Applied Information Technology And Computer Science Vol. 4 No. 4 (2023) 1503-1522

AITCS

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN :2773-5141

Aluna: A Student Management System for Tuition Uni Pintar

Muhammad Safwan Sofian[1], **Mohd Zainuri Saringat**[1*],

[1] Fakulti Sains Komputer dan Teknologi Maklumat
Universiti Tun Hussein Onn Malaysia, Parit Raja, 86400 Batu Pahat, Johor, Malaysia

DOI: https://doi.org/10.30880/aitcs.2023.04.02.085
Received 23 June 2023; Accepted 09 November 2023; Available online 30 November 2023

Abstract: The web-based Student Management System for Tuition Uni Pintar enhances the efficiency of existing manual processes by implementing a structured and systematic approach. Developed for teachers and students, Aluna replaces the conventional paper-based information storage with a robust system. Employing the Evolutionary Prototyping Model, the system utilizes software such as HTML, PHP, Java, JS, Visual Code, and a MySQL server for data storage. The implementation of Aluna optimizes the tuition center's operations, providing better systematic management and improving teacher efficiency. Accessible through laptops or computers, the system enables teachers to manage student information seamlessly. This report highlights the features and benefits of Aluna, emphasizing its potential to revolutionize tuition center management and enhance productivity.

Keywords: Student Management System, Tuition Uni Pintar, web-based system, Evolutionary Prototyping Model, data storage, efficiency, productivity.

1. Introduction

Aluna is a web-based Student Management System designed to support teachers in efficiently managing their students at Tuition Uni Pintar. The existing manual process of student management lacks a systematic approach, leading to challenges in storing and retrieving student information. Aluna aims to address these issues by providing a practical and organized solution. The system is developed using HTML, PHP, CSS, and XAMPP, with MySQL as the database. Accessible through web browsers, Aluna offers functions such as attendance recording, student information display, and grade entry. The secure storage of data in a centralized database eliminates the need for physical documents, saving space and resources. Tuition Uni Pintar is a learning centre catering high school students, aiming to enhance their academic achievements. The dedicated teachers strive to provide guidance and support to underachieving students.

The main objective of Aluna is to provide teachers with a user-friendly interface for managing student information systematically. The system allows teachers to record attendance, access comprehensive student profiles, and manage grades efficiently. Students have limited features,

*Corresponding author: zainuri@uthm.edu.my

including attendance marking, viewing test marks, and managing their profiles. The proposed system overcomes the limitations of the current manual process by centralizing information and providing a clean interface for easy access. With Aluna, teachers can effectively manage student data, saving time and effort in locating specific documents. The project's scope includes three main user roles: admin, teachers, and students. The admin has full control over managing student and teacher accounts, while teachers can manage their students' information and record attendance. Students can view their profiles, mark attendance, and access their test marks.

Aluna's web-based Student Management System aims to streamline student management processes at Tuition Uni Pintar. By providing a systematic approach to data management, the system enhances efficiency and reduces the reliance on physical documents. With its user-friendly interface, Aluna empowers teachers to effectively manage student information and ultimately supports the centre's mission of improving academic achievements.

2. Literature Review

2.1 Management Systems

A management system plays a crucial role in helping organizations effectively achieve their goals by managing various aspects of their operations. These objectives can encompass areas such as quality, efficiency, environmental performance, and workplace safety. In the case of Aluna, the system offers a systematic and improved approach to storing and managing tuition-related information and data. The complexity of a management system depends on the specific context of the organization. Smaller organizations may rely on strong leadership to define expectations and individual contributions without extensive documentation. On the other hand, highly regulated industries often necessitate comprehensive documentation and controls to meet legal obligations. Aluna, while not overly complex, incorporates all the necessary features to fulfill its intended purpose.

Web-based systems, like Aluna, present cost advantages and convenience. They can be accessed from any computer with an internet connection, eliminating the need for individual installations. Deploying web systems is also simplified as users only require a website address to access the system, streamlining the deployment process. Consequently, Aluna offers flexible accessibility due to its web-based nature. By leveraging the benefits of being a web system, Aluna enables users to access the system more conveniently. It provides a user-friendly interface that facilitates efficient storage and management of tuition-related information. The system streamlines administrative tasks, enhances data organization, and contributes to improved decision-making processes. Furthermore, Aluna's web-based architecture allows for scalability and adaptability as the organization's needs evolve over time.

In summary, Aluna serves as a valuable tool for efficiently managing tuition-related information. While it may not possess the complexity of more intricate systems, it offers all the necessary features required to fulfill its purpose. With its web-based accessibility, Aluna brings flexibility, convenience, and streamlined operations to Tuition Uni Pintar.

2.2 Web-Based Systems

A web-based system refers to a software program that is accessed via a network connection instead of being stored locally on a device. These systems operate within web browsers or can be client-based, where a portion of the program is downloaded to the user's desktop while processing occurs on an external server through the internet. While the terms web-based, internet-based, and cloud-based can be confusing, they all involve communication with users through computer screens and offer numerous advantages. Web-based systems provide users with easy access by simply entering a URL address in a search engine like Google, eliminating the need for complex connection procedures. This immediate accessibility offers immediate benefits. Recognizing the incredible advantages of web-based systems, Aluna, a student management system for Tuition Uni Pintar, has been developed to cater to the specific

needs of the tuition center. Aluna leverages the perks of a web system, providing essential features to support tuition management. Aluna streamlines the management of student-related information, ensuring efficient organization and facilitating administrative tasks. Its web-based nature allows for flexibility, scalability, and adaptability as the tuition center's requirements evolve. By leveraging the power of web technology, Aluna enhances connectivity and simplifies user interaction, ultimately improving the overall management processes of Tuition Uni Pintar.

Web-based systems offer convenient access to software programs through network connections. Aluna, as a web-based student management system, capitalizes on the benefits of web technology to support the efficient management of tuition-related information and tasks. Its implementation brings about improved connectivity, streamlined operations, and enhanced user experience for Tuition Uni Pintar.

2.3 Comparison of the existing system

The existing system study focuses on three selected systems: Bright Kids Tuition Centre Management Information System, Student Management System Development of Damya Deena Tuition Center, and Development of Tuition Centre Management System via Mobile App. These systems serve as valuable reference materials for the development of Aluna, the Student Management System for Tuition Uni Pintar. By studying these systems, insights and best practices can be gained to enhance the quality and functionality of Aluna. Leveraging the strengths and addressing the limitations observed in the reference systems, Aluna aims to provide a superior student management experience at Tuition Uni Pintar. The comparison between the 3 related systems and this project's system based on several features is summarized in Table 1.

