah 1. The only way to log in is through this

2.1.8 Site Adaptation Requirements

Since the system is a web-based system there are no particular adaptation requirements needed. It can run on handful number of platforms

2.2 Product Functions

- 1) UC001 – Sign In: This use case is used for every user in the system to enter the system
- 2) UC002 – Manage Academic Report: This use case is used for teacher to manage the academic report and marking of students
- 3) UC003 - View Academic Report: This use case is used for student to view their academic report

- 4) UC004 – Download Academic Report: This use case is used for student to download their academic report
- 5) UC005 – Manage Attendance Record: This use case is used for teacher to manage the attendance record including to take attendance during class
- 6) UC006 – View Attendance Record: This use case is used for student to views the attendance record
- 7) UC007 – Manage Timetable: This use case is used for student to manage the attendance record including to take attendance during class
- 8) UC008 – View Timetable: This use case is used for student to views the timetable
- 9) UC009 – Download Timetable: This use case is used for student to download the timetable
- 10) UC010 – Manage Tuition Record: This use case is used for teacher to manage the tuition record including approving or rejecting tuition payment
- 11) UC011 – View Tuition Record: This use case is used for student to view the tuition record
- 12) UC012 – Download Tuition Record: This use case is used for student to download the tuition record
- 13) UC013 – Make a Payment: This use case is used for student to download the tuition record
- 14) UC014 – Manage Profile: This use case is used for student and teacher to view and edit their profile
- 15) UC015 – Manage User: This use case is used for admin to manage user account both student and teacher
- 16) UC016 – View Student Info: This use case is used for teacher to view list of students and their detailed information

- 17) UC017 – View Teacher Info: This use case is used for student to view list of students and their detailed information

2.3 User Characteristics

Actors involved in the system are Student, Teacher and Admin. For student they will get access to most of their data by viewing or downloading it. As for teacher they mostly have access to managing use case means that they gain access to a lot of CRUD operations. Lastly, the admin has a role to manage all of the user account since creating users account is done by the admin. So, they have quite an important role in the system.

2.4 Constraints

Different from functional requirements, non-functional requirements act as the constraints of the system. Non-functional requirements more focused toward the system itself on what it can do to improve the functionality of the system. The system must achieve the requirements that are listed below.

- 1) Availability – The user should be able to access the system at any moment thus, the system must be available all day every day.

- 2) Security – The system should only allow authorized user from SMAS Muhammadiyah 1 to access the system. The system must provide protection to all of the data exist in the system.
- 3) Usability – The user should be able to explore the system with ease after inspecting the system a little bit. The system must be simple to use and comprehend.
- 4) Portability – The system should work with any web browser. The user shall be able to access the system from their preferred web browser

2.5 Assumption and Dependencies

The system can only be accessed by devices that can run a web browser. It is also recommended to access using a stable internet connection. In some rare cases if the system can not operate normally assumption can be made that the database might be the problem.

2.6 Apportioning of Requirements

If the requirements could not be made because of the project delayed it can be moved on to the next iteration

3. Specific Requirements

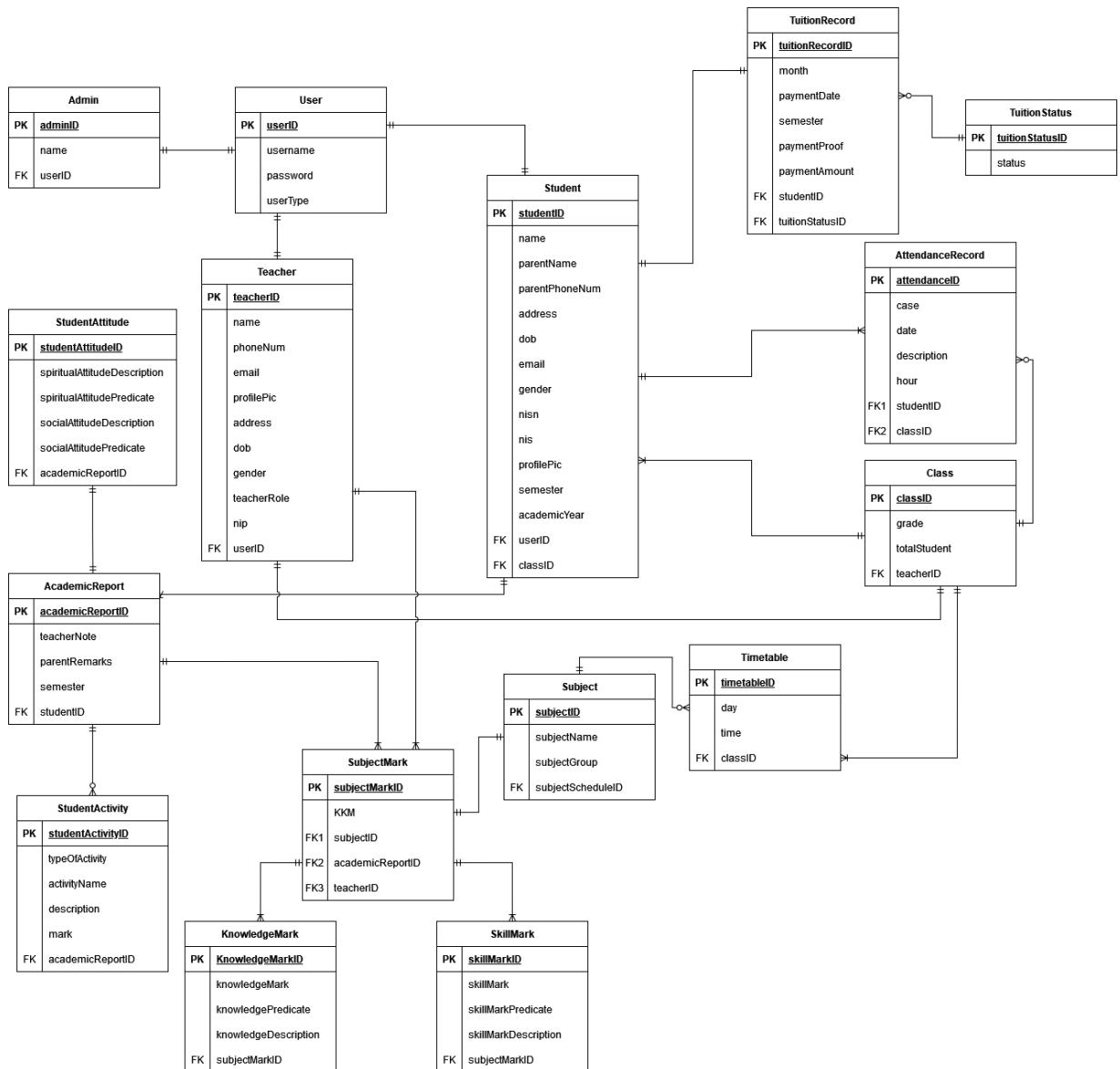


Figure 3.1: Domain Model of Student Academic Management System

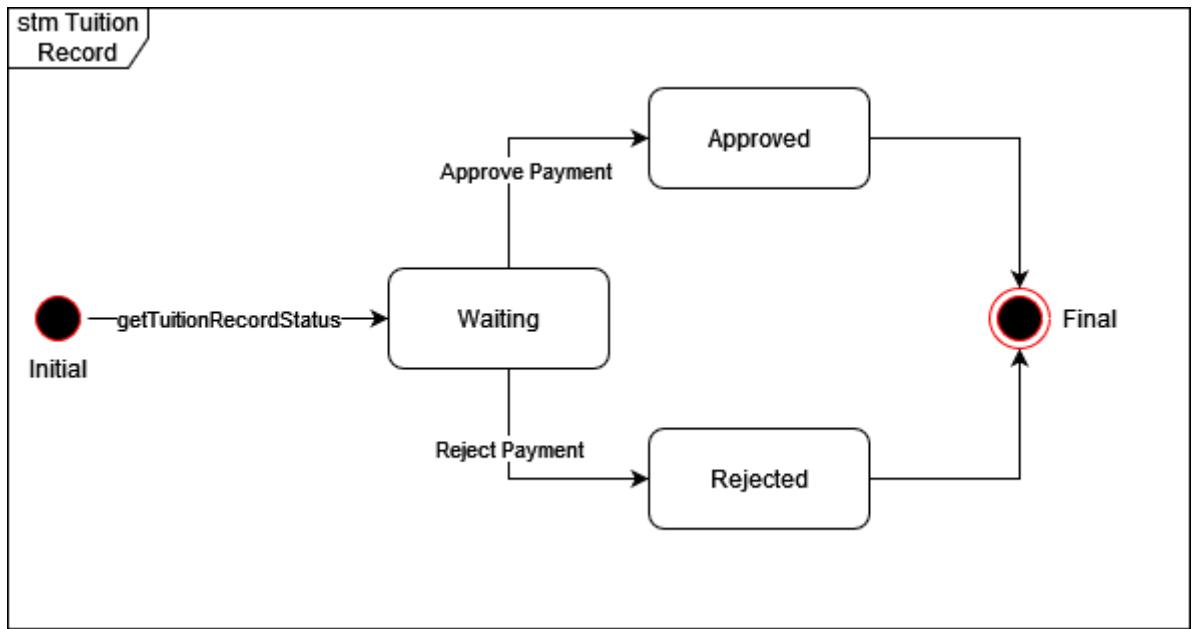


Figure 3.2: State Machine Diagram of Manage Tuition Record

3.1 External Interface Requirements

3.1.2 User Interfaces

SAMS is designed to be as intuitive as possible so that it will be user-friendly. The system also uses the Indonesian language because the users are Indonesian so it can make navigating the system easier

3.1.2.1 Login Interface

All of the users will have to go through this interface to enter the system. After the user logs in using valid credentials they will be redirected to each respective main interface

3.1.2.2 Main Interface for Student

This interface will be displayed after the student successfully logs in to the system. Here they can access to multiple function such as view academic reports, timetables and more.

3.1.2.3 Main Interface for Teacher

This interface will be displayed after the teacher successfully logs in to the system. Here they can access multiple functions such as creating academic reports, editing timetables, and more.

3.1.2.4 Main Interface for Admin

This interface will be displayed after the teacher successfully logs in to the system. Here they can access multiple functions such as creating a student account, creating a teacher account, creating an admin account, and more.

3.1.2 Hardware Interfaces

The system will work on any personal computer or laptop that can run the web browser required to access the system.

3.1.3 Software Interfaces

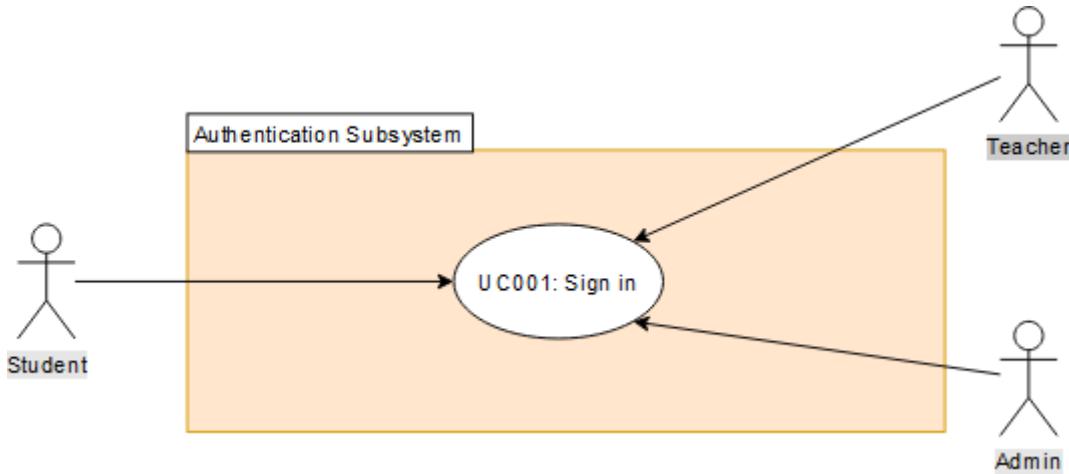
The software that is required to access the system can be any modern web browser. Options such as google chrome, Mozilla Firefox, and Opera are recommended

3.1.4 Communication Interfaces

Reliable TCP/IP protocols such as HTTP and FTP are used for communication over the internet to maintain maximum compatibility and reliability.

3.2 System Features

3.2.1 Module Authentication



3.2.1.1 UC001: Use Case Sign In

Table 3.1: Use Case Description for Sign In

Use Case ID:	UC-001
Use Case Name:	Sign In
Actors:	Admin/Teacher/Student
Description:	This use case is used for every user in the system to enter the system
Pre-conditions:	1. Existing User
Normal Flow:	<ol style="list-style-type: none">1. Go to the login page2. Fill in the username and password3. Click “Login” button
Alternative Flow:	-
Exception:	E.1 Error during validation 1. an error message will be displayed by the system 2. Users try to login again using the correct credentials or request a new account from an admin
Post-condition s:	The user successfully enters the system.
Related Requirement:	-

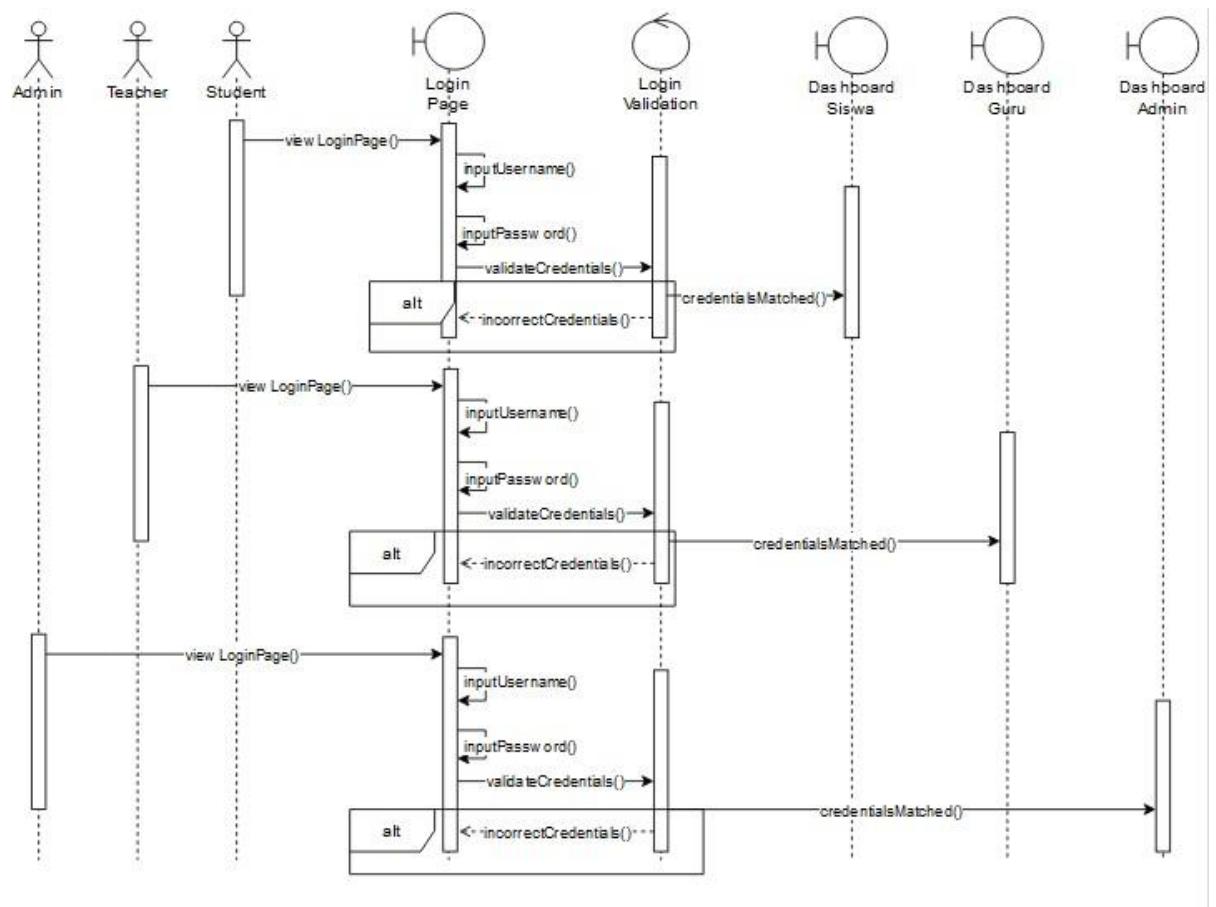


Figure 3.3: System Sequence Diagram of Sign In

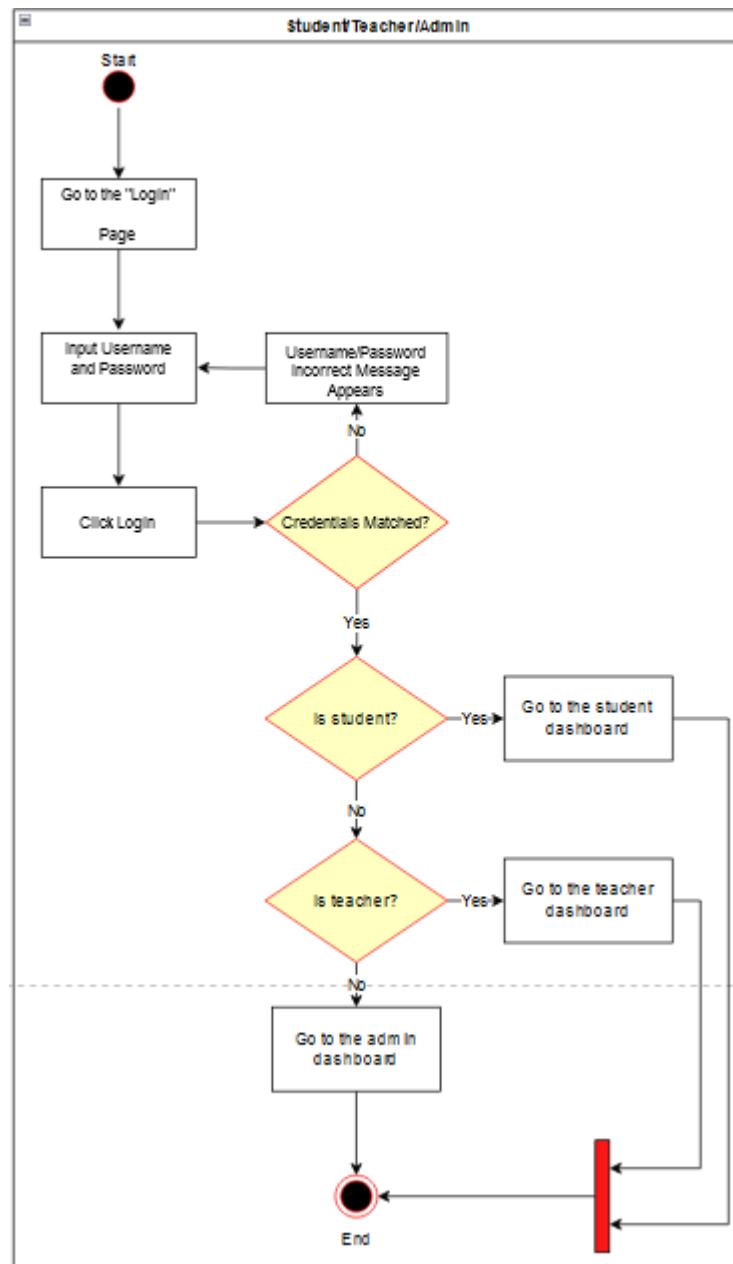
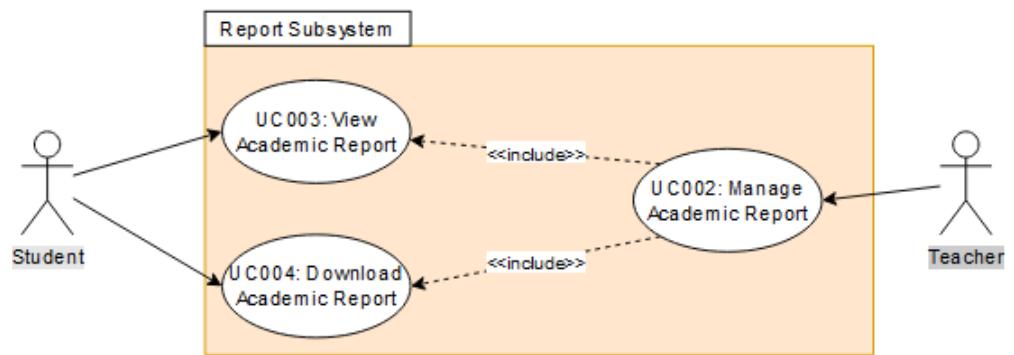


Figure 3.4: Activity Diagram of Sign In

3.2.2 Module Report



3.2.2.1

UC002: Use Case Manage Academic Record

Table 3.2: Use Case Description for Manage Academic Report

Use Case ID:	UC-002
Use Case Name:	Manage Academic Report
Actors:	Teacher
Description:	This use case is used for teacher to manage the academic report and marking of students
Pre-conditions:	2. Logged in to the system
Normal Flow:	<p>12. Go to the “Rapor” page</p> <p>13. If the teacher clicks “Penilaian” performs AF1</p> <p>14. If the teacher clicks “Edit Penilaian” performs AF2</p> <p>15. Select academic year</p> <p>16. Select semester</p> <p>17. Select class</p> <p>18. Select student</p> <p>19. The system displays the academic report</p> <p>20. If the teacher wants to download the academic report performs AF3</p> <p>21. If the teacher is a homeroom teacher and clicks “Buat Rapor” page performs AF4</p> <p>22. If the teacher wants to delete the academic report performs AF5</p>
Alternative Flow:	<p>6. Marking</p> <p>1.1 System redirects teacher to “Penilaian” page</p> <p>1.2 System displays the marking form</p> <p>1.3 Fill in the marking form</p> <p>1.4 System saves student mark</p> <p>7. Edit Marking</p> <p>7.1 System redirects teacher to “Edit Penilaian” page</p> <p>7.2 System displays the student that the teacher already marked</p> <p>7.3 Select student</p> <p>7.4 System displays the marking form</p> <p>7.5 Edit the mark</p> <p>8. Download Academic Report</p> <p>3.1 Click the “Unduh” button to download</p> <p>9. Create Academic Report</p> <p>9.1 System redirects teacher to “Buat Rapor” page</p> <p>9.2 Select Student</p> <p>9.3 Teacher fills in additional information</p> <p>9.4 Teacher clicks “Buat Rapor” button</p> <p>10. Delete Academic Report</p> <p>5.1 Click “Trash” icon to delete the academic report</p>
Exception:	<p>E.3 Required field(s) is/are empty in marking form</p> <ol style="list-style-type: none"> an error message will be displayed by the system Performs NF2/AF1 <p>E.4 Required field(s) is/are empty when creating Academic Report</p> <ol style="list-style-type: none"> an error message will be displayed by the system

	2. Performs NF9/AF4
Post-condition s:	<ul style="list-style-type: none"> 6. Teacher successfully views the academic report 7. Teacher successfully downloads the academic report 8. Teacher successfully marks the student 9. Teacher successfully edits the marks of the student 10. Teacher successfully creates the academic report
Related Requirement:	-

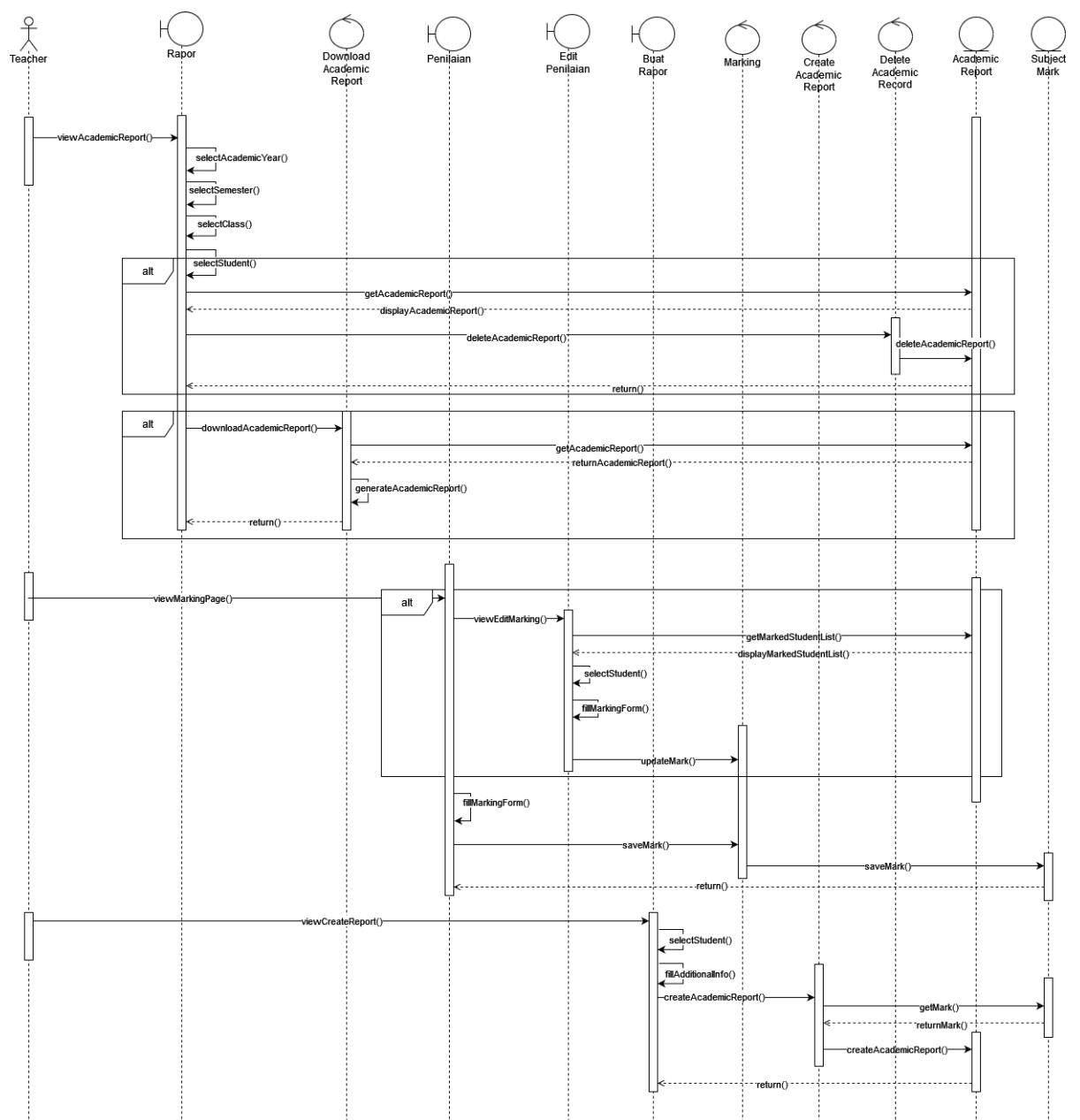


Figure 3.5: System Sequence Diagram of Manage Academic Report

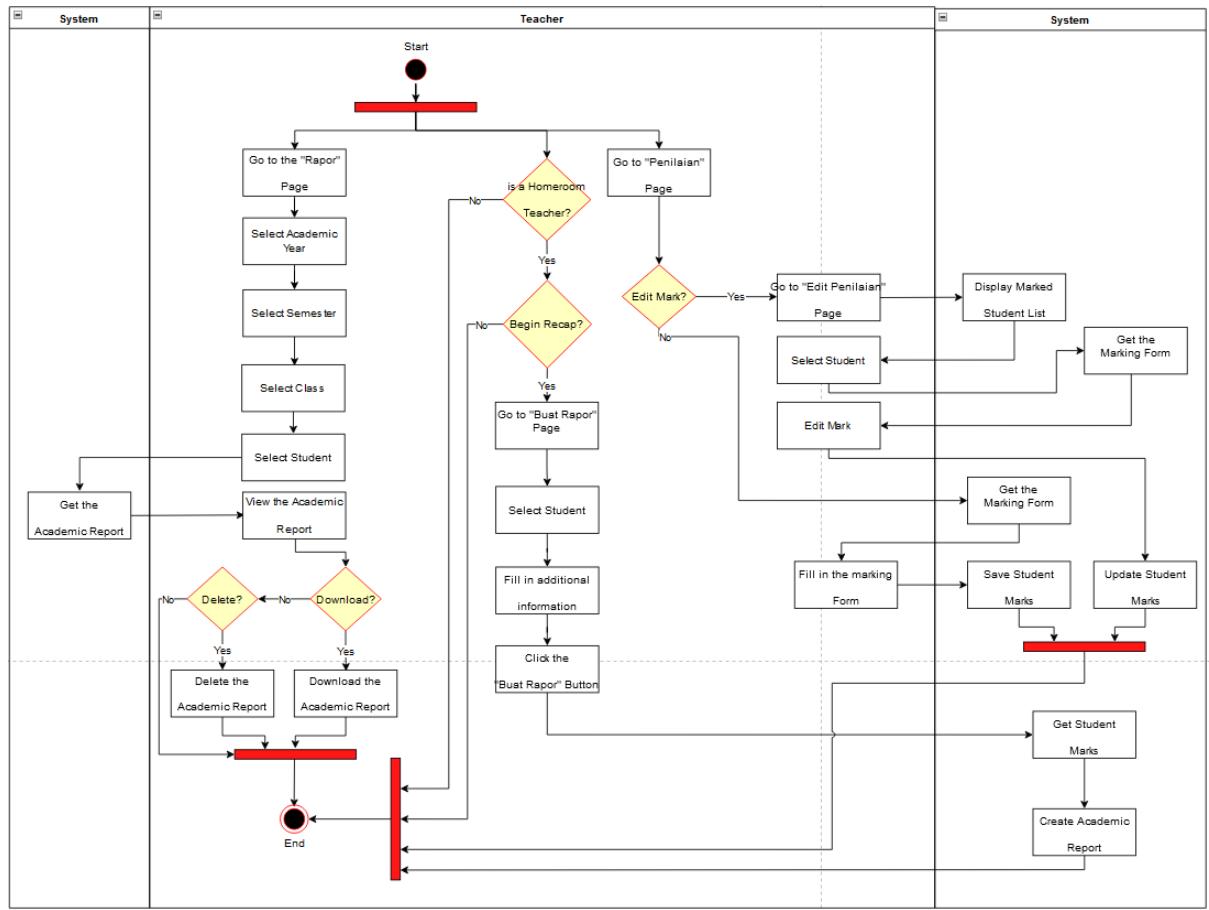


Figure 3.6: System Sequence Diagram of Manage Academic Report

3.2.2.2

UC003: Use Case View Academic Report**Table 3.3: Use Case Description for Sign In**

Use Case ID:	UC-003
Use Case Name:	View Academic Report
Actors:	Student
Description:	This use case is used for student to view their academic report
Pre-conditions:	<ul style="list-style-type: none"> 1. Logged in to the system
Normal Flow:	<ul style="list-style-type: none"> 1. Go to the “Rapor” page 2. Select academic year 3. Select semester 4. Select class 5. The system displays the academic report
Alternative Flow:	-
Exception:	-
Post-condition s:	<ul style="list-style-type: none"> 1. Student successfully views the academic report
Related Requirement:	-

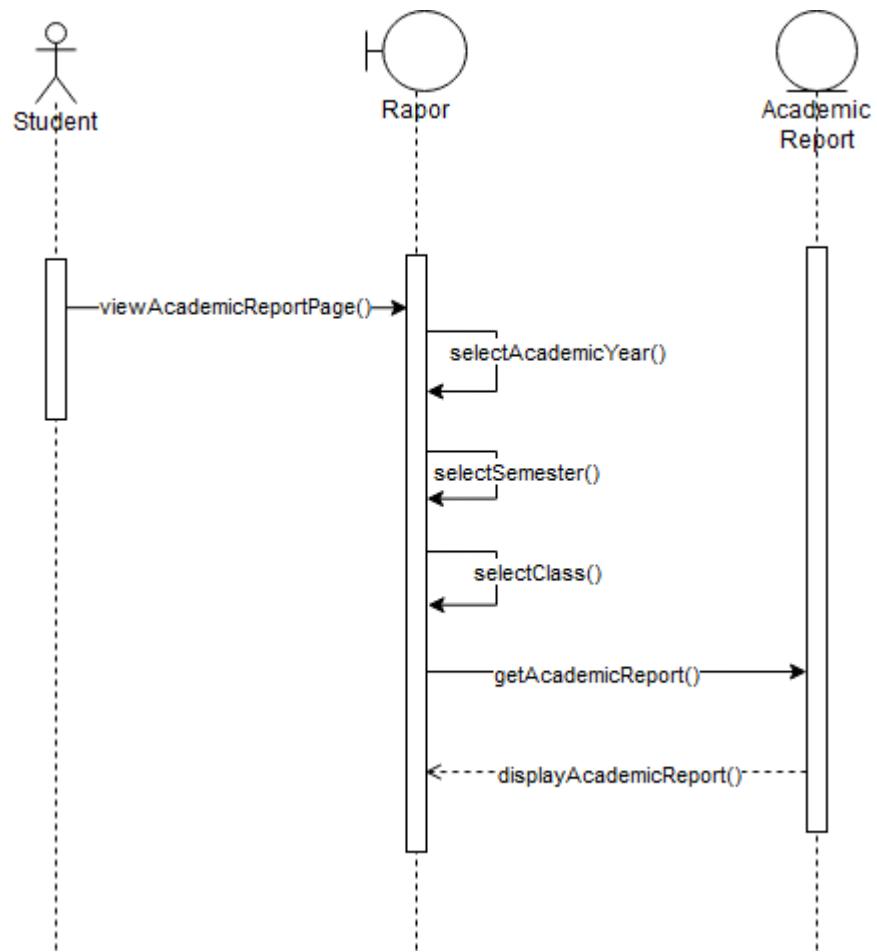


Figure 3.7: System Sequence Diagram of View Academic Report

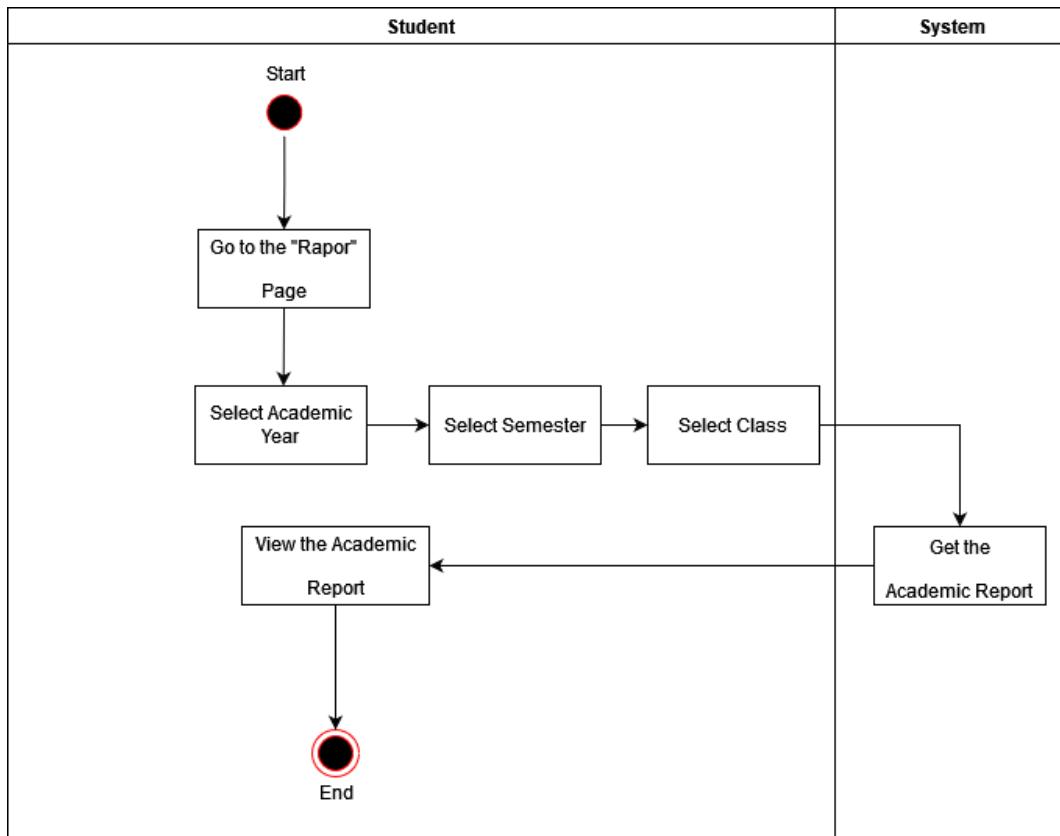


Figure 3.8: Activity Diagram of View Academic Report

3.2.2.2 UC004: Use Case Download Academic Report

Table 3.4: Use Case Description for Download Academic Report

Use Case ID:	UC-004
Use Case Name:	Download Academic Report
Actors:	Student
Description:	This use case is used for student to download their academic report
Pre-conditions:	1. Logged in to the system
Normal Flow:	<ol style="list-style-type: none"> 1. Go to the “Rapor” page 2. Select academic year 3. Select semester 4. Select class 5. The system displays the academic report 6. Click the “Unduh” Button to download
Alternative Flow:	-
Exception:	-
Post-condition s:	1. Student successfully downloads the academic report

