APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 5 No. 2 (2024) 520-537
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

E-Mengaji System for Yayasan Restu

Yumni Abd Aziz[1], Hairulnizam Mahdin[2]*

[1,2] *Fakulti Sains Komputer dan Teknologi Maklumat,*
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: hairuln@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2024.05.02.028

Article Info

Received: 13 June 2024
Accepted: 28 September 2024
Available online: 15 December 2024

Keywords

online class, management, class management

Abstract

The E-Mengaji system, developed to facilitate online Quranic learning by Yayasan Restu, serves as a comprehesive management tool for classes, students, and teachers. Currently relying on manual record-keeping, the system faces challenges such as inefficiencies, data loss risks, and scalability issues. The goal is to improve the operations' effectiveness and productivity while easing current difficulties. With a focus on educational technology and information management, the system comprises modules for user and class management, payment processing, and analytics. The system employs Laravel, an advanced technology, to address problems with traditional methods and uses a methodical approach with Agile design phases and rigorous testing to ensure a reliable and error-free system. E-Mengaji accommodates the unique needs of various organizations through scalability and customization, promoting improved operational agility and organizational performance. Overall, the E-Mengaji system has been successfully developed with a login module, manage user module, manage profile module, course registration module, calendar module, manage class module, manage feedback module, and attendance module

1. Introduction

The process of Quranic education and management is crucial for the spreading religious knowledge. Quranic education is the study and understanding of the Quran, which is a fundamental component of Islamic learning. This process is overseen by educational institutions or religious organizations. In the context of E-Mengaji, this initiative is led by Yayasan Restu, highlighting their commitment to making Quranic education accessible to a broader audience through online platforms.

The current manual record-keeping procedure utilized for student and class management has become complex and error-prone, resulting in inefficiencies and compliance issues. The rise in the utilization of online learning platforms has necessitated a sophisticated management system capable of handling the complexities associated with virtual education [1]. The project's objectives include designing and analyzing the system structurally, developing it using a web-based approach, and conducting functional and usability tests. This system is meticulously crafted to navigate the challenges of student enrollment, class schedules, and online payment processing, providing educators and administrators with a centralized platform to effortlessly oversee student records, class details, and financial transactions. The system aims to resolve existing challenges, improve the overall quality of Quranic education, and align with Yayasan Restu's mission. This introduction sets the stage for

a transformative approach to educational management, promoting efficiency, transparency, and an enhanced learning experience for both educators and students in the digital age.

2. Related Work

Three systems that are related to the proposed system were studied in comparison. The first system, Author–UTHM Academic Online Resources. It serves as an educational hub for students at Tun Hussein Onn University, Malaysia (UTHM), facilitating the distribution, assessment, and grading of students by lecturers. This web application acts as a platform where lecturers share subject resources, allowing students to access, download, and study at their own pace [2]. The system's advantage lies in its straightforward and clean user interface, offering flexibility in resource distribution, assignment uploads, and assessments. In contrast, the proposed system aims to simplify this application, focusing on essentials such as the dashboard and learning materials.

The second system is Edmodo, established by Jeff O'Hare and Nick Borg in 2008, is an educational technology company that provides schools and teachers with a platform for communication, collaboration, and coaching [3]. This system became one of the earliest tools introduced to students during their secondary school years. Despite being equipped with numerous features, some of these functionalities may prove unnecessary for students. These features include class, calendar, backpack, discover, and messages [4]. Despite the array of features, students often struggle to comprehend their optimal use [5]

The third system is Google Classroom by Google. It is one of the applications in Google Apps for Education (GAFE) suite, which is a comprehensive set of tools meant for educators and learners involved in learning and online collaboration [6]. It intends to facilitate the exchange of documents among teachers and students, expediting this collaborative process. [7]. The students are required to input a private code provided by the teacher to join a classroom. The application encompasses features such as assignments, grading, communication, and a to-do list.

Class management systems can facilitate various features such as attendance tracking, grading, and communication tools. Teachers can utilize these systems to create digital assignments, quizzes, and exams, while students can submit their work and track their progress through the platform. Additionally, class management systems often include features like discussion boards and live chats allowing for real-time interactions between students and teachers. Using e-learning platforms offers teachers and learners with an adaptable instrument that is accessible at anytime and anywhere. [8].

Table 1 System Comparison

Features/System	Author	Edmodo	Google Classroom	E-Mengaji
Technology:				
Web-based system	Yes	Yes	Yes	Yes
Online system	Yes	Yes	Yes	Yes
Modules:				
Login	Yes	Yes	Yes	Yes
Sign Up	Yes	Yes	Yes	Yes
Profile Management	No	Yes	Yes	Yes
Attendance	No	Yes	Yes	Yes
Class Management	Yes	Yes	Yes	Yes
Payment Management	No	No	No	Yes
Report and Analytics	No	Yes	Yes	Yes
Invitation Management	No	No	No	Yes
Feedback	No	Yes	Yes	Yes

3. Methodology/Framework

The main reason of selecting methodology is to facilitate productivity and workflow of the system development, particularly for large projects with several teams participating. It explains the concepts that serve as a guide for

the research procedures described in the article [9]. The Agile Model, Waterfall Model, and Scrum Model are some of the approaches that can be used depending on the amount of members, cost, and time. For this project, an Agile Model was chosen and thoroughly discussed.

3.1 Agile Methodology

Agile methodology covers a wide range of software development ideas and serves as a conceptual foundation for software engineering. It starts with an initial planning phase and continues through the deployment phase, involving iterative and incremental interactions throughout the project's life cycle. The major goal of agile approaches is to reduce overhead in the software development process, allowing for the seamless adoption of changes without risking the overall process or forcing extensive redo [10]. Agile methodologies consistently tackle quality concerns, emphasizing a continuous and repetitive approach. The Agile method places a strong emphasis on software quality, aligning with customer expectations, minimizing error rates, facilitating swift development, and embracing adaptability in a dynamic environment [11].