Table 1: Comparison of Existing System and Proposed System

Features	Aluna: Student Management System for Tuition Uni Pintar	Bright Kids Tuition Centre Management Information System	Student Management System Development of Damya Deena Tuition Center	Development of Tuition Centre Management System via Mobile App
User Login	Yes	Yes	Yes	Yes
Target User	All	All	All	All
Account Requirement	Yes	Yes	Yes	Yes
The ability to register an account	Yes (performed by admin)	Account registration is not available	Yes (only for teachers and students)	Yes
Academic result reporting	Yes	No	Yes	Yes
Timetable module	Yes	No	No	Yes
Class registration	Yes	Yes	Yes	No

Features	Aluna: Student Management System for Tuition Uni Pintar	Bright Kids Tuition Centre Management Information System	Student Management System Development of Damya Deena Tuition Center	Development of Tuition Centre Management System via Mobile App
Attendance Module	Yes (by ticking or scanning qr)	No	No	Yes (by scanning qr)
Class member management	Yes	No	Yes	No
Fee payment	No	No	Yes	Yes
Search function	Yes	Yes	No	No
Programming language used	HTML, CSS, JavaScript, PHP	HTML, CSS, JavaScript/JQuery	HTML, PHP	HTML, PHP, JavaScript
Platform support	Web	Web	Web	Mobile

3. Methodology

The evolutionary prototyping model was adopted for the development of Aluna, allowing for continuous modifications and enhancements throughout the project. This iterative approach ensures the system remains technically and practically sound. By incorporating client feedback, the prototype evolves and undergoes multiple iterations, integrating improved or additional functionality. Aluna has undergone two cycles of evolutionary prototyping, thoroughly reviewing and validating requirements, and gaining approval for the implementation of all features.

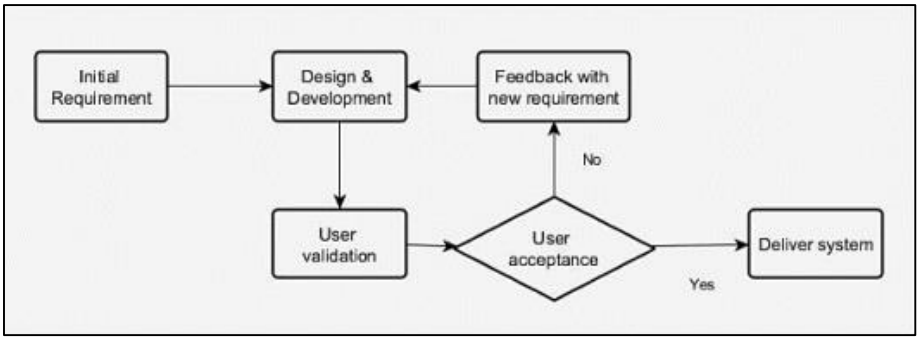

Figure 1: Evolutionary Prototyping Model

3.1 Project Planning

Table 2 shows the list of tasks performed at each phase in the Evolutionary Prototyping model for Aluna: A Student Management System For Tuition Uni Pintar.

Table 2: List of tasks in the Evolutionary Prototyping Model

Phase	Task
Initial Requirement	• Propose the project • Decide the scope, objectives, problem statements, significance and expected result of the project. • Determine the project schedule. • Interview the stakeholders. • Inspect the existing system. • Examine the software and hardware requirements. • Choose a suitable methodology. • Develop the use case diagram, activity diagram, sequence diagram, and class diagram.
Design & Development	• Design the system interfaces. • Design the database. • Develop the prototype. • Refine the prototype. • Develop the program code. • Develop and connect to the database. • Develop the system interfaces • System testing
User Validation	• Receive user validation
User Acceptance	• Achieve user acceptance
Feedback with New Requirement	• Collect feedback from stakeholders. • Examine new requirement if there is one.
Deliver System	• Deploy the system

3.2 Requirement Analysis

The requirement analysis helps in understanding and identifying the system requirements. Table 3 and Table 4 show the system's functional requirements and non-functional requirements respectively. The user requirement defines the system user demands and expectations for the system functions. Table 5 shows the system user requirements.

Table 3: Functional Requirements

No	Module	Function
1	Login	• The system shall allow all users to login with their correct email and password. • The system should only allow a valid email and password to be logged into the system. • The system shall alert if email or password field is left empty. • The system should redirect the user to their respective home page after a successful login.
2	Manage Attendance	• The system should record the attendance of students who are ticked. • The system should allow admin and teachers to toggle between the attendance table and the records table. • The system should take the attendance of the students who have scanned the qr code. • The system should allow teachers to display the qr code. • The system shall allow admin and teachers to print a report of the attendance record. • The system shall allow students to view their attendance and test marks.
3	Manage Student	• The system should display a table of student information. • The system shall allow admin and teachers to search students by their name or id. • The system should allow admin and teachers to insert, update or delete students and their information. • The system shall allow admin and teachers to insert, update or delete test marks. • The system should allow admin and teachers to print a report of the students.
4	Manage Teacher	• The system shall allow admin to insert, update or delete teacher and their information.

No	Module	Function
		• The system shall allow admin to search teachers by their name or id. • The system should allow the admin to print a report of the teachers.
5	Manage Timetable	• The system should allow students and teachers to view their timetable according to their tuition sessions.
6	Manage Class	• The system should allow admin to insert, update or delete the classes. • The system shall allow the admin to assign the students to classes. • The system shall allow the admin to print a report of all the classes.
7	Manage Profile	• The system should retrieve the current user id to properly display their current information. • The system shall allow admin to change their password. • The system should allow teachers and students to change their personal information and password.

Table 4: Non-Functional Requirements

No.	Requirement	Description
1	Operational	The system should work on any device with a browser and internet connection.
2	Performance	The system should provide a reasonable operational and response time.
3	Usability	The flow of the system should be easily understood by all types of users.
4	Security	The login module accurately verifies valid users and denies invalid users from system access.
5	Integrity	The system database is privately protected and secured from corruption and outside harm to an extent.
6	Availability	The system is readily operational and can be used anytime.

Table 5 shows the user requirements that define the system user demands and expectations for the system functions.