Related Requirement:	-
----------------------	---

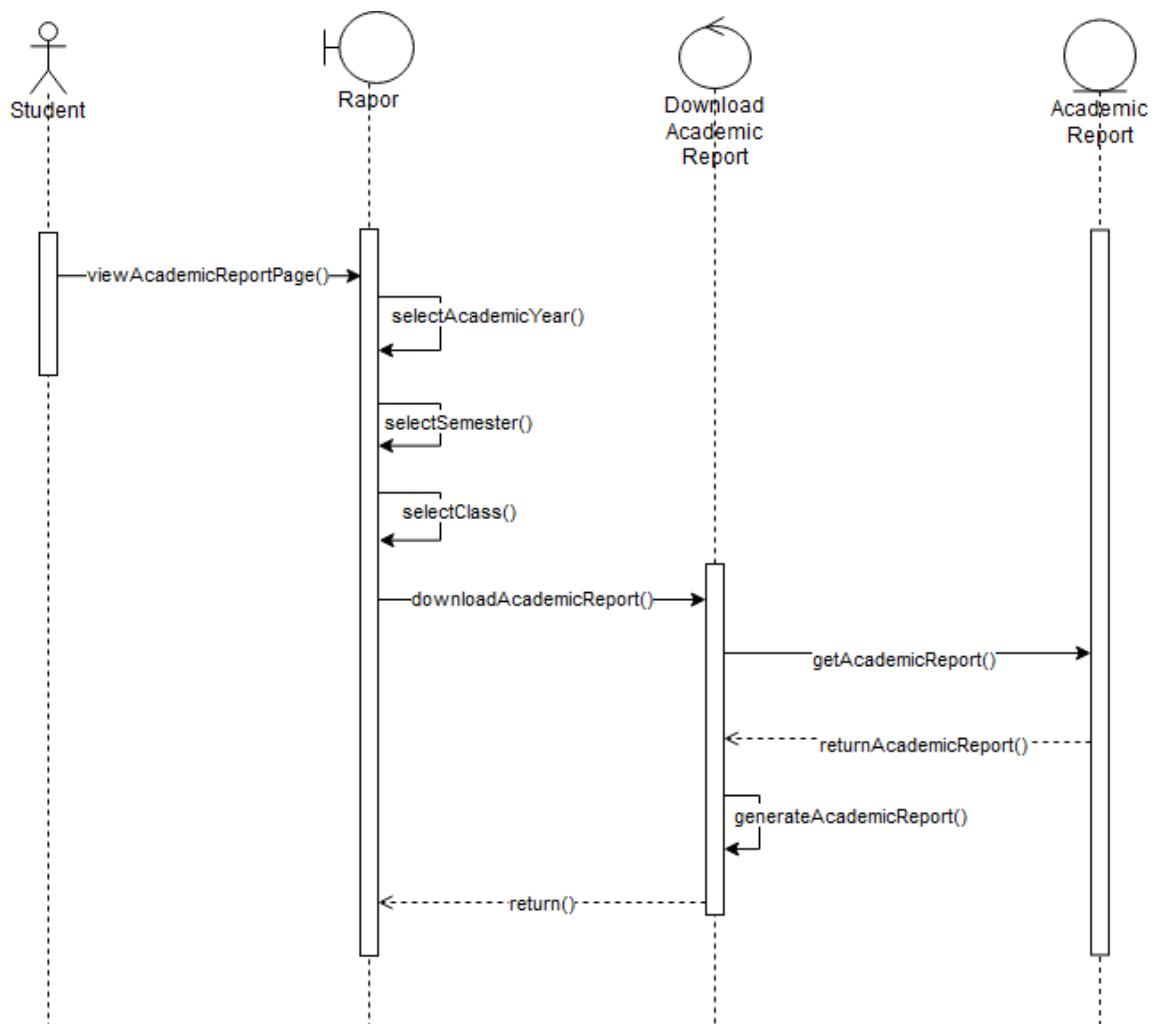


Figure 3.9: System Sequence Diagram of Download Academic Report

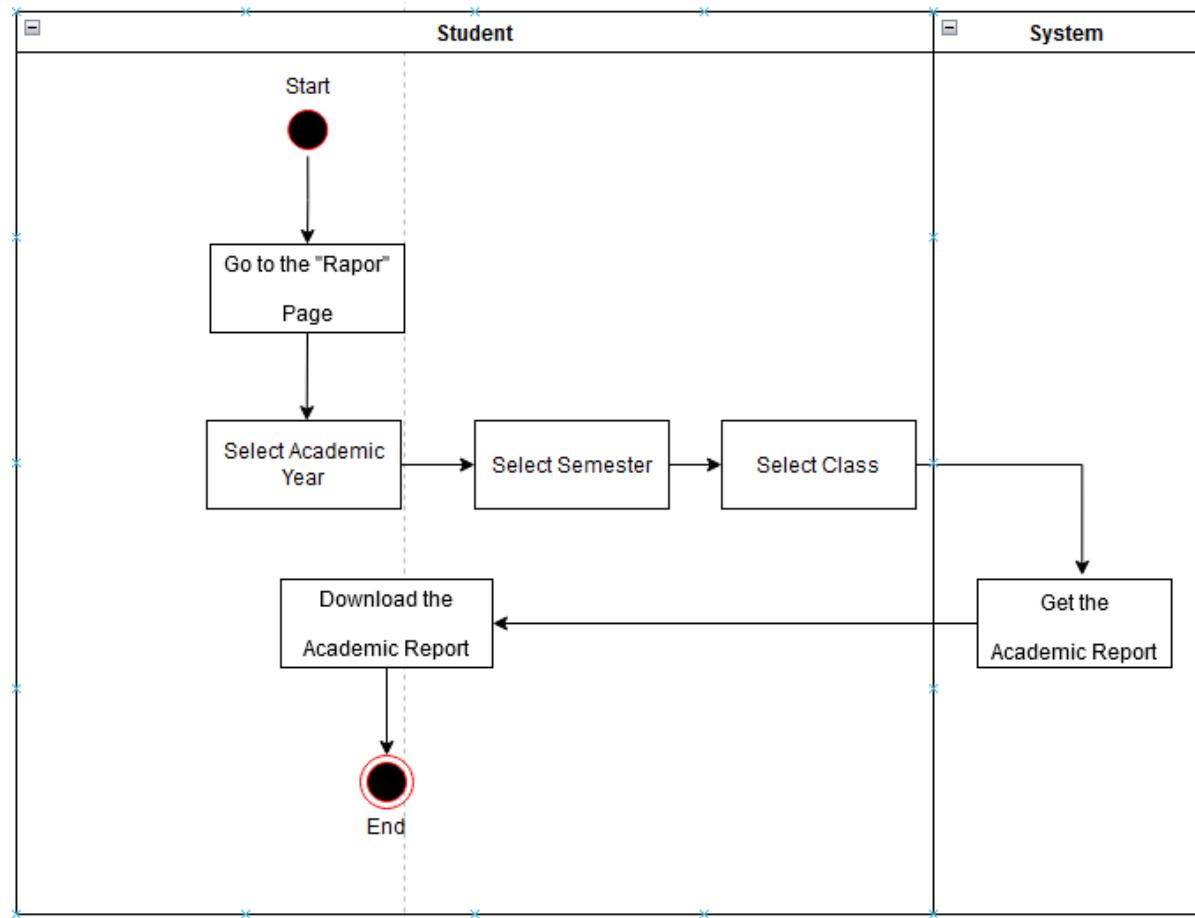
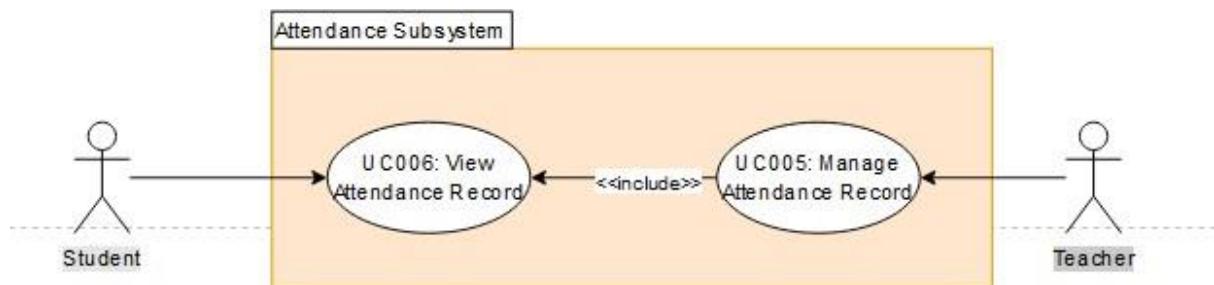


Figure 3.10: Activity Diagram of Download Academic Report

3.2.3 Module Attendance



3.2.3.1 UC005: Use Case Manage Attendance Record

Table 3.5: Use Case Description for Manage Attendance Record

Use Case ID:	UC-005
Use Case Name:	Manage Attendance Record
Actors:	Teacher
Description:	This use case is used for teacher to manage the attendance record including to take attendance during class
Pre-conditions:	<ul style="list-style-type: none"> 1. Logged in to the system
Normal Flow:	<ul style="list-style-type: none"> 1. Go to the “Absensi” page 2. Select date and hour 3. Select class 4. The system displays the attendance record 5. If the teacher wants to take attendance go to AF1 6. If the teacher clicks the “Ubah” button go to AF2
Alternative Flow:	<ul style="list-style-type: none"> 1. Attendance Checking <ul style="list-style-type: none"> 1.1 Teacher begins attendance checking 1.2 Fill in the case and description field for each student 2. Edit Attendance <ul style="list-style-type: none"> 2.1 Teacher edit the attendance record
Exception:	E.1 Required field(s) is/are empty when attendance checking <ul style="list-style-type: none"> 1. an error message will be displayed by the system 2. Performs NF5/AF1
Post-condition s:	<ul style="list-style-type: none"> 1. Teacher successfully checking student attendance 2. Teacher successfully edits the attendance record 3. Teacher successfully views the attendance record
Related Requirement:	-

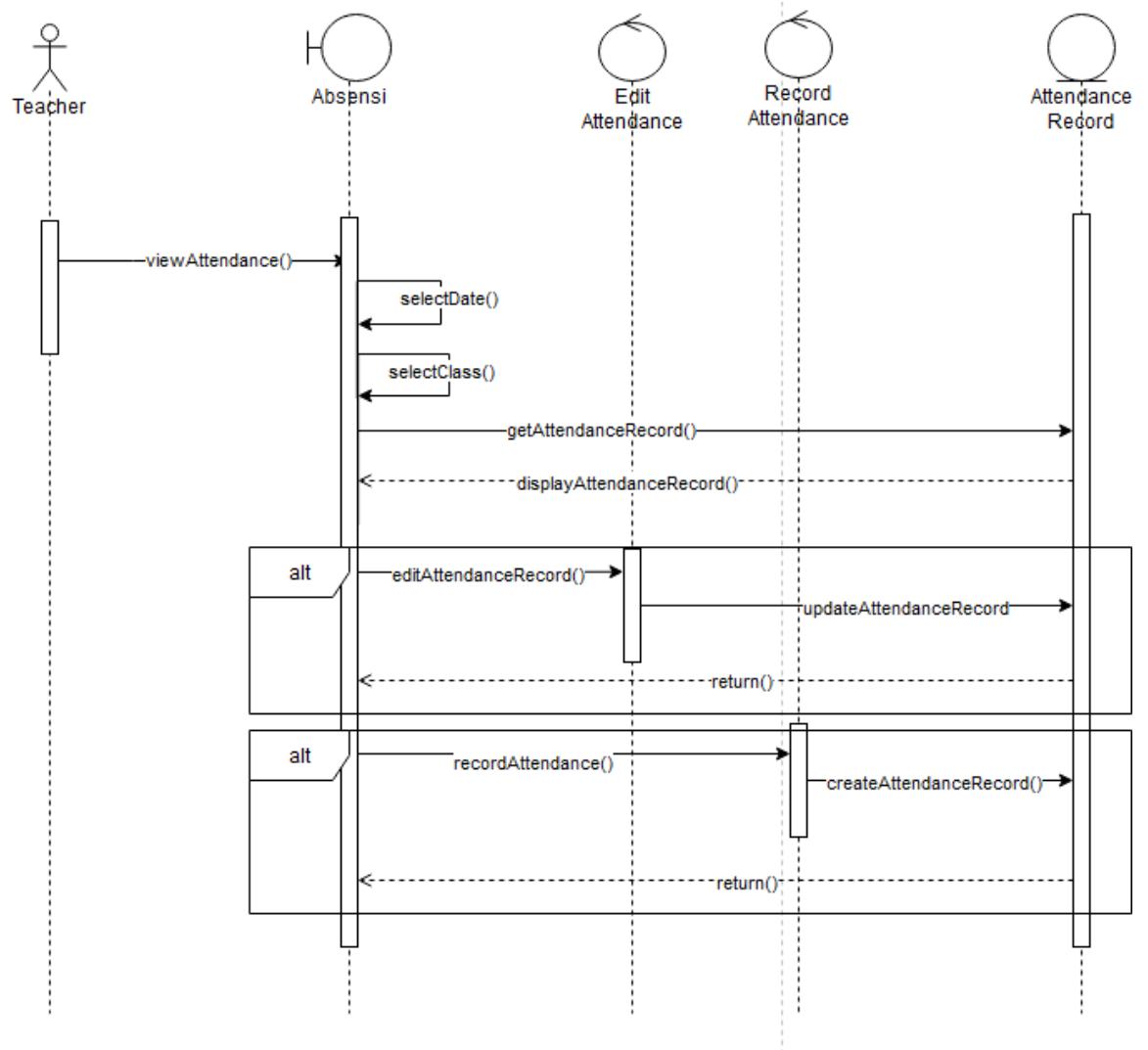


Figure 3.11: System Sequence Diagram of Manage Attendance Record

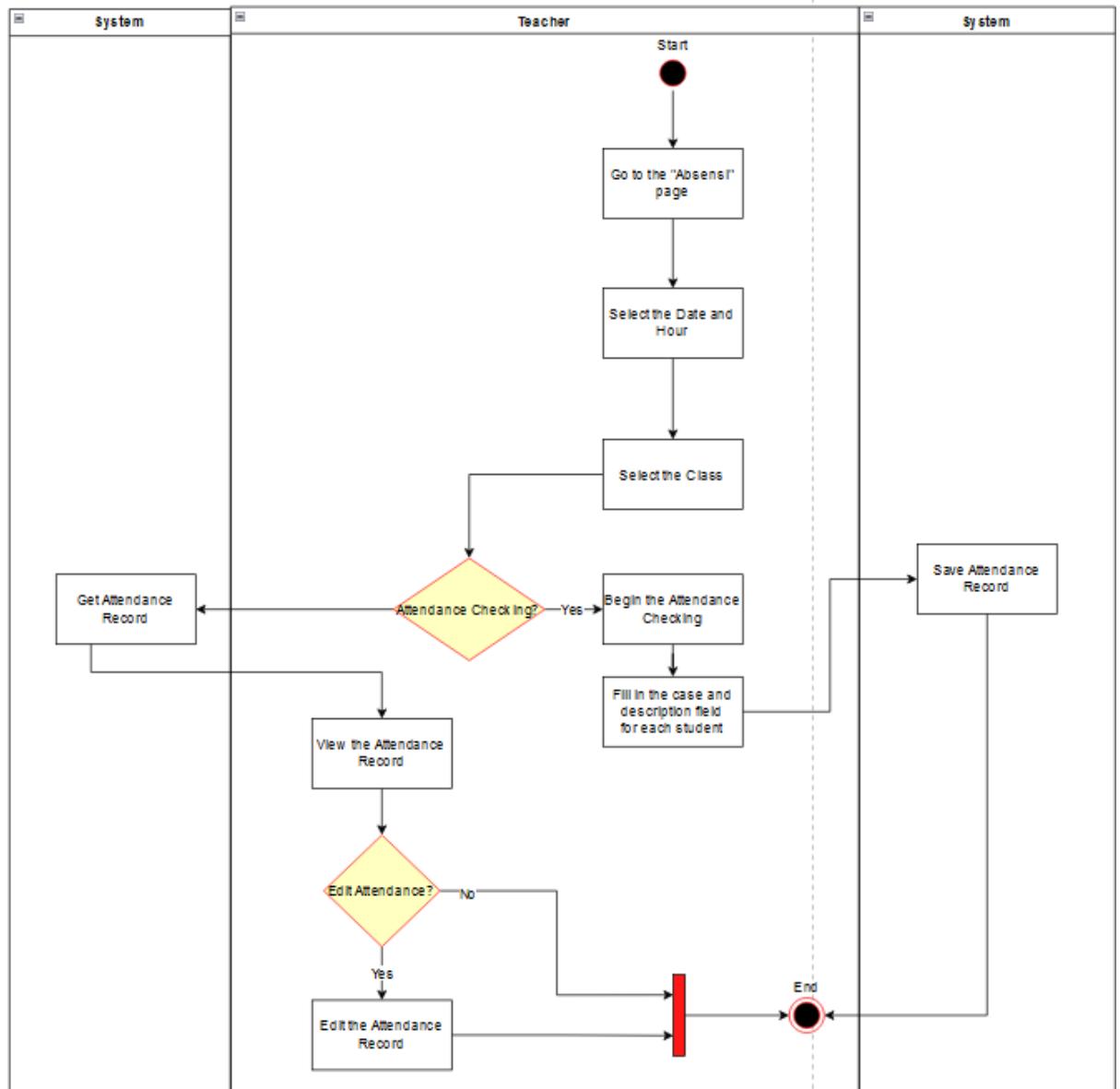


Figure 3.12: Activity Diagram of Manage Attendance Record

3.2.3.2

UC006: Use Case View Attendance Record**Table 3.6: Use Case Description for View Attendance Record**

Use Case ID:	UC-006
Use Case Name:	View Attendance Record
Actors:	Student
Description:	This use case is used for student to views the attendance record
Pre-conditions:	<ul style="list-style-type: none"> 1. Logged in to the system
Normal Flow:	<ul style="list-style-type: none"> 1. Go to the “Absensi” page 2. Select date 3. Select class 4. The system displays the attendance record
Alternative Flow:	-
Exception:	-
Post-condition s:	<ul style="list-style-type: none"> 1. Student successfully views the attendance record
Related Requirement:	-

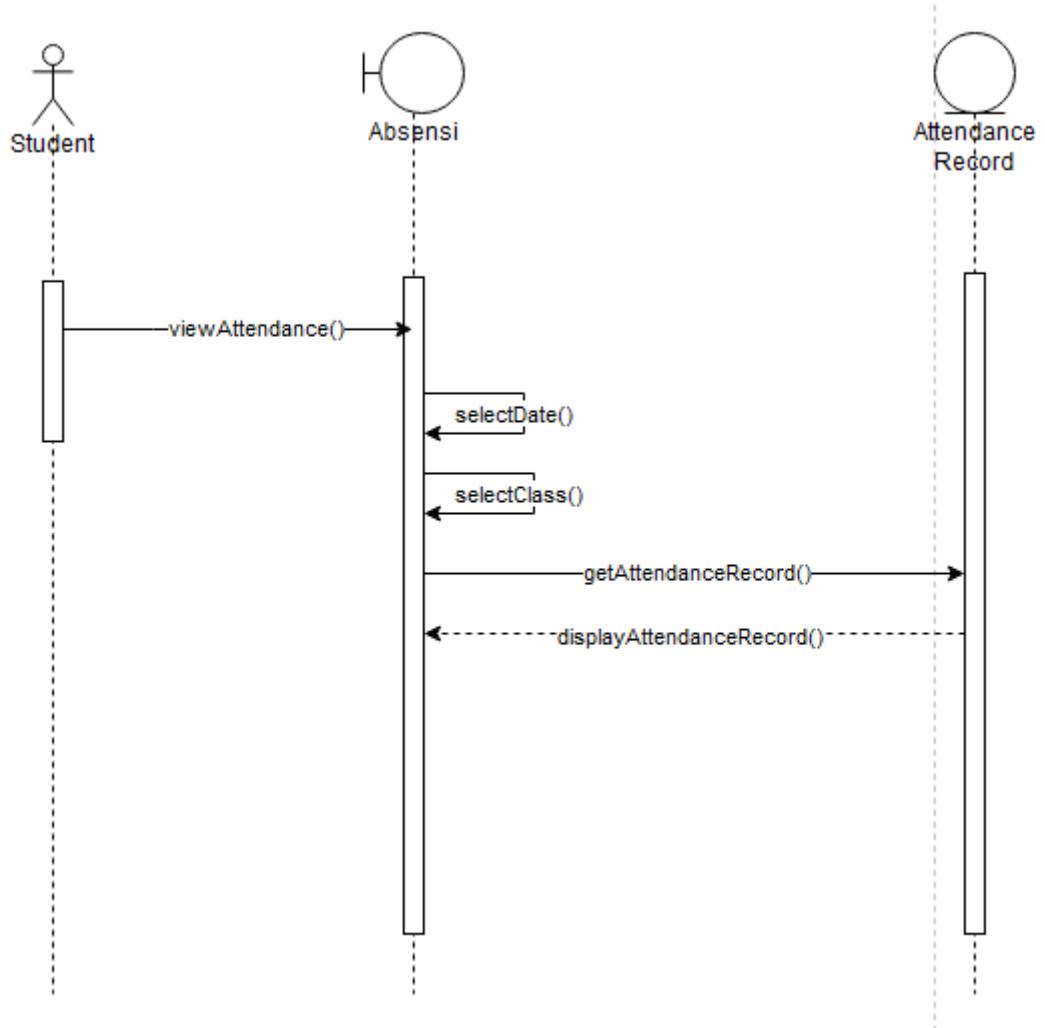


Figure 3.13: System Sequence Diagram of View Attendance Record

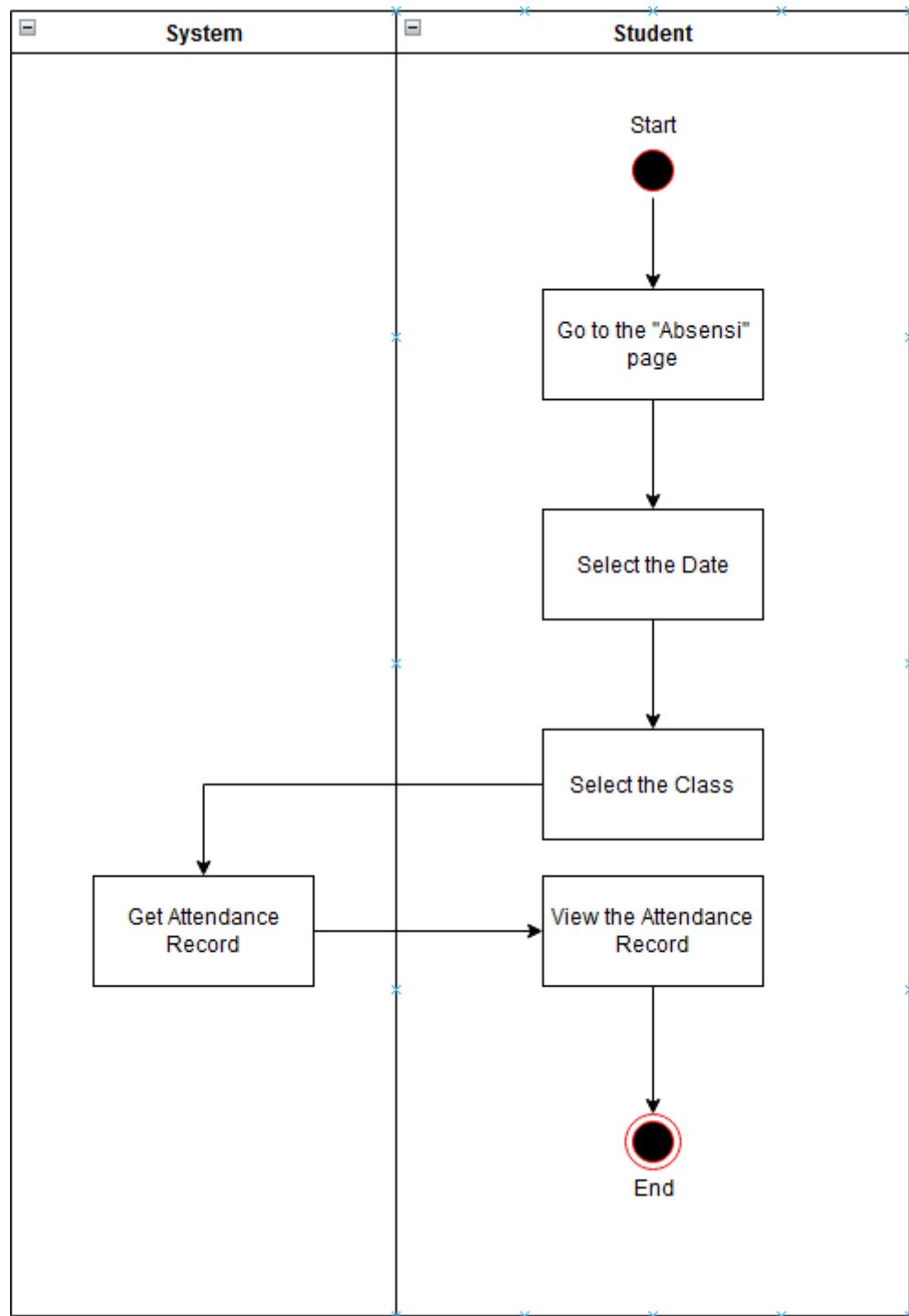
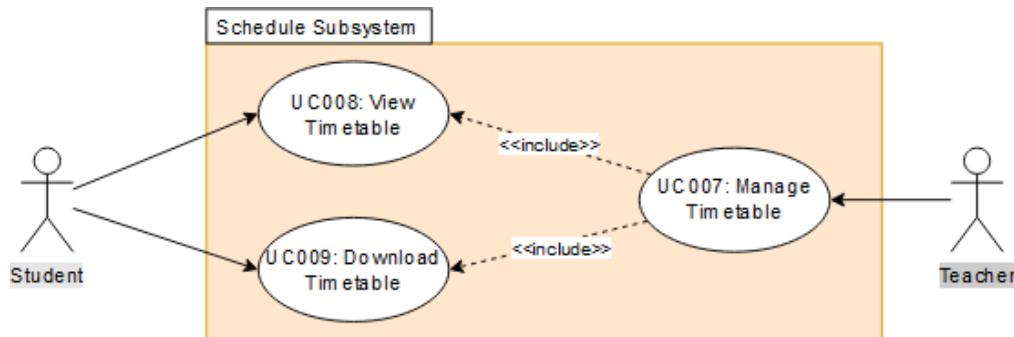


Figure 3.14: Activity Diagram of View Attendance Record

3.2.4 Module Schedule



3.2.4.1 UC007: Use Case Manage Timetable

Table 3.7: Use Case Description for Manage Timetable

Use Case ID:	UC-007
Use Case Name:	Manage Timetable
Actors:	Teacher
Description:	This use case is used for student to manage the attendance record including to take attendance during class
Pre-conditions:	1. Logged in to the system
Normal Flow:	<ol style="list-style-type: none"> 1. Go to the “Jadwal Pelajaran” page 2. Select class 3. The system displays the timetable 4. If the teacher wants to download timetable performs AF1 5. If the teacher is a homeroom teacher and want to edit timetable go to AF2
Alternative Flow:	<ol style="list-style-type: none"> 1. Create Timetable <ol style="list-style-type: none"> 1.1 Click “Ubah Jadwal Pelajaran” page 1.2 Select Class 1.3 Select subject 1.4 Select Day and Time 1.5 Click “Simpan” button 2. Download Timetable <ol style="list-style-type: none"> 2.1 The teacher clicks “Unduh” button
Exception:	E.1 Required field(s) is/are empty when editing Timetable <ol style="list-style-type: none"> 1. an error message will be displayed by the system 2. Performs NF5/AF2

Post-condition S:	1. Teacher successfully views the timetable 2. Teacher successfully edits the timetable 3. Teacher successfully downloads the timetable
Related Requirement:	-

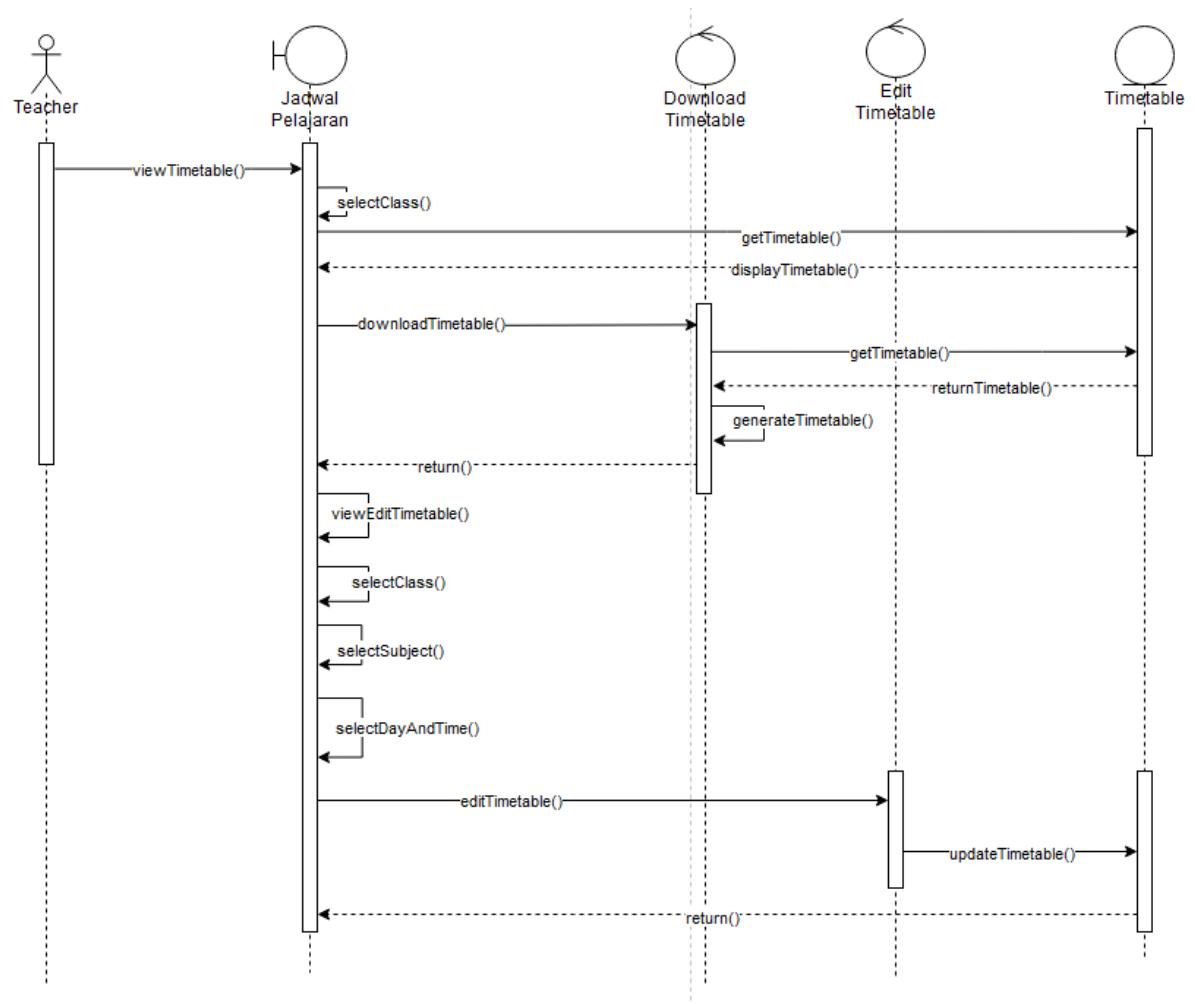


Figure 3.15: System Sequence Diagram of Manage Timetable

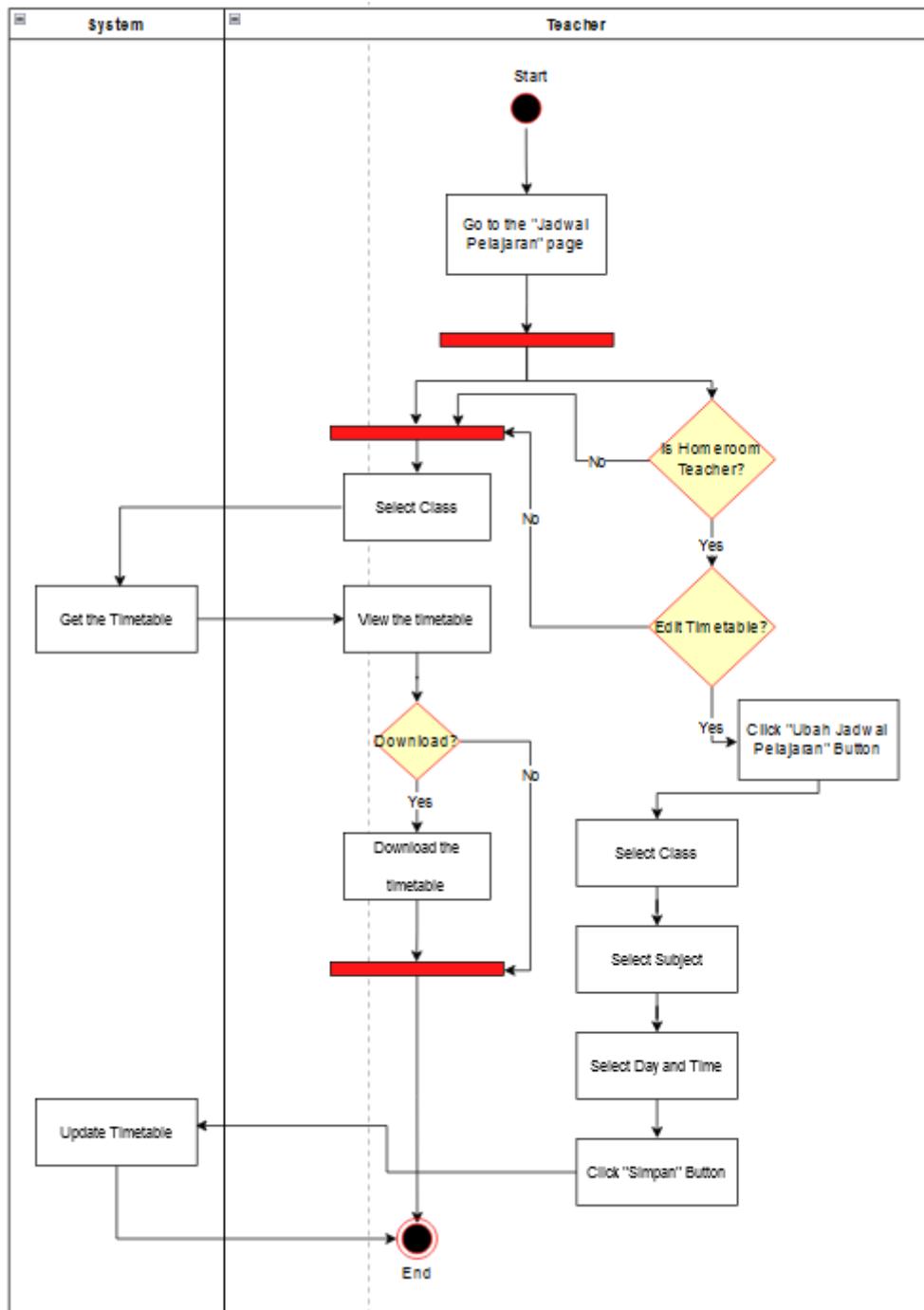


Figure 3.16: Activity Diagram of Manage Timetable

3.2.4.2

UC008: Use Case View Timetable

Table 3.8: Use Case Description for View Timetable

Use Case ID:	UC-008
Use Case Name:	View Timetable
Actors:	Student
Description:	This use case is used for student to views the timetable
Pre-conditions:	1. Logged in to the system
Normal Flow:	<ol style="list-style-type: none"> 1. Go to the “Jadwal Pelajaran” page 2. The system displays the timetable
Alternative Flow:	-
Exception:	-
Post-condition s:	1. Student successfully views the timetable
Related Requirement:	-

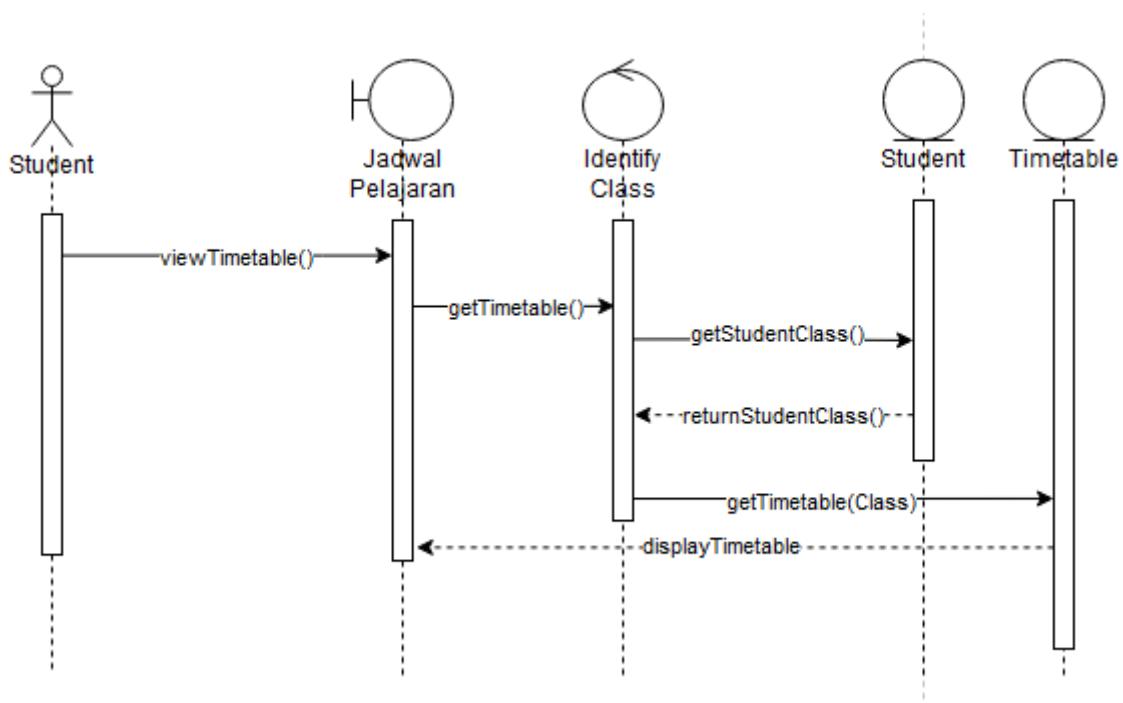


Figure 3.17: System Sequence Diagram of View Timetable

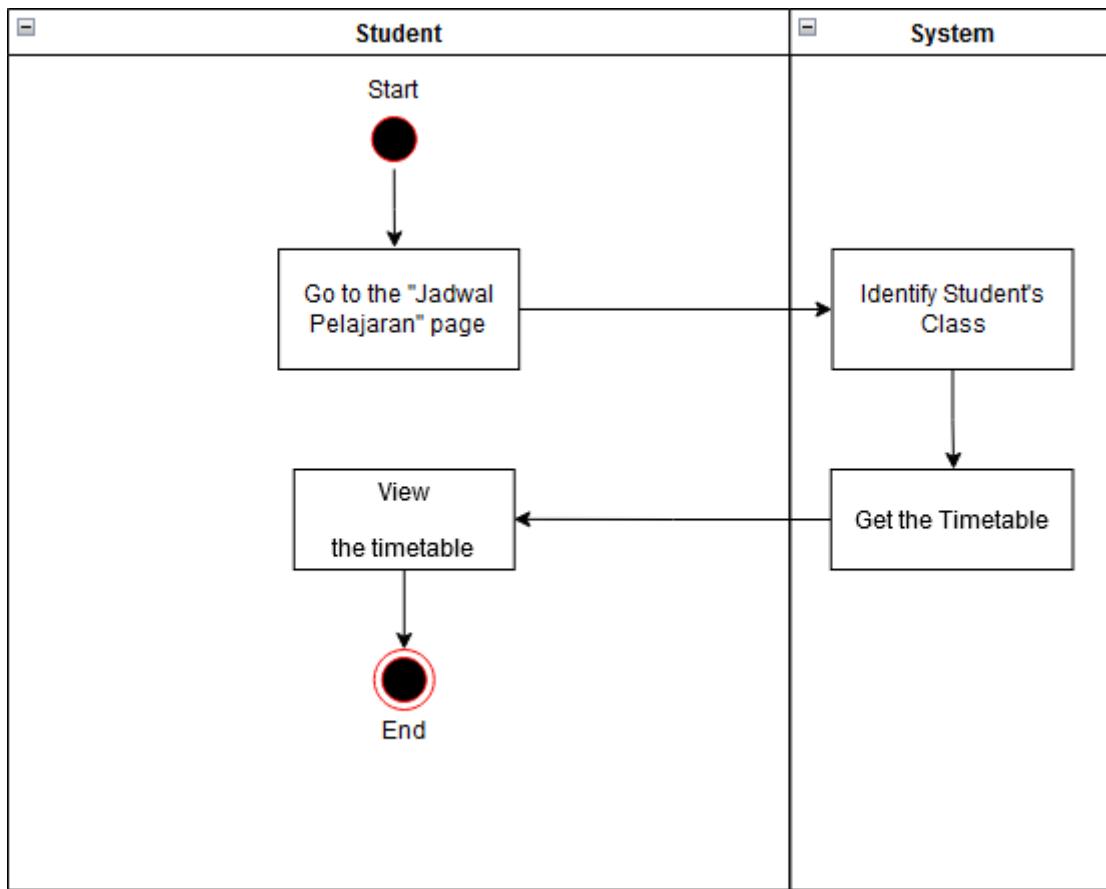


Figure 3.18: Activity Diagram of View Timetable

3.2.4.3 UC009: Use Case Download Timetable

Table 3.9: Use Case Description for Download Timetable

Use Case ID:	UC-009
Use Case Name:	Download Timetable
Actors:	Student
Description:	This use case is used for student to download the timetable
Pre-conditions:	1. Logged in to the system
Normal Flow:	<ol style="list-style-type: none"> 1. Go to the “Jadwal Pelajaran” page 2. The student clicks “Unduh” button
Alternative Flow:	-
Exception:	-
Post-condition s:	1. Student successfully download the timetable
Related Requirement:	-