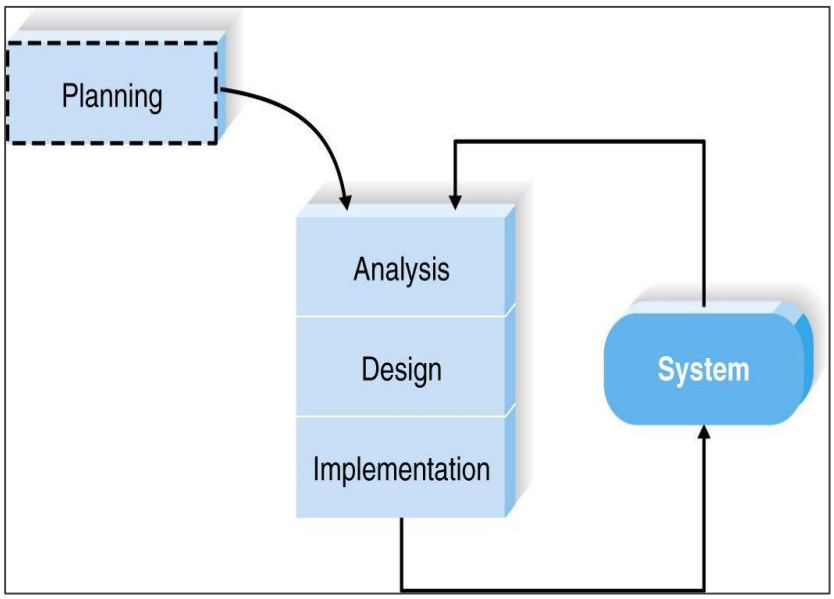


Fig. 1 *Agile Model [12]*

Table 2 Activities and Phases of Agile Method

Phase	Scope of Activity	Outcome
Planning	1. Recognize problem statements, project objectives, project scope, expected outcomes, and project significance. 2. Develop a project timeline. 3. Perform an online search for resources and articles related to the given title. 4. Conduct a comprehensive analysis of the features and functionality of current programs.	1. Project proposal 2. Literature review 3. Comparison between existing web-based system and proposed system.
Analysis	1. Interview with the personnel of the company. 2. Analyze hardware and software requirements. 3. Differentiate between functional and non-functional requirements.	1. Create a list of user requirements based on the interview session. 2. Specify hardware and software requirements. 3. Identify requirements encompassing both functional and non-functional aspects.
Design	1. Design CD, and DFD 2. Design a wireframe. 3. Design an ERD	1. CD, and DFD production. 2. Wireframe production. 3. ERD production.

Table 2:(cont) Activities and Phases of Agile Method

Phase	Scope of Activity	Outcome
Implementation	1. Develop an entire module. 2. Finalize the integration of the system. 3. Establish connectivity with the database.	1. Proposed system. 2. Error found and fixed.
Testing	1. Perform system testing. 2. Determine whether there is any potential for system upgrades.	1. Fix and improve the bugs. 2. Fix and ready to release the proposed system.

4. Analysis and Design

System analysis entails acquiring comprehensive and detailed insights into the system requirements. The process of requirement analysis is a systematic approach used to identify user needs for developing or updating an application. This encompasses the thorough analysis, documentation, validation, and management of software or system requirements. During this stage, the functional and non-functional requirements of the envisioned system are meticulously outlined and expounded upon.

4.1 Data Flow Diagram Context Diagram

Context diagrams show how it works around the system. The system interacts with two external entities, namely students and administrators. Figure 2 depicts the Data Flow Diagram Context Diagram (DFD CD) for the E-Mengaji System.

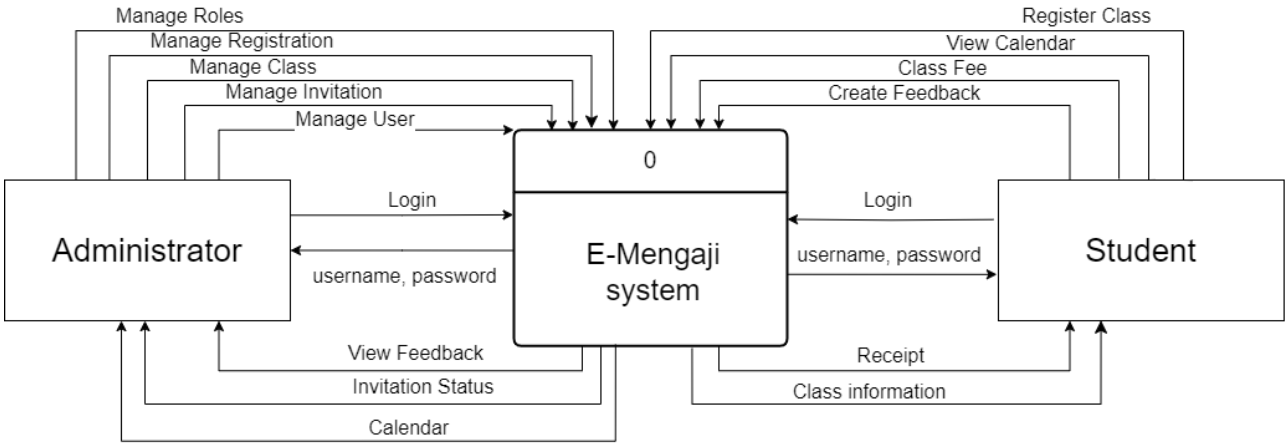


Fig. 2 *Data Flow Diagram Context Diagram (DFD CD)*

4.2 Data Flow Diagram (DFD) Level 0

The Data Flow Diagram (DFD) serves as a method for structured analysis and design. It functions as a visual instrument for illustrating logic models and conveying the transformation of data within a system. The DFD incorporates a mechanism for modeling data flow, facilitating decomposition to elucidate the intricacies of both data flows and functions. The DFD level 0 of the proposed system is shown in Figure 4.1.4. There is a total of eleven processes in DFD level 0 which are, login (process 1.0), manage registration (process 2.0), classes (process 3.0), payment (process 4.0), manage user (process 5.0), manage class (process 6.0), manage attendance (process 7.0), manage invitation (process 8.0), manage role (process 9.0), feedback (process 10.0) and calendar (process 11.0).