Table 5: User Requirements

No	User requirements
1	All users should be able to login with their correct email and password
2	All users should be brought to their respective homepages based on their roles after login
3	Admin should be able to insert, update or delete teachers and their information
4	Admin should be able to search teachers based on their names or id
5	Admin should be able to print a report of all the teachers
6	Admin and Teachers should be able to insert, update or delete students and their information.
7	Admin and Teachers should be able to search for students by their name or id
8	Admin and Teachers should be able to print a report of all students
9	Admin and Teachers should be able to insert, update or delete student test marks
10	Admin and Teachers should be able to print a report on all the test marks
11	Admin and Teachers should be able to search for each test by their test name or mark id
12	Admin and Teacher should be able to filter the student attendance based on the class, time and date
13	Admin and Teachers should be able to record student attendance by ticking the checkboxes
14	Admin and Teachers should be able to toggle between the attendance table and the records table
15	Admin and Teachers should be able to search attendance by student name or id
16	Admin and Teachers should be able to print a report of all student attendance
17	Teachers should be able to display the qr code for student attendance
18	Students should be able to take their attendance by scanning the qr code
19	Admin should be able to insert, update or delete classes
20	Admin should be able to assign students to any class
21	Admin should be able to print a report of all the classes
22	Teachers and Students should be able to view their timetable based on their tuition sessions
23	All users should be able to manage their account information
24	All users should be able to logout from the system

3.3 System Analysis

The structure and behavior of the system are analyzed and presented using the UML diagrams. UML Use Case diagram defines a set of behaviors of the system as well as how the actors interact with the behaviors. Figure 2 shows the Use Case Diagram for the system.

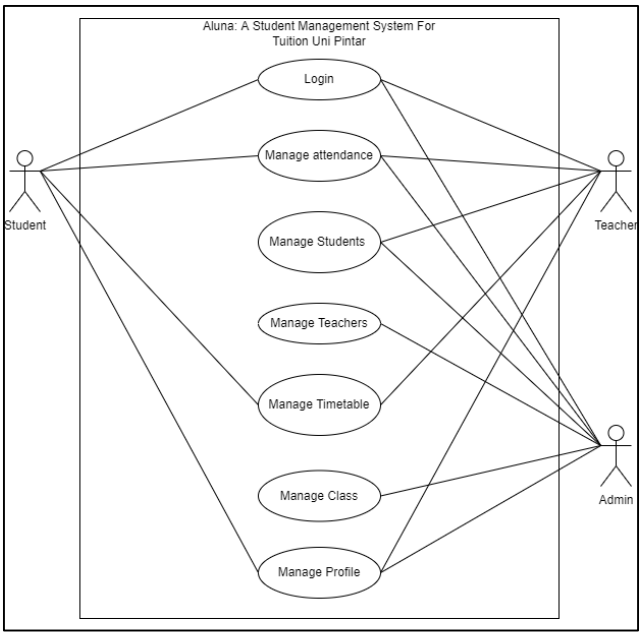

Figure 2: Use Case Diagram

Aluna is a comprehensive system designed to cater to the needs of three distinct user groups: administrators, students, and teachers. To ensure security and personalized access, all users are required to log in using their unique credentials. Once logged in, each user is directed to their respective homepage, tailored to their specific role and responsibilities within the system.

The administrator holds the highest level of access and authority in Aluna. Their primary responsibility is to manage various aspects of the system. This includes overseeing student and teacher profiles, organizing classes, and maintaining accurate attendance records. The administrator plays a crucial role in ensuring smooth operations and effective management of the entire academic ecosystem. Teachers, another user group in Aluna, have access to features that aid them in their teaching responsibilities. They can conveniently view their personalized timetable, allowing for effective scheduling and planning. Additionally, teachers have the ability to manage students within their assigned classes, enabling them to monitor attendance and take their attendance using a qr code. Students, the final user group, are provided with a user-friendly interface that grants them access to specific functionalities. While their access is limited compared to administrators and teachers, students still have important tools at their disposal. They can conveniently scan their attendance, ensuring accurate records. Additionally, students can access their test marks and view their personalized timetable, keeping them informed and organized.

Regardless of user type, Aluna ensures that all users have the ability to manage their accounts. This includes updating personal information, such as contact details or profile pictures, as well as modifying login credentials to maintain account security.

The class diagram shows the static view of the structure of the system and how the system components interact with each other. The class diagram is also useful for visualizing, describing, and documenting various system components. The class diagram for Aluna: A Student Management System For Tuition Uni Pintar is shown in Figure 3.

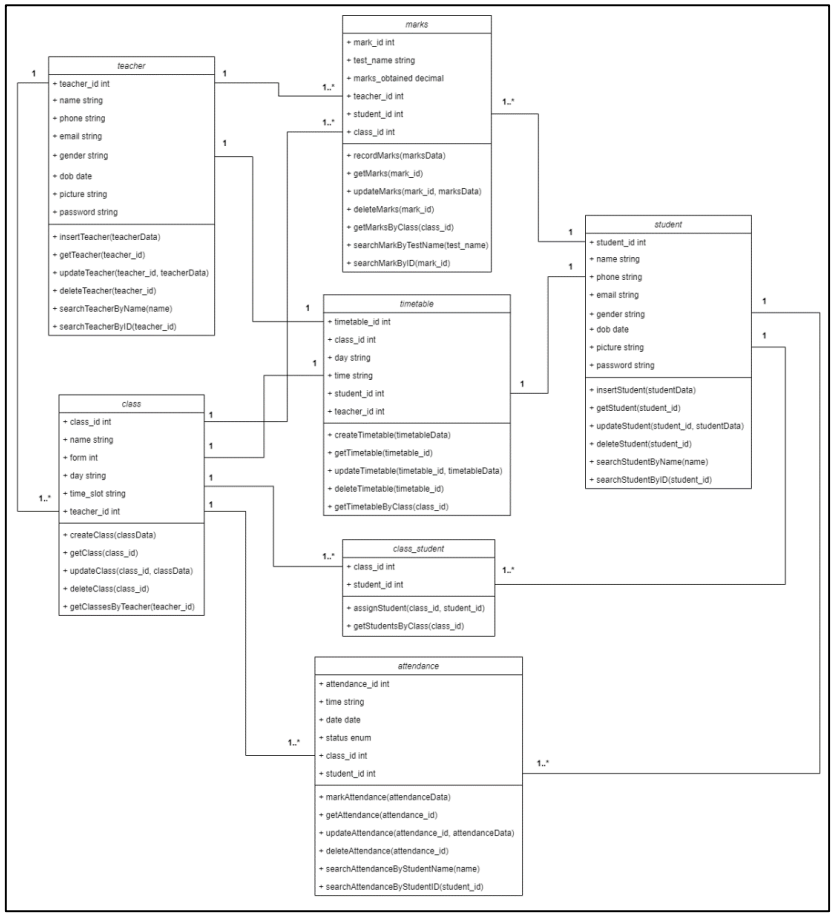

Figure 3: Class Diagram

3.4 System Design

The system architecture is a conceptual diagram that depicts the organization and operation of numerous system components and subsystems, including hardware, software, network devices, and other equipment. Figure 4 shows the system architecture for the system.