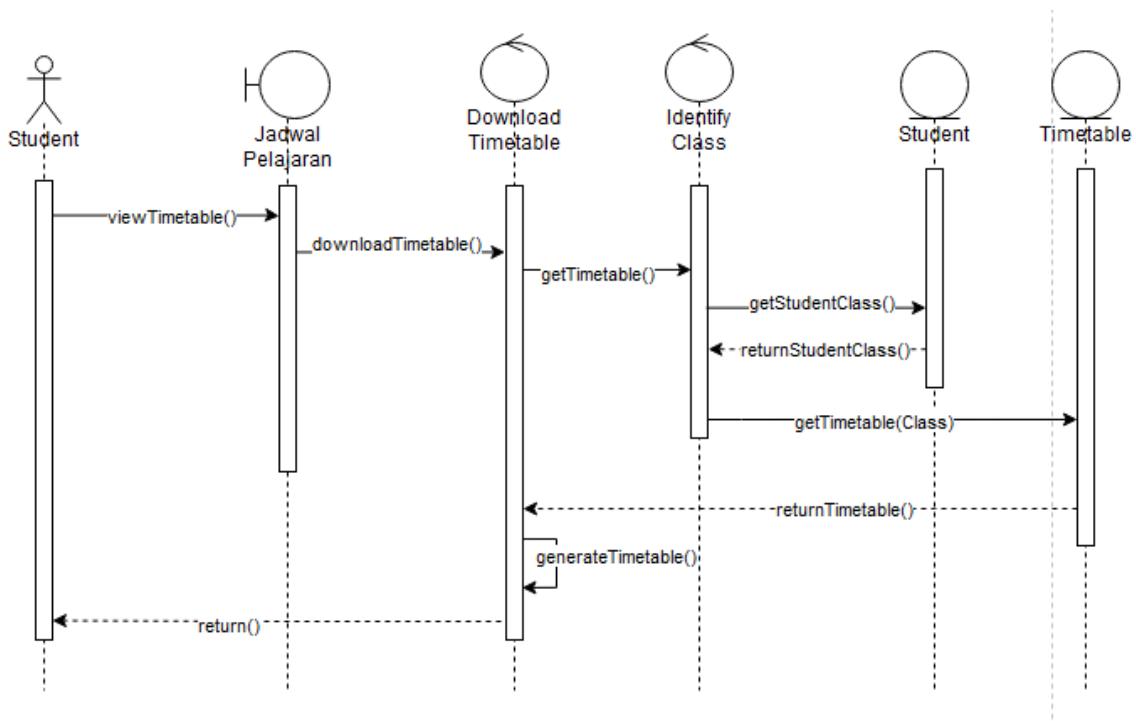


Figure 3.19: System Sequence Diagram of Download Timetable

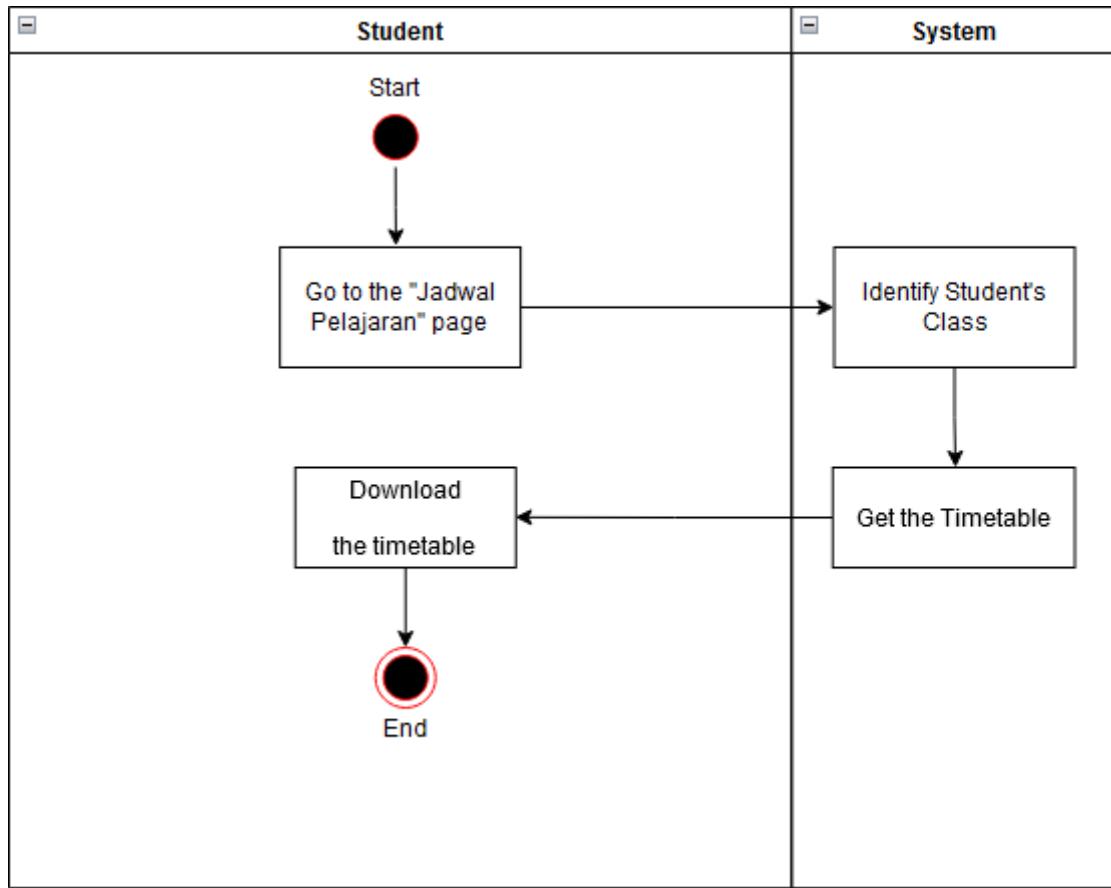
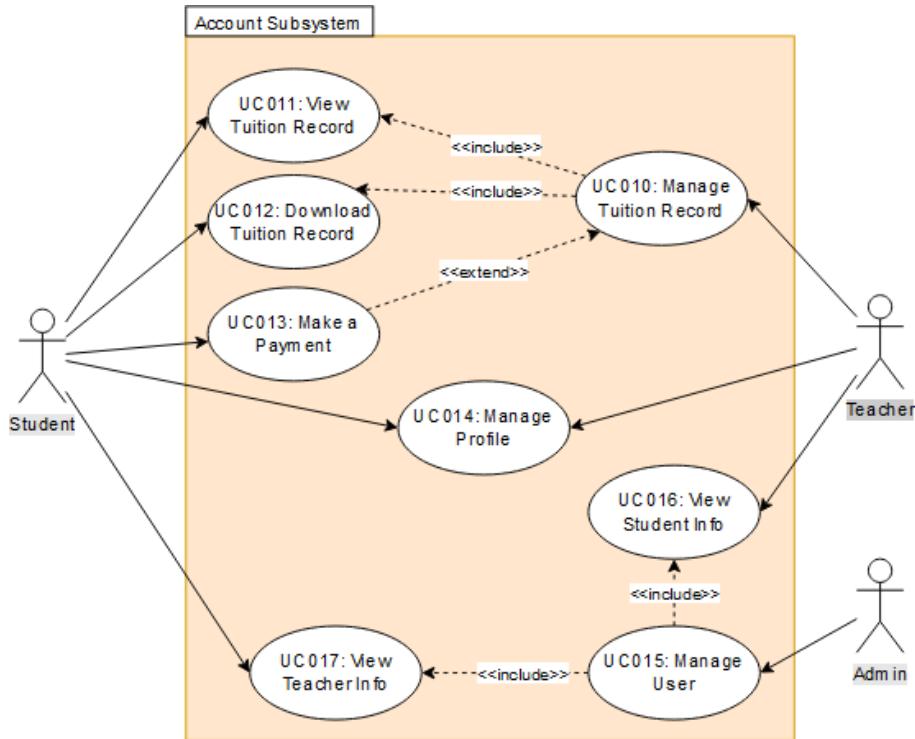


Figure 3.20: Activity Diagram of Download Timetable

3.2.5 Module Account



3.2.5.1 UC010: Use Case Manage Tuition Record

Use Case ID:	UC-010
Use Case Name:	Manage Tuition Record
Actors:	Teacher
Description:	This use case is used for teacher to manage the tuition record including approving or rejecting tuition payment
Pre-conditions:	1. Logged in to the system
Normal Flow:	<ol style="list-style-type: none"> 1. Go to the “SPP” page 2. Select academic year 3. Select class 4. Select student 5. The system displays the tuition record 6. If the teacher wants to approve the payment performs AF1 7. If the teacher wants to reject the payment performs AF2
Alternative Flow:	<ol style="list-style-type: none"> 1. Approve payment <ol style="list-style-type: none"> 1.1 Change the status to paid 2. Download Timetable <ol style="list-style-type: none"> 2.1 Change the status to outstanding
Exception:	-

Post-condition S:	1. Teacher successfully views the tuition record 2. Teacher successfully approves or reject the payment
Related Requirement:	-

Table 3.10: Use Case Description for Manage Tuition Record

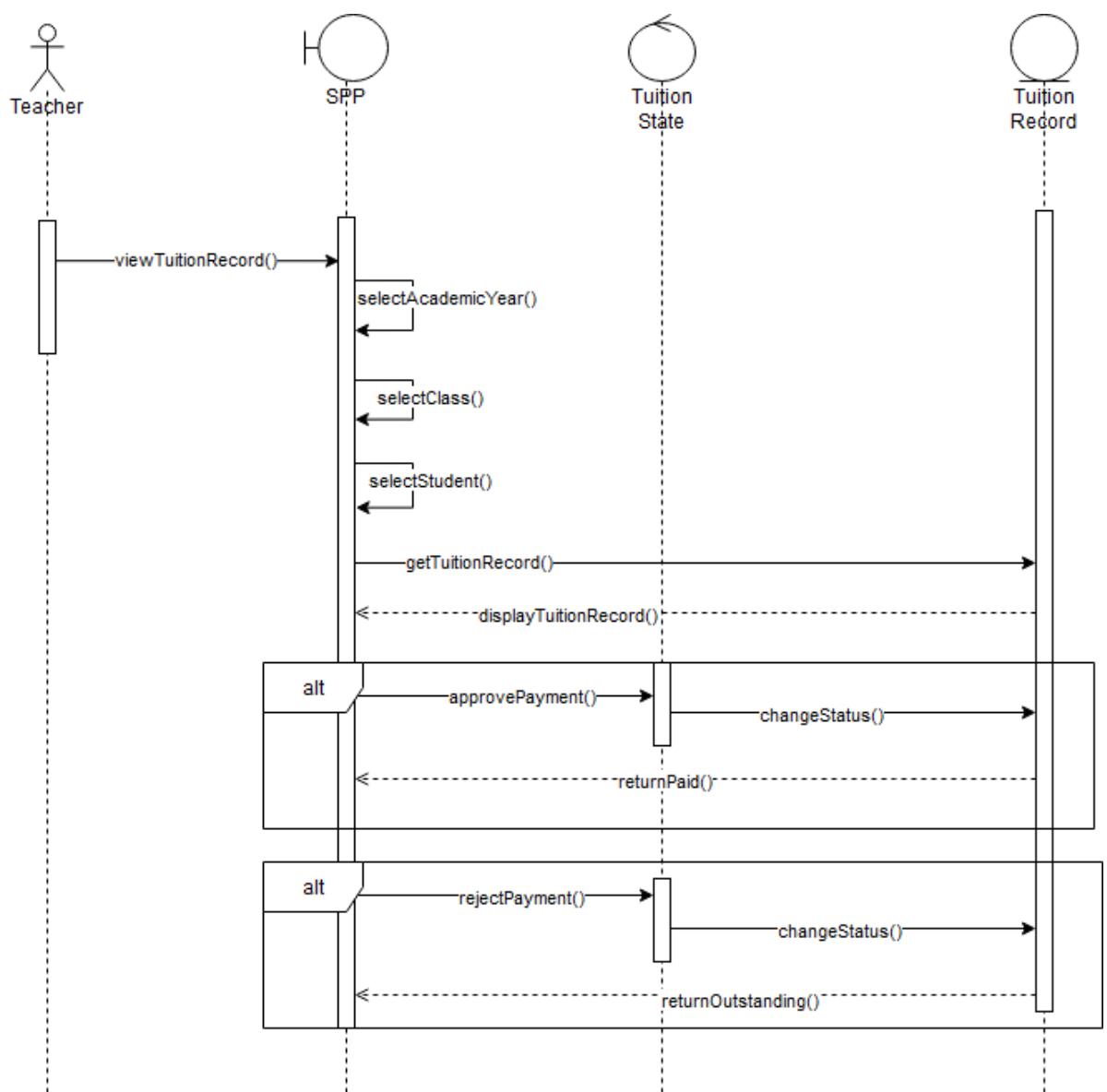


Figure 3.21: System Sequence Diagram of Manage Tuition Record

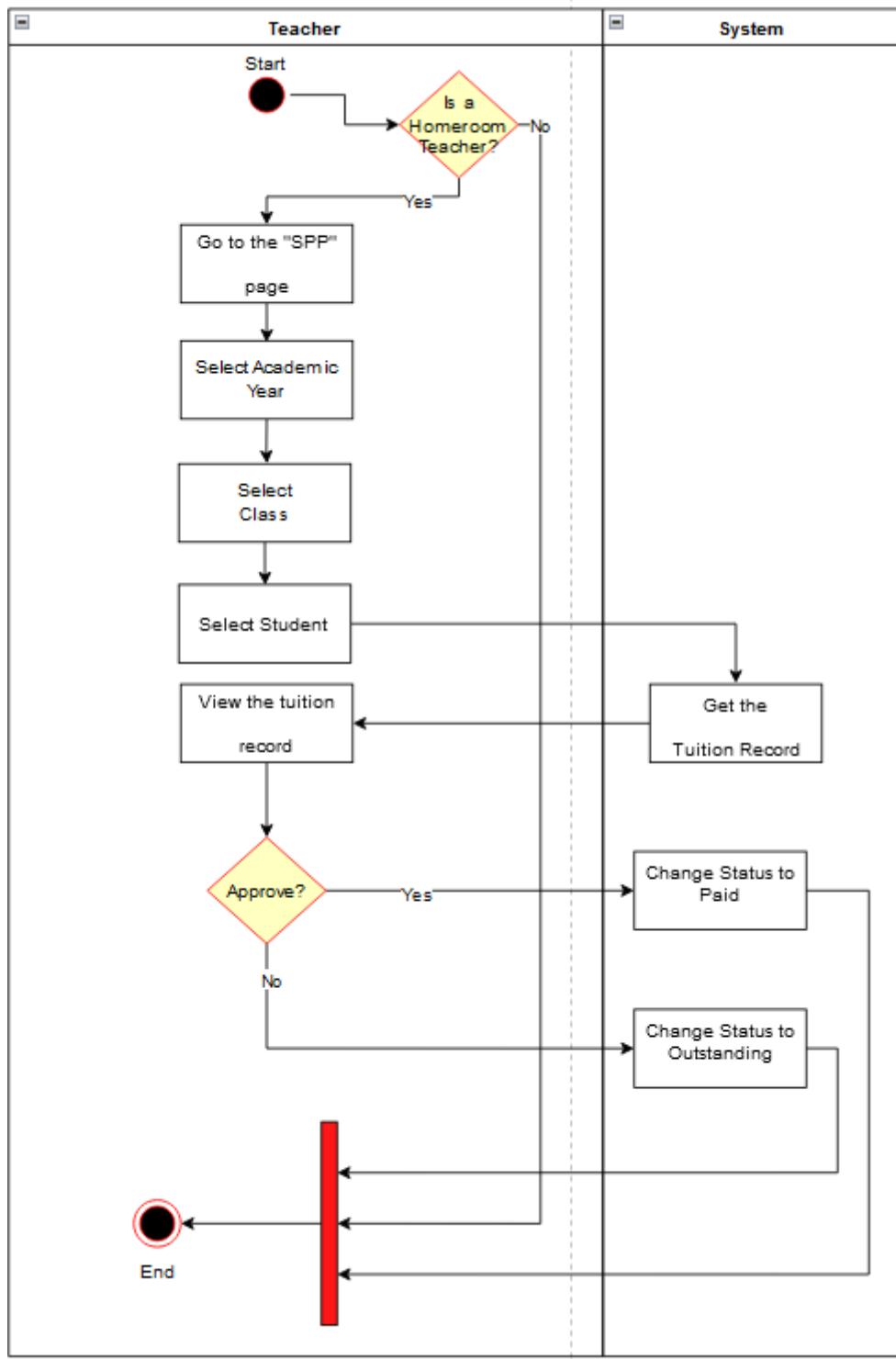


Figure 3.22: Activity Diagram of Manage Tuition Record

3.2.5.2 UC011: Use Case View Tuition Record

Table 3.11: Use Case Description for View Tuition Record

Use Case ID:	UC-011
Use Case Name:	View Tuition Record
Actors:	Student
Description:	This use case is used for student to view the tuition record
Pre-conditions:	1. Logged in to the system
Normal Flow:	<ol style="list-style-type: none"> 1. Go to the “SPP” page 2. Select academic year 3. Select class 4. Select student 5. The system displays the tuition record
Alternative Flow:	-
Exception:	-
Post-condition s:	1. Student successfully views the tuition record
Related Requirement:	-

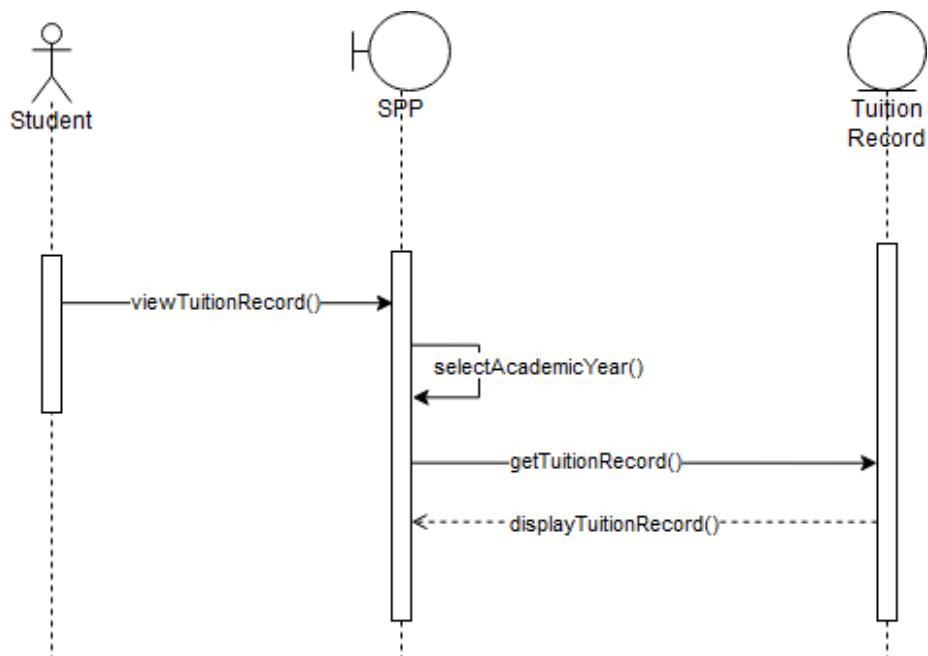


Figure 3.23: System Sequence Diagram of View Tuition Record

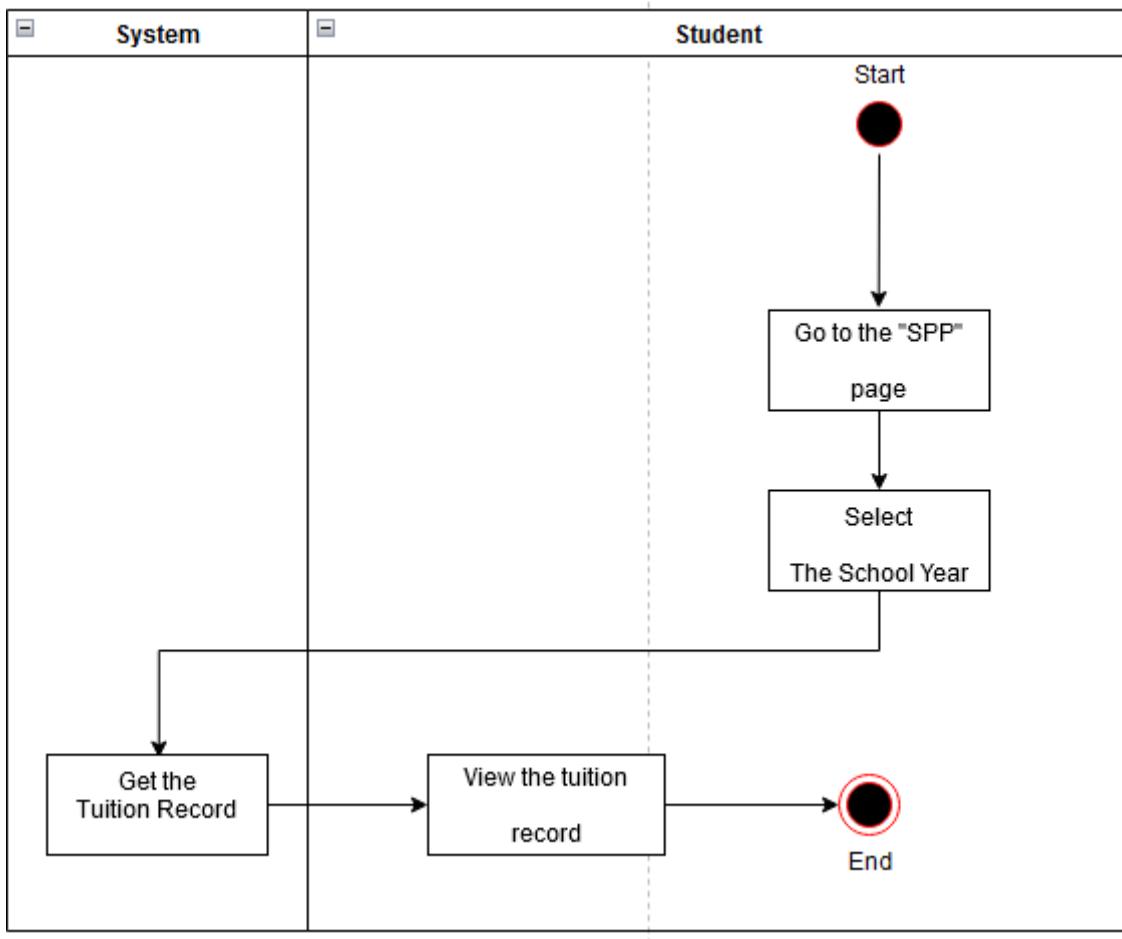


Figure 3.24: Activity Diagram of View Tuition Record

3.2.5.3 UC012: Use Case Download Tuition Record

Table 3.12: Use Case Description for Download Tuition Record

Use Case ID:	UC-012
Use Case Name:	Download Tuition Record
Actors:	Student
Description:	This use case is used for student to download the tuition record
Pre-conditions:	<ul style="list-style-type: none"> 1. Logged in to the system
Normal Flow:	<ul style="list-style-type: none"> 1. Go to the “SPP” page 2. Select academic year 3. Select class 4. Select student 5. Click the “Unduh” button to download the tuition record
Alternative Flow:	-
Exception:	-
Post-condition s:	<ul style="list-style-type: none"> 1. Student successfully download the tuition record
Related Requirement:	-

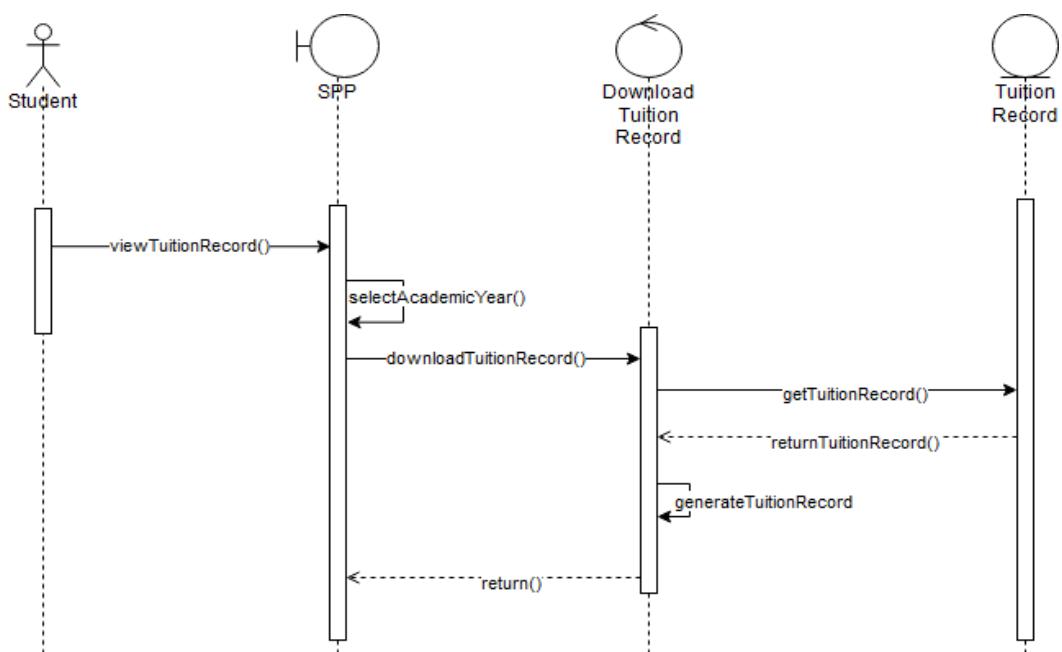


Figure 3.25: System Sequence Diagram of Download Tuition Record

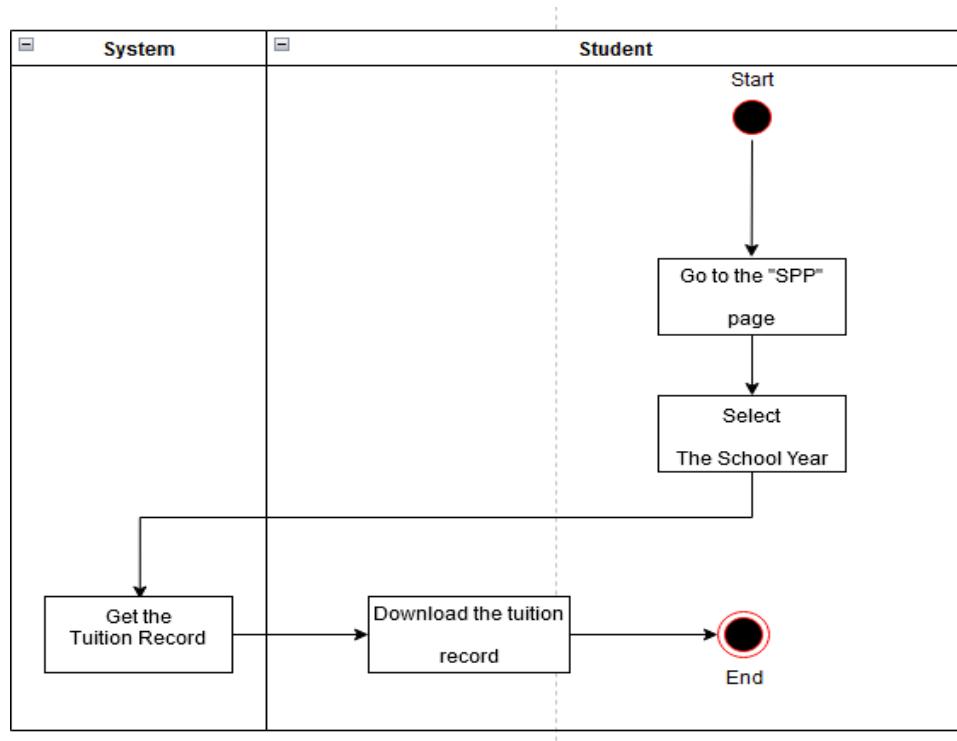


Figure 3.26: Activity Diagram of Download Tuition Record

3.2.5.4 UC013: Use Case Make a Payment

Table 3.13: Use Case Description for Make a Payment

Use Case ID:	UC-013
Use Case Name:	Make A Payment
Actors:	Student
Description:	This use case is used for student to download the tuition record
Pre-conditions:	<ul style="list-style-type: none"> 1. Logged in to the system
Normal Flow:	<ul style="list-style-type: none"> 1. Go to the “SPP” page 2. Select “Pembayaran SPP” 3. Select the month to be paid 4. Upload proof of payment
Alternative Flow:	-
Exception:	E.1 Unsupported uploaded file extension <ul style="list-style-type: none"> 1. An error message will be displayed by the system 2. Performs NF4
Post-condition s:	<ul style="list-style-type: none"> 1. Student successfully uploads the proof of payment
Related Requirement:	-

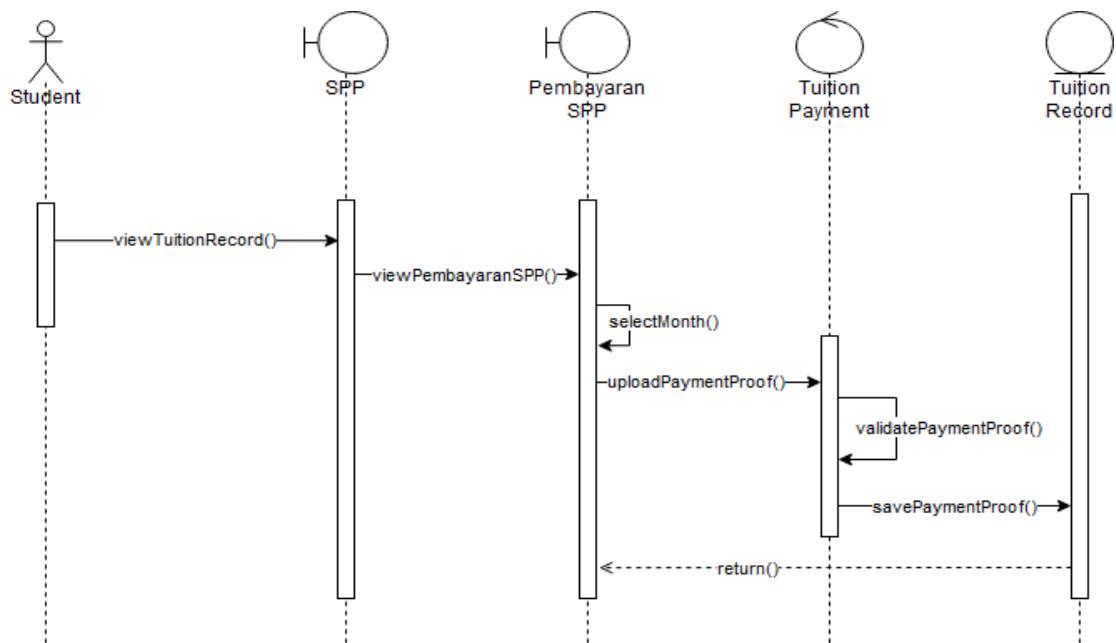


Figure 3.27: System Sequence Diagram of Make a Payment

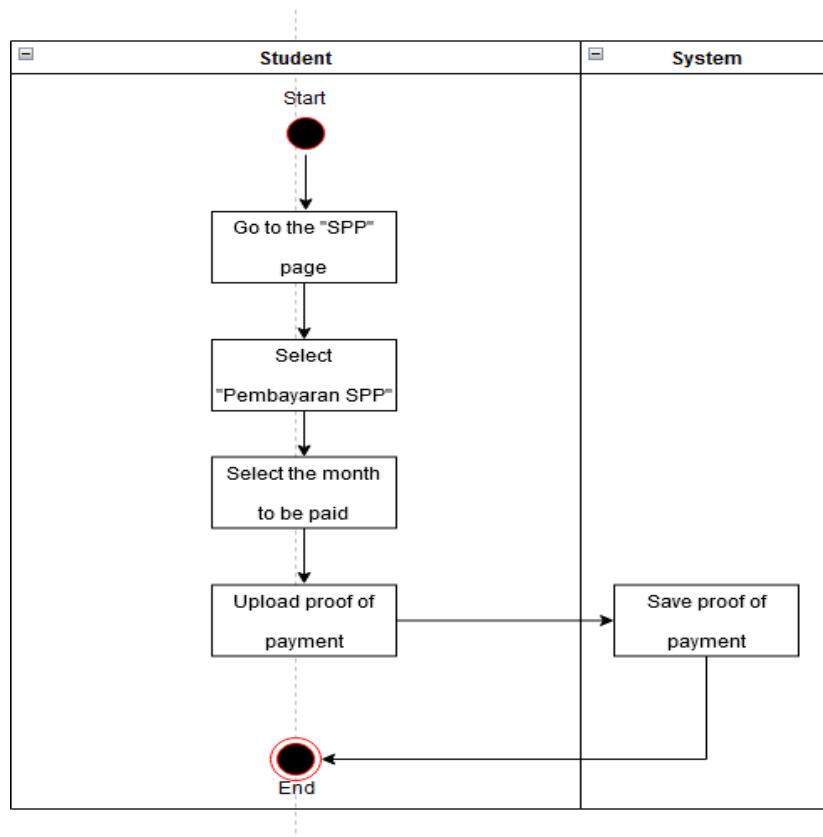


Figure 3.28: Activity Diagram of Make a Payment

3.2.5.5

UC014: Use Case Manage Profile**Table 3.14: Use Case Description for Manage Profile**

Use Case ID:	UC-014
Use Case Name:	Manage Profile
Actors:	Student/Teacher
Description:	This use case is used for student and teacher to view and edit their profile
Pre-conditions:	<ul style="list-style-type: none"> 1. Logged in to the system
Normal Flow:	<ul style="list-style-type: none"> 1. Go to the “Profil” page 2. The system display user profile 3. If the user wants to edit their profile performs AF1
Alternative Flow:	-
Exception:	-
Post-condition S:	<ul style="list-style-type: none"> 1. Student/teacher successfully views their profile info 2. Student/teacher successfully edit their profile info
Related Requirement:	-

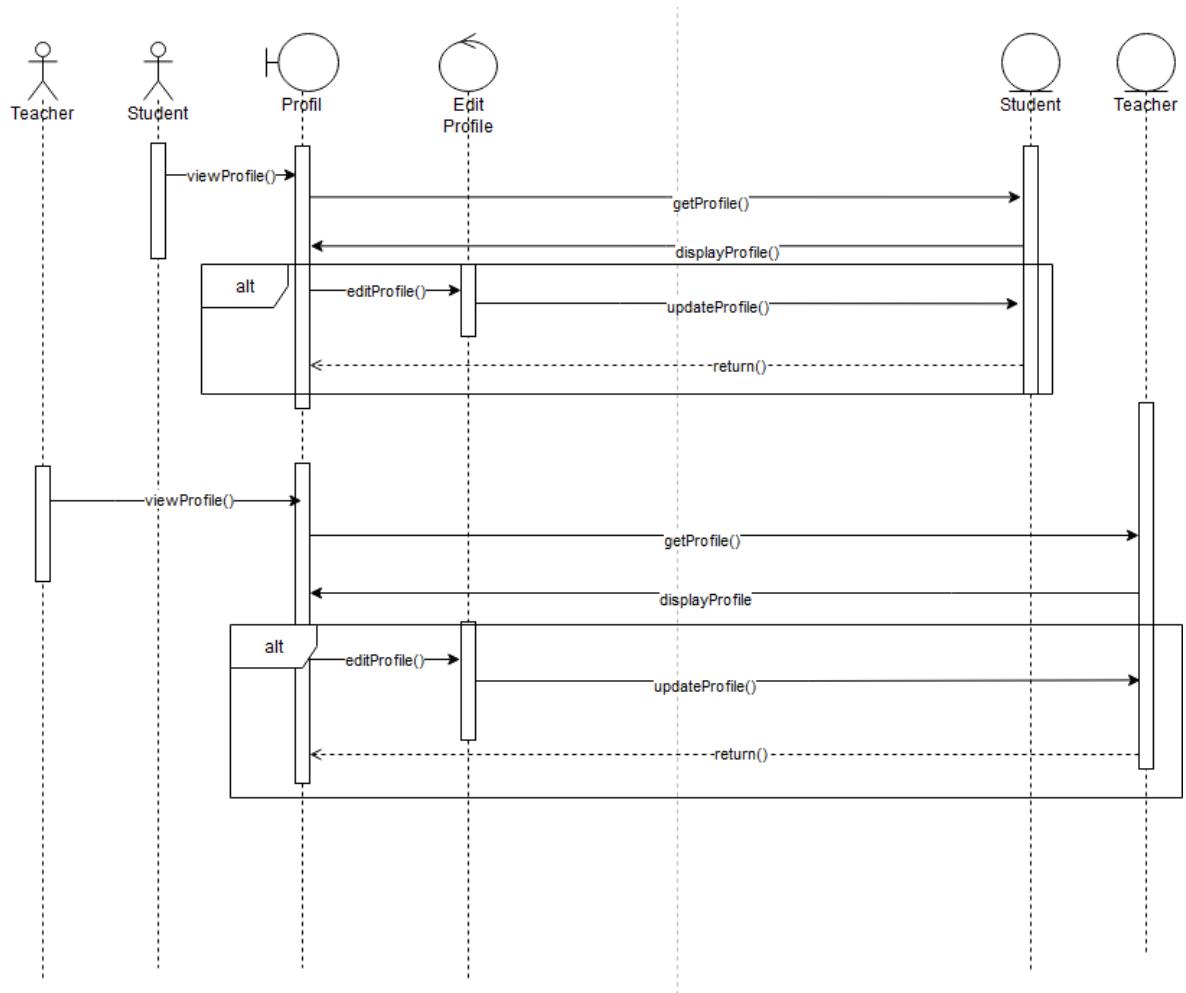


Figure 3.29: System Sequence Diagram of Manage Profile

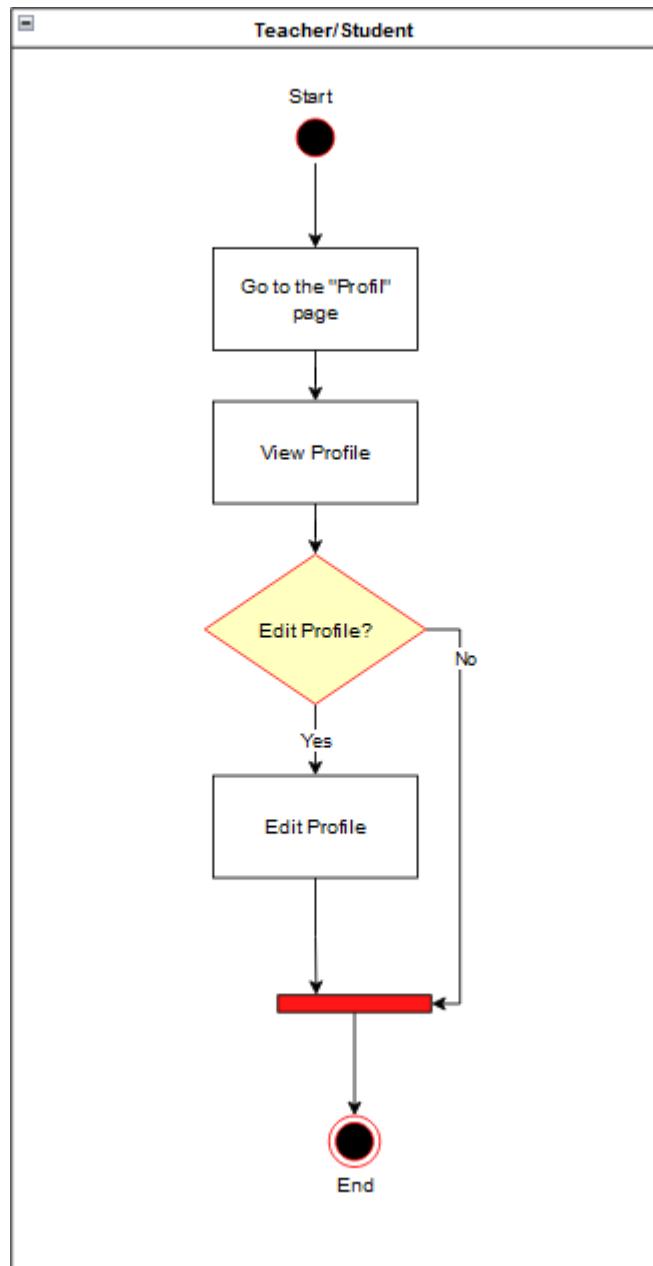


Figure 3.30: Activity Diagram of Manage Profile

3.2.5.6

UC015: Use Case Manage User

Use Case ID:	UC-015
Use Case Name:	Manage User
Actors:	Admin
Description:	This use case is used for admin to manage user account both student and teacher
Pre-conditions:	<ul style="list-style-type: none"> 1. Logged in to the system
Normal Flow:	<ul style="list-style-type: none"> 1. Go to the “Akun Manajemen” page 2. The system displays the list of user page 3. If the selected user type is Teacher performs AF1 4. If the selected user type is student performs AF2 5. If admin want to add a new student account performs AF3 6. If admin want to add a new teacher account performs AF4 7. If admin want to add a new admin account performs AF5
Alternative Flow:	<ul style="list-style-type: none"> 1. View Teacher List <ul style="list-style-type: none"> 1.1 The system display teacher account list 1.2 If the admin wants to delete the account performs AF7 1.3 If the admin wants to assign the teacher as a homeroom teacher performs AF6 2. View Student List <ul style="list-style-type: none"> 2.1 The system display student account list 2.2 If the admin wants to delete the account performs AF7 3. Add student account <ul style="list-style-type: none"> 3.1 Go to “Tambah Akun Siswa” page 3.2 Fill in the Create Student Account Form 3.3 Click “Tambah” to create the student account 4. Add teacher account <ul style="list-style-type: none"> 4.1 Go to “Tambah Akun Guru” page 4.2 Fill in the Create Teacher Account Form 4.3 Click “Tambah” to create the student account 5. Add admin account <ul style="list-style-type: none"> 5.1 Go to “Tambah Akun Admin” page 5.2 Fill in the Create Admin Account Form 5.3 Click “Tambah” to create the student account 6. Assign homeroom teacher <ul style="list-style-type: none"> 6.1 Click the dropdown button to select which class will the teacher assigned to 7. Delete account <ul style="list-style-type: none"> 7.1 Delete the users account by clicking the “Trash” icon
Exception:	<p>E.1 The required fields when creating user account is empty</p> <ul style="list-style-type: none"> 1. An error message will be displayed by the system 2. Performs NF5/NF6/NF7 <p>E.2 Validation error during creating user account</p> <ul style="list-style-type: none"> 1. An error message will be displayed by the system 2. Performs NF5/NF6/NF7