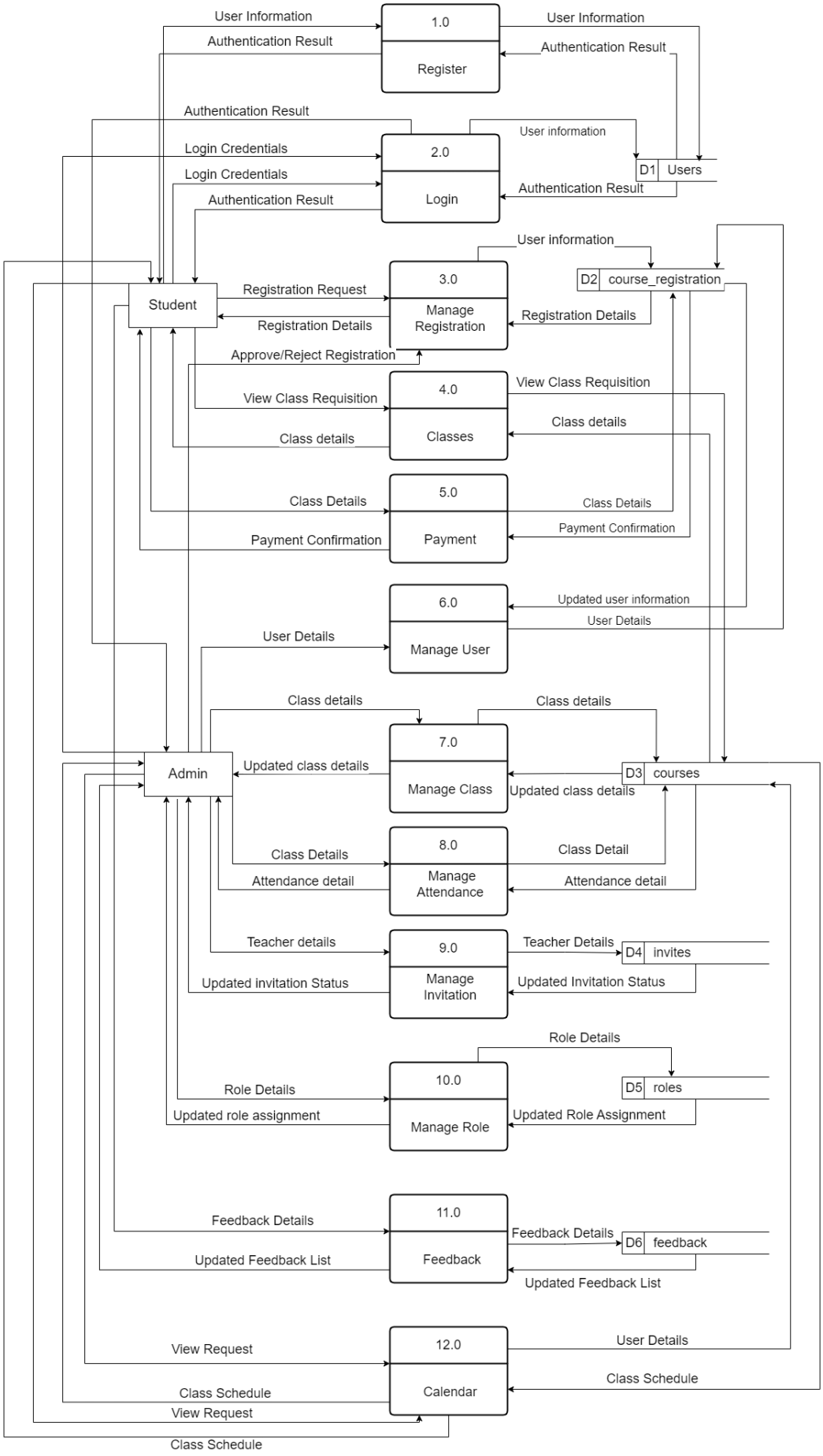


Fig. 3 *Data Flow Diagram Level 0 (DFD 0)*

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 520-537

4.3 Entities Relationship Diagram

An Entity-Relationship Diagram (ERD) shows the links between collections of entities in the database. In this context, an entity is an object or data component, whereas an entity set is a collection of comparable entities. These entities may have features that help to characterize their qualities. The system employs one-to-zero or many and one-to-one or many relationships. For instance, an administrator can add one or multiple vessels, while a vessel is associated with one or multiple vessel schedules. The ERD of the proposed system is represented in Figure 4.