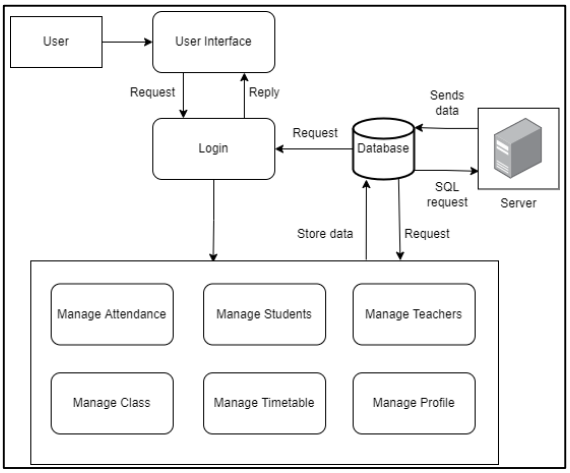

Figure 4: System architecture

The database is designed to receive and store data for the application. In this section, the database schema will describe the entities that hold data for the application. The database schema for the application is shown:

i. **Student** (student_id, name, phone, email, gender, dob, picture, password)

ii. **Teacher** (teacher_id, name, phone, email, gender, dob, picture, password)

iii. **Admin** (admin_id, email, password)

iv. **Attendance** (attendance_id, class_id, time, date, student_id, status)

v. **Class** (class_id, name, form, teacher_id, day, time_slot)

vi. **Class_Student** (class_id, student_id)

vii. **Marks** (mark_id, test_name, marks_obtained, class_id, teacher_id, student_id)

viii. **Timetable** (timetable_id, class_id, day, time, student_id, teacher_id)

4. Results and Discussion

Functionality testing and user acceptance testing are the two types of testing done. This section presents the result and discussion of the proposed system. These include test cases for proposed system results and user acceptance testing results. It details the specs and needs of the system. It is specifically employed during testing to determine whether the system requirements have been met. This helps with testing and requirements tracing to ensure that all requirements are tested.

4.1 Implementation

The Aluna student management system was implemented using HTML, CSS, JavaScript, and PHP. The user interface was developed without specific frameworks, allowing for customization and tailored user experience. JavaScript and PHP were used for backend functionalities, overcoming challenges through iterative testing. This implementation facilitated seamless integration between the frontend and backend components, ensuring efficient system operations within the student management context.

Figures 5 and 6 show one important aspect is the implementation of password hashing. The user's password is hashed using the **password_hash()** function in PHP. By applying a one-way hash algorithm, the original value of the password is obscured, providing an additional layer of security. This ensures that even if there is a breach or unauthorized access to the system, the actual passwords of the users remain protected.

```php
// Hash the password
$hashedPassword = password_hash($password, PASSWORD_DEFAULT);
```

Figure 5: Password Hashing Implementation

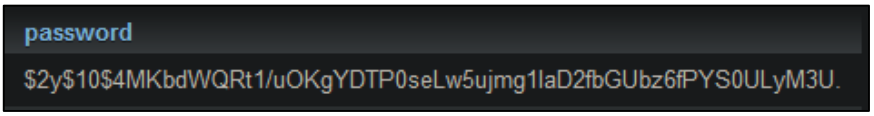

Figure 6: Encrypted Password Display (Database)

In Figure 7, an important implementation in the code involves the use of parametrized statements to enhance the security of database operations, specifically during the insertion of student records. This approach utilizes the `prepare()` function to create a parameterized SQL statement where question marks serve as placeholders for the values. By employing the `bind_param()` function, the actual values are securely bound to the placeholders, ensuring that they are treated as data rather than executable SQL code. This implementation effectively mitigates the risks associated with SQL injection attacks, as the

values are handled securely, and malicious code injection is prevented. As a result, the system's database operations maintain a higher level of security and integrity.

```php
// Prepare a parameterized statement for inserting the student record into the database
$stmt = $con->prepare("INSERT INTO student (student_id, name, phone, email, gender, dob, password) VALUES (?, ?, ?, ?, ?, ?, ?)");

// Bind parameters to the statement
$stmt->bind_param("sssssss", $id, $name, $phone, $email, $gender, $dob, $hashedPassword);
```

Figure 7: Parametrized Statements Implementation

Figure 8 shows a security measure to prevent caching of the login page. This is achieved by sending specific HTTP headers in the response. The Cache-Control header with directives like "no-store, no-cache, must-revalidate, max-age=0" instructs the browser not to store a cached version of the page and to revalidate it on every access. The Cache-Control header with "post-check=0, pre-check=0" disables browser caching checks. The Pragma header with the value "no-cache" ensures that intermediary servers and browsers do not cache the page. These headers collectively prevent the browser and intermediate systems from storing a cached version of the login page, reducing the risk of unauthorized access or outdated information being displayed.

```php
// Prevent caching of the login page
header("Cache-Control: no-store, no-cache, must-revalidate, max-age=0");
header("Cache-Control: post-check=0, pre-check=0", false);
header("Pragma: no-cache");
```

Figure 8: Security Headers for Preventing Caching

In this section, we delve into the implementation of various modules within the system, encompassing functionalities such as login, teacher management, student management, class management, attendance management, timetable management, and profile management.

Figure 9 presents the login interface for all users of the system. To access the system, users are required to enter their email and password in the designated fields. The login process includes validations to check for empty fields. If the login credentials are incorrect, an error message will be displayed indicating the login failure. Conversely, upon successful login, a confirmation message will be shown, indicating that the user has been successfully authenticated.