Post-condition S:	1. Admin successfully create user account 2. Admin successfully views user list 3. Admin successfully assign teacher as a homeroom teacher
------------------------------------	--

Table 3.15: Use Case Description for Manage User

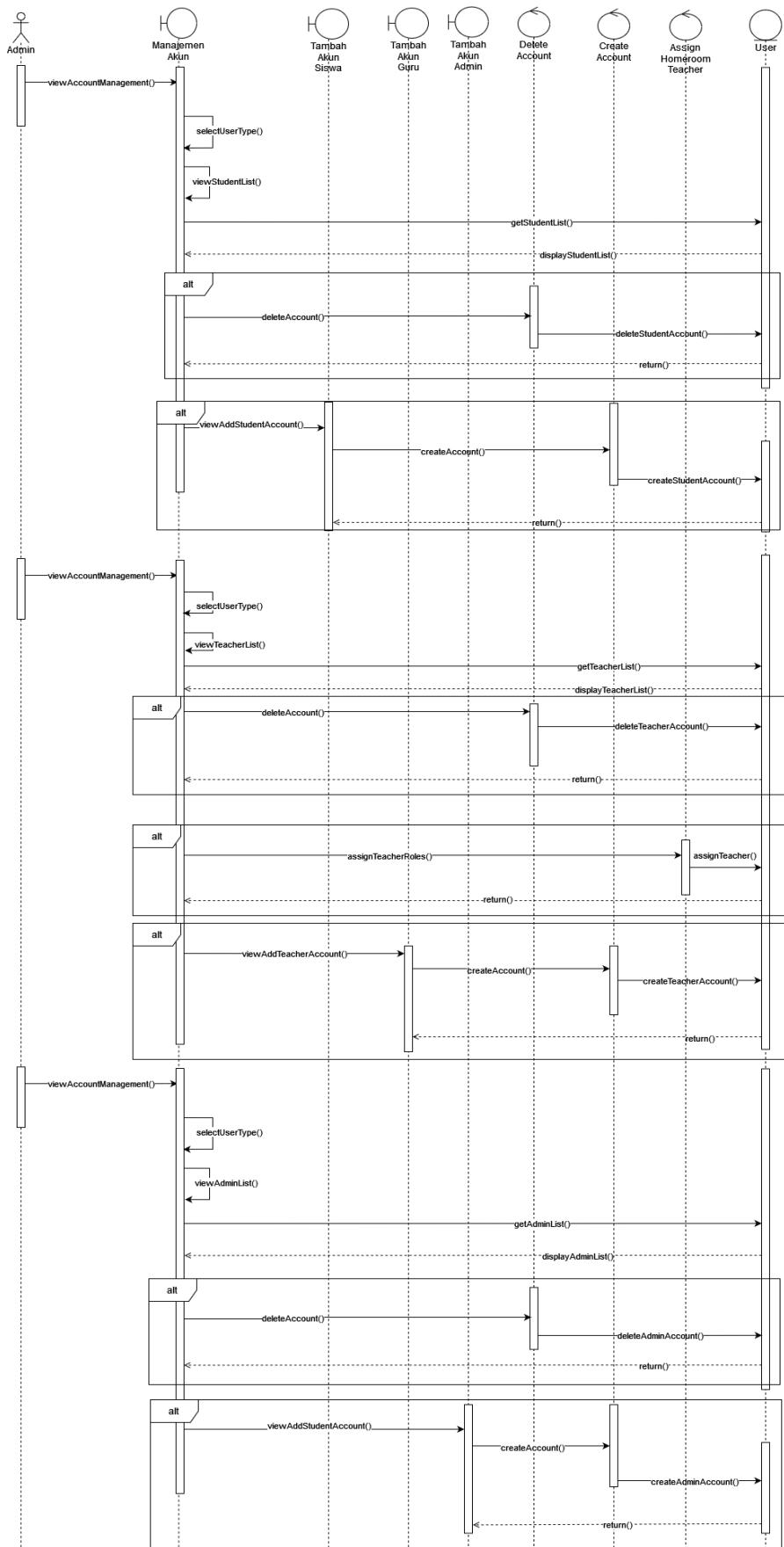


Figure 3.31: System Sequence Diagram of Manage User

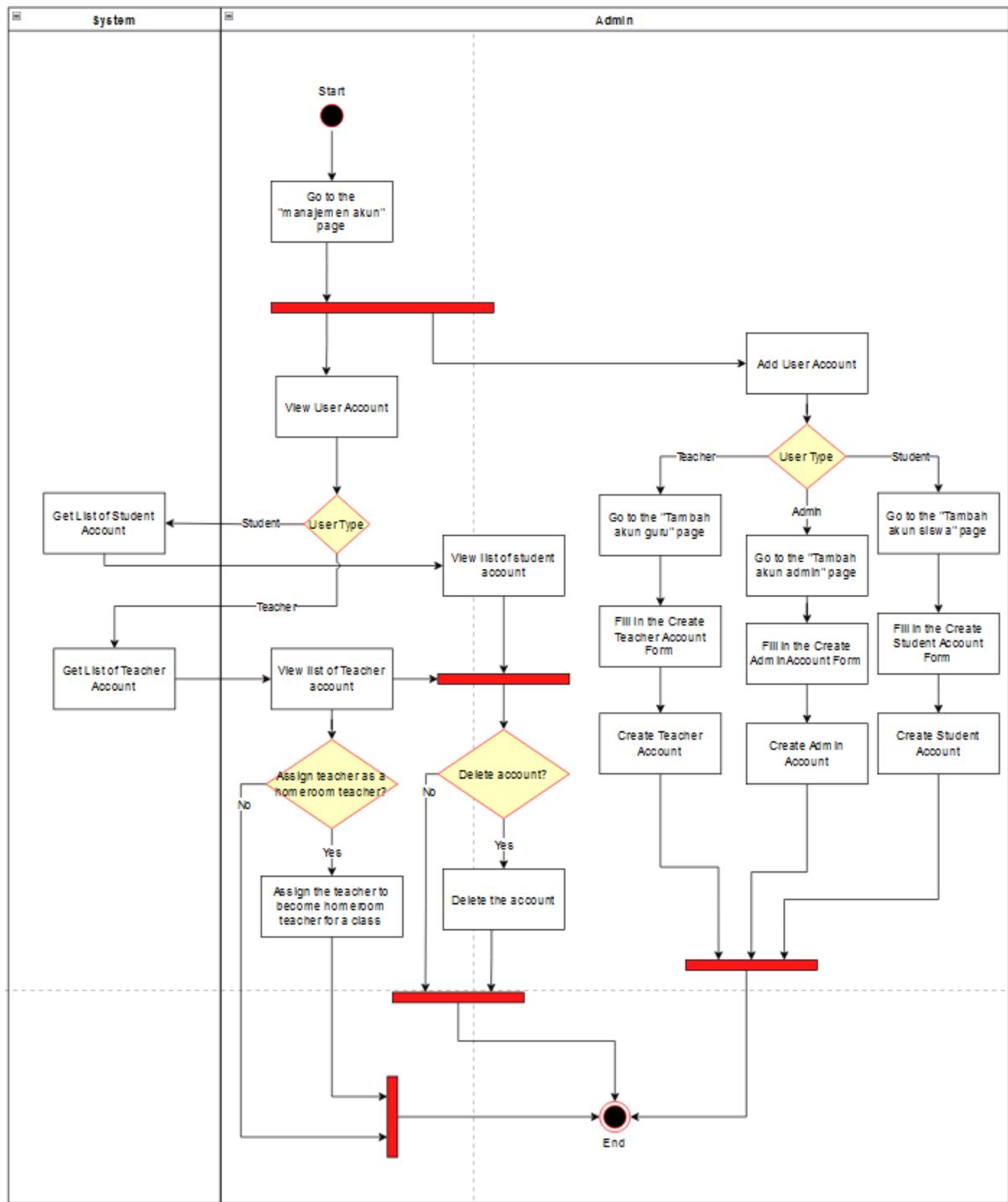


Figure 3.32: Activity Diagram of Manage User

3.2.5.7

UC016: Use Case View Student Info

Table 3.16: Use Case Description for View Student Info

Use Case ID:	UC-016
Use Case Name:	View Student Info
Actors:	Teacher
Description:	This use case is used for teacher to view list of students and their detailed information
Pre-conditions:	1. Logged in to the system
Normal Flow:	<ol style="list-style-type: none"> 1. Go to the “Daftar Siswa” page 2. Select the class 3. Select the academic year 4. Select the semester 5. The system display student list 6. If the teacher wants to view detailed information of the student performs AF1
Alternative Flow:	<ol style="list-style-type: none"> 1. View Student Detailed Info <ol style="list-style-type: none"> 1.1 Select one of the students from the list 1.2 The system displays the detailed information about the selected student
Exception:	-
Post-condition s:	<ol style="list-style-type: none"> 1. Teacher successfully views student list 2. Teacher successfully views the detailed information of a student
Related Requirement:	-

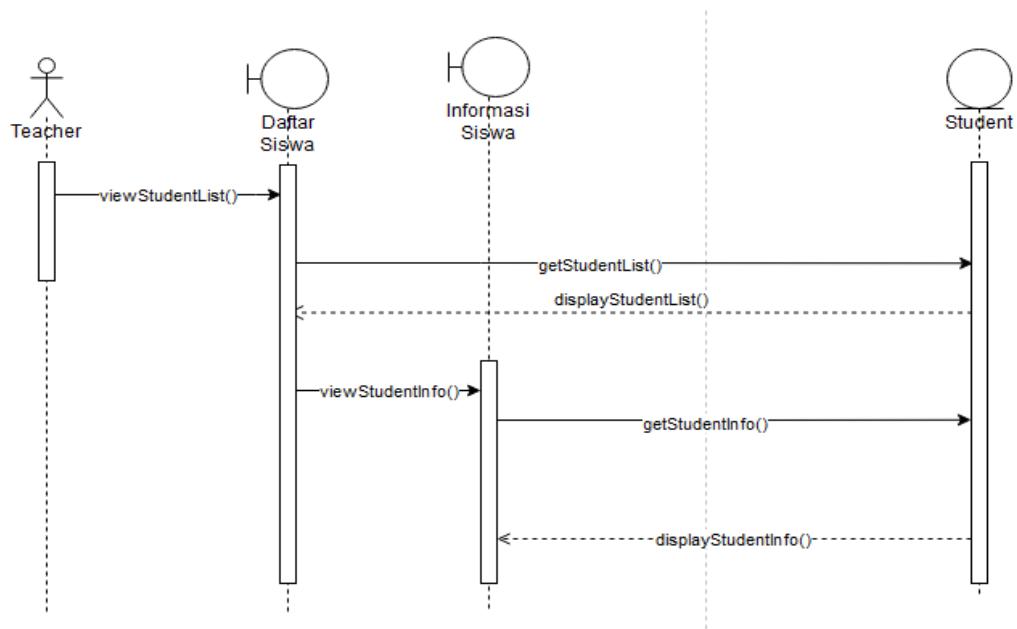


Figure 3.33: System Sequence Diagram of View Student Info

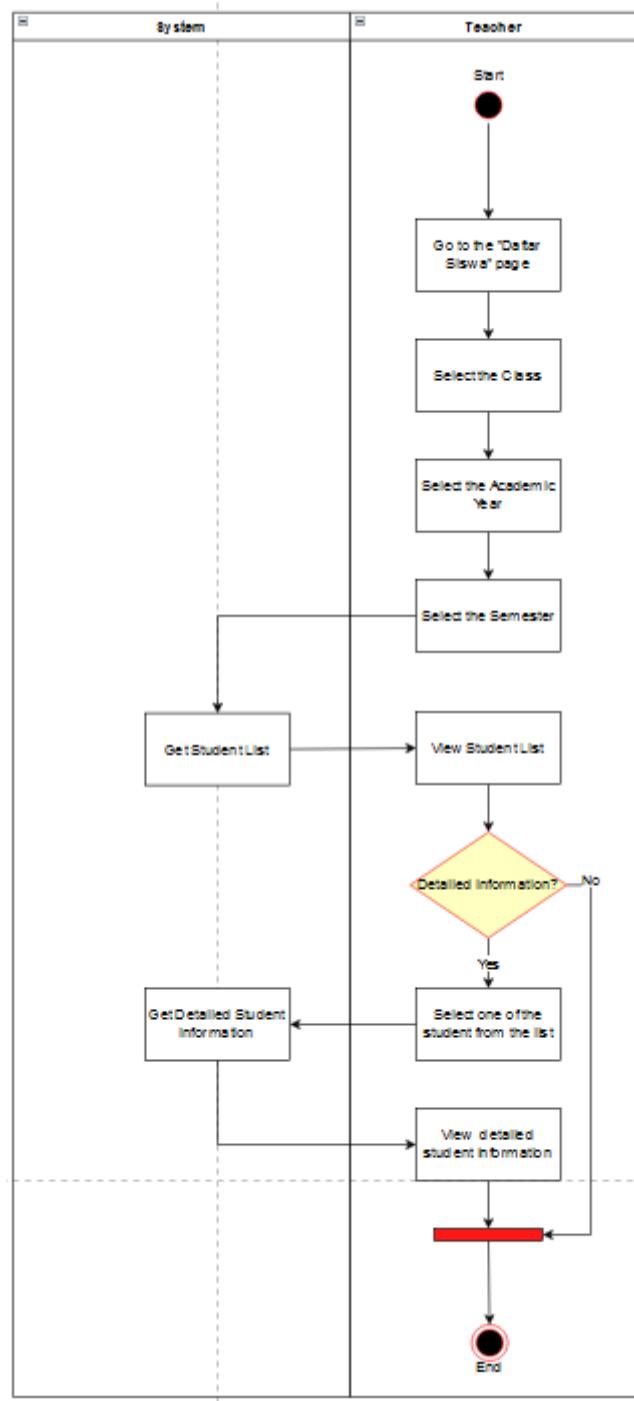


Figure 3.34: Activity Diagram of View Student Info

3.2.5.8

UC017: Use Case View Teacher Info

Table 3.17: Use Case Description for View Teacher Info

Use Case ID:	UC-017
Use Case Name:	View Teacher Info
Actors:	Student
Description:	This use case is used for student to view list of students and their detailed information
Pre-conditions:	1. Logged in to the system
Normal Flow:	<ol style="list-style-type: none"> 1. Go to the “Daftar Guru” page 2. The system displays teacher list 3. If the student wants to view detailed information of the teacher performs AF1
Alternative Flow:	<ol style="list-style-type: none"> 1. View Teacher Detailed Info <ol style="list-style-type: none"> 1.1 Select one of the teachers from the list 1.2 The system displays the detailed information about the selected teacher
Exception:	-
Post-condition s:	<ol style="list-style-type: none"> 1. Student successfully views teacher list 2. Student successfully views the detailed information of a teacher
Related Requirement:	-

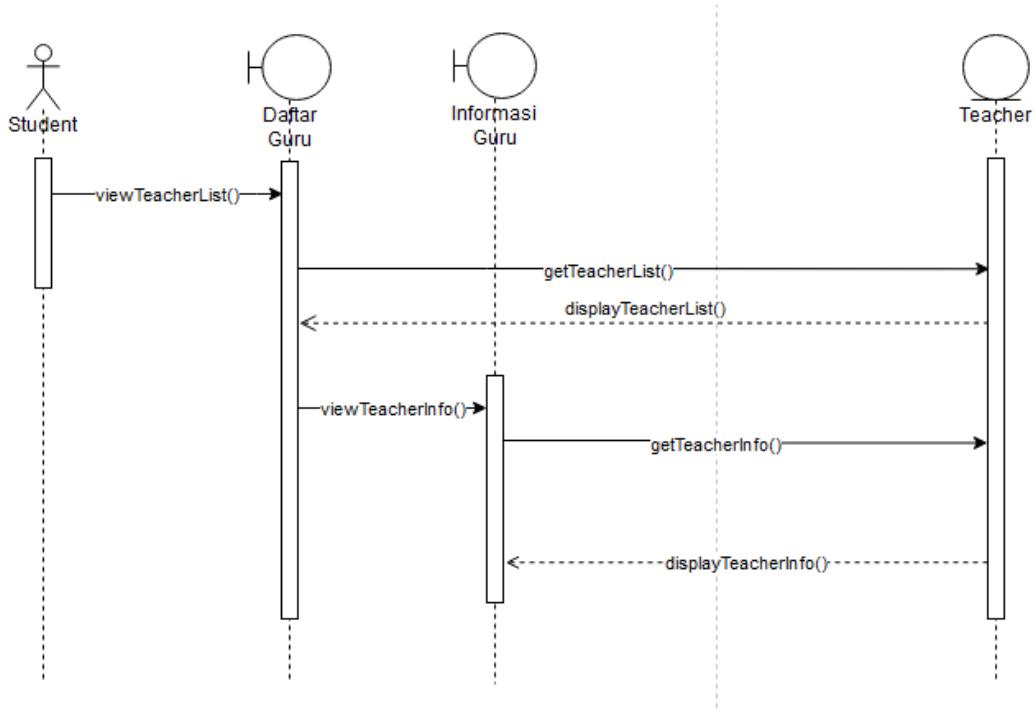


Figure 3.35: System Sequence Diagram of View Teacher Info

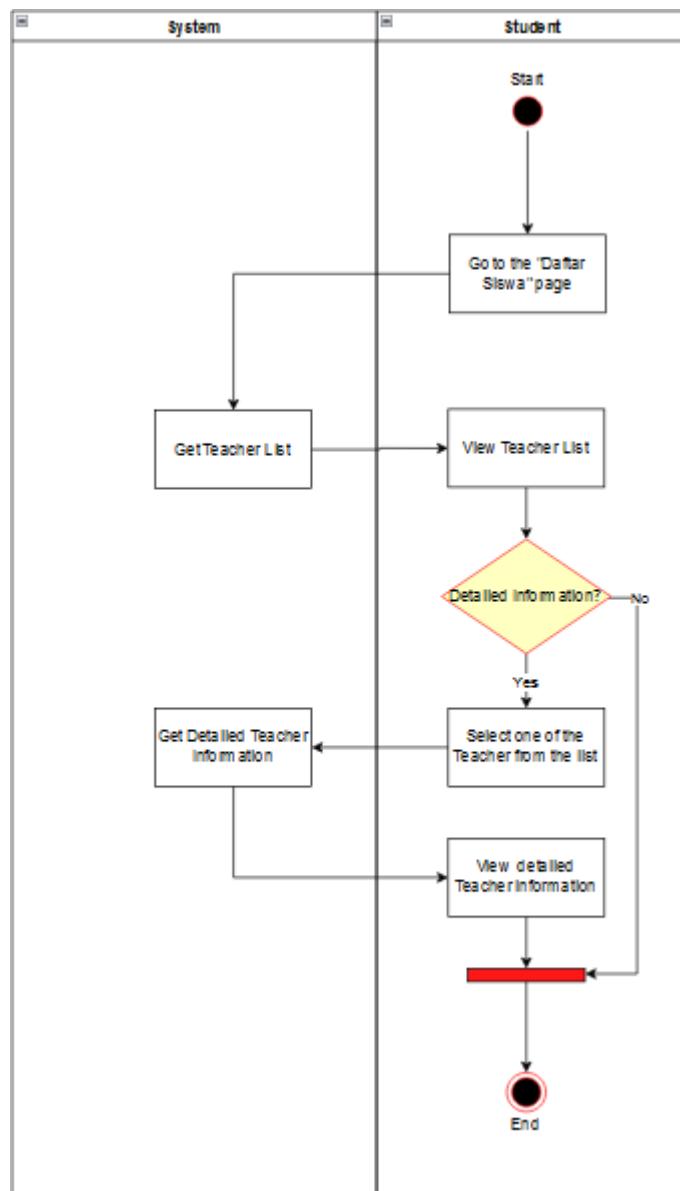


Figure 3.36: Activity Diagram of View Teacher Info

3.3 Performance Requirements

- 1) Workload – The system should be able to support all of the students, teachers and admin of SMAS Muhammadiyah 1
- 2) Scalability – The system should be able to scale up based on the demand and workload
- 3) Response Time – The system should be able to load fast and with minimum delay
- 4) Safety – The system should be able to protect the data during any action that requires the data to be transferred.

3.4 Design Constraints

The system should be accessible with any web browser using internet connection

3.5 Software System Attributes

- 1) Availability – The user should be able to access the system at any moment thus, the system must be available all day every day.
- 2) Security – The system should only allow authorized user from SMAS Muhammadiyah 1 to access the system. The system must provide protection to all of the data exist in the system.
- 3) Usability – The user should be able to explore the system with ease after inspecting the system a little bit. The system must be simple to use and comprehend.

- 4) Portability – The system should work with any web browser. The user shall be able to access the system from their preferred web browser

3.6 Other Requirements

No other requirements for this project

Appendix B Software Design Document



SCSJ3323: Software Design and Architecture

Software Design Document

Student Academic Management System

Version 1.0

23 June 2022

Department and Faculty

Prepared by: Muhammad Darlen Sava

Revision Page

a. Overview

This is the first draft of the Software Requirements Specification (SRS) for Student Academic Management System (SAMS).

b. Target Audience

Student, Teacher and Admin of SMAS Muhammadiyah 1

c. Project Team Members

Muhammad Darlen Sava

d. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Muhammad Darlen Sava	Initial Draft	23 June 2022

Note:

This template is an annotated outline for a software design document adapted from the IEEE Recommended Practice for Software Design Descriptions. The IEEE Recommended Practice for Software Design Descriptions have been reduced in order to simplify this assignment while still retaining the main components and providing a general idea of a project definition report. Please refer to IEEE Std 1016•1998 1 for the full IEEE Recommended Practice for Software Design Descriptions. Examples of models are from Satzinger (2011). Compiled by Shahliza Abdul Halim, PhD and checked by Shahida Sulaiman, PhD on 2 May 2016.

Table of Contents

1	Introduction		1
1.1	Purpose		1
1.2	Scope		1
1.3	Definitions, Acronyms and Abbreviations		1
1.4	Reference Materials		1
1.5	System Overview		2
2	System Architectural Design		3
2.1	Architectural Style and Rationale		3
2.2	Component Model		4
2.3	Use Case Diagram		5
3	Detailed Description of Modules		7
3.1	Complete Package Diagram		7
3.2	Modules Detailed Descriptions		8
3.2.1	Module Authentication		8
	3.2.1.1	Package Diagram	8
	3.2.1.2	Class Diagram	9
	3.2.1.3	Sequence Diagrams	9
3.2.2	Module Report		10
	3.2.2.1	Package Diagram	11
	3.2.2.2	Class Diagram	12
	3.2.2.3	Sequence Diagrams	13
3.2.3	Module Attendance		15
	3.2.3.1	Package Diagram	16
	3.2.3.2	Class Diagram	17
	3.2.3.3	Sequence Diagrams	17

3.2.4	Module Schedule		19
	3.2.4.1	Package Diagram	20
	3.2.4.2	Class Diagram	21
	3.2.4.3	Sequence Diagrams	21
3.2.5	Module Account		23
	3.2.5.1	Package Diagram	24
	3.2.5.2	Class Diagram	25
	3.2.5.3	Sequence Diagrams	26
4	Data Design		32
4.1	Data Description		32
4.2	Data Dictionary		35
5	User Interface Design		43
5.1	Overview of User Interface		43
5.2	Screen Images		45
6	Requirements Matrix		63
7	Appendices		65
Appendices (if any)			

1. Introduction

1.1 Purpose

This SDD describes the design for the Student Academic Management System which is a system dedicated to SMAS Muhammadiyah 1 Banda Aceh. This SDD is required to determine the design requirements of the stakeholder, it also acts as a reference point for the development of the project.

1.2 Scope

The current system of SMAS Muhammadiyah is still involving manual task execution with a decentralized database. So, this project only focused on the conversion of manually managing student data to a website that stores all of the student's data, which can be accessed by teachers, students, and their parents. As such, SMAS Muhammadiyah will only provide the student's data throughout the development of the project. The users in this system will be the teachers, students/parents, and admins of SMAS Muhammadiyah. The goal of this project is to create an effective solution to enable the institution to convert its manual student management system into a computerized system.

1.3 Definitions, Acronyms and Abbreviation

1. SAMS - Student Academic Management System
2. SRS - Software Requirements Specification
3. SDD - Software Design Document

1.4 References

My.utm.my. (n.d.). Retrieved May 16, 2022, from <https://my.utm.my/home>

Applikasi Rapor online - solusi. Sekawan Media. (2022, May 16). Retrieved May 16, 2022, from <https://www.sekawanmedia.co.id/solusi/aplikasi-rapor-online/>

Miller, J., Kostogriz, A., & Gearon, M. (Eds.). (2009). Culturally and linguistically diverse classrooms: New dilemmas for teachers. Bristol, UK: Multilingual Matters.

Dinustek. (n.d.). *Solusi Rapor Sekolah online.* evaluation. Retrieved May 16, 2022, from <https://evaluation.id/>

1.5 Overview

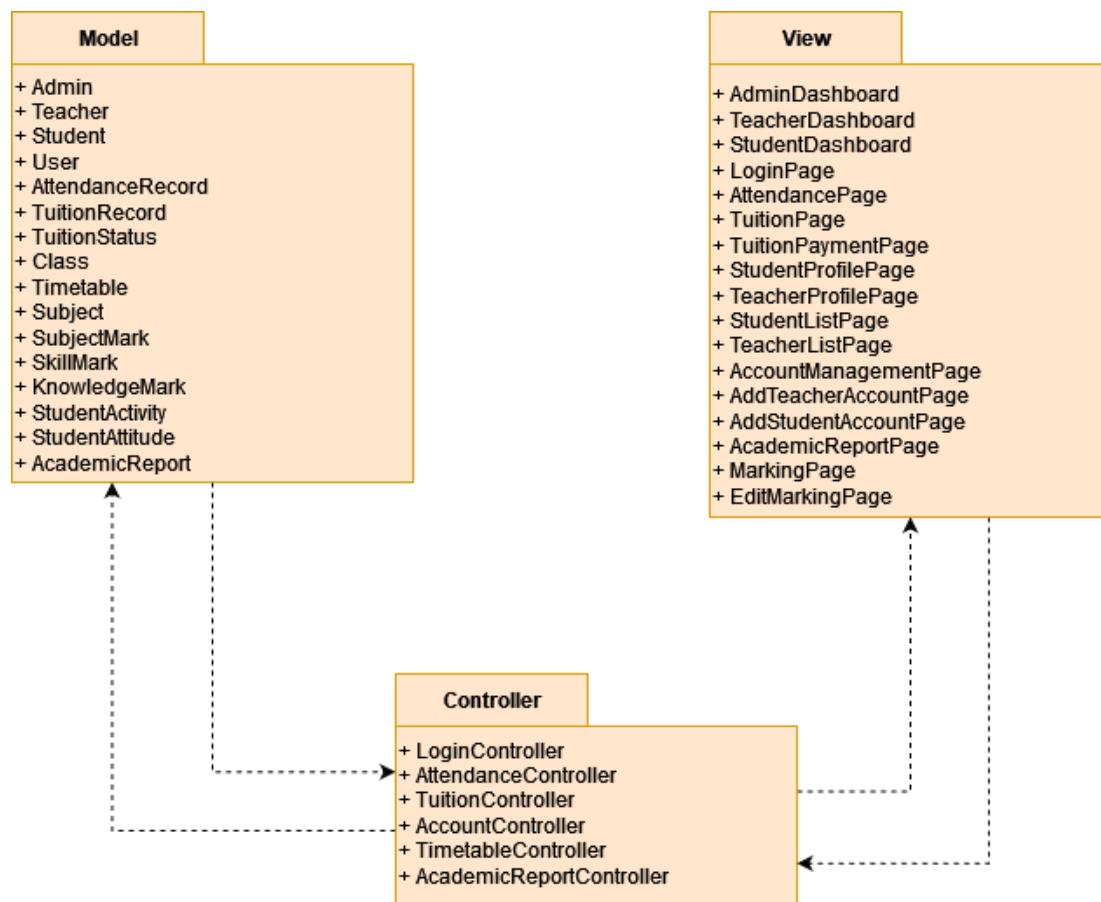
This SDD is divided into seven chapters. The first chapter is the introduction which consists of the purpose and scope of the project. It also includes all the acronyms/abbreviation descriptions and references that exist in this SDD and a short overview of the whole project. For the second chapter, The SDD focused on the system architectural design of the project. This chapter delves into the chosen architecture style and the reasoning behind it. In the third chapter, the detailed description of modules for the project are stated. It consists of packages diagrams for each module, class and sequence diagram for each use cases. Chapter 4 of this SDD includes the design of data which consist of Entity Relationship Diagram and data dictionary. The fifth chapter shows the mockup user interface design for the system. In the final chapter it is stated which system component satisfy the SRS using the requirement matrix table.

2. System Architectural Design

2.1 Architecture Style and Rationale

The architecture used for this proposed system is MVC or Model-View-Controller. In this model the code structure separated into different part which are mode, view and controller. The model in MVC is used as a bridge to transfer data between view and controller. As the name suggest the view in the MVC is for the user interface. And the controller is used for calling the functions.

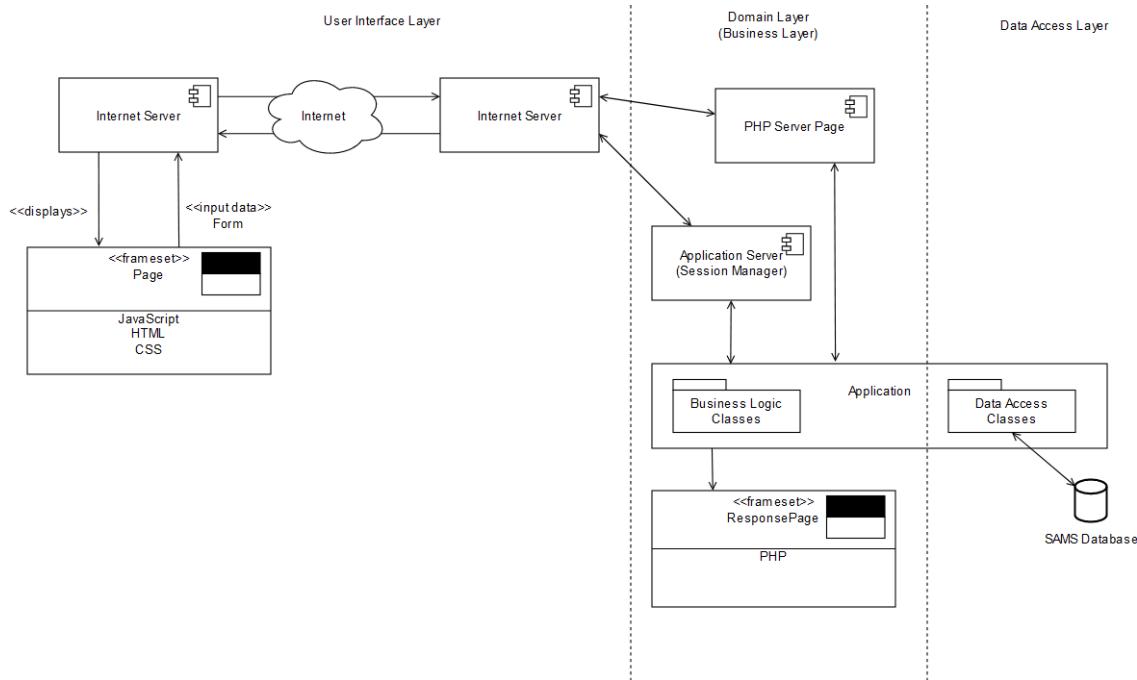
One of the most important benefits of using MVC in a web-based app is how flexible it is which led to faster process when developing. Since its flexible, it means that changes made will not entirely affect other part of the system. Additionally, it has the benefit of helping the developers repeating the same code because model which represents data is separated from the logic in the controller. All of these benefits will greatly help developing SAMS. Following is the MVC diagram of SAMS.



AM000 Architecture Model of SAMS

2.2 Component Model

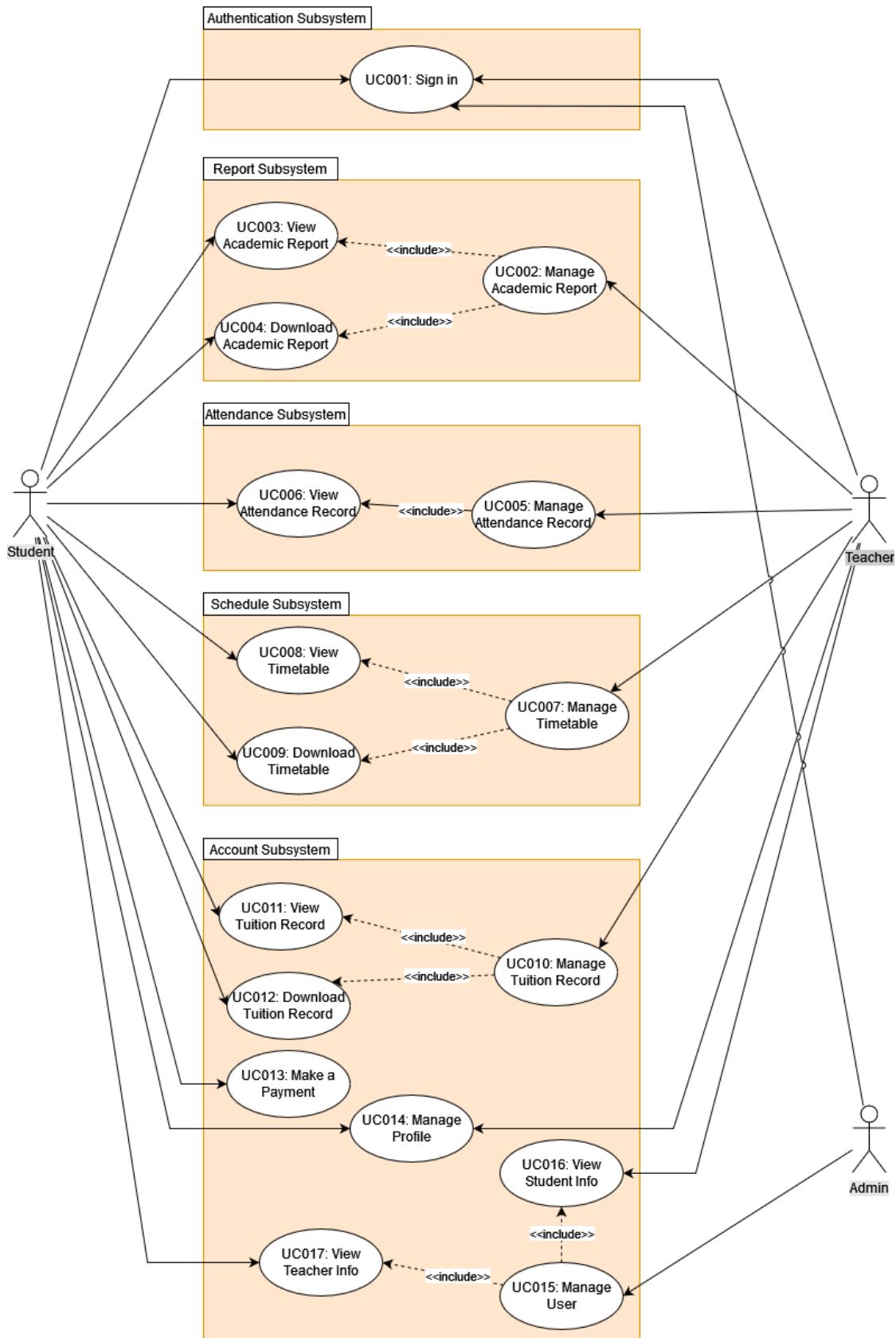
The figure below depicts the component model of Student Academic Management System.



CM000 Component Model of SAMS

The diagram separated into three layers; The user interface layer, domain or business layer and data access layer. A stable internet connection is required to have a seamless experience when using SAMS. JavaScript, HTML and CSS is used for user interface layer. Combined, they make a user interface to display the system that the end-user can also interact with such as taking the input from the user. This input then get send to internet with internet server and end up in the domain layer. The input that the internet server sends then got separated into either PHP server page or Application Server. In application server the system is expected to run since the server provide the settings to run the application. As for the PHP server page, it is used for taking request from the user and then send it to the programs inside the application. PHP is used in the frameset or response page and acts to respond to any task that the user performs.