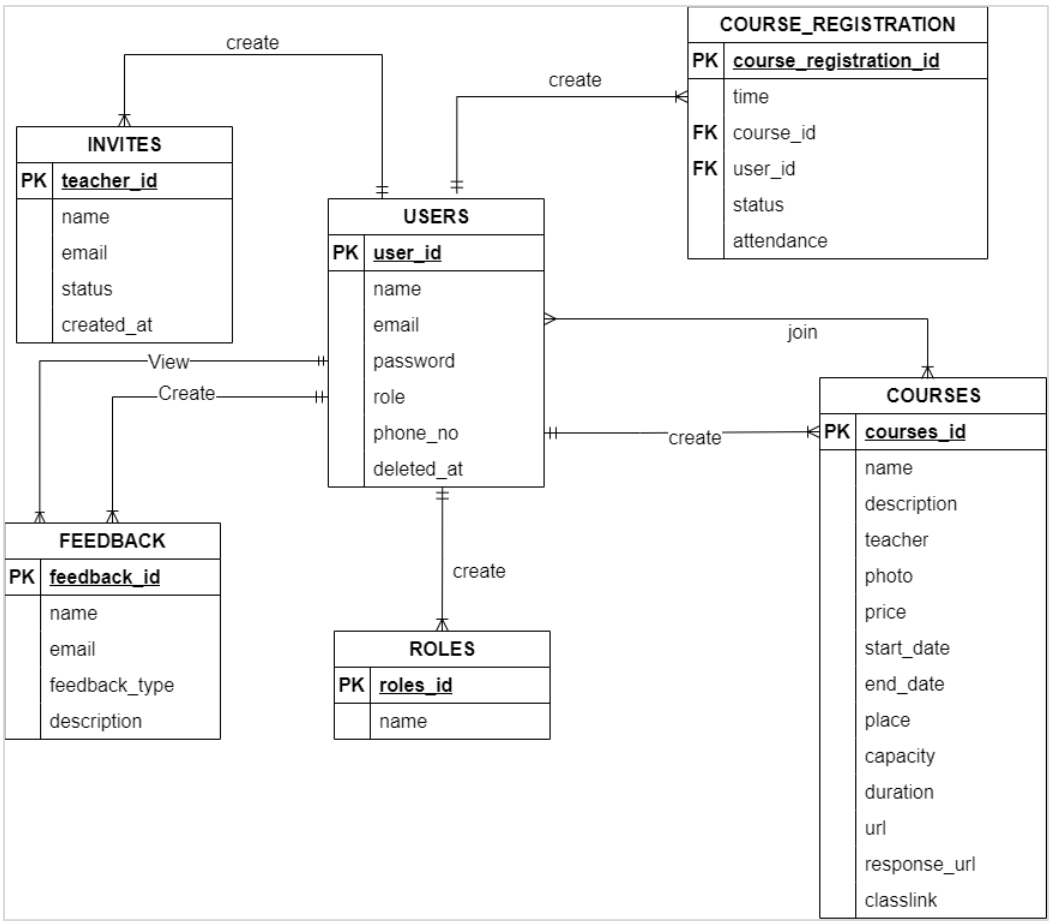


Fig. 4 *ERD of the proposed system*

4.4 Graphical User Interface

The User Interface (UI) serves as the location where human-computer interaction takes place. In the context of this envisioned system, customers will engage with it through web pages. This segment focuses on designing and elucidating the system's user interface to provide developers with a visual representation of the layout during the system development process. The interface presented in this section is crafted using Figma software.

4.4.1 Student User Interface

The interfaces listed below have been built for student users of the system based on their special needs. They are designed by using Figma.

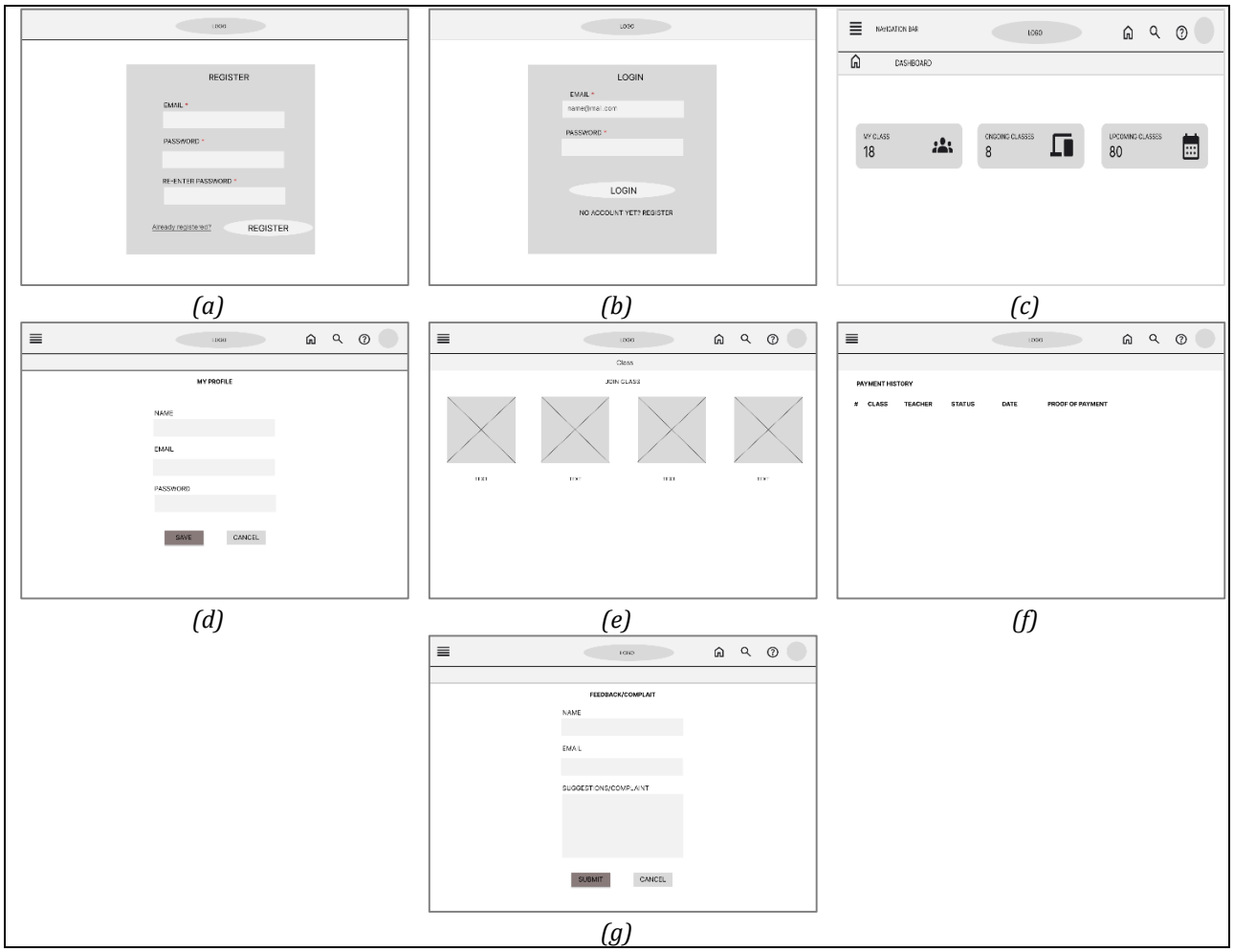
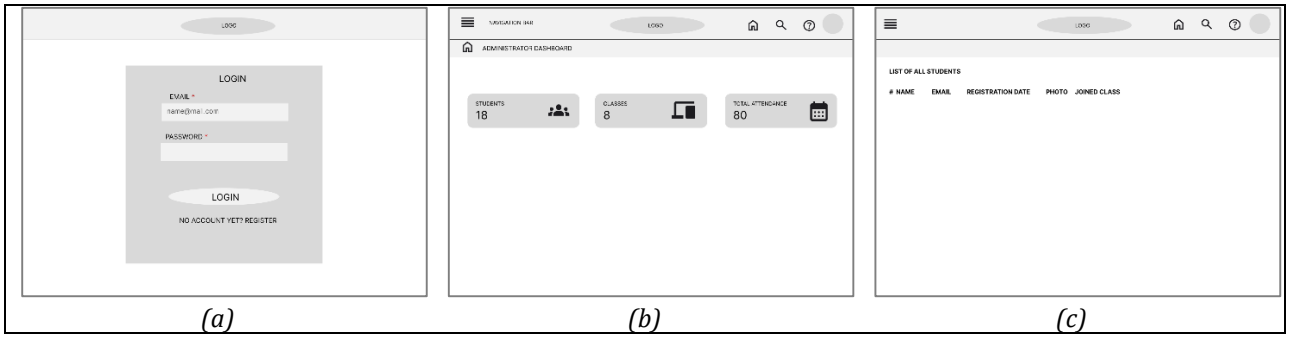


