UTM UNIVERSITI TEKNOLOGI MALAYSIA
INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING
ISSN 2180-4370
Journal Homepage : https://ijic.utm.my/

STUDENT ACADEMIC MANAGEMENT SYSTEM

Muhammad Darlen Sava
Faculty of Computing
Universiti Teknologi Malaysia
81310 Skudai, Johor, Malaysia
d.sava2001@graduate.utm.my

Mohd Shahizan bin Othman
Faculty of Computing
Universiti Teknologi Malaysia
81310 Skudai, Johor, Malaysia
shahizan@utm.my

***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.**

I. INTRODUCTION

With the advent of modern technology, many tasks have been automated across numerous sectors, including academia. Globalization, technological advancements, and research into improved teaching environments have led to the development of comprehensive systems for managing student, teacher, and institutional data. However, some areas, such as student academic management, still rely on manual processes, particularly in Aceh, Indonesia. This paper proposes a web-based Student Academic Management System to simplify teachers' tasks, save time, and provide student data access to students and their parents.

SMAS Muhammadiyah, a private high school in Banda Aceh City, Indonesia, still uses manual methods for managing academic data. Teachers manually update report cards every semester, resulting in periods of inaccessibility for students and parents. Furthermore, tuition management and attendance tracking are also done manually, making the process inefficient and prone to human error. Decentralized data storage and manual timetable management add to the challenges.

This project aims to facilitate the transition from a manual student management system to an automated one. The project aims to identify system requirements, design and develop the student academic management system based on stakeholder requirements, and test system functionalities. The project focuses on digitizing student data management using a website for SMAS Muhammadiyah 1. The users will be teachers, students, and school admins. This project will enhance institutional efficiency, boost teachers' productivity, facilitate data access for students and parents, and reduce the risk of human error.

II. LITERATURE REVIEW

The proposed system is informed by a case study from SMAS MUHAMMADIYAH 1 BANDA ACEH, a private high school in Indonesia. Their current manual operations for managing academic data highlight the need for a more streamlined, centralized process.

The current system at SMAS MUHAMMADIYAH 1 BANDA ACEH relies on various platforms such as Microsoft

Word, Microsoft Excel, and WhatsApp, which leads to data decentralization. The proposed system aims to address this by establishing a centralized database.

We explored three similar systems: the UTM Academic Management System, the Evaluation application, and the Sekawan Media Aplikasi Rapor Online. Each system has unique features, but none encompasses all the desired functionalities for our proposed system.

A comparative table was created to contrast the existing systems against the proposed one, highlighting the latter's comprehensive feature set.

TABLE I. COMPARISON BETWEEN EXISTING SYSTEM AND PROPOSED SYSTEM

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
Tuition Record Management	Yes	No	No	Yes
Attendance Record Management	No	No	Yes	Yes
Centralized Database	Yes	Yes	Yes	Yes

III. SYSTEM DEVELOPMENT METHODOLOGY

Selecting the appropriate system development methodology is critical to any project, as various methods offer unique benefits and disadvantages.

The Agile Scrum methodology, a part of the Software Development Life Cycle (SDLC), was chosen for this project due to its popularity, focus on teamwork, customer satisfaction, and iterative nature. This methodology enables rapid, incremental system delivery, allowing stakeholders to provide ongoing feedback and modify requirements as necessary. Agile methodology incorporates six key stages: Requirements, Design, Development, Testing, Deployment, and Feedback. Each stage will be explained below:

1) *Requirements:* This stage involves collecting stakeholder requirements crucial for shaping the project's direction and cost.

2) *Design:* This phase delves deeper into the project's details, incorporating scope, objectives, tools, and system requirements.

3) Development: The system is transformed from requirements and design into a functional system.

4) Testing: The product is tested and prepared for deployment.

5) Deployment: The system is released, with continuous updates possible for newly found bugs.

6) *Feedback:* Users' feedback is gathered for system improvement.

Various technologies are employed in this project, including Windows 10, Google Drive, Figma, Draw.io, Visual Studio Code, Laravel, PHP, Amazon S3, Amazon EC2, Vue.js, Tailwind, PostgreSQL, and pgAdmin.

The system necessitates both software and hardware requirements. The software requirements include Windows 8 or later, macOS Sierra 10.12 or later, and 64-bit Ubuntu 14.04+; along with compatible web browsers. The hardware requirements entail a suitable processor, 512MB (2GB for 64-bit) RAM, and 200MB hard drive space.