2.3 Use Case Diagram

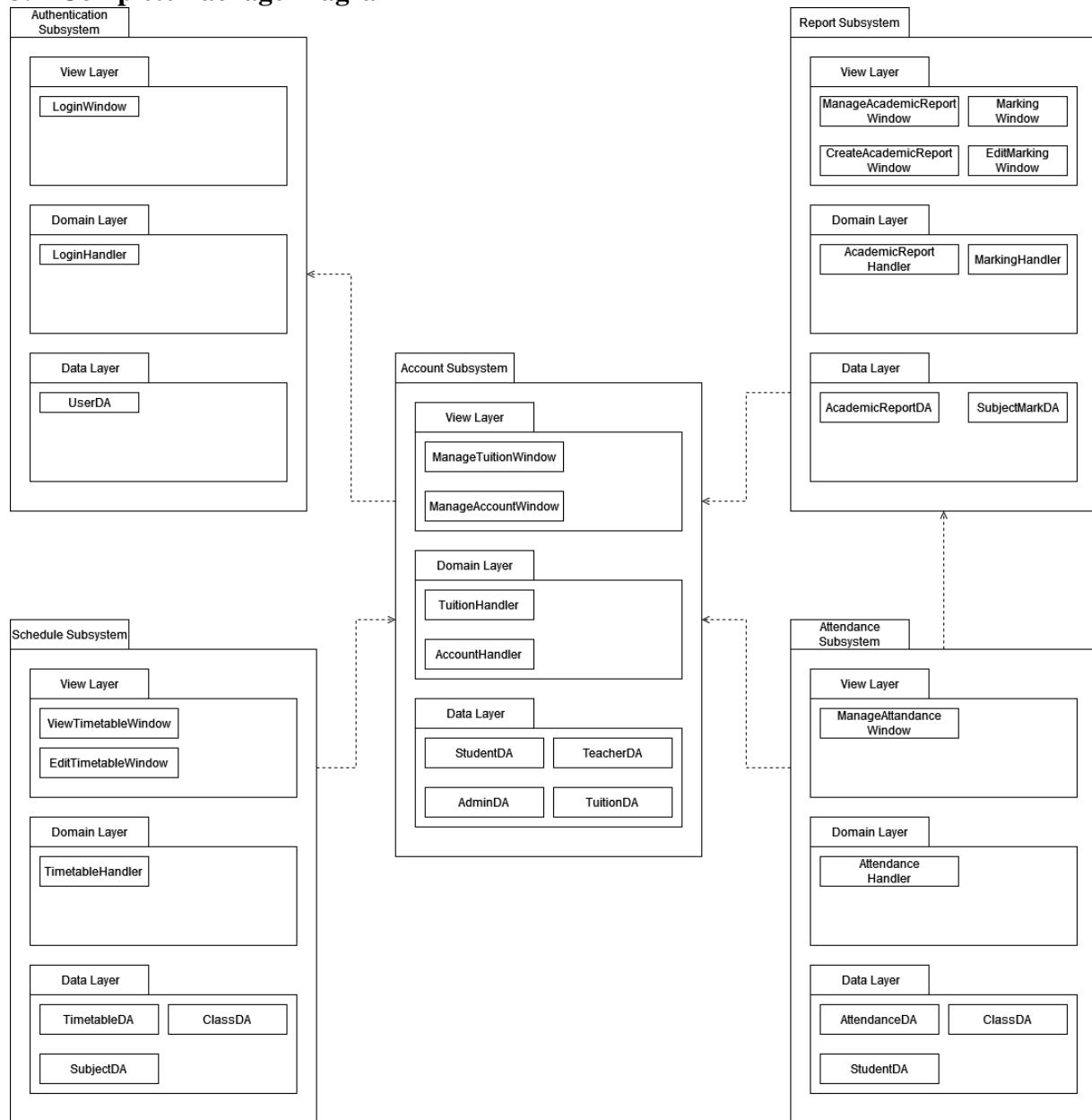


UCD000 Use Case Diagram of SAMS

The figure above shows all of the functions that exist in the proposed system. Besides the use case and subsystems there are three users that directly involved in the system. The arrow shows users can only access certain functions. There are 17 use cases that divided into 5 subsystems. There are multiple use cases that included into another use case, this is because certain user like student can only access some of the original functions. The description on what the actors and use case's role in the system will discussed in the next subchapter.

3. Detailed Description of Components

3.1 Complete Package Diagram



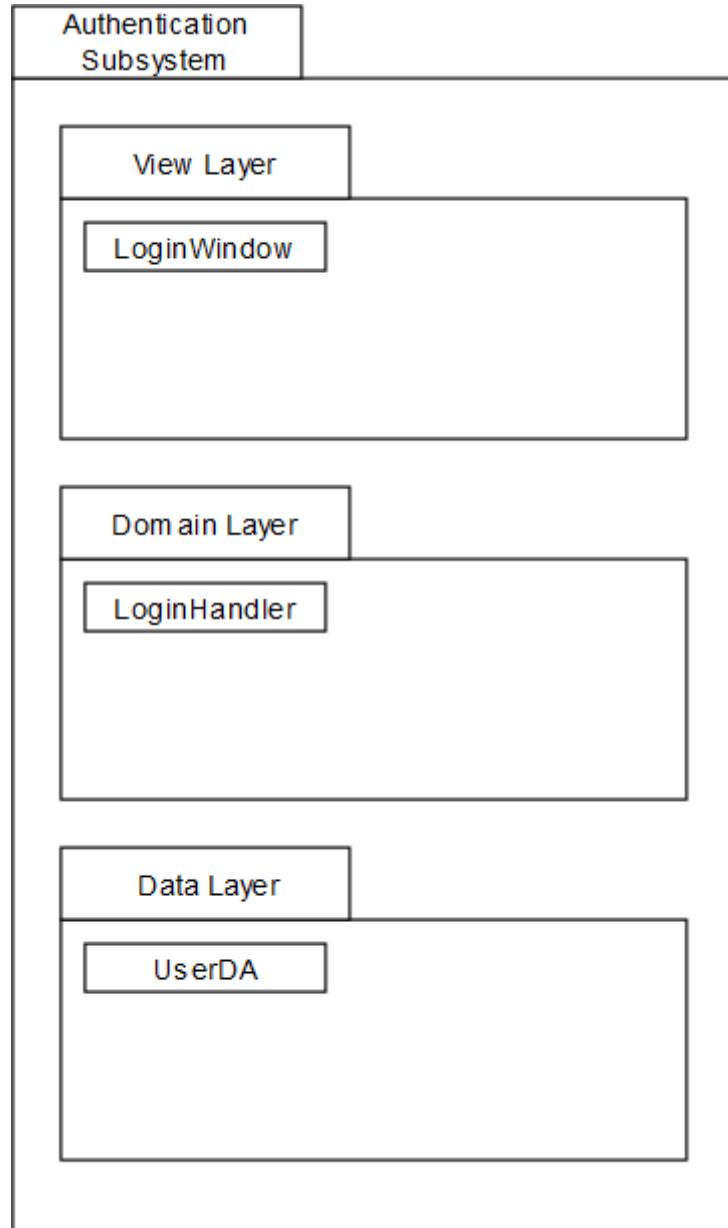
CPD000 Complete Package Diagram of SAMS

3.2 Detailed Description

Below is the detailed description of SAMS for each of the subsystems.

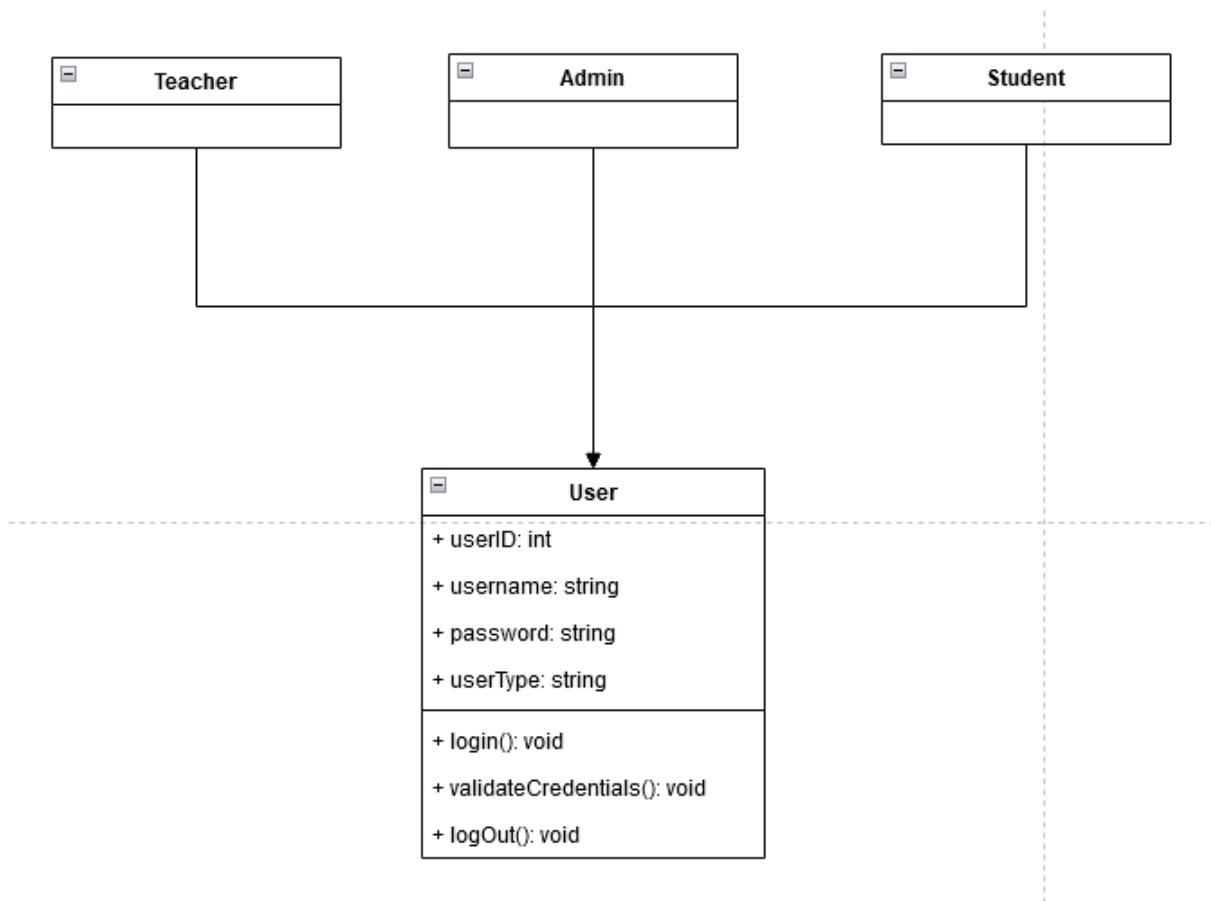
3.2.1 Subsystem Authentication

3.2.1.1 P001: Package Authentication



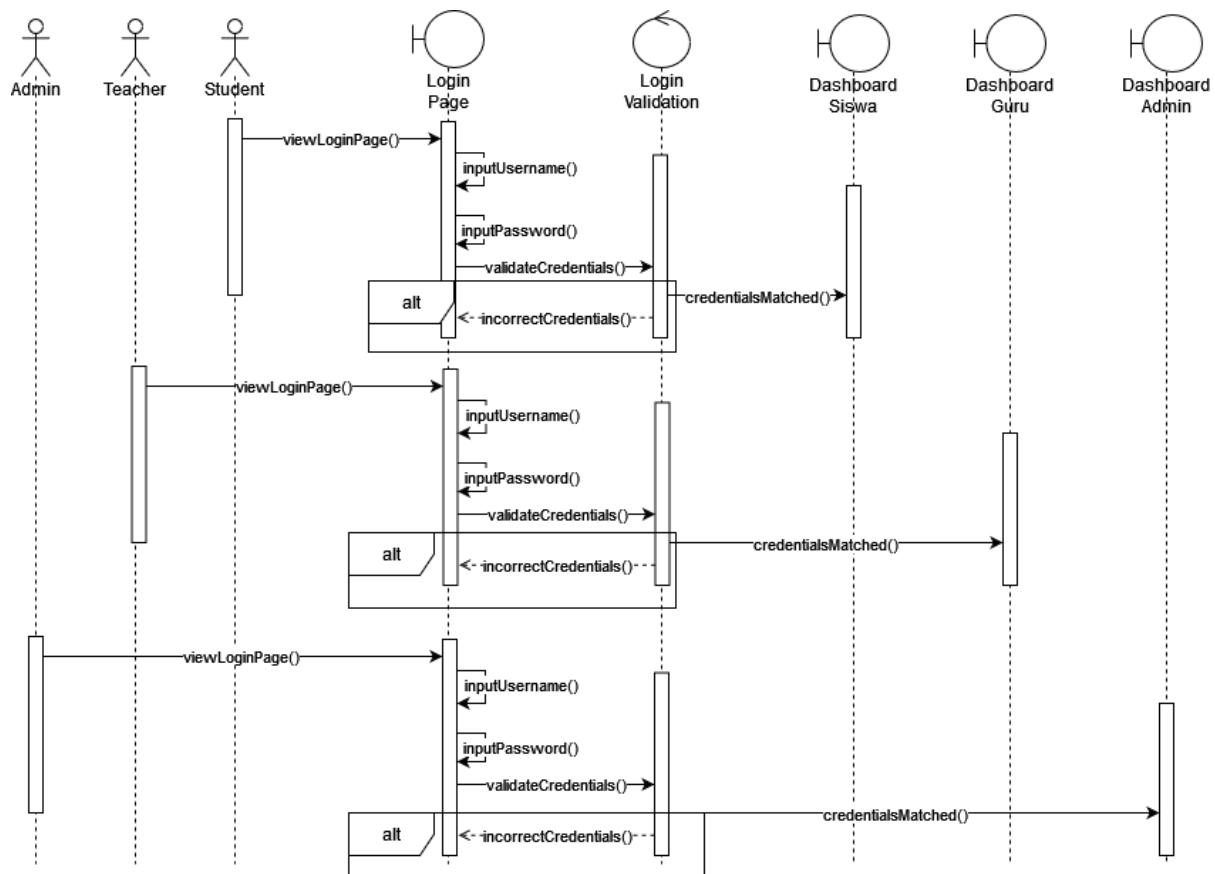
PD001 Package Diagram of Authentication Subsystem

3.2.1.2 Class Diagram



CD001 Class diagram for Authentication Subsystem

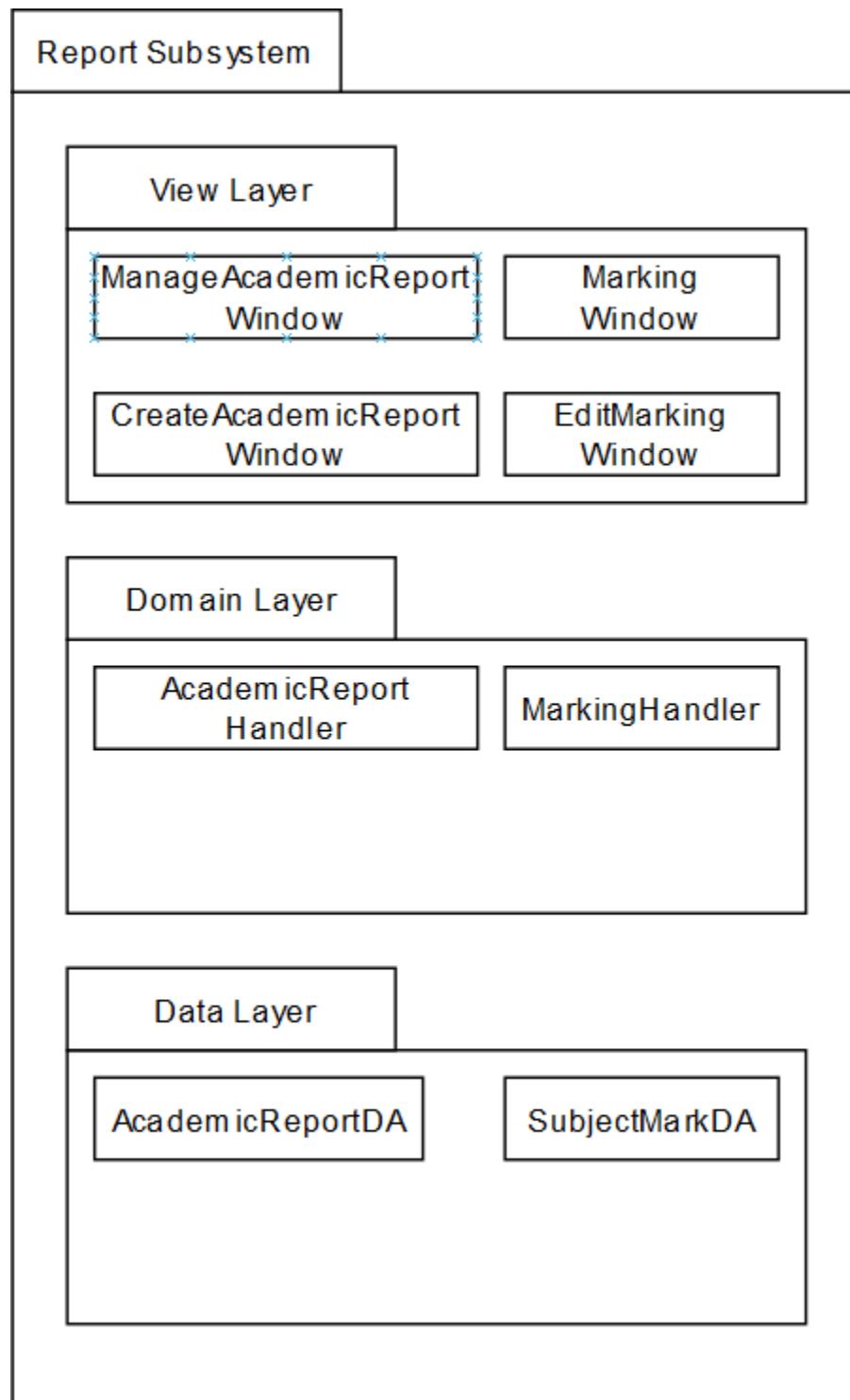
3.2.1.3 Sequence Diagrams



SD001 Sequence Diagram of Sign In

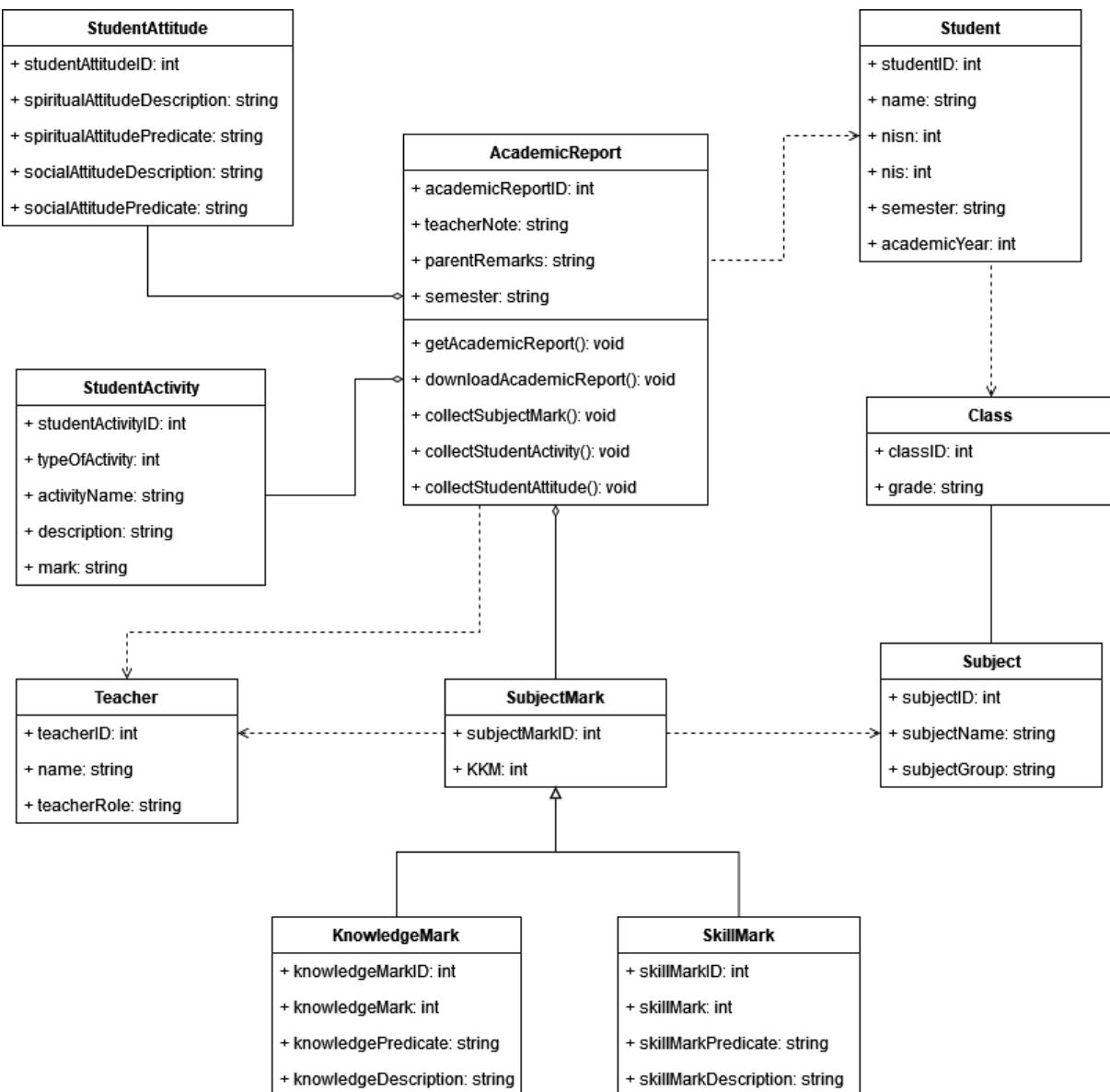
3.2.2 Subsystem Report

3.2.2.1 P002: Package Report



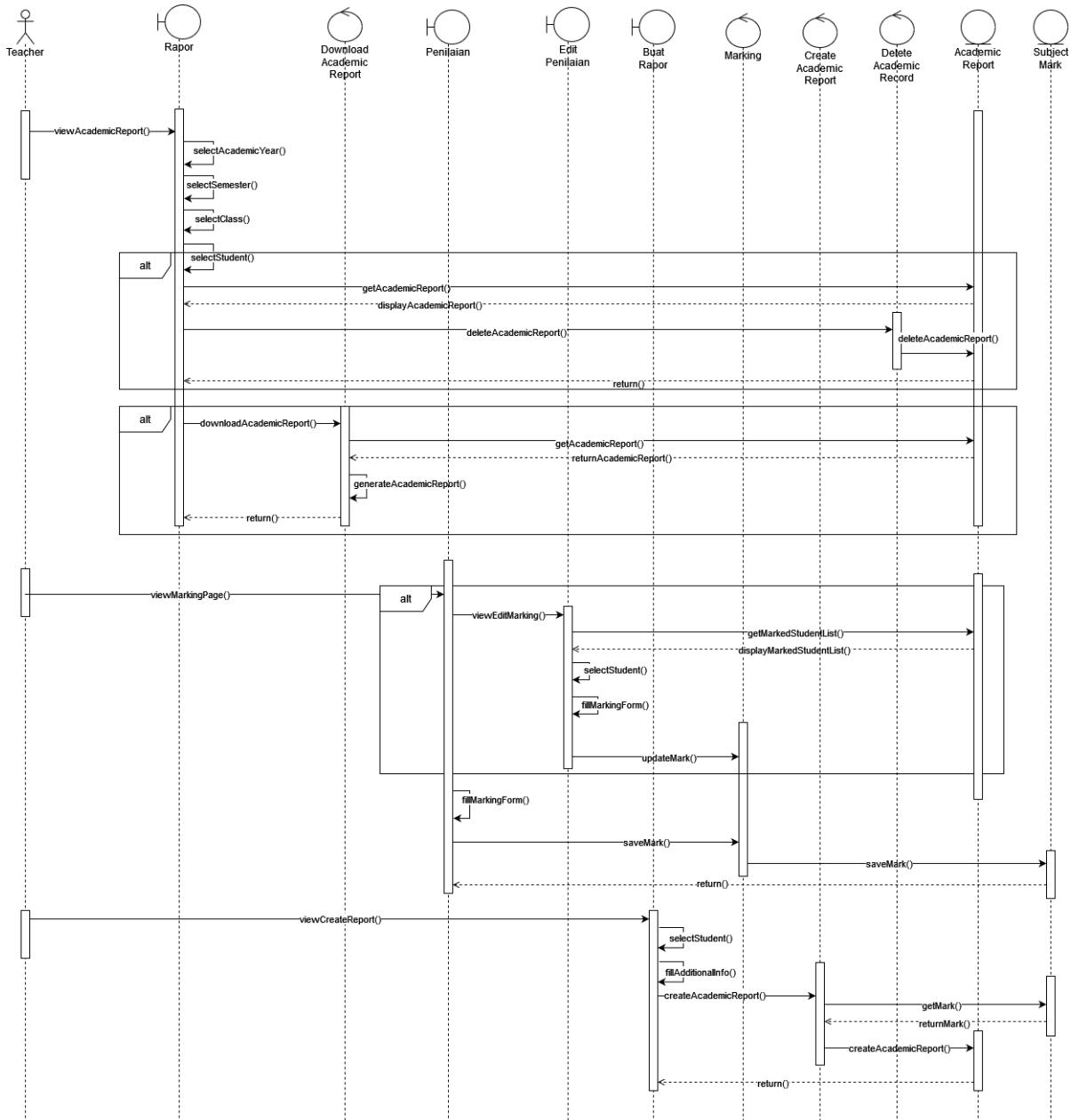
PD002 Package Diagram of Report Subsystem

3.2.2.2 Class Diagram

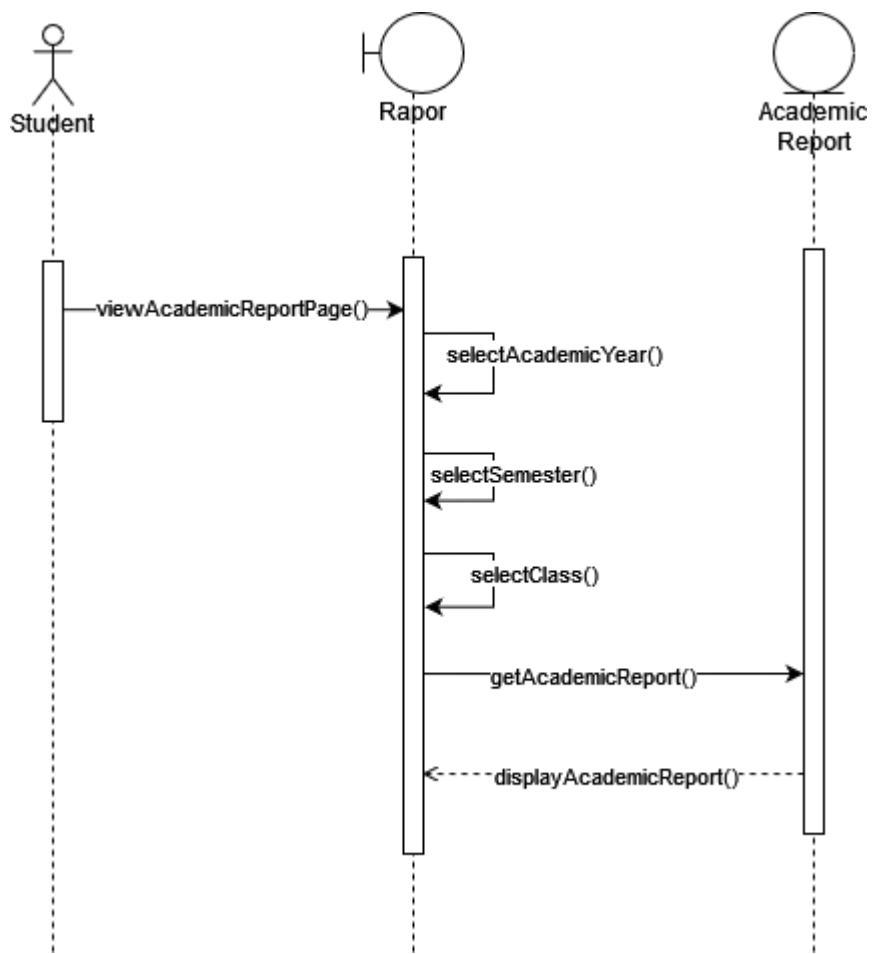


CD002 Class Diagram for Report Subsystem

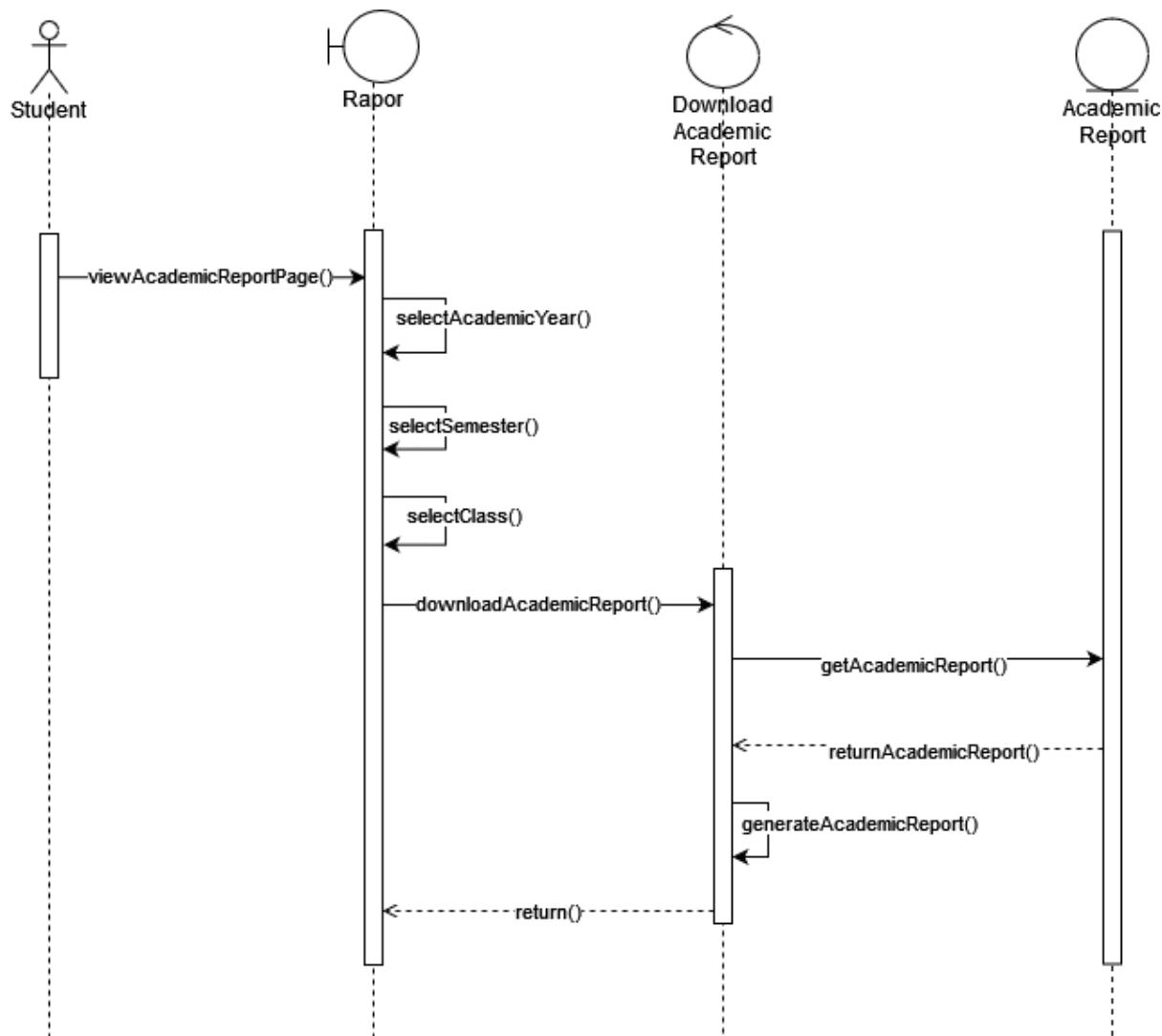
3.2.2.3 Sequence Diagrams



SD002 Sequence Diagram of Manage Academic Report



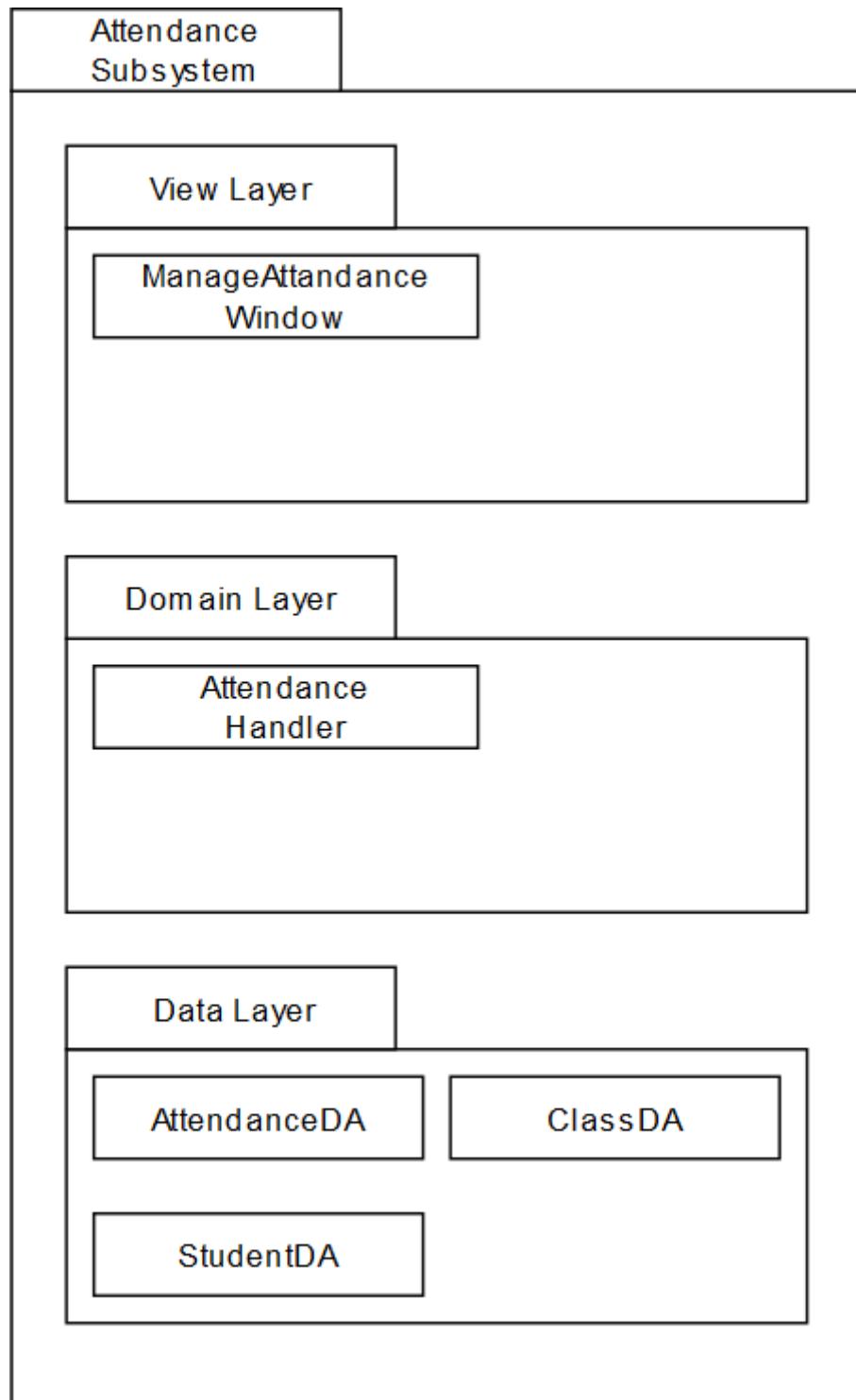
SD003 Sequence Diagram of View Academic Report



SD004 Sequence Diagram of Download Academic Report

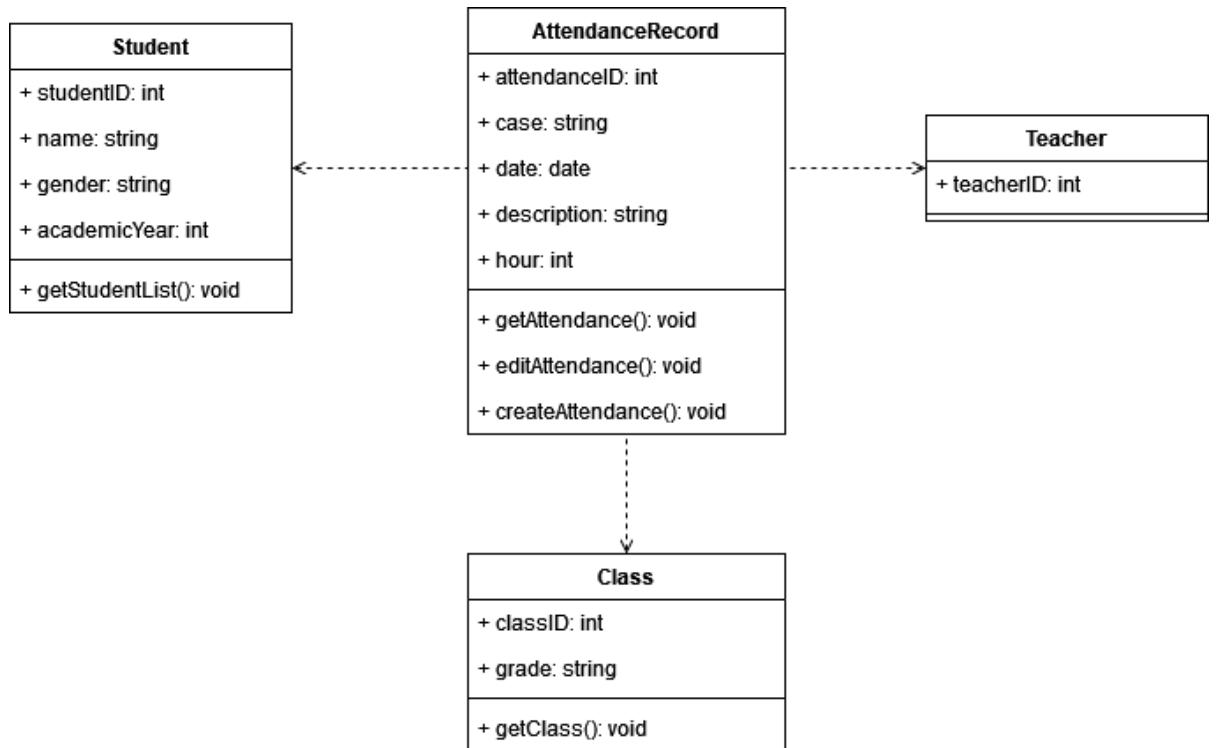
3.2.3 Subsystem Attendance

3.2.3.1 P003: Package Attendance



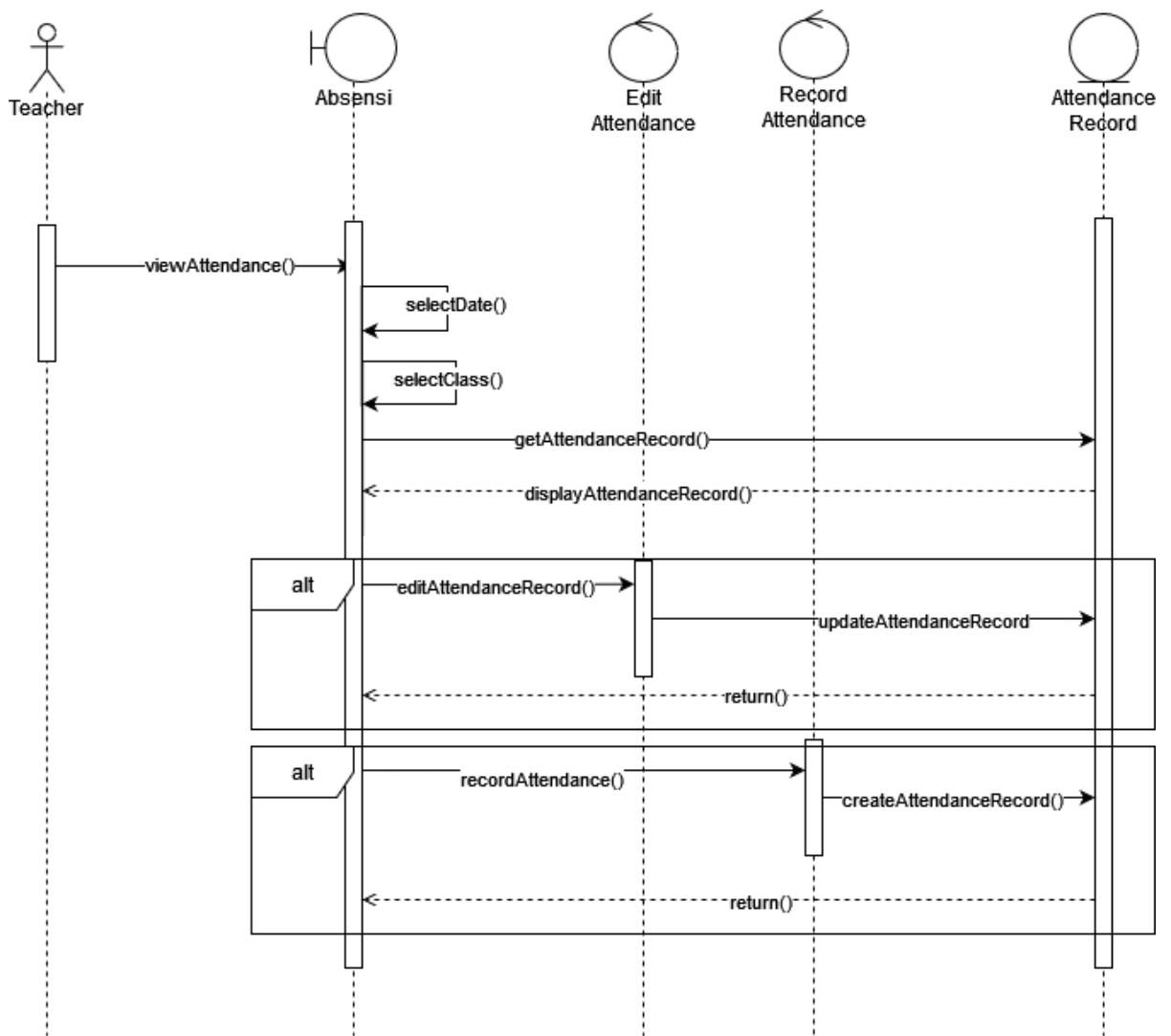
PD003 Package Diagram of Attendance Subsystem