Fig. 5 *(a)Registration interface; (b) Login interface; (c)Homepage Interface; (d)Edit Profile interface; (e)Class interface; (f) Payment History Interface; (g)Feedback Interface*

4.4.2 Admin User Interface

The interfaces listed below have been built for administrators of the system based on their special needs. They are designed by using Figma.



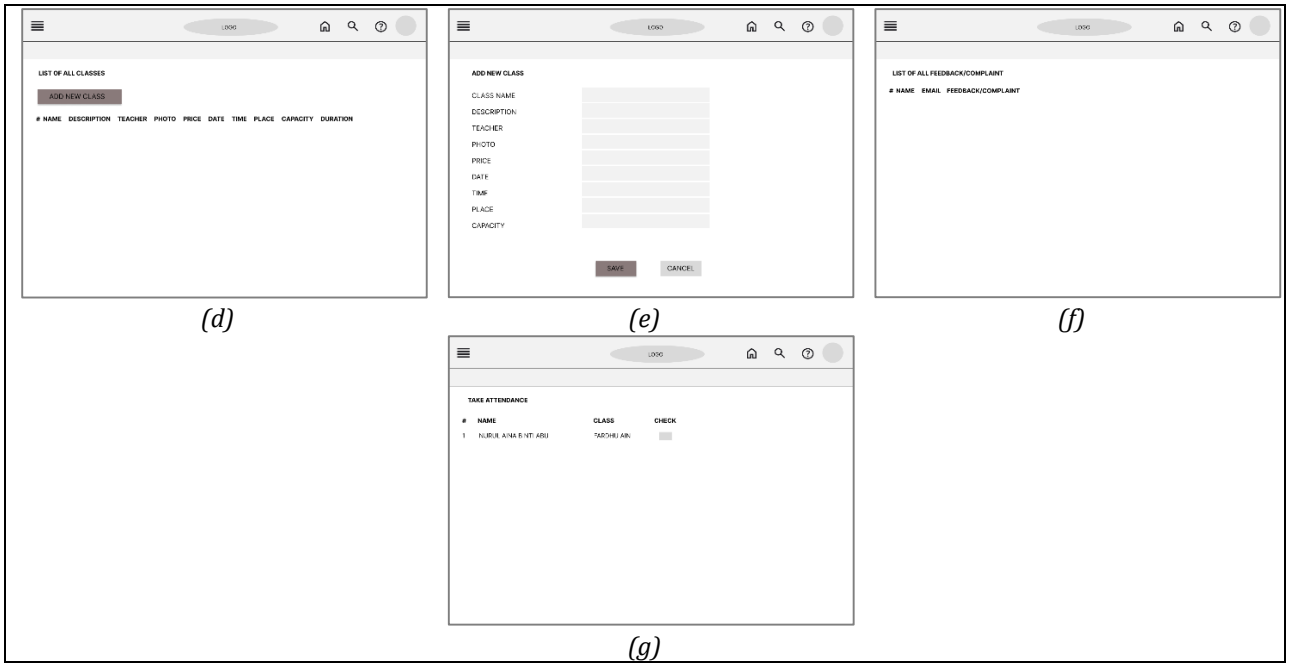


Fig. 6 *(a) Login interface; (b) Dashboard interface; (c)Manage Student Interface; (d) Manage Classes interface; (e)Create New Class interface; (f) Manage Feedback Interface; (g)Attendance Interface*

5. Implementation and Testing

The system has been primarily developed as a Laravel-based web application, utilizing the robustness and scalability of this popular PHP framework. This architecture enhances maintainability and structural clarity, simplifying the management of complex applications. For the front end, JavaScript, CSS and HTML are used for the creation of an interactive and responsive user interface.

On the backend, Laravel's features such as Eloquent ORM, Blade templating engine, and built-in secure authentication functions are extensively employed. Middleware is used to filter requests, and service containers manage class dependencies and perform dependency injection. Database management is handled via HeidiSQL, an open-source tool for administering MySQL [13]. The system is tested locally with XAMPP, a cross-platform web server solution, and Visual Studio Code is the primary code editor. Additionally, future scalability and improvements have been considered throughout the system's implementation, leveraging Laravel's modular nature and compatibility with various libraries and tools.