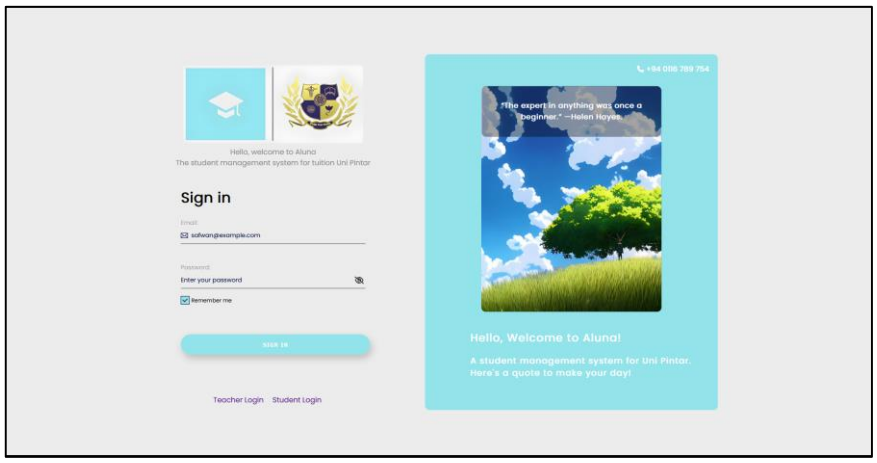

Figure 9: Login Interface

The interface depicted in Figure 10 represents the Manage Attendance module within the system. This module is accessible to all three types of users: admin, teacher, and student. The admin user can manage the attendance records, while the teacher user can mark attendance by selecting

checkboxes or using QR codes. On the other hand, the student user can scan QR codes and view their attendance status. This module provides a comprehensive interface for efficiently managing attendance records, catering to the specific needs of each user role.

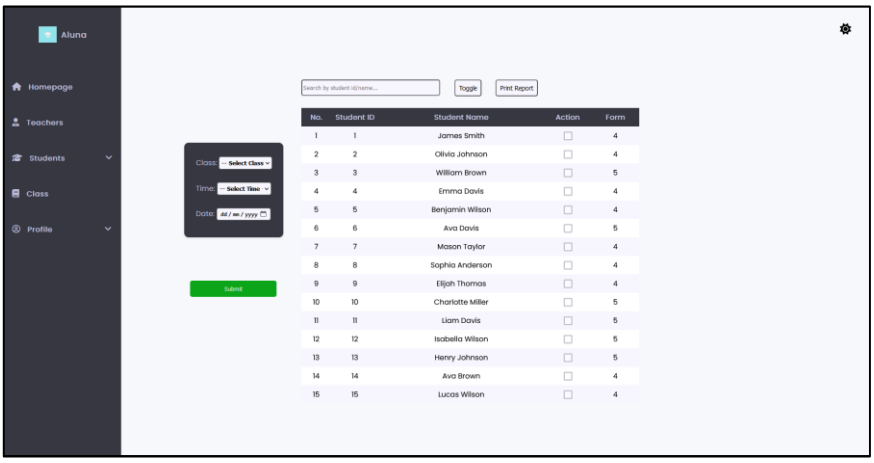

Figure 10: Manage Attendance Interface (Admin)

The interface displayed in Figure 11 represents the Manage Student module, which is accessible to both admin and teachers. This module facilitates various student management operations, including the ability to add, update, or delete student records and modify their information. Furthermore, users can generate comprehensive reports from the student data table for convenient reference. By providing a user-friendly interface, the Manage Student module streamlines the process of handling student-related tasks, ensuring efficient management of student information for both admin and teachers.

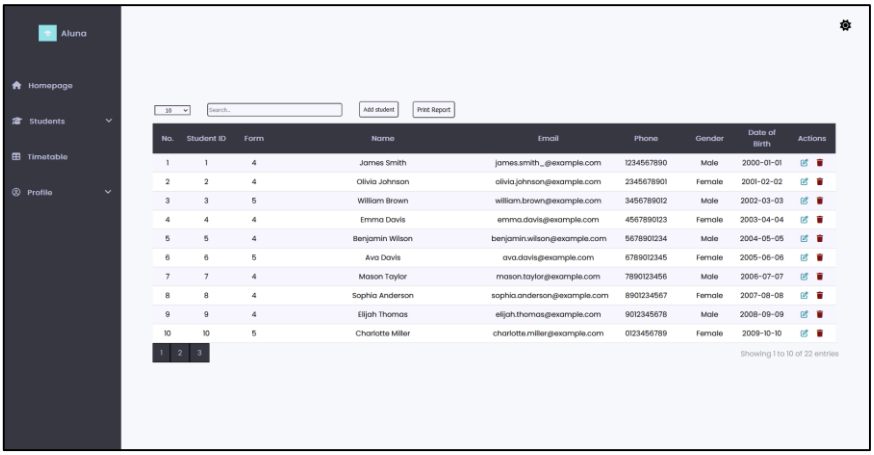

Figure 11: Manage Students Interface

Figure 12 illustrates the user interface of the Manage Teachers module, specifically designed for the admin role. This module empowers the admin to efficiently manage teacher-related tasks, including adding, updating, and deleting teacher records. Through this interface, the admin can conveniently handle teacher information and ensure accurate and up-to-date records within the system. By providing a dedicated module for managing teachers, the system streamlines administrative processes and enhances the overall efficiency of teacher management. The admin can easily navigate the interface and perform necessary actions to effectively oversee and maintain the teacher database.

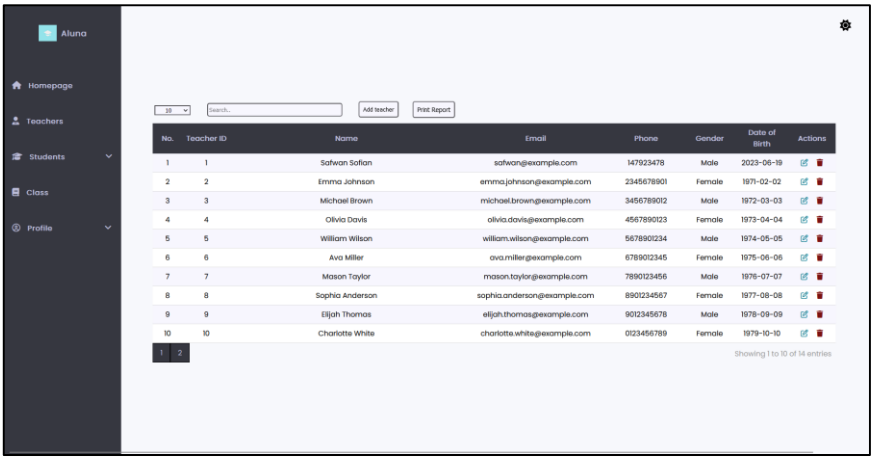

Figure 12: Manage Teachers Interface

Figure 13 presents the user interface of the Manage Timetable module, catering to both students and teachers. Upon logging into the system, the module dynamically retrieves and displays relevant class information for each user based on their role. If the user is a teacher, the timetable will showcase the classes they teach, while for students, it will feature the classes they attend. The timetable is structured according to the specific time and day of each class, providing a personalized and organized schedule for every user. By offering a personalized timetable, the system enhances user convenience and facilitates efficient management of classes and schedules. Users can easily refer to their timetable to stay updated on their upcoming classes and effectively plan their academic activities.