3.2.3.2 Class Diagram

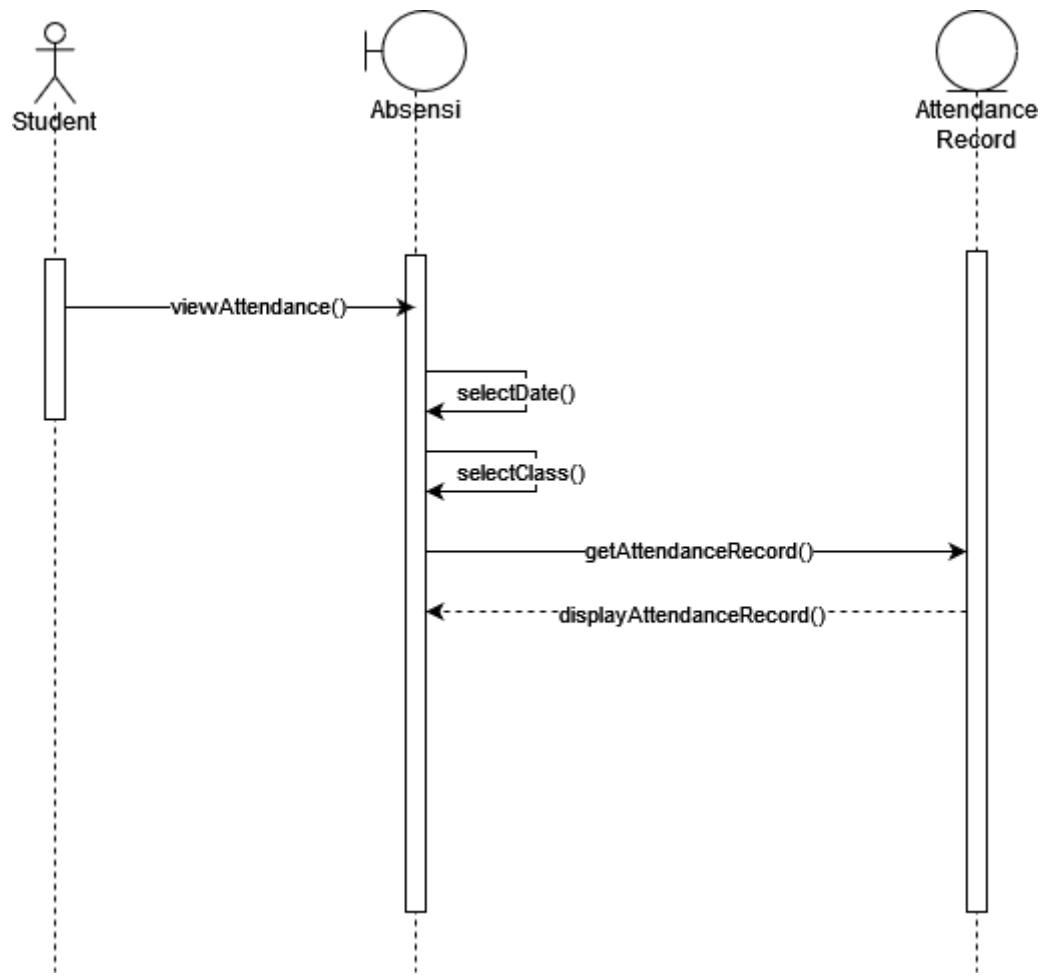


CD003 Class Diagram for Attendance Subsystem

3.2.3.3 Sequence Diagrams



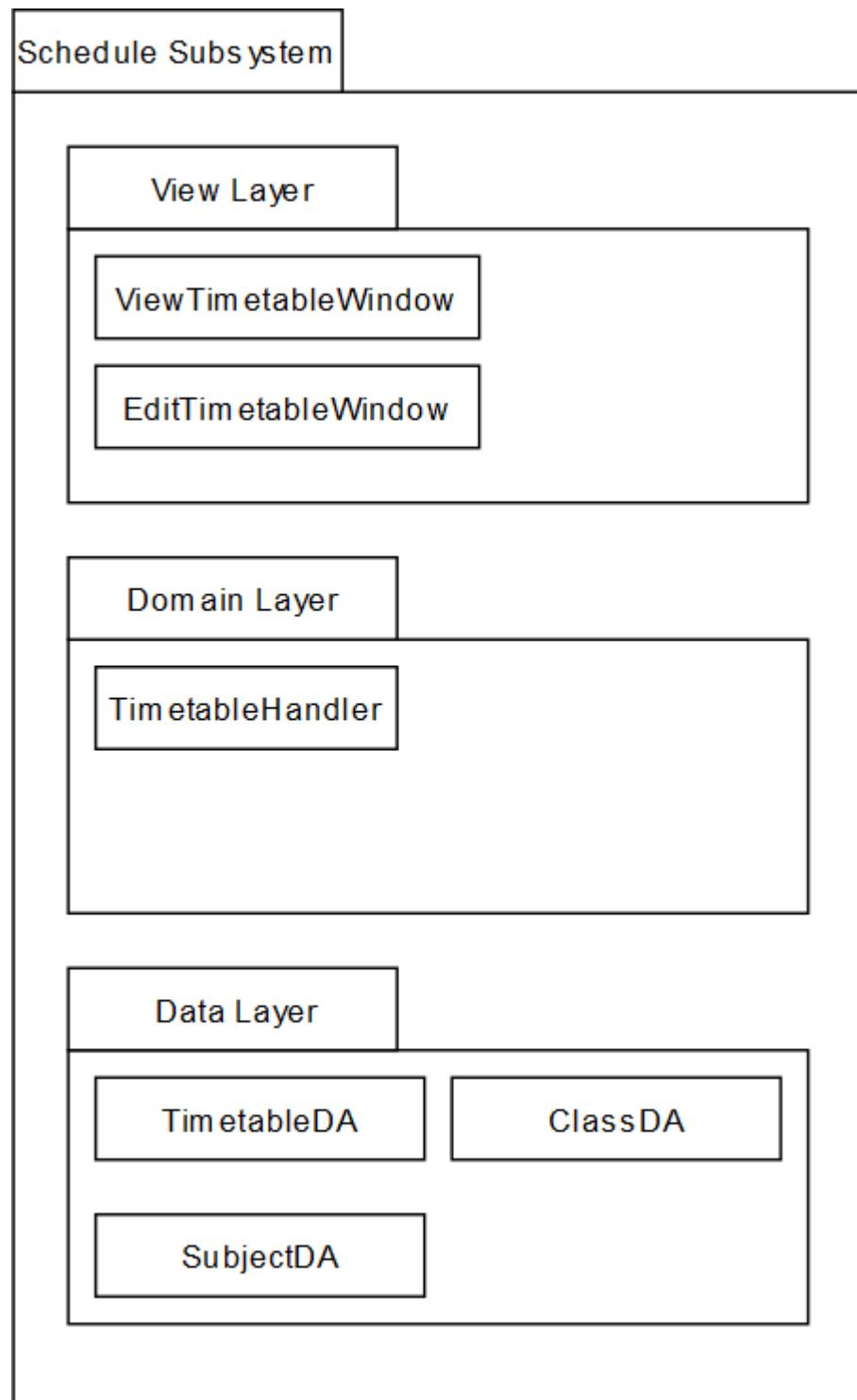
SD005 Sequence Diagram of Manage Attendance Record



SD006 Sequence Diagram of View Attendance Record

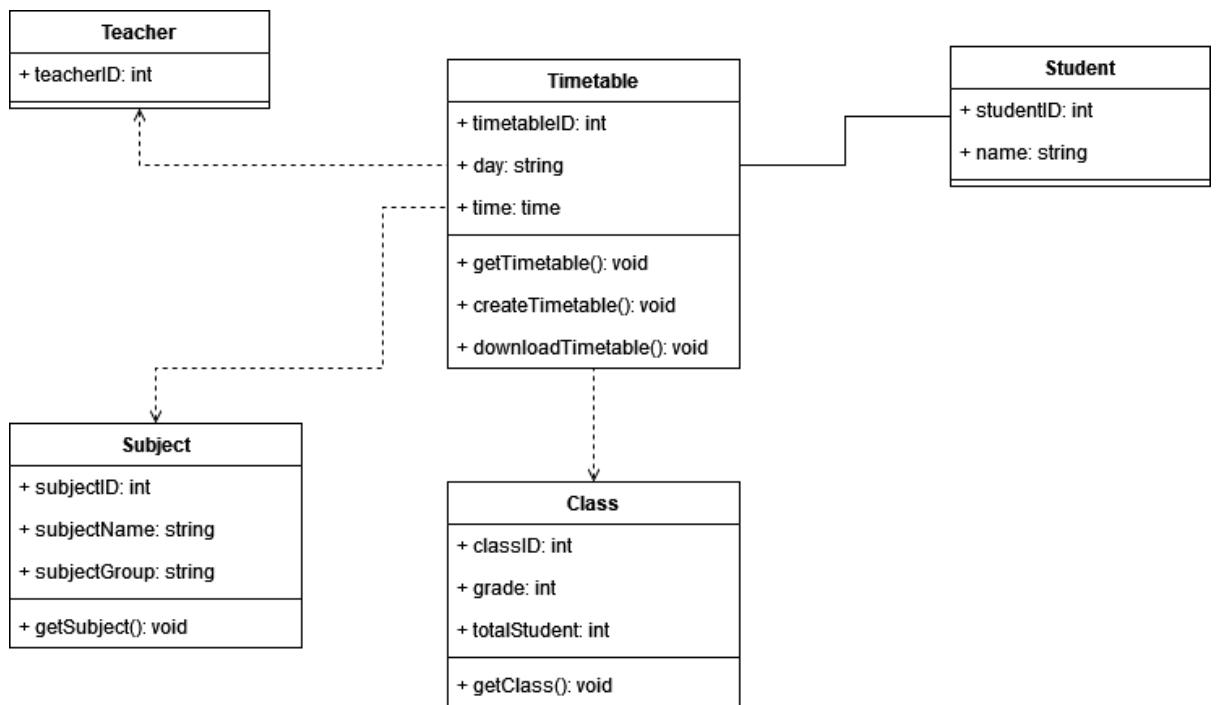
3.2.4 Subsystem Schedule

3.2.4.1 P004: Package Schedule



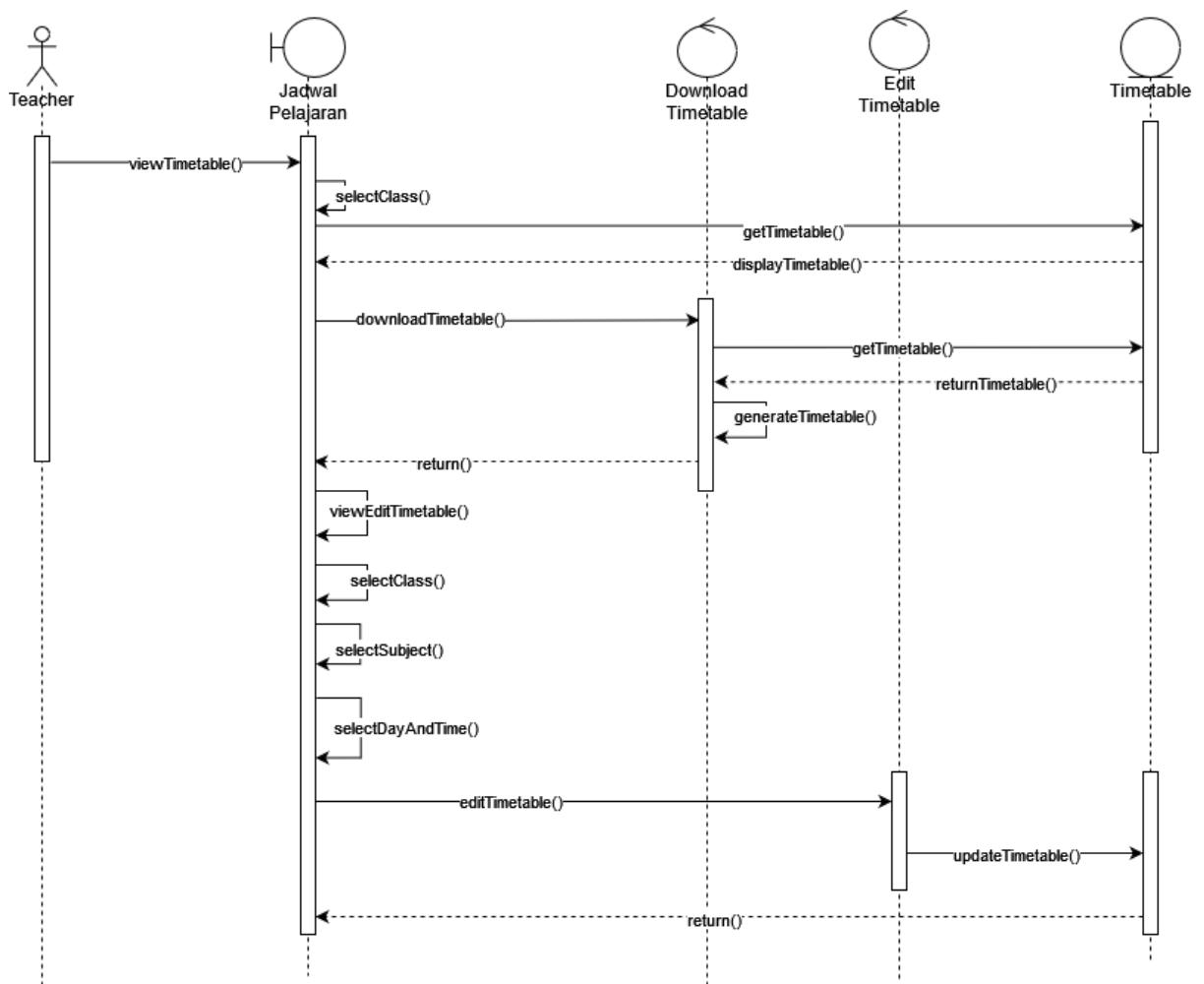
PD004 Package Diagram of Schedule Subsystem

3.2.4.2 Class Diagram

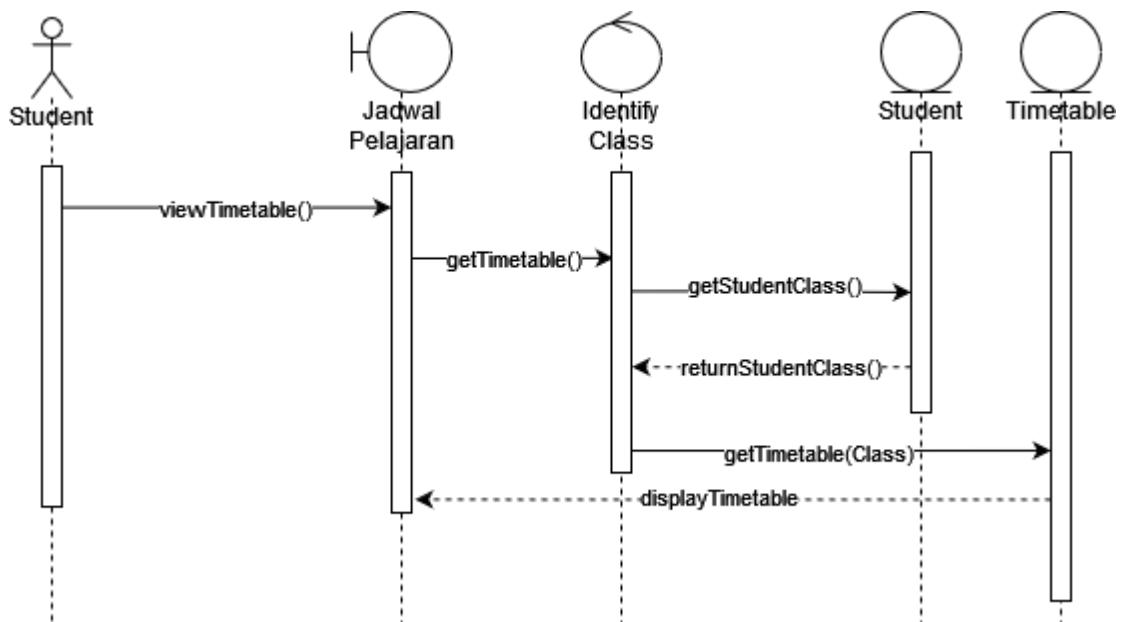


CD004 Class Diagram for Schedule Subsystem

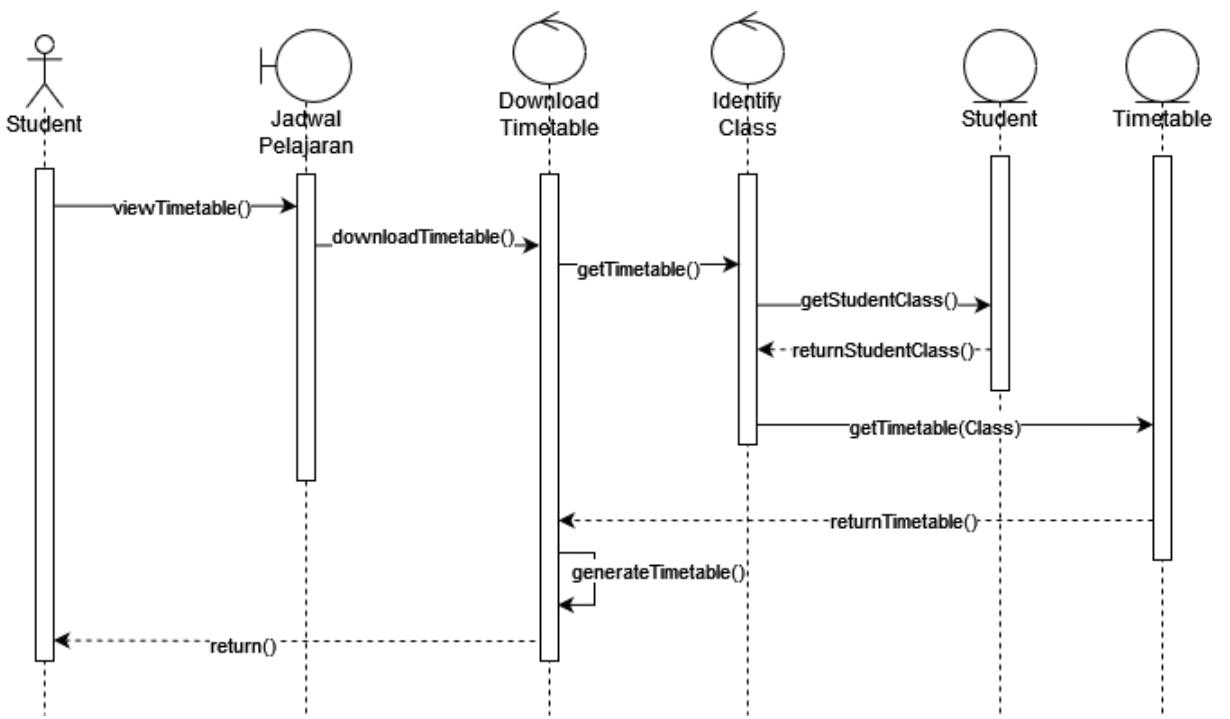
3.2.4.3 Sequence Diagrams



SD007 Sequence Diagram of Manage Timetable



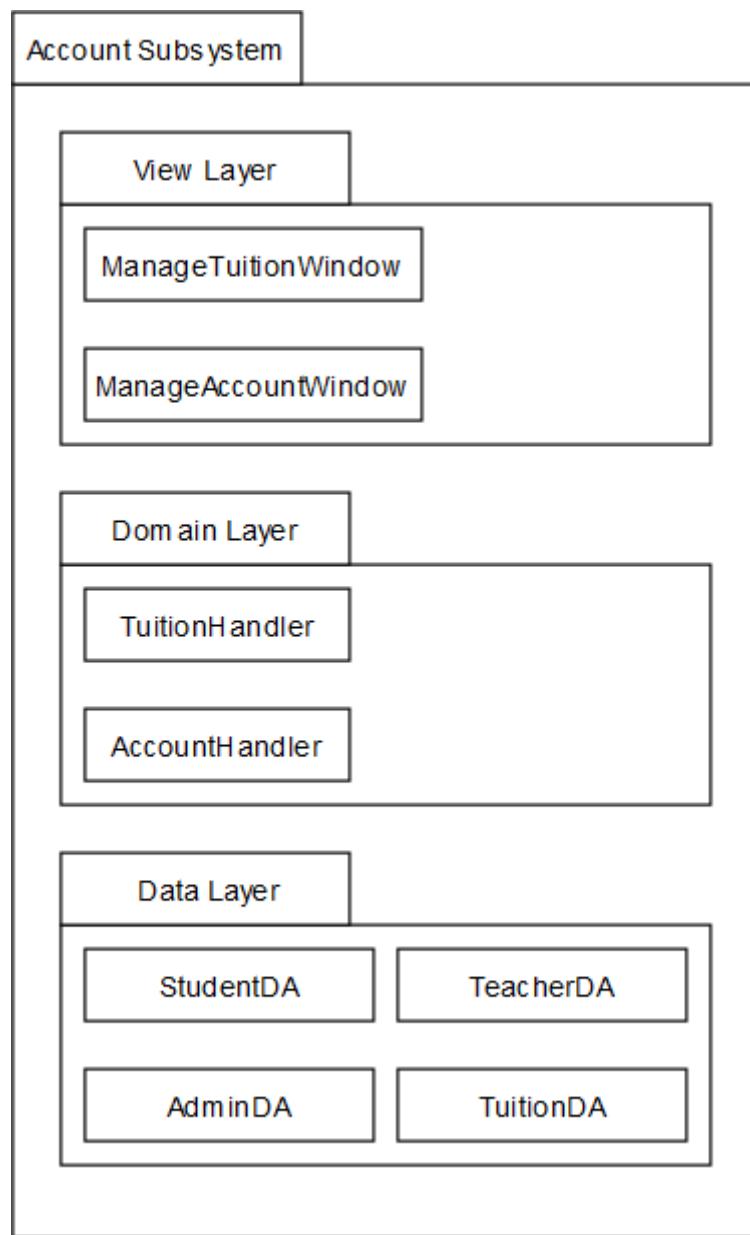
SD008 Sequence Diagram of View Timetable



SD009 Sequence Diagram of Download Timetable

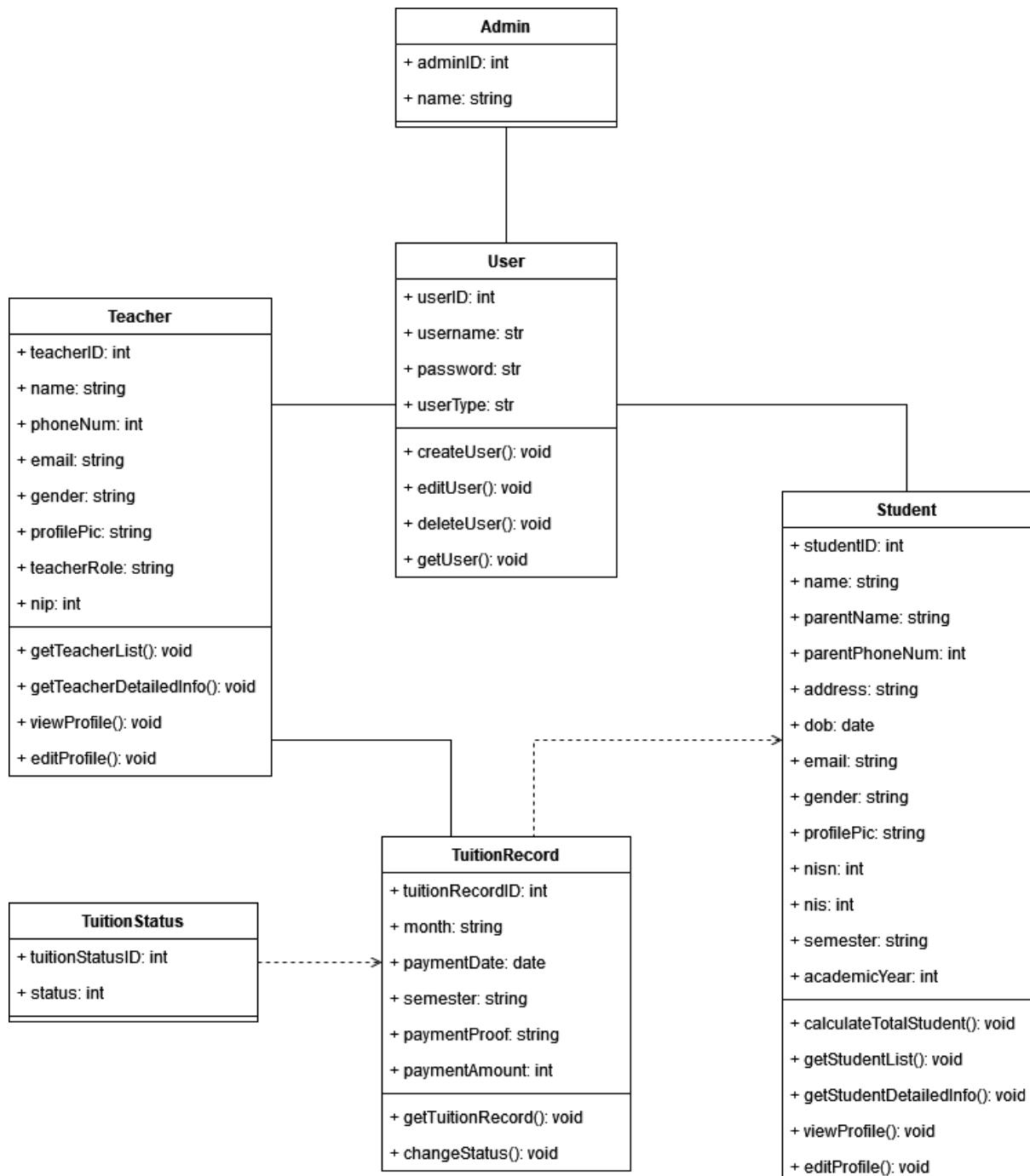
3.2.5 Subsystem Account

3.2.5.1 P005: Package Account



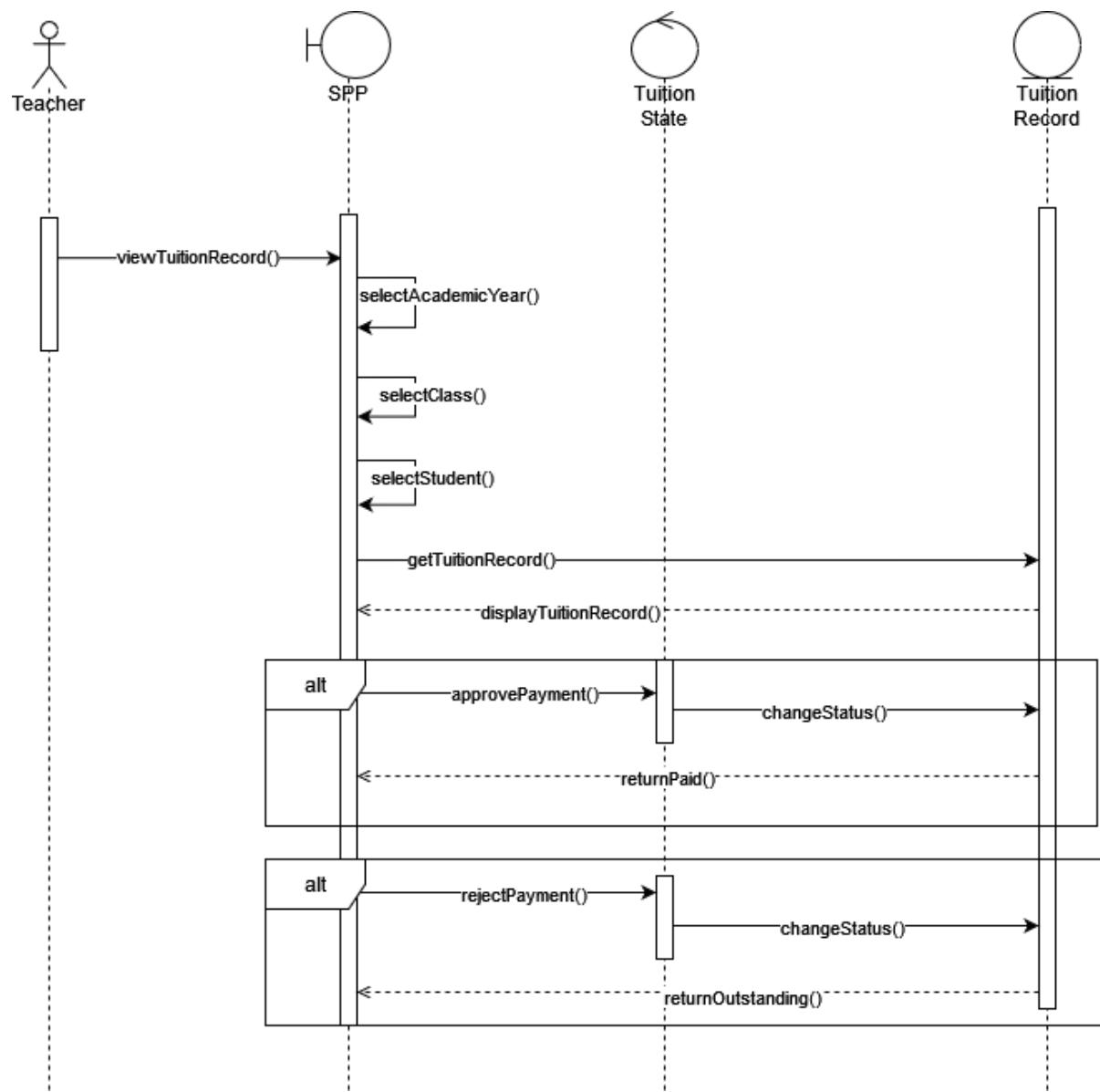
PD005 Package Diagram of Account Subsystem

3.2.5.2 Class Diagram

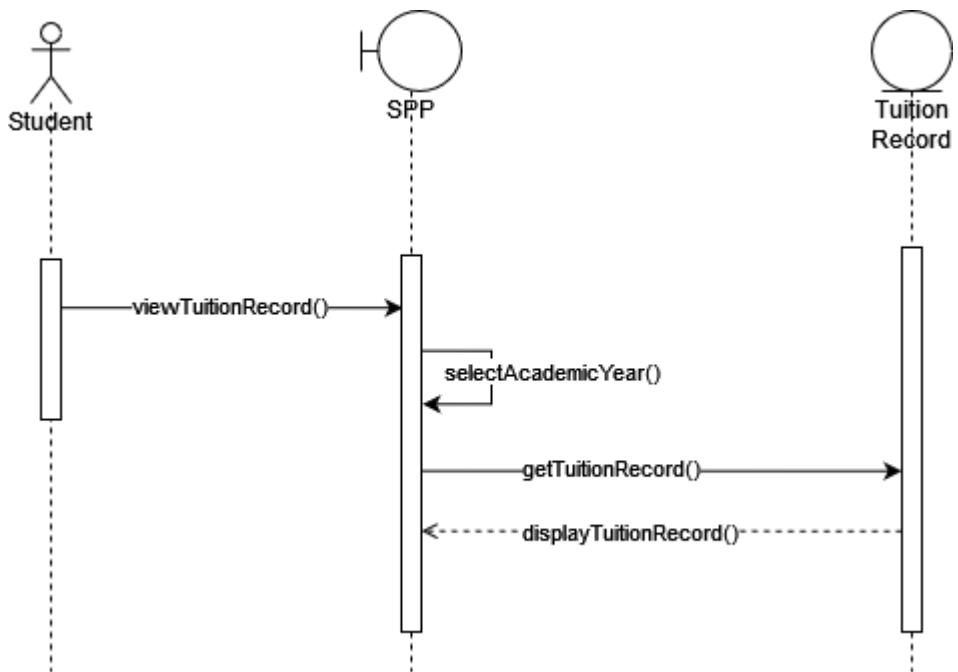


CD005 Class Diagram for Account Subsystem

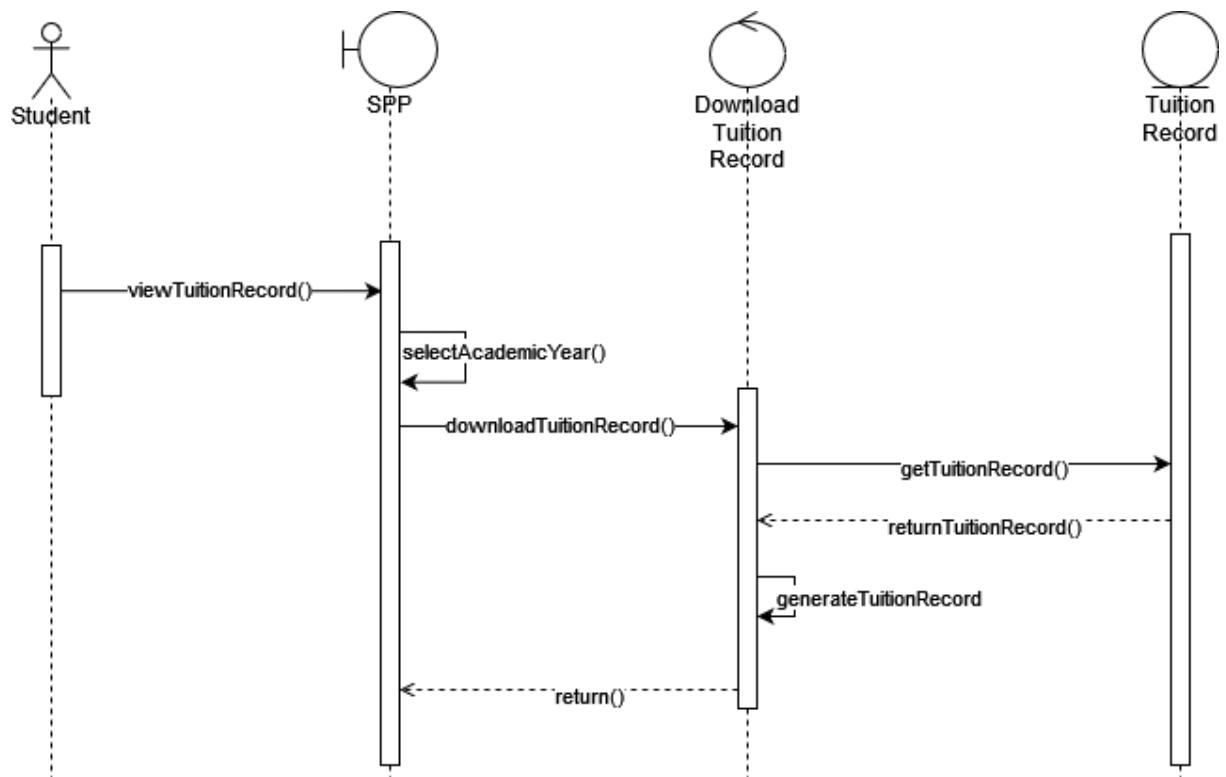
3.2.5.3 Sequence Diagrams



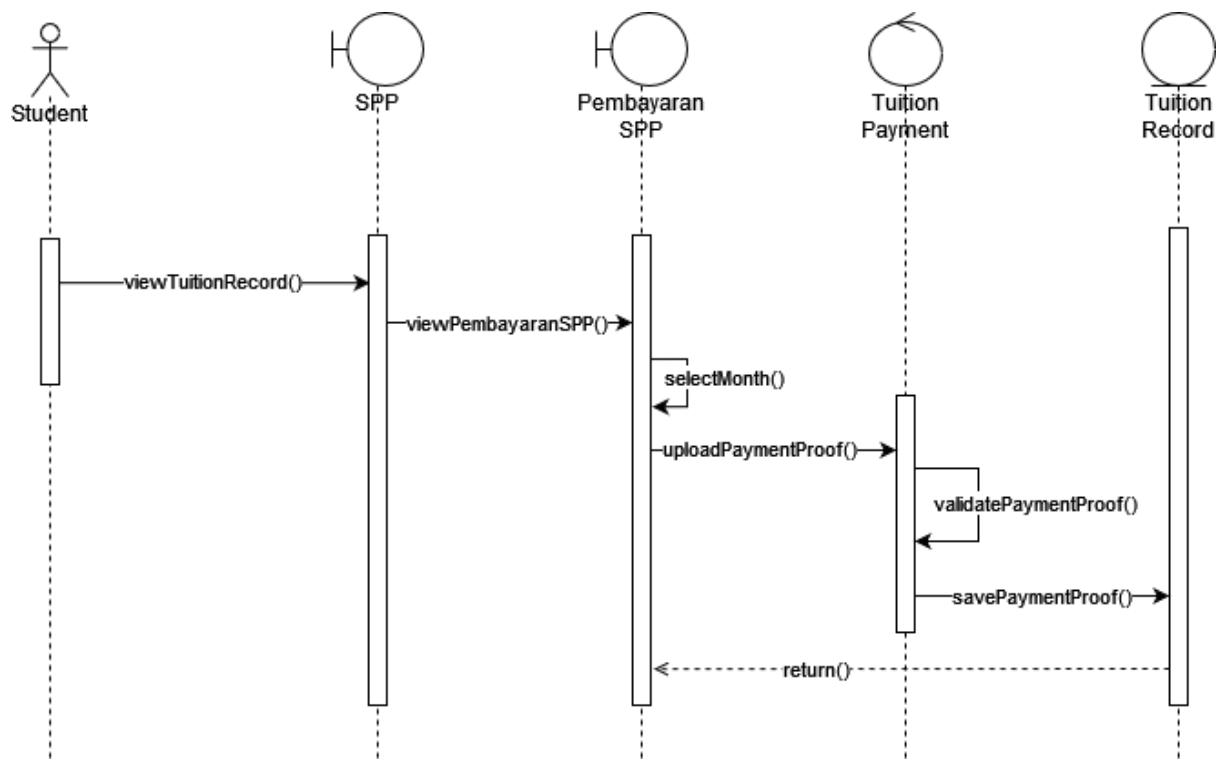
SD010 Sequence Diagram of Manage Tuition Record