5.1 Database Configuration

HeidiSQL tool is selected as the database administration tool for this proposed web. The establishment of a connection between a PHP script and a database is required to access and add content to the MySQL database. For running the function and retrieving the username or password for a specific user, each module in the web-based system should connect to a database. The database configuration needs to be initialized to MySQL as shown in Fig. 6.

```
'connections' => [
        'mysql' => [
            'driver' => 'mysql',
            'url' => env('DATABASE_URL'),
            'host' => env('DB_HOST', '127.0.0.1'),
            'port' => env('DB_PORT', '3306'),
            'database' => env('DB_DATABASE', 'forge'),
            'username' => env('DB_USERNAME', 'forge'),
            'password' => env('DB_PASSWORD', ''),
            'unix_socket' => env('DB_SOCKET', ''),
            'charset' => 'utf8mb4',
            'collation' => 'utf8mb4_unicode_ci',
            'prefix' => '',
            'prefix_indexes' => true,
            'strict' => true,
```

```
            'engine' => null,
            'options' => extension_loaded('pdo_mysql') ? array_filter([
                PDO::MYSQL_ATTR_SSL_CA => env('MYSQL_ATTR_SSL_CA'),
            ]) : [],
        ],
```

Fig. 6 *Database Configuration*

5.1.1 Profile Management Module

Profile Management involves managing individual user profiles, typically focusing on personal information and preferences. It allows users to view and update their details.

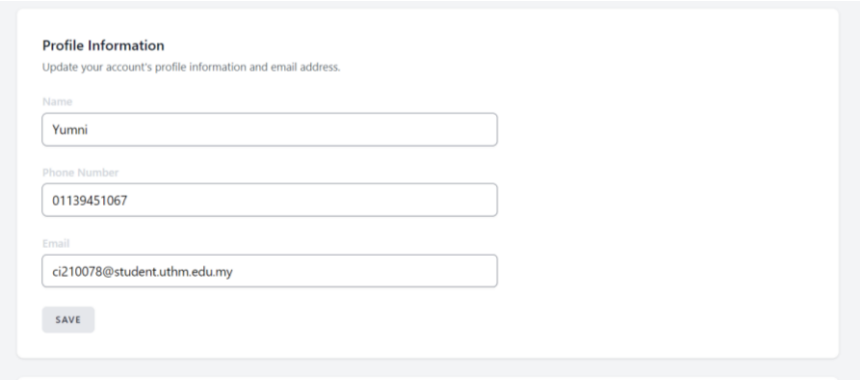


Fig. 7 *Update Profile Interface*

5.1.2 Manage Class

Class Management involves managing classes, usually performing CRUD operations (Create, Read, Update, Delete) to efficiently handle class information.

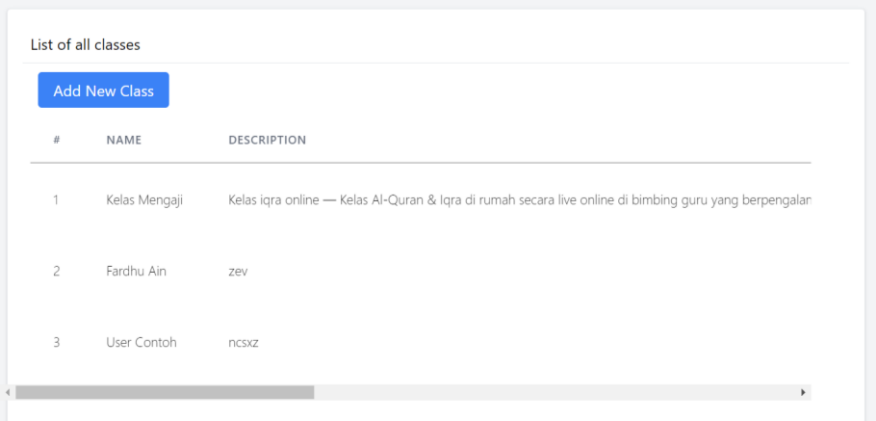


Fig. 8 *Manage Class Interface*

5.1.3 Manage Attendance

This function manages the attendance records for a specific course. It first fetches the course details, including related user registrations, using the course ID. If the course is not found, it returns a 404 error. Then, it retrieves the attendance list for the course and displays the attendance records on the admin's attendance list page.

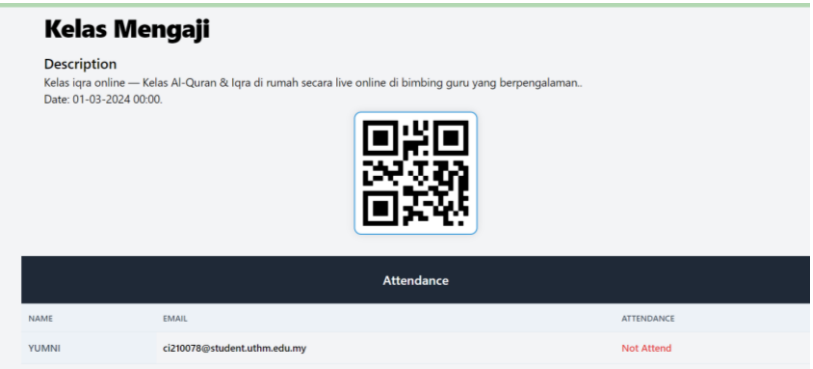


Fig. 9 *Update Profile Interface*

5.1.4 Manage Invitation

This function handles the sending of invitations. It validates the input data (name and email), creates a new invite record in the database with a status of 'pending,' and sends an invitation email to the provided email address. Finally, it returns to the previous page with a success message.