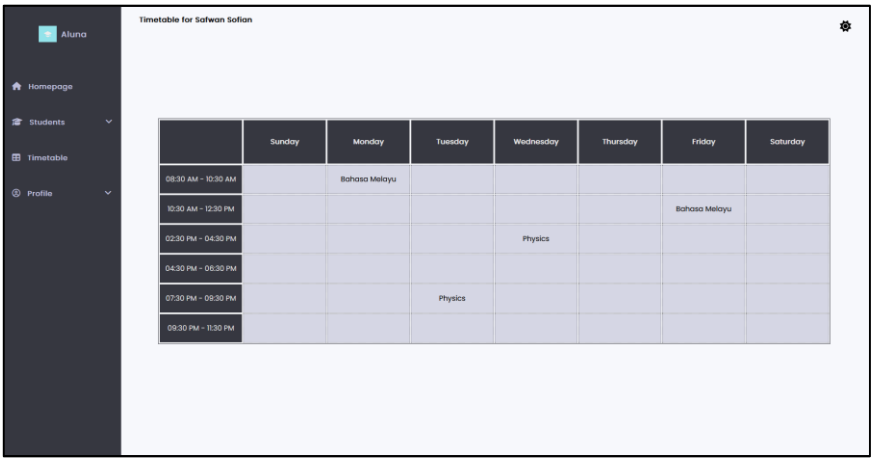

Figure 13: Manage Timetable Interface

Figure 14 illustrates the user interface of the Manage Class module, exclusively accessible to the admin role. Within this module, the admin is granted the authority to perform various operations related to class management. Specifically, the admin can insert new classes into the system, update existing class information, and delete classes as necessary. Furthermore, the admin possesses the capability to assign students to their respective classes, ensuring accurate and organized student allocation. By offering these functionalities, the Manage Class module empowers the admin to efficiently handle class-related tasks, streamline the student allocation process, and maintain a well-structured academic environment within the system.

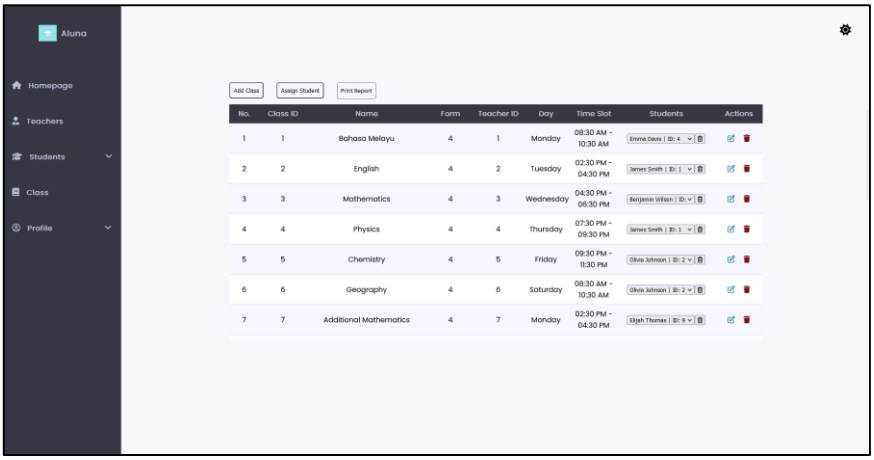

Figure 14: Manage Class Interface

Figure 15 depicts the user interface of the Manage Profile module. This module is designed to cater to the needs of individual users based on their logged-in credentials. Upon accessing this page, the system retrieves and displays the user's existing information from the database. Users are presented with their current profile details, such as name, email, and other relevant data. Additionally, users are granted the flexibility to update and modify their profile information, enabling them to keep their records accurate and up to date. Furthermore, the module offers users the option to change their password, ensuring account security and providing a convenient way to manage login credentials. The Manage Profile module prioritizes user-centric functionality, facilitating a personalized and customizable experience for everyone within the system.

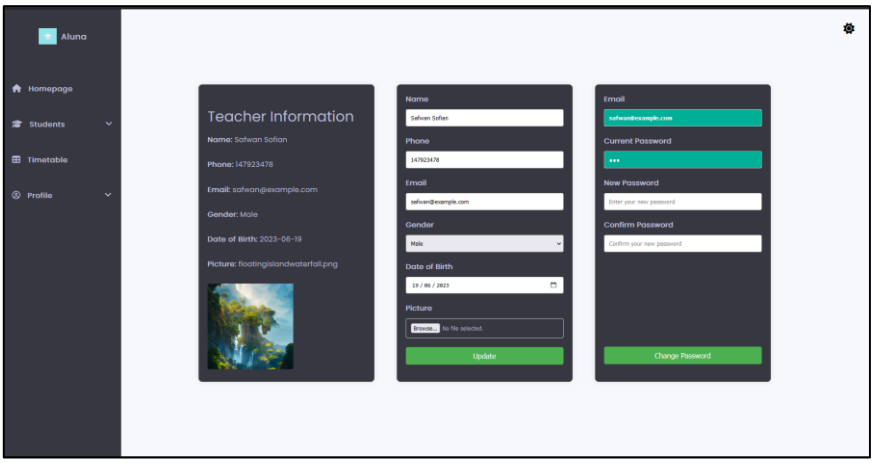

Figure 15: Manage Profile Interface

In conclusion, the Aluna Student Management System offers a comprehensive set of modules to facilitate efficient and streamlined management of various aspects related to students, teachers, classes, attendance, timetable, and user profiles. These modules provide a user-friendly interface, enabling administrators, teachers, and students to carry out their respective tasks effectively. The system empowers administrators with capabilities to manage student and teacher records, assign students to classes, and oversee overall system operations. Teachers benefit from features such as attendance management and timetable viewing, enhancing their ability to track and manage student activities. Students, on the other hand, can access their personal profiles, view attendance records, and manage their own information. The integration of these modules ensures a cohesive and intuitive user experience, ultimately supporting the goals of Tuition Uni Pintar in fostering academic achievements and providing a seamless administrative process.