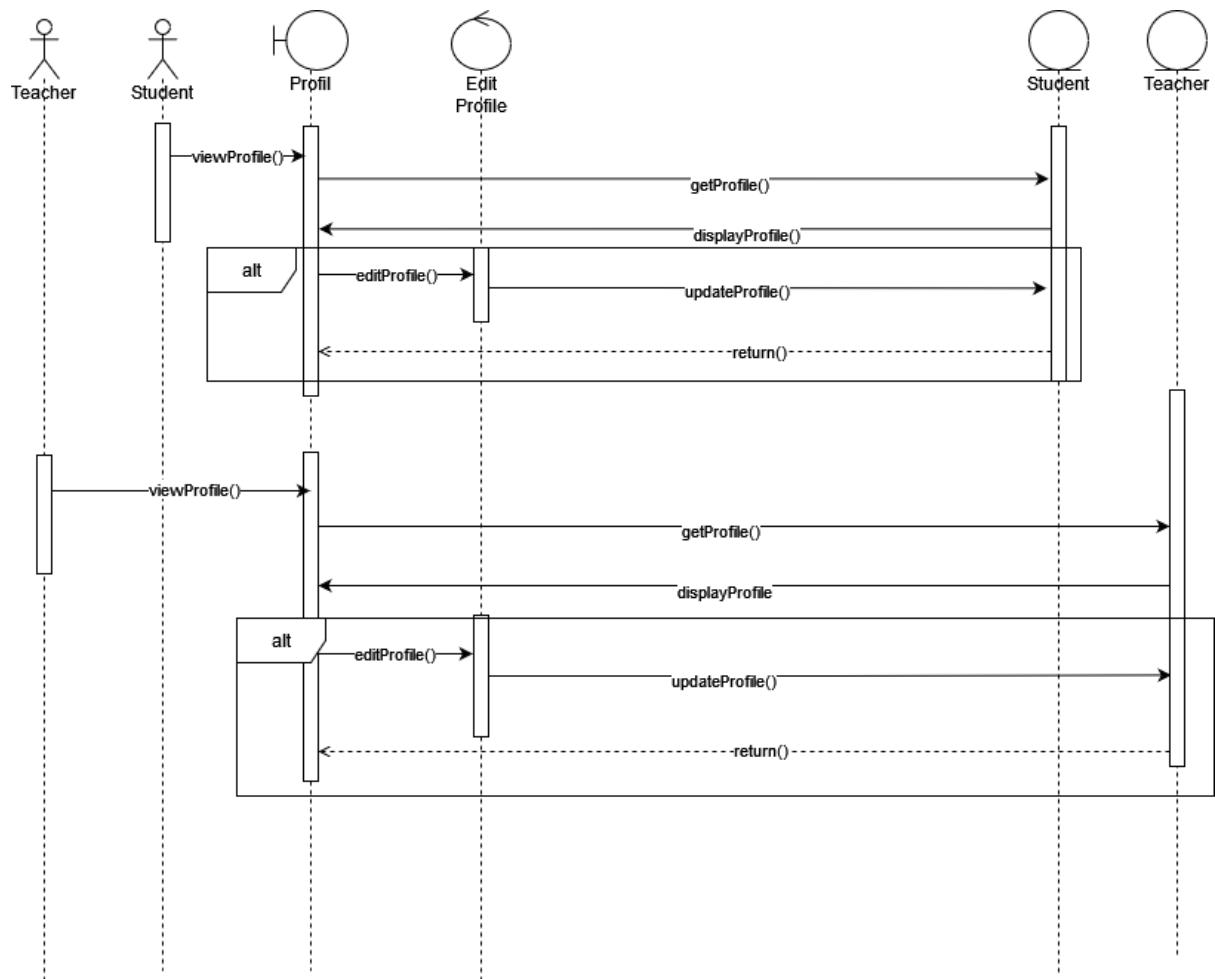
SD011 Sequence Diagram of View Tuition Record



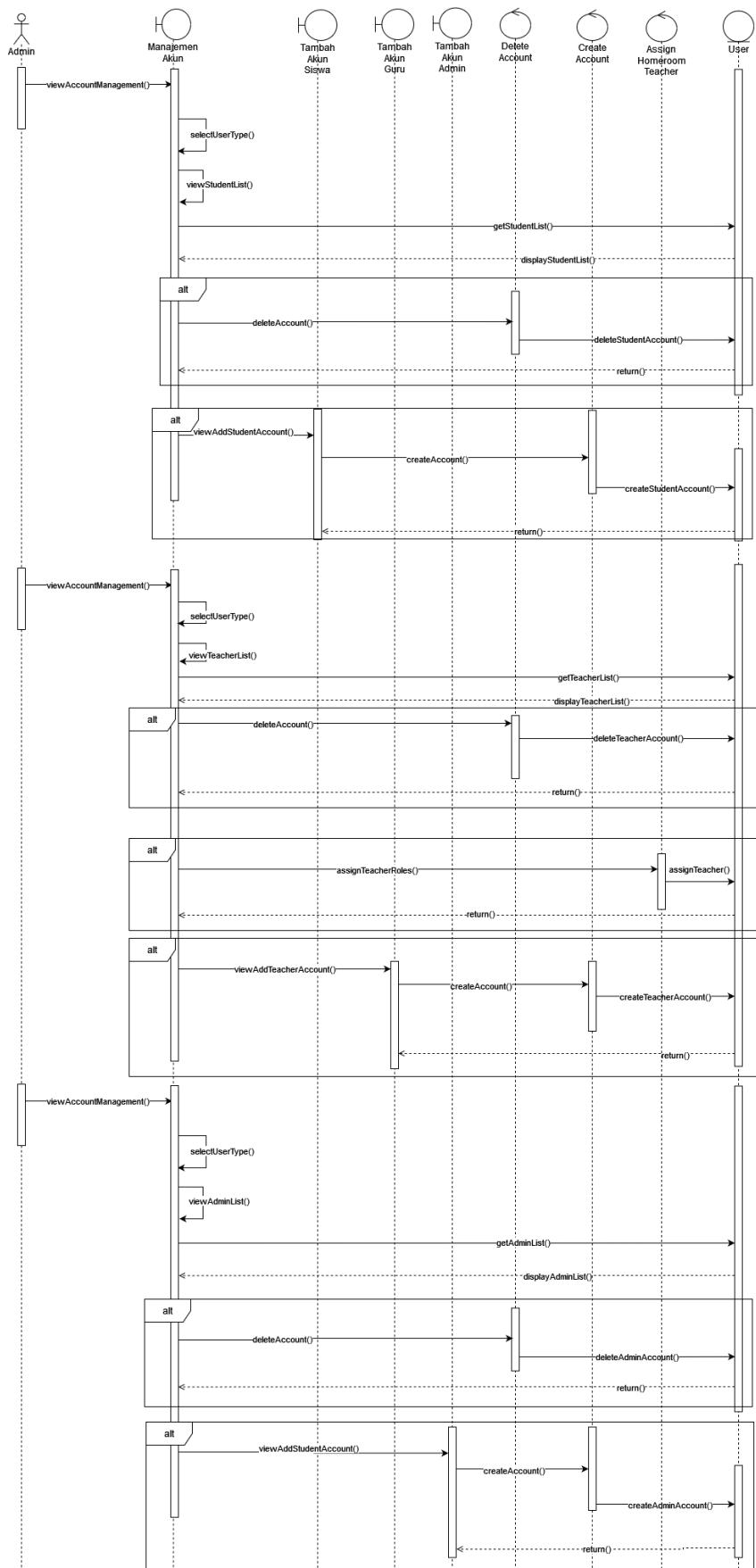
SD012 Sequence Diagram of Download Tuition Record



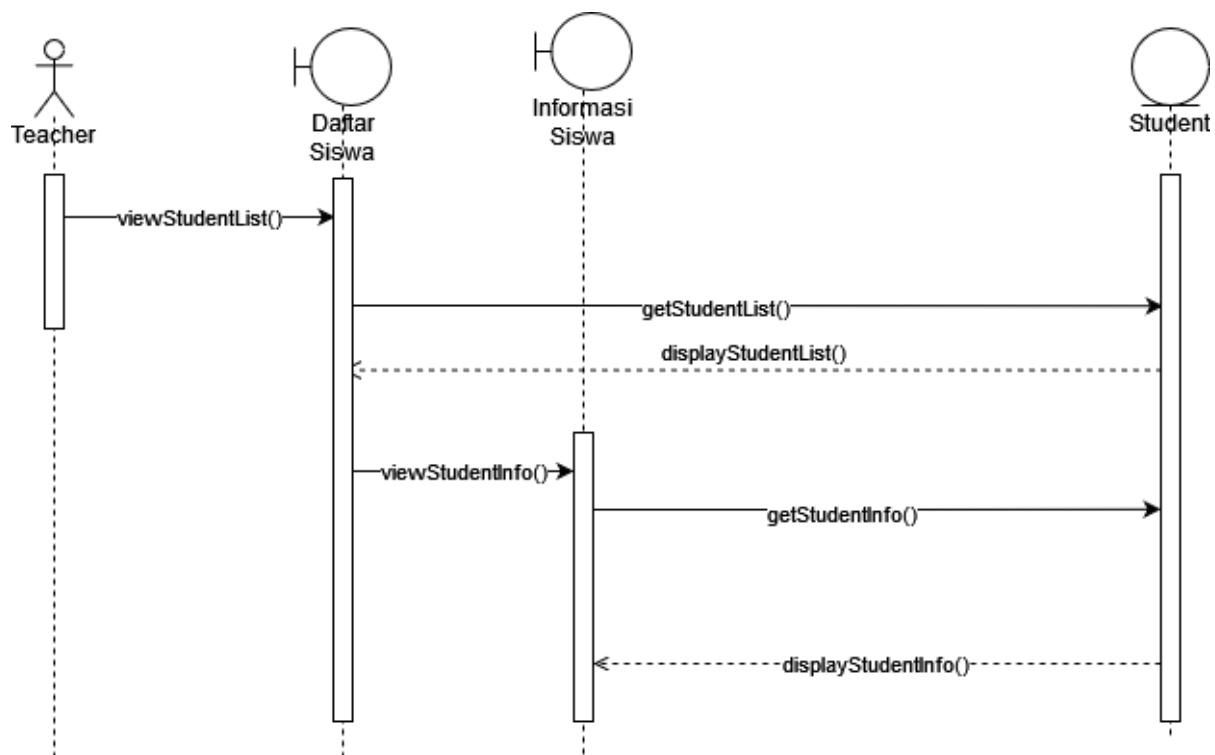
SD013 Sequence Diagram of Make a Payment



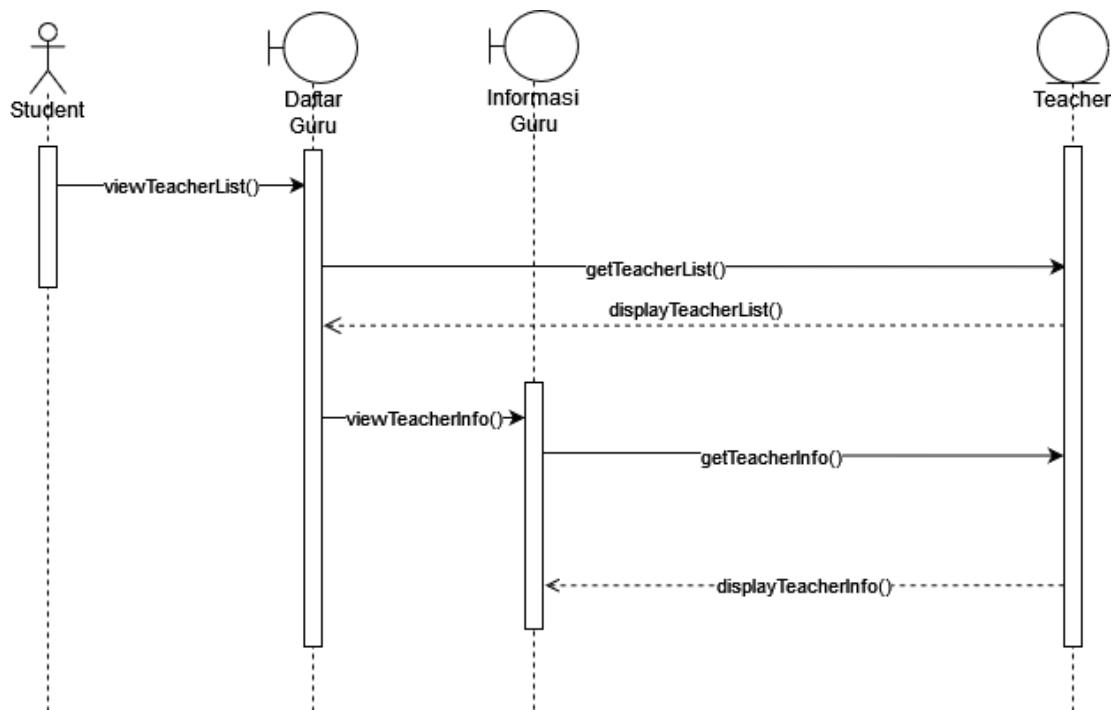
SD014 Sequence Diagram of Manage Profile



SD015 Sequence Diagram of Manage User



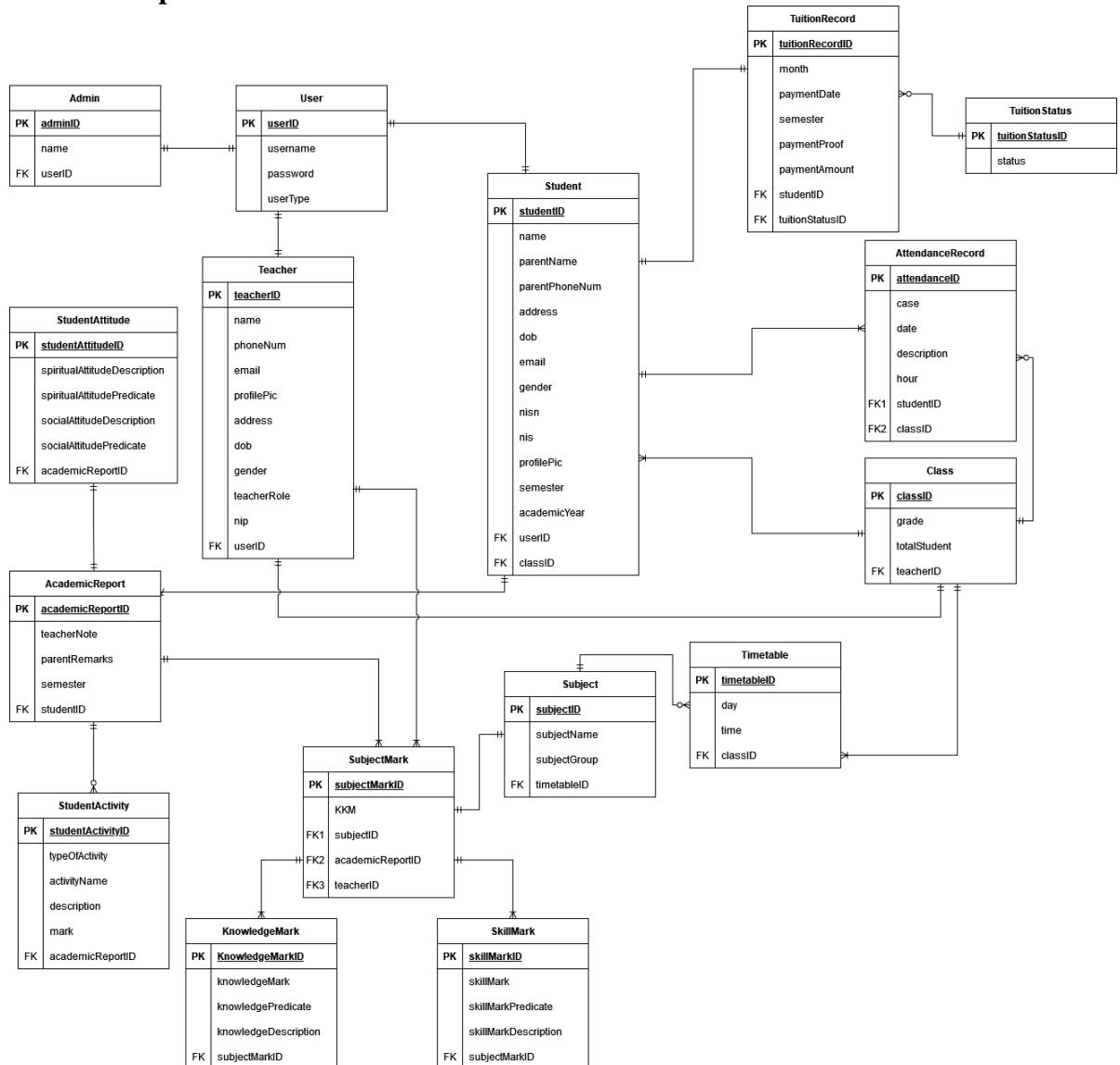
SD0016 Sequence Diagram of View Student Info



SD0017 Sequence Diagram of View Teacher Info

4. Data Design

4.1 Data Description



ERD001 Entity Relationship Diagram of Student Academic Management System

Table 4.1 Data Description of Student Academic Management System

Entity Name	Description
-------------	-------------

User	This entity stores username and password of all users. It also stores the information about which user type is the account (Admin, Teacher or Student)
Teacher	This entity stores all of the teacher user data. They consist of several personal information and also some information that can be showed when the student views the teacher list and their detailed information.
Student	This entity stores all of the student user data. They consist of several personal information and also some information that can be showed when teacher view the student list and their detailed information.
Admin	This entity stores all of the admin user data.
TuitionRecord	This entity stores the tuition record data that is important in managing the tuition record. It holds data such as the month that need to be paid, the payment date, proof and amount when student paid their tuition.
TuitionStatus	This entity stores the status of tuition payment. Teacher can either approved or disapprove it after checking the proof image.
AttendanceRecord	This entity stores the attendance data that will be filled when a teacher took an attendance. The student can also view the attendance data of their own within a range of dates.
Class	This entity stores all the data of classrooms. They mainly consist of every classroom that exist in the school with grade as the classifier. It also stores total student of that certain classroom.
Timetable	This entity stores the timetable data. The data stored here is used to view the timetable by student or teacher. The assigned homeroom teacher will fill their classroom timetable using day time and the subject and it will be automatically generated.
Subject	This entity stores the subject data. It consists of every subject in the school and will likely never change since adding a new

	subject is a rare occasion. So, the data here is already exist beforehand
SubjectMark	This entity stores the subject's mark data. The mark for the subject is separated into two types of marks then will be collected by this entity to combine them when creating academic report later
KnowledgeMark	This entity stores the knowledge type mark of a subject. It consists of the mark itself, the predicate in which has a range from the best "A" to the worst "D". It also contains the description from the teacher of how well the students did in their subject.
SkillMark	This entity stores the knowledge type mark of a subject. It consists of the mark itself, the predicate in which has a range from the best "A" to the worst "D". It also contains the description from the teacher of how well the students did in their subject.
AcademicReport	This entity stores the academic report data that will be created by assigned homeroom teacher when the recap day has begun. It will collect all of the marks, activity, attitude and some student's information which then will generate the final report of that semester. The
StudentAttitude	This entity stores the student's attitude data. The homeroom teacher can input the student's attitude during the semester either in spiritual aspect or social aspect
StudentActivity	This entity stores the student's activity data. The homeroom teacher can input the student's activity during the semester either extracurricular or any achievement they got.

4.2 Data Dictionary

Teacher Table			
Field Name	Datatype	Constraint	Description
teacherID	BIGINT	Primary Key	Unique ID of the Teacher Table
name	CHAR	not null	Name of the Teacher
phoneNum	INT	not null	Phone number of the Teacher
email	VARCHAR	not null	Email of the Teacher
profilePic	VARCHAR	nullable	The url of the teacher profile picture
address	VARCHAR	not null	Address of the Teacher
dob	DATE	not null	Date of Birth of the Teacher
gender	CHAR	not null	Gender of the Teacher
teacherRole	CHAR	nullable	Signified the role of the teacher as a homeroom teacher of a class or not
nip	INT	not null	NIP of the Teacher
userID	BIGINT	Foreign Key	Unique ID of the User table

Student Table			
Field Name	Datatype	Constraint	Description
studentID	BIGINT	Primary Key	Unique ID of the Student table
name	CHAR	not null	name of the student

parentName	CHAR	not null	Student's parent(s) name
parentPhoneNum	INT	not null	Student's parent(s) phone number
address	VARCHAR	not null	Address of the student
dob	DATE	not null	Date of Birth of the student
email	VARCHAR	not null	Email of the student
gender	CHAR	not null	Gender of the student
nism	INT	not null	“Nomor Induk Siswa Nasional” of the student
nis	INT	not null	“Nomor Induk Sekolah” of the student
profilePic	VARCHAR	nullable	The url of the teacher profile picture
semester	VARCHAR	not null	Current semester of the student
academicYear	INT	not null	Current academic year of the student
userID	BIGINT	Foreign Key	Unique ID of the User table
classID	BIGINT	Foreign Key	Unique ID of the Class table

User Table			
Field Name	Datatype	Constraint	Description
userID	BIGINT	Primary Key	Unique ID of the User table
username	VARCHAR	not null	username of the user
password	VARCHAR	not null	password of the user

Admin Table			
Field Name	Datatype	Constraint	Description
adminID	BIGINT	Primary Key	Unique ID of the Admin table
name	CHAR	not null	Name of the admin
userID	BIGINT	Foreign Key	Unique ID of the User table

TuitionRecord Table			
Field Name	Datatype	Constraint	Description
tuitionRecordID	BIGINT	Primary Key	Unique ID of the TuitionRecord table
month	TINYINT	not null	The month of the tuition to be paid
paymentDate	DATE	not null	Payment date for the tuition
semester	VARCHAR	not null	Semester of the tuition
paymentProof	VARCHAR	not null	The url of the uploaded image for the tuition's proof of payment
paymentAmount	BIGINT	not null	The total amount of tuition that is needed to be paid
studentID	BIGINT	Foreign Key	Unique ID of the Student table

TuitionStatus Table			
Field Name	Datatype	Constraint	Description
tuitionStatusID	BIGINT	Primary Key	Unique ID of the TuitionStatus table

status	CHAR	not null	The status of approval for the tuition payment
--------	------	----------	--

AttendanceRecord Table			
Field Name	Datatype	Constraint	Description
attendanceID	BIGINT	Primary Key	Unique ID of the Attendance table
case	CHAR	not null	Case of attendance (Present, Not Present without reason, Sick, Leave permission)
date	DATE	not null	Date of the attendance
description	TEXT	nullable	Detail on why the student did not attend the class
hour	INT	nullable	School's hour that usually consist of up to 8 hours per day
studentID	BIGINT	Foreign Key	Unique ID of the Student table
classID	BIGINT	Foreign Key	Unique ID of the Class table

Class Table			
Field Name	Datatype	Constraint	Description
classID	BIGINT	Primary Key	Unique ID of the Class table
grade	VARCHAR	not null	Grade of the class
totalStudent	INT	not null	Total student in the class
teacherID	BIGINT	Foreign Key	Unique ID of the Teacher table

Timetable Table			
Field Name	Datatype	Constraint	Description
timetableID	BIGINT	Primary Key	Unique ID of the Timetable table
day	CHAR	not null	Day of the subject
time	TIME	not null	Time of the subject
classID	BIGINT	Foreign Key	Unique ID of the Class table

Subject Table			
Field Name	Datatype	Constraint	Description
subjectID	BIGINT	Primary Key	Unique ID of the Subject table
subjectName	VARCHAR	not null	Name of the subject
subjectGroup	CHAR	not null	Group of the subject
subjectScheduleID	BIGINT	Foreign Key	Unique ID of the SubjectSchedule table

SubjectMark Table			
Field Name	Datatype	Constraint	Description
subjectMarkID	BIGINT	Primary Key	Unique ID of the Subject Mark table
KKM	INT	not null	KKM (Kriteria Ketuntasan Minimal) or Minimum Completeness Criteria is set by the school to determine whether a student pass the subject or not

subjectID	BIGINT	Foreign Key	Unique ID of the Subject table
academicReportID	BIGINT	Foreign Key	Unique ID of the AcademicReport table
teacherID	BIGINT	Foreign Key	Unique ID of the Student table

SkillMark Table			
Field Name	Datatype	Constraint	Description
skillMarkID	BIGINT	Primary Key	Unique ID of the SkillMark table
skillMark	INT	not null	Skill Mark of a student for a subject
skillMarkPredicate	CHAR	not null	The predicate of the Skill mark that the student got for a subject
skillMarkDescription	TEXT	nullable	The Description of the Skill mark that the student got for a subject
subjectMarkID	BIGINT	Foreign Key	Unique ID of the SubjectMark table

KnowledgeMark table			
Field Name	Datatype	Constraint	Description
knowledgeMarkID	BIGINT	Primary Key	Unique ID of the KnowledgeMark table

knowledgeMark	INT	not null	Knowledge mark of a student for a subject
knowledgePredicate	CHAR	not null	The predicate of the knowledge mark that the student got for a subject
knowledgeDescription	TEXT	nullable	The Description of the knowledge mark that the student got for a subject
subjectMarkID	BIGINT	Foreign Key	Unique ID of the SubjectMark table

AcademicReport Table			
Field Name	Datatype	Constraint	Description
academicReportID	BIGINT	Primary Key	Unique ID of the AcademicReport table
teacherNote	TEXT	not null	Note from the teacher about the student
semester	VARCHAR	not null	The semester of the student
studentID	BIGINT	Foreign Key	Unique ID of the Student table

StudentActivity			
Field Name	Datatype	Constraint	Description
achievementID	BIGINT	Primary Key	Unique ID of the Achievement table
typeOfActivity	INT	nullable	Type of activity that the student got the achievement for
description	TEXT	nullable	Details of the achievement

mark	CHAR	nullable	The mark for the activity
academicReportID	BIGINT	Foreign Key	Unique ID of the AcademicReport table

StudentAttitude Table			
Field Name	Datatype	Constraint	Description
studentAttitudeID	BIGINT	Primary Key	Unique ID of the StudentAttitude table
spiritualAttitudeDescription	TEXT	not null	The description of the student's spiritual attitude
spiritualAttitudePredicate	CHAR	not null	The predicate of the student's spiritual attitude
socialAttitudeDescription	TEXT	not null	The description of the student's social attitude
socialAttitudePredicate	CHAR	not null	The predicate of the student's social attitude
academicReportID	BIGINT	Foreign Key	Unique ID of the AcademicReport table

5. User Interface Design

5.1 Overview of User Interface

Page navigation diagram is provided below to describe the flow of the user interface. There are three users/actors in the system which mean there will be three different dashboards for each user. The following figures are the page navigation design for teacher, student and admin.

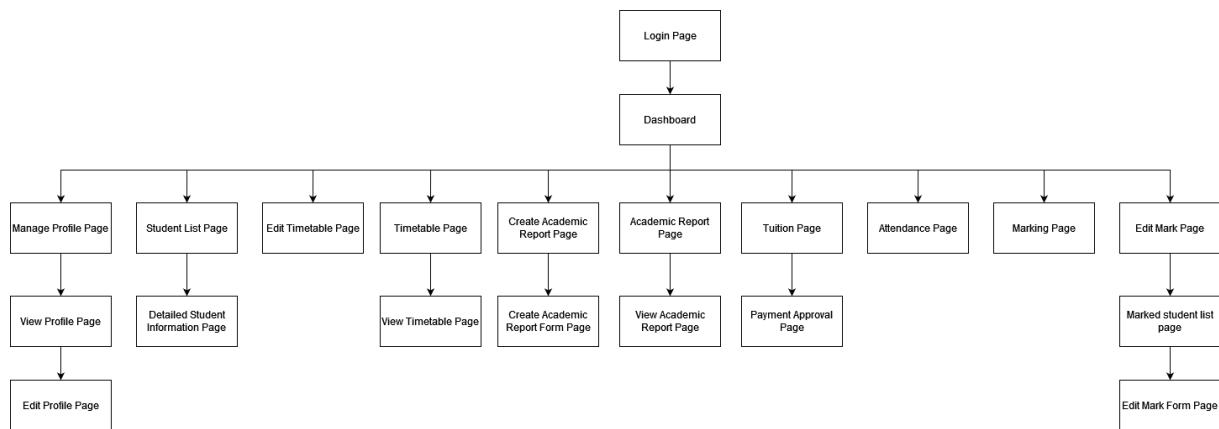


Figure 5.1 Page Navigation for Teacher in SAMS

After entering the system successfully through login, the teacher will be redirect to their own dashboard. From this dashboard the teacher can access all kind of different functions from creating academic report page to managing their profile.

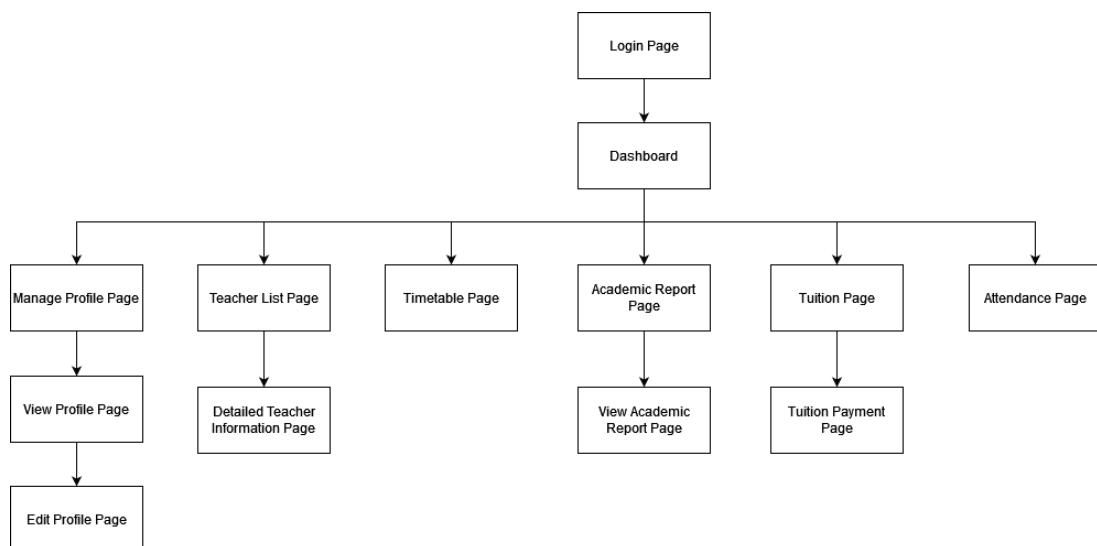


Figure 5.2: Page Navigation for Student in SAMS

After the student login into the system, they also welcomed with their own dashboard. From the dashboard the student can access manage profile page, teacher list page, timetable page, academic report page, tuition page and attendance page.

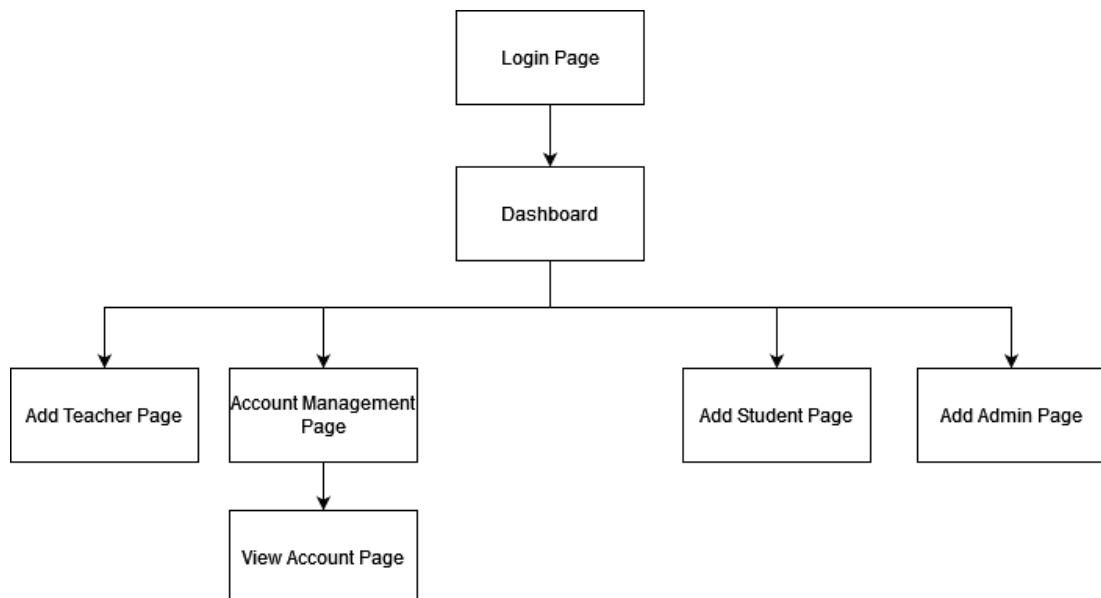


Figure 4.8: Page Navigation for Admin in SAMS

From the figure above, the admin can access to multiple pages after successfully enter the system. These pages are added teacher page, account management page, add student page and add admin page.

5.2 Screen Images

5.2.1 Login UI

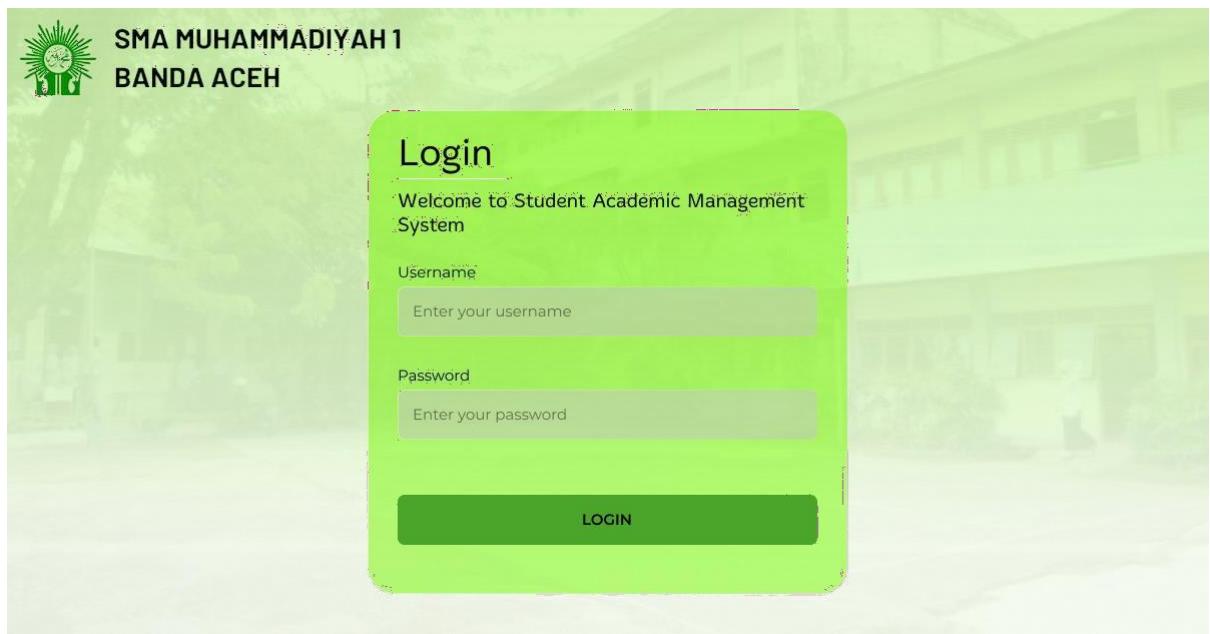


Figure 5.1: Login Page of SAMS

5.2.2 Admin UI



Figure 5.2: Admin Dashboard of SAMS

Manajemen Akun				
Pengguna		Guru		
•	Nama	NIP	Wali Kelas	<input type="text"/>
	Herika Harahap, SE, M.Pd. herika@example.com	19770314 200504 2 002	XI-MIPA	
	Juairiah, S.Pd. juairiah@example.com	19770314 200504 2 002	T/A	
	Cut Nur Ely cultur@example.com	19770314 200504 2 002	X-IPS	
	Dra. Nurmaliah nurmaliah@example.com	19770314 200504 2 002	T/A	
	Melly Hastuty, S.Pd. melly@example.com	19770314 200504 2 002	T/A	
Tampilkan per halaman		5		
				1 dari 25 < >

Figure 5.3: Teacher Account Management of SAMS

Manajemen Akun				
Pengguna		Siswa		
•	Nama	NIS/NISN	Kelas	Semester
	Egita Hayatun egita@example.com	9415/53196112	XI-MIPA	Ganjil
	Elisa elisa@example.com	9416/51005903	XI-MIPA	Genap
	Erni Wahyuni erni@example.com	9417/51748425	X-IPS	Ganjil
	Fauzi Murtaza murtaza@example.com	9418/53544074	XI-MIPA	Ganjil
	Muhammad Irvandi irvandi@example.com	9420/43616991	XI-MIPA	Genap
Tampilkan per halaman		5		
				1 dari 50 < >

Figure 5.4: Student Account Management of SAMS

The screenshot shows a web-based application interface for managing accounts at SMA Muhammadiyah 1 Banda Aceh. On the left, a vertical sidebar menu lists several options: Dashboard, Manajemen Akun, Tambah Akun Guru (which is highlighted with a green background), Tambah Akun Siswa, and Tambah Akun Admin. The main content area is titled "Tambah Akun Guru". It contains a form with the following fields:

- Username
- Kata Sandi (with a note: "Kata sandi harus lebih dari 7 karakter")
- Kata Sandi (with a note: "Kata sandi harus lebih dari 7 karakter")
- Wali Kelas: A dropdown menu.

A large green "Tambah" button is located at the bottom right of the form.

Figure 5.5: Add Teacher Account of SAMS

The screenshot shows the "Tambah Akun Siswa" (Add Student Account) form within the same application interface. The sidebar menu on the left is identical to Figure 5.5. The main content area is titled "Tambah Akun Siswa". The form includes the following fields:

- Username
- Kata Sandi (with a note: "Kata sandi harus lebih dari 7 karakter")
- Masukan Kembali Kata Sandi (with a note: "Kata sandi harus lebih dari 7 karakter")
- Tahun Ajaran: A dropdown menu.
- Semester: A dropdown menu.

A large green "Tambah" button is located at the bottom right of the form.