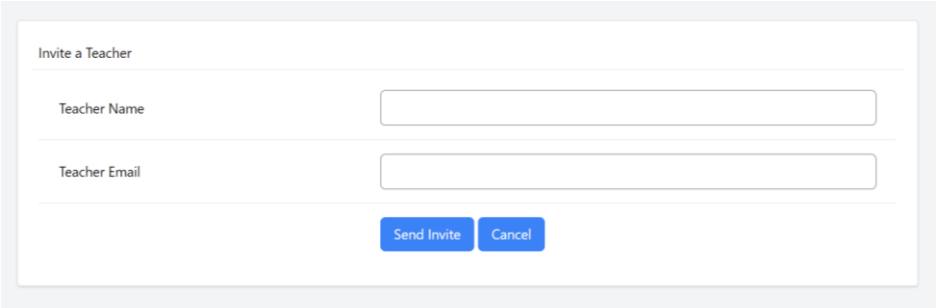


Fig. 10 *Send Invitation Interface*

5.1.5 Manage Users

User Management allows administrators to oversee all user accounts in the system. Admin can view a list of users, create new ones, update existing user details like name, email, and roles, and delete users if needed.

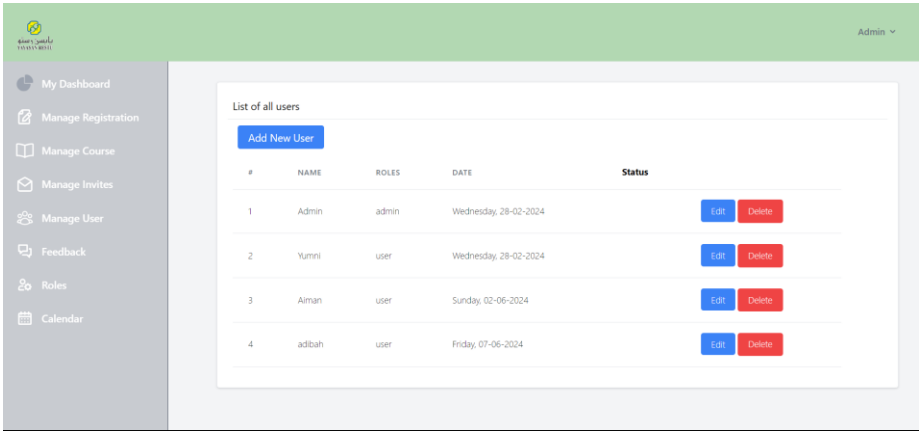


Fig 11 *View User Interface*

5.1.6 Feedback

In this module, the student can give feedback regarding the system, teacher, or the class they joined. The administrator is able to view the feedback given by users.

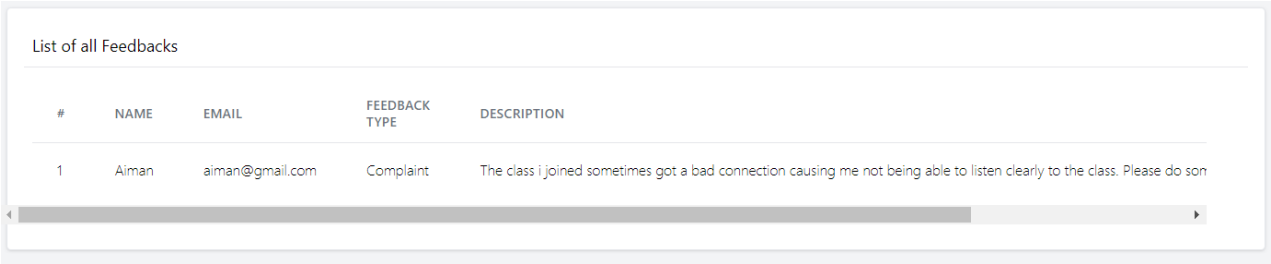


Fig. 12 *View Feedback Interface*

5.1.7 Manage Roles

This function retrieves all roles from the database and displays them on the admin's role management page, allowing administrators to view and manage roles.

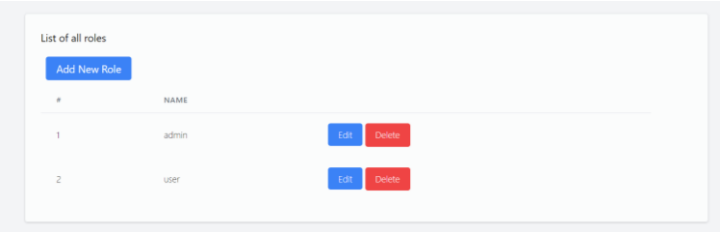


Fig. 13 *View Roles Interface*

5.1.8 Calendar

This function fetches specific details (ID, name, start date, end date) of all courses from the database and displays them on the calendar index page, allowing users to view course schedules.

Fig. 14 *View Calendar Interface*

5.1.9 Payment Module

In the payment module, students can make payments to join the class they want. The ToyyibPay API is used to create a bill. It then redirects the user to the payment page for the generated bill code.

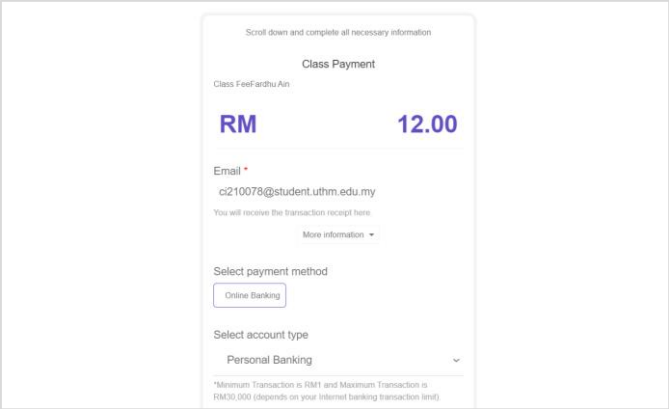


Fig. 15 *Payment Interface*

5.2 Testing

Testing a system is a form of software testing that utilizes scripts to uncover software defects. It encompasses testing both the system and the application. Application testing ensures the smooth functioning of the application to deliver the intended results and meet the specified goals and scope for end users. Consequently, testing is integrated into the project's scope and objectives right from the start.