IV. REQUIREMENT ANALYSIS AND DESIGN

The system's functional and non-functional requirements are examined from the perspective of stakeholders. Functional requirements describe the system's main features and behaviors, with diagrams to illustrate these. Non-functional requirements address constraints, focusing on system availability, security, usability, and portability. Below is the use case diagram of the system:

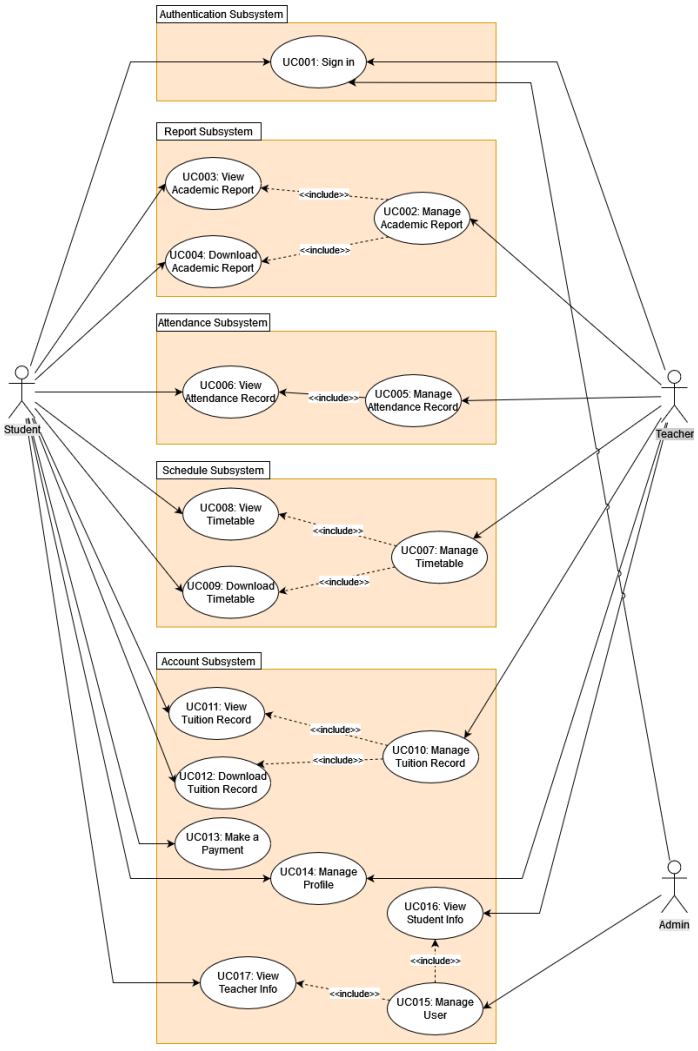

FIGURE I. USE CASE DIAGRAM OF STUDENT ACADEMIC MANAGEMENT SYSTEM

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 discussed in the next subchapter.

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.

The proposed system uses the Model-View-Controller (MVC) architecture, ensuring flexibility and facilitating code repetition. This design contributes to efficient system development, since alterations to one component don't necessarily affect others.

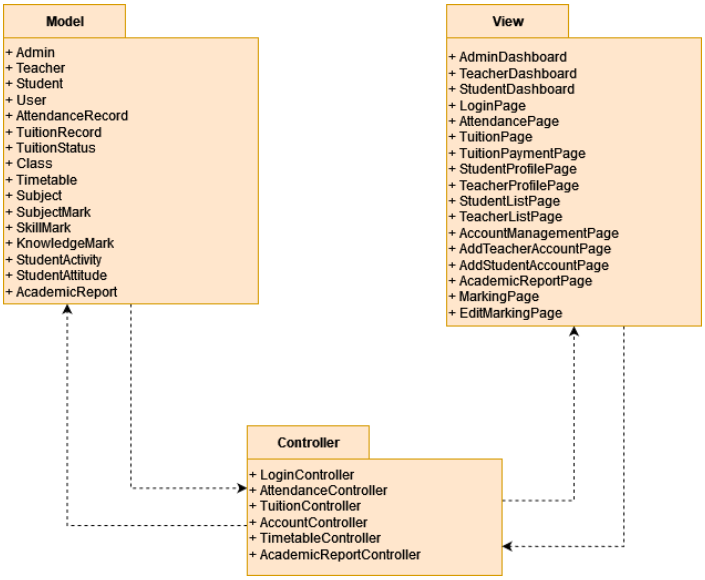

FIGURE II. ARCHITECTURE MODEL DIAGRAM OF STUDENT ACADEMIC MANAGEMENT SYSTEM

V. IMPLEMENTATION AND TESTING

The SAMS website is developed using the Vue.js framework, a widely-used JavaScript framework, to construct the frontend interfaces. This framework allows for creating intuitive, engaging user experiences.

On the backend, Laravel, which is a PHP-based web framework, is used. Laravel is recognized for its elegant syntax and capacity to facilitate tasks such as routing, sessions, and caching, which are essential for robust web applications.

Inertia.js, an innovative technology, is used as the bridge between the back and front ends. It enables developers to build single-page applications using classic server-side routing and controllers.

The website's visual appeal and responsive design are enhanced using PrimeVue and Tailwind CSS. PrimeVue provides a rich set of open-source Vue components, while Tailwind CSS is a utility-first CSS framework that allows for deep design customization.

For managing the data, PostgreSQL is chosen as the database system. As an object-relational database system, PostgreSQL offers advanced features such as complex queries, foreign keys, views, transactional integrity, and multi-version concurrency control.