Figure 5.6: Add Student Account of SAMS

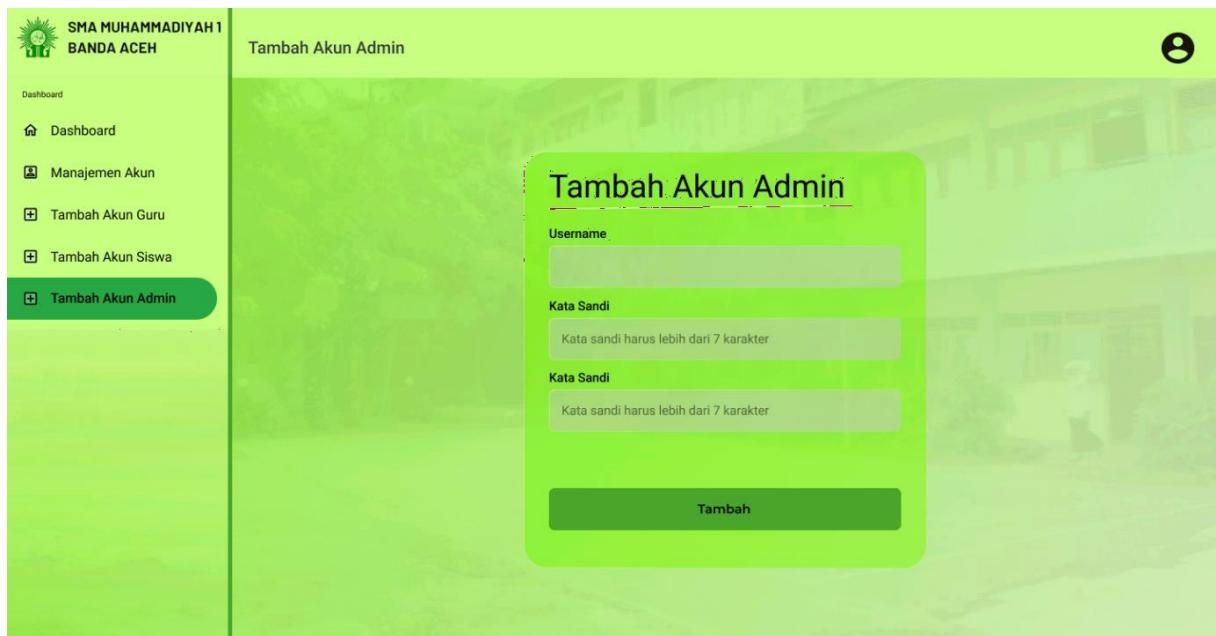


Figure 5.7: Add Admin Account of SAMS

5.2.3 Teacher UI



Figure 5.8: Teacher Dashboard of SAMS

Figure 5.9: Manage Teacher Profile of SAMS

Figure 5.10: Manage Academic Report of SAMS

No	Mata Pelajaran	KKM	Pengetahuan		Keterampilan	
			Nilai	Predikat	Nilai	Predikat
Kelompok A (Umum)						
1	Pendidikan Agama dan Budi Pekerti	70	85	B	85	B
2	Pendidikan Pancasila dan Kewarganegaraan	70	85	B	85	B
3	Bahasa Indonesia	70	85	B	85	B
4	Matematika	70	83	B	84	B
5	Sejarah Indonesia	75	83	B	80	B
6	Bahasa Inggris	75	90	B	90	B
Kelompok B (Umum)						
1	Seni Budaya	75	85	B	83	B
2	Pendidikan Jasmani, Olah Raga, dan Kesehatan	75	85	B	83	B
3	Prakarya dan Kewirausahaan	80	85	B	85	B
4					
Kelompok C (Peminatan)						
1	Biologi	75	84	B	82	B
2	Kimia	70	82	B	82	B
3	Fisika	70	85	B	84	B

Figure 5.11: View Academic Report of SAMS

Buat Rapor			
Kelas XI-MIPA - Semester 1/Ganjil			
Nama	NIS/NISN	Peminatan	
Sheila Faiqah Aurora sheila@gmail.com	9460/0053357399	MIPA	Buat Rapor
Elisa elisa@example.com	9416/51005903	MIPA	Buat Rapor
Erni Wahyuni erni@example.com	9417/51748425	MIPA	Buat Rapor
Fauzi Murtaza murtaza@example.com	9418/53544074	MIPA	Buat Rapor
Muhammad Irvandi irvandi@example.com	9420/43616991	MIPA	Buat Rapor

Figure 5.12: Create Academic Report of SAMS

Buat Rapor			
Sheila Faiqah Aurora			
<input type="button" value="Kembali"/>			
Ekstrakurikuler	<input type="text" value="Pramuka"/> <input type="button" value="+"/>	Nilai	<input type="button" value="A"/> <input type="button" value="▼"/>
<div style="border: 1px solid black; padding: 5px;"> Sangat Terampil dan mampu menggunakan kompas, semaphore, simpul, dan teknik kepramukaan lainnya, namun perlu ditingkatkan lagi dalam baris berbaris dan kedisiplinan </div>			
Prestasi	<input type="text"/> <input type="button" value="+"/>		
<div style="border: 1px solid black; height: 40px; margin-top: 10px; padding: 5px;"> <i>Keterangan</i> </div>			
<input type="button" value="Selanjutnya →"/>			

Figure 5.13: Form 1 for Create Academic Report of SAMS

Buat Rapor

Sheila Faiqah Aurora

Kembali

Perilaku Spiritual	Nilai A
Sangat Terampil dan mampu menggunakan kompas, semaphore, simpul, dan teknik kepramukaan lainnya, namun perlu ditingkatkan lagi dalam baris berbaris dan kedisiplinan	
Perilaku Sosial	Nilai A
Keterangan	
Catatan Wali Kelas	

Selanjutnya →

Figure 5.14: Form 2 for Create Academic Report of SAMS

Buat Rapor

Sheila Faiqah Aurora

Apakah anda yakin ingin membuat rapor?

Buat Rapor

Kembali

Figure 5.15: Confirmation for Create Academic Report of SAMS

The screenshot shows the 'Penilaian' (Assessment) section of the SAMS application. On the left sidebar, under the 'Penilaian' category, the 'Edit Penilaian' option is selected. The main form displays a student record for 'Sheila Faiqah Aurora'. The fields include: Kelas (XI-MIPA), Semester (Ganjil), Siswa (Sheila Faiqah Aurora), Tahun Pelajaran (2021/2022), Mata Pelajaran (Matematika), KKM (70), Aspek (Keterampilan), and Nilai (84). A descriptive text box states: 'Siswa sudah terampil dalam menyelesaikan soal-soal materi matriks terus berlatih untuk soal-soal pembuktian induksi matematika.' A green 'Simpan' button is located at the bottom right.

Figure 5.16: Marking of SAMS

The screenshot shows the 'Buat Rapor' (Create Report) section of the SAMS application. Under the 'Edit Penilaian' category in the sidebar, the 'Edit Penilaian' option is selected. The main area displays a list titled 'Siswa yang sudah dinilai' (Students already assessed) with five entries:

Nama	NIS/NISN	Aspek	Action
Sheila Faiqah Aurora sheila@gmail.com	9460/0053357399	Keterampilan	Ubah Nilai
Elisa elisa@example.com	9416/51005903	Pengetahuan	Ubah Nilai
Erni Wahyuni erni@example.com	9417/51748425	Keterampilan	Ubah Nilai
Fauzi Murtaza murtaza@example.com	9418/53544074	Pengetahuan	Ubah Nilai
Muhammad Irvandi irvandi@example.com	9420/43616991	Keterampilan	Ubah Nilai

At the bottom, there are pagination controls ('Tampilkan per halaman' dropdown set to 5, page '1 dari 1') and navigation arrows.

Figure 5.17: View Edit Marking of SAMS

Edit Penilaian

Kelas: XI-MIPA Semester: Ganjil

Siswa: Sheila Faiqah Aurora Tahun Pelajaran: 2021/2022

Mata Pelajaran: Matematika KKM: 70 Aspek: Keterampilan

Nilai: 84

Deskripsi:
Siswa sudah terampil dalam menyelesaikan soal-soal materi matriks terus berlatih untuk soal-soal pembuktian induksi matematika.

Ubah

Figure 5.18: Edit Marking Form of SAMS

Daftar Siswa

Sortir: Tahun Ajaran: 2021/2022 Semester: Ganjil Kelas: XI MIPA

Nama	NIS/NISN	Peminatan	Aksi
Sheila Faiqah Aurora sheila@gmail.com	9460/0053357399	MIPA	Lihat Detail
Elisa elisa@example.com	9416/51005903	MIPA	Lihat Detail
Erni Wahyuni erni@example.com	9417/51748425	MIPA	Lihat Detail
Fauzi Murtaza murtaza@example.com	9418/53544074	MIPA	Lihat Detail
Muhammad Irvandi irvandi@example.com	9420/43616991	MIPA	Lihat Detail

Tampilan per halaman: 5 1 dari 1 < >

Figure 5.19: List of students of SAMS

The screenshot shows the 'Detail' view of a student's profile. At the top right is a back button labeled 'Kembali'. On the left, there is a circular placeholder for a profile picture. To its right, the student's name 'Nama: Sheila Faiqah Aurora' is displayed. Below the name, the student's class is listed as 'Kelas: XI-MIPA'. Further down, the NIS/NISN is given as 'NIS/NISN: 9460/0053357399'. Address, birth date, phone number, and email are also provided.

Kelas: XI-MIPA
Nama Wali Murid: A. Hamid
Nomor HP Wali Murid: (+62)8119728722
Alamat: Jln. Ujong Batee Setui Banda Aceh
Tanggal Lahir: 12/12/2005
Nomor HP: (+62)8119728721
Email: sheila@example.com

Figure 5.20: Detailed info of students of SAMS

The screenshot displays the 'Jadwal Pelajaran' (Timetable) for the XI MIPA class. The top row shows the days of the week: Senin, Selasa, Rabu, Kamis, Jumat, and Sabtu. The columns represent time slots 1 through 8. Each cell contains the subject taught at that specific time on that day. For example, on Monday (Senin), all slots are filled with 'Matematika'. On Tuesday (Selasa), all slots are filled with 'Biologi'. On Wednesday (Rabu), all slots are filled with 'Matematika'. On Thursday (Kamis), all slots are filled with 'Biologi'. On Friday (Jumat), all slots are filled with 'Matematika'. On Saturday (Sabtu), all slots are filled with 'Biologi'.

Kelas		1	2	3	4	5	6	7	8
Senin	XI MIPA	Matematika							
Selasa		Biologi							
Rabu		Matematika							
Kamis		Biologi							
Jumat		Matematika							
Sabtu		Biologi							

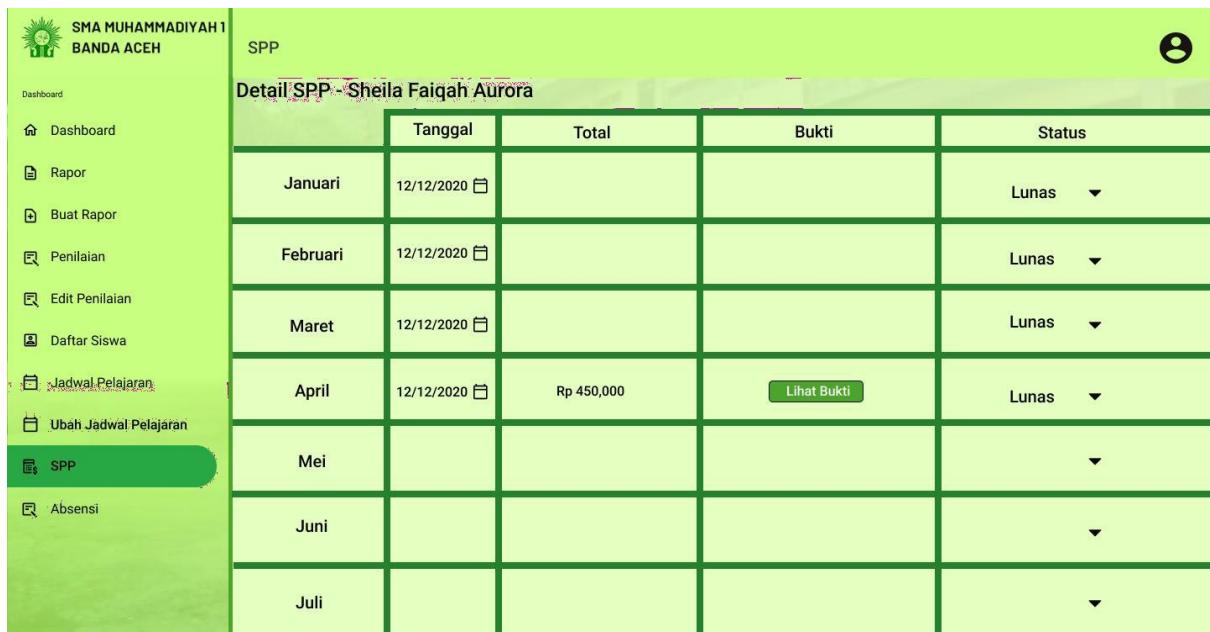
Figure 5.21: View timetable of SAMS

Jadwal Pelajaran - Kelas XI-MIPA								
	1	2	3	4	5	6	7	8
Senin	Matematika ▾							
Selasa	Biologi ▾							
Rabu	Matematika ▾							
Kamis	Biologi ▾							
Jumat	Matematika ▾							
Sabtu	Biologi ▾							

Figure 5.22: Edit timetable of SAMS

SPP			
Sortir: Tahun Ajaran: 2021/2022 Semester: Ganjil Kelas: XI MIPA			
Nama	NIS/NISN	Peminatan	Lihat SPP
Sheila Faiqah Aurora sheila@gmail.com	9460/0053357399	MIPA	Lihat SPP
Elisa elisa@example.com	9416/51005903	MIPA	Lihat SPP
Erni Wahyuni erni@example.com	9417/51748425	MIPA	Lihat SPP
Fauzi Murtaza murtaza@example.com	9418/53544074	MIPA	Lihat SPP
Muhammad Irvandi irvandi@example.com	9420/43616991	MIPA	Lihat SPP

Figure 5.23: Manage tuition record of SAMS



The screenshot shows a table titled "Detail SPP - Sheila Faiqah Aurora". The columns are "Tanggal" (Date), "Total", "Bukti" (Evidence), and "Status". The rows represent months from January to July. The "Total" column for April shows "Rp 450,000" and contains a button labeled "Lihat Bukti" (View Evidence). The "Status" column for all months shows "Lunas" (Paid) with a dropdown arrow.

	Tanggal	Total	Bukti	Status
Januari	12/12/2020			Lunas ▾
Februari	12/12/2020			Lunas ▾
Maret	12/12/2020			Lunas ▾
April	12/12/2020	Rp 450,000	Lihat Bukti	Lunas ▾
Mei				▼
Juni				▼
Juli				▼

Figure 5.24: Change tuition status of SAMS



The screenshot shows a table titled "Absensi". The columns are "Nama" (Name), "NIS/NISN", "S/I/A/H" (Attendance Status), and "Keterangan" (Remarks). The rows list five students: Sheila Faiqah Aurora, Elisa, Erni Wahyuni, Fauzi Murtaza, and Muhammad Irvandi. Each row includes their NISN, attendance status (e.g., H = Hadir, S = Sakit, I = Izin), and a remarks field. A legend at the top right defines the status abbreviations: S = Sakit, I = Izin, A = Alpha, H = Hadir. A "Simpan" (Save) button is located at the top right of the form.

Nama	NIS/NISN	S/I/A/H	Keterangan
Sheila Faiqah Aurora sheila@gmail.com	9460/0053357399	H ▾	
Elisa elisa@example.com	9416/51005903	H ▾	
Erni Wahyuni erni@example.com	9417/51748425	H ▾	
Fauzi Murtaza murtaza@example.com	9418/53544074	S ▾	
Muhammad Irvandi irvandi@example.com	9420/43616991	I ▾	

Figure 5.25: Record attendance of SAMS

5.2.4 Student UI



Figure 5.26: Student dashboard of SAMS

This screenshot shows the "Profil" (Profile) section of the SAMS system. It displays a student's profile information. On the left is the vertical navigation menu. The main area shows a circular placeholder for a profile picture, followed by the student's name, Nama: Sheila Faiqah Aurora. Below the name are several details: Kelas: XI-MIPA, NIS/NISN: 9460/0053357399, Alamat: Jln. Ujong Batee Setui Banda Aceh, Tanggal Lahir: 12/12/2005, Nomor HP: (+62)8119728721, and Email: sheila@example.com. To the right of these details are the Wali Murid's information: Nama Wali Murid: A. Hamid and Nomor HP Wali Murid: (+62)8119728722. At the bottom right is a green button labeled "Ubah Profil" (Change Profile).

Figure 5.27: Manage student profile of SAMS

The screenshot shows the SAMS application interface for viewing academic reports. The top navigation bar includes the school logo (SMA Muhammadiyah 1 Banda Aceh), user profile, and a search icon. The left sidebar has a green header labeled 'Rapor' and contains links for Dashboard, Jadwal Pelajaran, SPP, Absensi, and Daftar Guru. The main content area is titled 'Rapor' and displays a form for selecting the academic year (2021/2022), semester (Ganjil), and class (XI MIPA). Below this, a table provides student details: Nama Sekolah : SMA Muhammadiyah 1 Banda Aceh; Alamat : Jln. Ujung Batee Setia Banda Aceh; Kelas : XI MIPA; Semester : 1 (Satu) / Ganjil; Nama : Awal Hidayat Marpaung; Tahun Pelajaran : 2021/2022; Nomor Induk/NISN : 9490/ 0067683477. A section titled 'Deskripsi Pengetahuan dan Keterampilan' (Description of Knowledge and Skills) is shown with three rows for subjects: Pendidikan Agama dan Budi Pekerti, Pendidikan Pancasila dan Kewarganegaraan, and Bahasa Indonesia, each with sub-sections for Pengetahuan and Keterampilan.

Figure 5.28: View academic report of SAMS

The screenshot shows the SAMS application interface for viewing academic reports. The layout is identical to Figure 5.28, with the school logo, user profile, and search icon at the top. The left sidebar has a green header labeled 'Rapor' and contains links for Dashboard, Jadwal Pelajaran, SPP, Absensi, and Daftar Guru. The main content area is titled 'Rapor' and displays a form for selecting the academic year (2021/2022), semester (Ganjil), and class (XI MIPA). Below this, a central message box displays the text 'Rapor Tidak Ditemukan' (Report Not Found).

Figure 5.29: View academic report (when there is no academic report) of SAMS

The screenshot shows a green-themed user interface for the SAMS application. On the left, a sidebar lists navigation options: Dashboard, Rapor, Jadwal Pelajaran (highlighted with a red oval), SPP, Absensi (highlighted with a red oval), and Daftar Guru.

The main content area is titled "Daftar Siswa" and "Kelas: XI MIPA". It features a large green grid table with columns labeled 1 through 8. Rows represent days of the week: Senin, Selasa, Rabu, Kamis, Jumat, and Sabtu. Each cell contains either "Matematika" or "Biologi". A green button in the top right corner says "Unduh".

	1	2	3	4	5	6	7	8
Senin	Matematika							
Selasa	Biologi							
Rabu	Matematika							
Kamis	Biologi							
Jumat	Matematika							
Sabtu	Biologi							

Figure 5.30: View timetable of SAMS

The screenshot shows a green-themed user interface for the SAMS application. On the left, a sidebar lists navigation options: Dashboard, Rapor, Jadwal Pelajaran, SPP, Absensi (highlighted with a red oval), and Daftar Guru.

The main content area is titled "Absensi". It includes a search bar with "Tanggal" (12/12/2021 - 15/01/2022) and "Kelas" (XI MIPA). To the right, a legend defines symbols: S = Sakit, J = Izin, A = Alpha, and H = Hadir.

The table displays three rows of attendance statistics:

- Total Sakit: 3 Hari
- Total Izin: - Hari
- Total Alpha: - Hari

Figure 5.31: View attendance of SAMS

Daftar Guru			
	Nama	NIP	
	Herika Harahap, SE, M.Pd. herika@example.com	19770314 200504 2 002	Lihat Detail
	Juairiah, S.Pd. juairiah@example.com	19770314 200504 2 002	Lihat Detail
	Cut Nur Ely cuthur@example.com	19770314 200504 2 002	Lihat Detail
	Dra. Nurmaliah nurmaliah@example.com	19770314 200504 2 002	Lihat Detail
	Melly Hastuty, S.Pd. melly@example.com	19770314 200504 2 002	Lihat Detail

Figure 5.32: List of teachers of SAMS

Daftar Guru			
	Detail		
	Nama: Herika Harahap, SE, M.Pd.		Kembali
	NIP: 19770314 200504 2 002		
	Alamat: Jln. Ujong Batee Setui Banda Aceh		
	Nomor HP: (+62)8119728721		
	Email: herika@example.com		

Figure 5.33: Teacher's detailed information of SAMS

Tanggal	Total	Bukti	Status
Januari	12/12/2021		
Februari	12/12/2021		
Maret	12/12/2021		
April	12/12/2021	Rp 450,000	Lihat Bukti
Mei			
Juni			
Juli			

Figure 5.34: View tuition record of SAMS

Tanggal	Total	Bukti
Januari	12/12/2021	
Februari	12/12/2021	
Maret	12/12/2021	
April	12/12/2021	Rp 450,000
Mei		
Juni		
Juli		

Figure 5.35: Make a payment of SAMS

6. Requirements Matrix

Requirement matrix is used to determine which component of the system satisfies the requirements mentioned in the SRS. In this case, the following table shows the use case stated in the SRS is compared with the sequence diagram.

	SD 001	SD 002	SD 003	SD 004	SD 005	SD 006	SD 007	SD 008	SD 009	SD 010	SD 011	SD 012	SD 013	SD 014	SD 015	SD 016	SD 017
UC 001	X																
UC 002		X															
UC 003			X														
UC 004				X													
UC 005					X												
UC 006						X											
UC 007							X										
UC 008								X									
UC 009									X								
UC 010										X							
UC 011											X						
UC 012												X					
UC 013													X				

UC 014												X			
UC 015													X		
UC 016														X	
UC 017															X

7. Appendices

Provide appendices if any.

Appendix C Software Testing Documentation



Software Testing Documentation

Project Title

Version 1.0

24 June 2023

Department and Faculty

Prepared by: Muhammad Darlen Sava

Revision Page

a. Overview

This is the first draft of the Software Requirements Specification (SRS) for Student Academic Management System (SAMS).

b. Target Audience

Student, Teacher and Admin of SMAS Muhammadiyah 1

c. Project Team Members

Muhammad Darlen Sava

d. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Muhammad Darlen Sava	Initial Draft	24 June 2023

Note:

This template is an annotated outline for a software testing document. It is based on IEEE standards 829, 1008, 1012 and 1012a. This document covers: unit testing (the verification of individual sub-systems or components of the system against their specifications), integration testing (the testing of inter-operating sub-systems or components against their specifications) and system testing (both verification against the system specification, and validation against the user requirements). This template has been simplified and customized to meet the need of SCSJ2203 course at Faculty of Computing, UTM. Compiled by Ruhaidah Samsudin, PhD and checked by Masitah Ghazali, PhD and Shahida Sulaiman, PhD (revised on 14 May 2016).

Table of Contents

1	Introduction	1
1.1	Purpose	1
1.2	Scope	1
1.3	Definitions, Acronyms and Abbreviations	1
1.4	Reference Materials	1
1.5	System Overview	2
2	Test Cases	
2.1	Test TC001 for Module Authentication: <Sign In (UC001)>	3
2.2	Test TC002 for Module Report: <Manage Academic Report (UC002)>	3
2.3	Test TC003 for Module Report: <View Academic Report (UC003)>	7
2.4	Test TC004 for Module Report: <Download Academic Report (UC004)>	7
2.5	Test TC003 for Module Report: <Manage Attendance Record (UC005)>	8
2.6	Test TC003 for Module Report: <View Attendance Record (UC006)>	8
2.7	Test TC003 for Module Report: <Manage Timetable (UC007)>	9
2.8	Test TC003 for Module Report: <View Timetable (UC008)>	10
2.9	Test TC003 for Module Report: <Download Timetable (UC009)>	10
2.10	Test TC003 for Module Report: <Manage Tuition Record (UC010)>	10
2.11	Test TC003 for Module Report: <View Tuition Record (UC011)>	11
2.12	Test TC003 for Module Report: <Download Tuition Record (UC012)>	12

2.13	Test TC003 for Module Report: <Make a Payment (UC013)>	12
2.14	Test TC003 for Module Report: <Manage Profile (UC014)>	13
2.15	Test TC003 for Module Report: <Manage User (UC015)>	14
2.16	Test TC003 for Module Report: <View Teacher Info (UC016)>	14
2.17	Test TC003 for Module Report: <View Student Info (UC017)>	15

1. Introduction

1.1 Purpose

This STD describes the test design for the Student Academic Management System which is a system dedicated to SMAS Muhammadiyah 1 Banda Aceh. This STD is required to determine the test design for the development phase, it also acts as a reference point when testing the system during development.

1.2 Scope

The current system of SMAS Muhammadiyah is still involving manual task execution with a decentralized database. So this project only focused on the conversion of manually managing student data to a website that stores all of the student's data, which can be accessed by teachers, students, and their parents. As such, SMAS Muhammadiyah will only provide the student's data throughout the development of the project. The users in this system will be the teachers, students/parents, and admins of SMAS Muhammadiyah. The goal of this project is to create an effective solution to enable the institution to convert its manual student management system into a computerized system.

1.3 Definitions, Acronyms and Abbreviation

- 1. SAMS - Student Academic Management System
- 2. STD - Software Testing Documentation

1.4 References

My.utm.my. (n.d.). Retrieved May 16, 2022, from <https://my.utm.my/home>

Aplikasi Rapor online - solusi. Sekawan Media. (2022, May 16). Retrieved May 16, 2022, from <https://www.sekawanmedia.co.id/solusi/aplikasi-rapor-online/>

Miller, J., Kostogriz, A., & Gearon, M. (Eds.). (2009). Culturally and linguistically diverse classrooms: New dilemmas for teachers. Bristol, UK: Multilingual Matters.

Dinustek. (n.d.). Solusi Rapor Sekolah online. evaluation. Retrieved May 16, 2022, from <https://evaluation.id/>

1.5 System Overview

This STD is divided into four chapters. The first chapter is the introduction which consists of the purpose and scope of the project. It also includes all the acronyms/abbreviation descriptions and references that exist in this STD and a short overview of the whole project. For the second chapter, The STD focused on the test cases for the system. This chapter delves into the test cases that will be used when testing the system. In the third chapter, the analysis of the test approach is described. It consists of which technique or method that is used when testing the system and whether it will be successful or a failure. Chapter 4 of this SDD includes the design of data which consist of Entity Relationship Diagram and data dictionary. The fifth chapter shows the mockup user interface design for the system. In the final chapter it is stated which system component satisfy the SRS.

2. Test Cases

2.1 Test TC001 for Module Authentication: <Sign In (UC001)>

TC001_01

Action	Input data	Expected result	Actual result
Input Username	abc123	Username Field is filled	Username Field is filled
Input Password	qwerty123	Password is field is filled	Password is field is filled
Click Login	-	Successfully enter the system	Successfully enter the system

2.2 Test TC002 for Module Report: <Manage Academic Report (UC002)>

TC002_01

Action	Input data	Expected result	Actual result
Click the “Rapor” menu	-	The list of students displayed	The list of students displayed
Input the academic year, semester, class and name	Year: 2021/2022 Semester : Ganjil Class: XI MIPA Name: Sheila	The specific student is displayed	The specific student is displayed
Click the “Tinjau”/”Download” button or the “Trash” icon	-	The academic report is displayed/downloaded/deleted	The academic report is displayed/downloaded/deleted

TC002_02

Action	Input data	Expected result	Actual result
Click the “Buat Rapor” menu	-	The list of students displayed	The list of students displayed
Search the specific student	Sheila	The specific student is displayed	The specific student is displayed
Click the “Buat Rapor” button	-	The create academic report form is displayed	The create academic report form is displayed

Input the extracurricular	Pramuka	The extracurricular is field is filled	The extracurricular is field is filled
Select the extracurricular mark	A	The extracurricular mark is selected	The extracurricular mark is selected
Click the “Selanjutnya” Button	-	The next create academic report form is displayed	The next create academic report form is displayed
Select the spiritual attitude mark	A	The spiritual attitude mark is selected	The spiritual attitude mark is selected
Fill in the spiritual attitude description	Selalu berdoa sebelum melakukan kegiatan	The spiritual attitude is field is filled	The spiritual attitude is field is filled
Select the social attitude mark	A	The social attitude mark is selected	The social attitude mark is selected
Fill in the social attitude description	Mulai konsisten perilaku jujur	The social attitude is field is filled	The social attitude is field is filled
Click the “Selanjutnya” button	-	The confirmation page is displayed	The confirmation page is displayed
Click the “Buat Rapor” button	-	The academic report is created	The academic report is created

TC002_03

Action	Input data	Expected result	Actual result
Click the “Penilaian” menu	-	The marking form is displayed	The marking form is displayed
Select the academic year, semester, class, subject, aspect.	Year: 2021/2022 Semester: Ganjil Class: XI MIPA Name: Sheila Subject: Matematika Aspect: Keterampilan	The form option is selected	The form option is selected
Input the “KKM”, mark and description field	KKM: 70 Mark: 84 Description: Siswa terampil	KKM, description and mark field are filled	KKM, description and mark field are filled

	dalam mengerjakan soal		
Click the “Simpan” button	-	The mark is saved	The mark is saved

TC002_04

Action	Input data	Expected result	Actual result
Click the “Ubah Penilaian” menu	-	The marked student list is displayed	The marked student list is displayed
Select the student and mark aspect.	Name: Sheila Aspect: Keterampilan	The form option is selected	The form option is selected
Click the “Ubah Nilai” button	-	The edit mark form is displayed	The edit mark form is displayed
Select the academic year, semester, class, subject, aspect.	Year: 2021/2022 Semester: Ganjil Class: XI MIPA Name: Sheila Subject: Matematika Aspect: Keterampilan	The form option is selected	The form option is selected
Input the “KKM”, mark and description field	KKM: 70 Mark: 84 Description: Siswa terampil dalam mengerjakan soal	KKM, description and mark field are filled	KKM, description and mark field are filled
Click the “Simpan” button	-	The mark is saved	The mark is saved

2.3 Test TC003 for Module Report: <View Academic Report (UC003)>
TC003_01

Action	Input data	Expected result	Actual result
Click the “Rapor” menu	-	The list of students displayed	The list of students displayed
Input the academic year, semester and class	Year: 2021/2022 Semester: Ganjil Class: XI MIPA	The specific student is displayed	The specific student is displayed
Click the “Tinjau” Button	-	The academic report is displayed	The academic report is displayed

2.4 Test TC004 for Module Report: <Download Academic Report (UC004)>
TC004_01

Action	Input data	Expected result	Actual result
Click the “Rapor” menu	-	The list of students displayed	The list of students displayed
Input the academic year, semester and class	Year: 2021/2022 Semester: Ganjil Class: XI MIPA	The specific student is displayed	The specific student is displayed
Click the “Download” Button	-	The academic report is downloaded	The academic report is downloaded

2.5 Test TC005 for Module Attendance: <Manage Attendance Record (UC005)>

TC005_01

Action	Input data	Expected result	Actual result
Click the “Absensi” menu	-	The list of students displayed	The list of students displayed
Select the date, total hour and class	Date: 12/12/2021 Total hour: 2 Class: XI MIPA	The form option is selected	The form option is selected
Select one of the student presences	S	The form option is selected	The form option is selected
Click the “Simpan” Button	-	The attendance record is saved	The attendance record is saved

2.6 Test TC006 for Module Attendance: <View Attendance Record (UC006)>

TC006_01

Action	Input data	Expected result	Actual result
Click the “Absensi” menu	-	The attendance record view displayed	The attendance record view displayed
Select the date range, and class	Date: 12/12/2021 – 15/01/2022 Class: XI MIPA	The attendance record is displayed	The attendance record is displayed

2.7 Test TC007 for Module Schedule: <Manage Timetable (UC007)>

TC007_01

Action	Input data	Expected result	Actual result
Click the “Jadwal Pelajaran” menu	-	The timetable UI is displayed	The timetable UI is displayed
Select the class	Class: XI MIPA	The Timetable is displayed	The Timetable is displayed

TC007_02

Action	Input data	Expected result	Actual result
Click the “Ubah Jadwal Pelajaran” menu	-	The edit timetable UI is displayed	The edit timetable UI is displayed
Select the hour and subject	Hour: 1 Subject: Matematika	The form option is selected	The form option is selected
Click the “Simpan” Button	-	The attendance record is saved	The attendance record is saved

2.8 Test TC008 for Module Schedule: <View Timetable (UC008)>

TC008_01

Action	Input data	Expected result	Actual result
Click the “Jadwal Pelajaran” menu	-	The timetable is displayed	The timetable is displayed

2.9 Test TC009 for Module Schedule: <Download Timetable (UC009)>

TC009_01

Action	Input data	Expected result	Actual result
Click the “Jadwal Pelajaran” menu	-	The timetable UI is displayed	The timetable UI is displayed
Click the “Unduh” button	-	The Timetable is downloaded	The Timetable is downloaded

2.10 Test TC010 for Module Account: <Manage Tuition Record (UC010)>

TC010_01

Action	Input data	Expected result	Actual result
Click the “SPP” menu	-	The list of students displayed	The list of students displayed
Select the academic year, semester, class and name	Year: 2021/2022 Semester: Ganjil Class: XI MIPA Name: Sheila	The specific student is displayed	The specific student is displayed
Choose one of the students and click “Lihat SPP”	-	The student's tuition record is displayed	The student's tuition record is displayed
Click the “Lihat Bukti” button	-	The uploaded payment proof by the student is displayed	The uploaded payment proof by the student is displayed
Change the status to approve the payment	Lunas	The Tuition status is approved	The Tuition status is approved

2.11 Test TC011 for Module Account: <View Tuition Record (UC011)>

TC011_01

Action	Input data	Expected result	Actual result
Click the “SPP” menu	-	The tuition record UI is displayed	The tuition record UI is displayed
Select the academic year, semester	Year: 2021/2022 Semester: Ganjil	The specific tuition record is displayed	The specific tuition record is displayed
Click the “Lihat Bukti” button	-	The uploaded payment proof by the student is displayed	The uploaded payment proof by the student is displayed

2.12 Test TC012 for Module Account: <Download Tuition Record (UC012)>

TC012_01

Action	Input data	Expected result	Actual result
Click the “SPP” menu	-	The tuition record UI is displayed	The tuition record UI is displayed
Select the academic year, semester	Year: 2021/2022 Semester: Ganjil	The specific tuition record is displayed	The specific tuition record is displayed
Click the “Download” button	-	The tuition record is downloaded	The tuition record is downloaded

2.13 Test TC013 for Module Account: <Make a Payment (UC013)>

TC013_01

Action	Input data	Expected result	Actual result
Click the “SPP” menu	-	The tuition record UI is displayed	The tuition record UI is displayed
Click the “Pembayaran SPP” button	-	The tuition record is downloaded	The tuition record is downloaded
Select the academic year, semester	Year: 2021/2022 Semester: Ganjil	the academic year and semester are selected	the academic year and semester are selected
Check the month that want to be paid	Januari. Februari	The specific month is checked	The specific month is checked
Click “Upload Bukti” Button	-	The prompt to upload the payment proof is displayed	The prompt to upload the payment proof is displayed
Upload the proof of payment file	Payment.png	The file is uploaded	The file is uploaded
Click “Simpan” button	-	The payment proof is saved	The payment proof is saved

2.14 Test TC014 for Module Account: <Manage Profile (UC014)>

TC014_01

Action	Input data	Expected result	Actual result
Click Profile menu	-	The user profile is displayed	The user profile is displayed
Click “Ubah Profile” Button	-	The user can edit their profile	The user can edit their profile
Change the email address	sheila@example.com	The edit email address field is filled	The edit email address field is filled
Click “Simpan” button	-	The profile is updated	The profile is updated

2.15 Test TC015 for Module Account: <Manage User (UC015)>

TC015_01

Action	Input data	Expected result	Actual result
Click the “Manajemen Akun” menu	-	The account management UI is displayed	The account management UI is displayed
Select the user type	Guru	The list of teacher user account displayed	The list of teacher user account displayed
Select the homeroom teacher	XI MIPA	The selected teacher became the homeroom of the selected class	The selected teacher became the homeroom of the selected class
Click “Trash” icon	-	The user account is deleted	The user account is deleted

2.16 Test TC016 for Module Account: <View Student Info (UC016)>

TC016_01

Action	Input data	Expected result	Actual result
Click the “Daftar Siswa” menu	-	The list of students is displayed	The list of students is displayed
Select the academic year, semester and class	Year: 2021/2022 Semester: Ganjil Class: XI MIPA	The list of specified students displayed	The list of specified students displayed
Click “Lihat Detail” button	-	The detailed information of the student is displayed	The detailed information of the student is displayed

2.17 Test TC017 for Module Account: <View Teacher Info (UC017)>

TC017_01

Action	Input data	Expected result	Actual result
Click the “Daftar Teacher” menu	-	The list of students is displayed	The list of students is displayed
Click “Lihat Detail” button	-	The detailed information of the student is displayed	The detailed information of the student is displayed

Appendix D Gannt Chart



Figure D.1: PSM 1 Gantt Chart

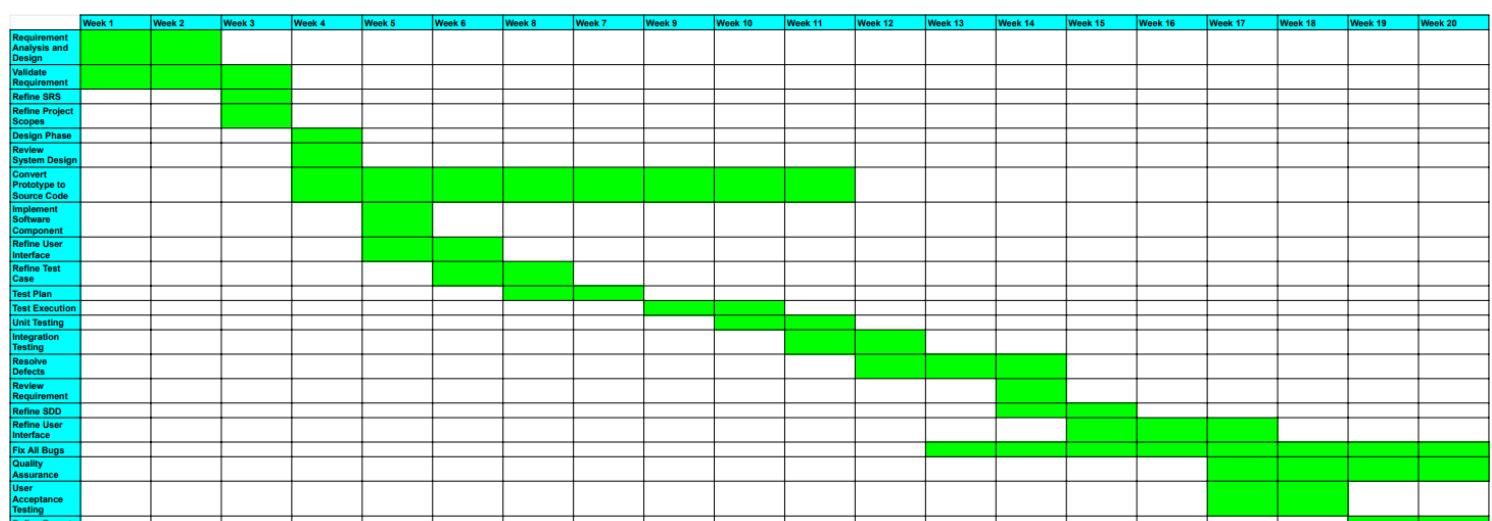


Figure D.2: PSM 2 Gantt Chart