4.2 Testing

The test case is based on the functional requirements of the proposed system. It involved comparing a system's behaviour and functionality to the expected results to ensure that it serves its intended purpose and can function properly. Table 6 shows the summary of the test case based on the functional testing.

Table 6: Summary of test case

Test ID	Requirement ID	Test Case Description	Result (Pass/Fail)
TC01	**FR01**	**Login**	
TC01-01	FR01-01	The system shall allow all users to login with their correct email and password.	Pass
TC01-02	FR01-02	The system should only allow a valid email and password to be logged into the system.	Pass
TC01-03	FR01-03	The system shall alert if email or password field is left empty.	Pass
TC01-04	FR01-04	The system should redirect the user to their respective homepage after a successful login.	Pass
TC02	**FR02**	**Manage Attendance**	
TC02-01	FR02-01	The system should record the attendance of students who are ticked.	Pass
TC02-02	FR02-02	The system should allow admin and teachers to toggle between the attendance table and the	Pass
TC02-03	FR02-03	The system should take the attendance of the students who have scanned the qr code.	Pass
TC02-04	FR02-04	The system should allow teachers to display the qr code.	Pass
TC02-05	FR02-05	The system shall allow students to view their attendance and test marks.	Pass
TC02-06	FR02-06	The system shall allow admin and teachers to print a report of the attendance record.	Pass
TC03	**FR03**	**Manage Student**	
TC03-01	FR03-01	The system should display a table of student information.	Pass
TC03-02	FR03-02	The system shall allow admin and teachers to search students by their name or id.	Pass

Test ID	Requirement ID	Test Case Description	Result (Pass/Fail)
TC03-03	FR03-03	The system should allow admin and teachers to insert, update or delete students and their	Pass
TC03-04	FR03-04	The system shall allow admin and teachers to insert, update or delete test marks.	Pass
TC03-06	FR03-06	The system should allow admin and teachers to print a report of the students.	Pass
TC04	**FR04**	**Manage Teacher**	
TC04-01	FR04-01	The system shall allow admin to insert, update or delete teacher and their information.	Pass
TC04-02	FR04-02	The system shall allow admin to search teachers by their name or id.	Pass
TC04-03	FR04-03	The system should allow the admin to print a report of the teachers.	Pass
TC05	**FR05**	**Manage Timetable**	
TC05-01	FR05-01	The system should allow students and teachers to view their timetable according to their	Pass
TC06	**FR06**	**Manage Class**	
TC06-01	FR06-01	The system should allow admin to insert, update or delete the classes.	Pass
TC06-02	FR06-02	The system shall allow the admin to assign the students to classes.	Pass
TC06-03	FR06-03	The system shall allow the admin to print a report of all the classes.	Pass
TC07	**FR07**	**Manage Profile**	
TC07-01	FR07-01	The system should retrieve the current user id to properly display their current information.	Pass
TC07-02	FR07-02	The system shall allow admin to change their password.	Pass
TC07-03	FR07-03	The system should allow teachers and students to change their personal information and	Pass

User acceptance testing is then performed with the anticipated user. The system's module and interface design are the aspects that are being tested. The test will be conducted via a questionnaire form that utilized Google Forms. There is a coordinator who will test the system from the website.

Students and teachers will test the system to their satisfaction. The main module of the system that will test is manage teachers, manage students, manage class, manage attendance, and manage timetable module. Each of the modules has obtained feedback from the user.

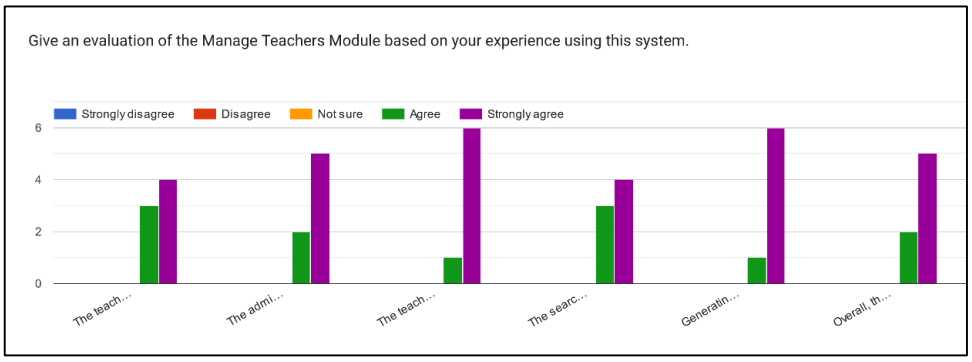

Figure 16: Feedback for manage teachers module

The system incorporates a rating scale from 1 to 5, where users can express their level of agreement or disagreement regarding their opinion on the system. This range corresponds to responses ranging from "strongly disagree" to "strongly agree." Notably, the feedback for the "Manage Students" functionality is gathered from both teachers and students since they share some common features and interactions within this module. The feedback received from users indicates a high level of satisfaction with the system's functionality, as many users consistently rate each module with a score of 5, indicating strong agreement with its performance.

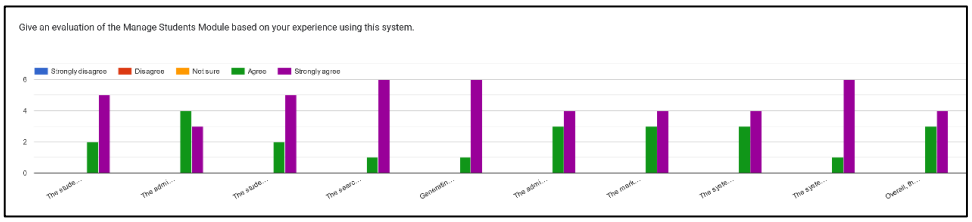

Figure 17: Feedback for manage students module

The manage class module empowers the administrator to perform essential tasks such as inserting, updating, and deleting classes, as well as assigning students to those classes. User feedback regarding this module has consistently indicated high ratings, signifying that it effectively fulfils its intended purpose and delivers an excellent user experience. The positive feedback received further validates the module's functionality and affirms its ability to meet the needs and expectations of the system users.