As for code development, rigorous standards are followed to ensure the code's clarity, efficiency, and maintainability. Regular code reviews and testing are incorporated into the development process to catch and resolve issues early, ultimately ensuring a high-quality and reliable system.

Below are figures that display the User Interfaces of some of the main functions.

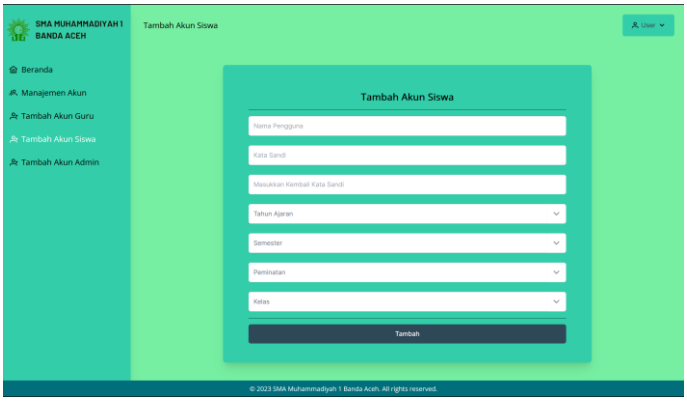

FIGURE III. ADD STUDENT PAGE FOR USER ADMIN

The figure above shows the user interface for add student page for admin. Here the admin can add a new student account with the corresponding inputs. Later the student can fill additional info in their profile page.

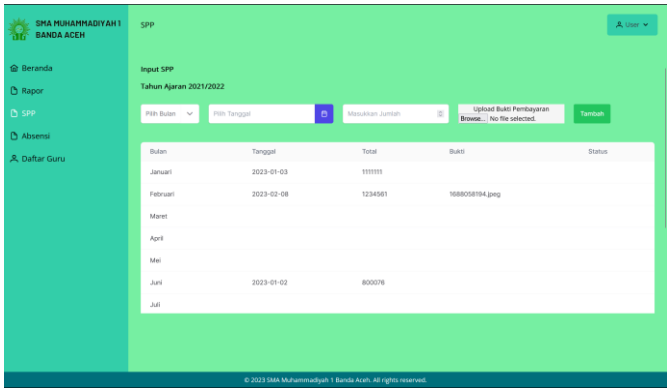

FIGURE IV. MANAGE TUITION PAGE FOR USER STUDENT

The figure above shows the user interface for manage tuition page for student. Here the student can view their existing tuition record or make a payment by providing the corresponding inputs and uploading the proof of payment. Later the status of the tuition will be verified by the teacher.

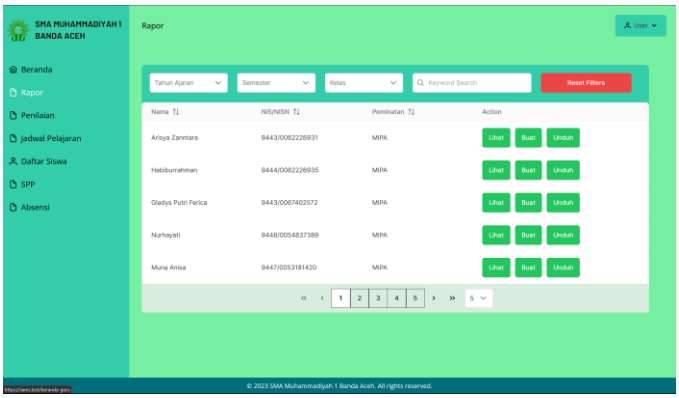

FIGURE V. MANAGE TUITION PAGE FOR USER STUDENT

The figure above shows the user interface for manage academic report page for teacher. Here the teacher can view the student list. The teacher can also use the filter and search function to find the corresponding student. Then they can choose whether to view the academic record of that student or to create the academic report.

The system is assessed using several methodologies, including Black box, White box, and User testing to uncover errors and bugs. These tests ensure that the system aligns closely with the requirements.

1) *Black Box Testing:* Black box testing involves examining the functionality of the application without considering its internal structures or workings. This type of testing focuses on the system's input and output. It checks the system's behavior when performing specific sequences (System Flow), its response to different inputs and expected outputs (Input Output Verification), and its reactions to incorrect data and error messages (Error Message Testing).

2) *White Box Testing:* White box testing requires knowledge of the system's internal structure, design, and implementation. It's used to validate the system's internal operations, including code structures, code flows, and internal system logic. Specific examples are given under the User Testing subchapter.

3) *User Testing:* Real users perform User Testing to evaluate the system. It provides direct insight into user interaction with the system and the satisfaction level of these users. The testing result is presented in Table II.

TABLE II. 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

VI. CONCLUSION

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.

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.

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:

1) Integrated payment system for tuition payment, so that the payment system is more efficient and secure

2) Integration with third party LMS (Learning Management System)

3) Integration with email services such as notification.

ACKNOWLEDGMENT

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 family that help kept my spirit up until this moment.

REFERENCES

[1] *My.utm.my*. (n.d.). Retrieved May 16, 2022, from https://my.utm.my/home

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

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

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