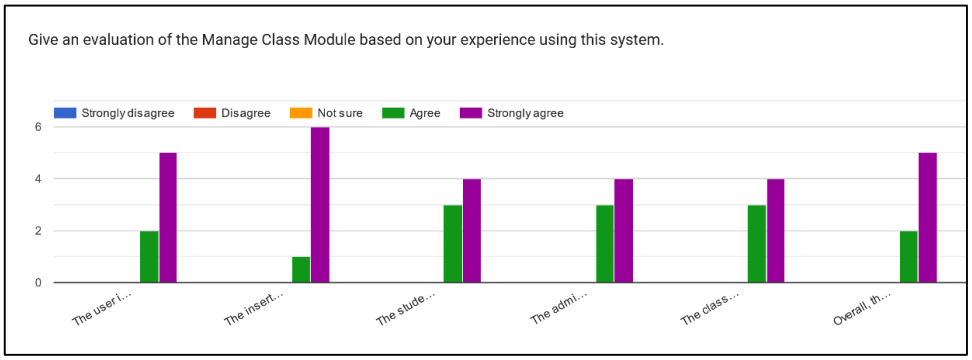

Figure 18: Feedback for manage class module

The manage attendance module serves as a platform for both teachers and administrators to oversee student attendance. Administrators have the capability to manage attendance records, while teachers possess additional functionality, such as generating QR codes for attendance tracking. Feedback regarding this module has been consistently positive, with high evaluations indicating user satisfaction. This positive reception underscores the module's effectiveness in facilitating efficient attendance management and reflects its ability to meet the requirements and expectations of both teachers and administrators.

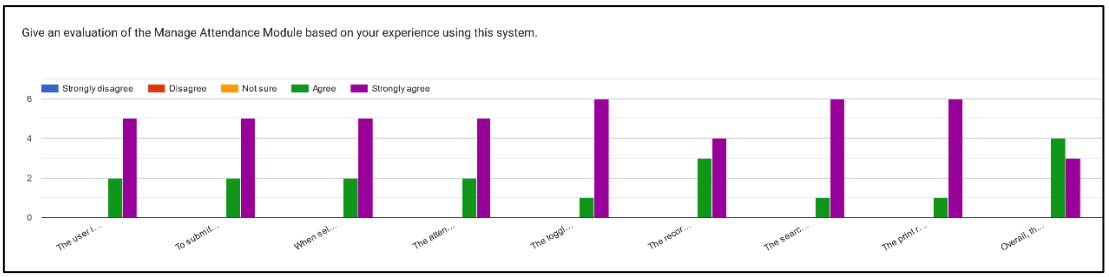

Figure 19: Feedback for manage attendance module

The manage timetable module showcases personalized timetables for teachers and students based on their respective roles and class schedules. This module has received excellent feedback, commending both its user-friendly interface and reliable functionality. Users appreciate the module's effectiveness in presenting timetables and enhancing their overall experience with the system.

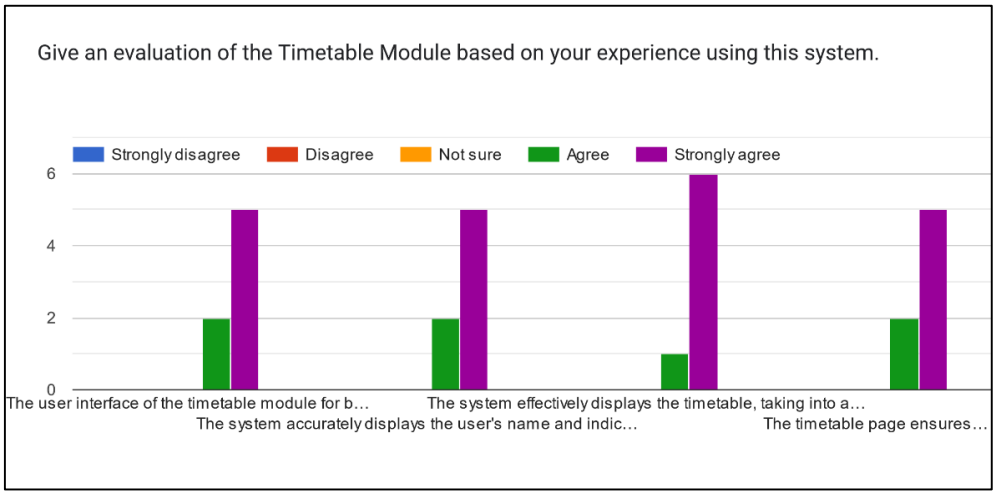

Figure 20: Feedback for manage timetable module

The feedback received for the various modules in the system reflects a high level of satisfaction and appreciation among the users. The positive ratings and comments attest to the system's effectiveness in fulfilling its intended purposes and providing a great user experience. The functionalities, from managing students and classes to attendance and timetable management, have garnered praise for their reliability, user-friendliness, and overall performance. This feedback serves as valuable validation of the system's successful implementation and its ability to meet the needs and expectations of its users. The constructive feedback received also provides valuable insights for further enhancements and improvements in the future. Overall, the positive feedback reinforces the success of the system and the positive impact it has on the users involved in the Tuition Uni Pintar community.

5. Conclusion

In conclusion, Aluna, the web-based Student Management System, offers a practical solution to address the challenges faced in managing student information at Tuition Uni Pintar. By utilizing technologies such as HTML, PHP, CSS, and MySQL, Aluna provides a user-friendly interface for teachers to efficiently record attendance, access comprehensive student profiles, and manage grades. The system's centralized database eliminates the need for physical documents, optimizing space and resources. With its focus on streamlining processes and enhancing efficiency, Aluna aligns with Tuition Uni Pintar's goal of supporting underachieving students and improving academic achievements. By implementing Aluna, the center can effectively manage student data, saving time and facilitating a more organized approach to student management.

Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support and encouragement throughout the process of conducting this study.

References

[1] Nuseibeh, B., & Easterbrook, S. (2000). Requirements engineering: a roadmap. In Proceedings of the Conference on the Future of Software Engineering (pp. 35-46). ACM.

[2] Nacheva, R. (2017). Prototyping approach in user interface development. In Second Conference on Innovative Teaching Methods (ITM 2017) 28-29 JUNE 2017, VARNA (Vol. 28, p. 78).

[3] Mohd, R. A. Y., & Ramli, A. A. (2022). Student Management System Development of Damya Deena Tuition Center (SPP Damya Deena). Applied Information Technology and Computer Science, 3(1), 889-901.

[4] Chai, W. C., & Mostafa, S. A. (2021). Bright Kids Tuition Centre Management Information System. Applied Information Technology and Computer Science, 2(2), 937-957.

[5] Liu, Z., Wang, H., & Zan, H. (2010). Design and implementation of student information management system. In *2010 International symposium on intelligence information processing and trusted computing* (pp. 607-610). IEEE.