5.2.1 Functional Testing

Table 3 displays the test cases for every module within the system, totaling 8 test cases for these modules. The objective of these tests is to confirm whether the administrator can successfully register an account, login to the system, and determine if the system appropriately denies access upon entering incorrect credentials. Table 3 indicates that all three tests have successfully passed.

Table 3 List of test cases

Test Cases ID	Description	Status
TC_100	**Register**	
TC_100_01	The user must be able to register on the system.	PASS
TC_100_02	The user must be able to input their name, email, phone number, and password into the system.	PASS
TC_100_03	Email and password must all be validated by the system	PASS
TC_100_04	If the email address and password invalid, the system will not permit registration.	PASS
TC_200	**Login**	
TC_200_01	The user must can log in to the system.	PASS
TC_200_02	The user must be able to enter their email and password into the system	PASS
TC_200_03	The system must confirm the email and password.	PASS
TC_200_04	If the user enters an invalid email or password, the system must display an error.	PASS
TC_200_05	If the email and password are incorrect, the system will not permit login.	PASS
TC_300	**Update Profile Information**	
TC_300_01	The system shall display the administrator or customer change account details page based on user selection.	PASS
TC_300_02	The system must permit the user to enter fresh personal information.	PASS
TC_300_03	The system must enable users to enter new passwords.	PASS
TC_300_04	The system must save the new password entered by the users.	PASS
TC_300_05	The system will show the user's most recent information.	PASS
TC_300_06	If the details and password are incorrect, the system will not permit the altered account details process.	PASS

Table 3:(cont) List of test cases

Test Cases ID	Description	Status
TC_400	**Manage Class**	
TC_400_01	The system shall display the classes that exist in the system.	PASS
TC_400_02	The user can add a new class to the system.	PASS
TC_400_03	The user can update the class information in the system.	PASS
TC_400_04	The user can remove the class in the system.	PASS
TC_400_05	After adding, deleting, or changing the class information, the system must update the class page.	PASS
TC_400_06	If the text field is empty, the system will not let the user create a new class.	PASS
TC_400_07	If the text box has blank spaces, the system will not let the user modify the class details.	PASS
TC_500	**Manage Users Details**	
TC_500_01	The system shall display the user's details	PASS
TC_500_02	The system may let the user enter new user information.	PASS
TC_500_03	The user can remove user information from the system.	PASS
TC_500_04	The user can update user information through the system.	PASS
TC_500_05	If the text field is blank, the system will not permit the user to add, delete, or amend user details.	PASS
TC_600	**Manage Role**	
TC_600_01	The system shall display the roles that exist in the system.	PASS
TC_600_02	The user can add a new role in the system.	PASS
TC_600_03	The user can update the role information in the system.	PASS
TC_600_04	The user can remove the role in the system.	PASS
TC_600_05	After adding, deleting, or changing the role information, the system must update the role information.	PASS
TC_600_06	If the text field is empty, the system will not let the user create a new role.	PASS
TC_600_07	If the text box has blank spaces, the system will not let the user modify the role details.	PASS
TC_700	**Manage Feedback**	
TC_700_01	The system shall display the feedback details	PASS
TC_700_02	The user can create new feedback in the system	PASS
TC_800	**Make Payment**	
TC_800_01	On the registration page, the system must show the class details, including the class name and the total cost.	PASS
TC_800_02	The system must let the user choose their preferred bank of choice.	PASS
TC_800_03	The user can enter payment information into the system.	FAIL
TC_800_04	A transaction receipt will be sent by the system to the user's registered email address.	PASS
TC_800_05	Before allowing the user to submit, the system must confirm that all of the required fields on the payment form are filled out.	PASS

5.2.2 User Acceptance Testing (UAT)

UAT (User Acceptance Testing) is a testing approach in which the end-user assesses and receive the software system before it is published to public. Conducted after functional, and system testing, UAT is the final phase of testing. For UAT, a small group of users, including administrators and students, are randomly selected to complete a survey evaluating the system's user acceptability. This testing involved fifteen users: one administrator and fourteen users. The collected data is analyzed and presented in a graph, where a ranking of 1 indicates highly unsatisfied and a ranking of 5 signifies very satisfied.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 520-537

5.2.3 Student UAT

Table 4 provides the results and input from fifteen users' user interface evaluations. The proposed system's simplicity of use, layout, interface design, navigation, instruction, and instructive text were all rated as satisfactory by the majority of users. No one was dissatisfied or extremely dissatisfied with the suggested system's user interface design. In addition, Table 5 displays the results and consequences of the system feature evaluations. All of the results from Tables 4 and 5 were converted into the charts in Figures 16 and 17.

Table 4 Result of user interface evaluation from student

No.	Features	Ranking					Total
		1	2	3	4	5	
1.	The system is easy to understand.	-	-		3	10	13
2.	Satisfaction with the layout	-	-	4	2	7	13
3.	Satisfaction with the interface design (color, background)	-	-	3	3	7	13
4.	Satisfaction with the navigation of the system.	-	-	-	2	11	13
5.	Satisfaction with the instruction, and informative text in the system.	-	-	-	2	11	13

Table 5 Result of system functionality evaluation from student

No.	Features	Ranking					Total
		1	2	3	4	5	
1.	Able to register	-	-	-	3	10	13
2.	Able to login	-	-	-	3	10	13
3.	Able to manage profile information	-	-	-	3	10	13
4.	Able to view classes and schedule	-	-	-	3	10	13
5.	Easy to join class	-	-	-	3	10	13
6.	Redirect to the correct payment gateway	-	-	-	3	10	13
7.	View payment history	-	-	-	3	10	13
8.	Able to generate receipt in PDF, and download it	-	-	-	3	10	13
9.	Logout the system without any issue	-	-	-	3	10	13

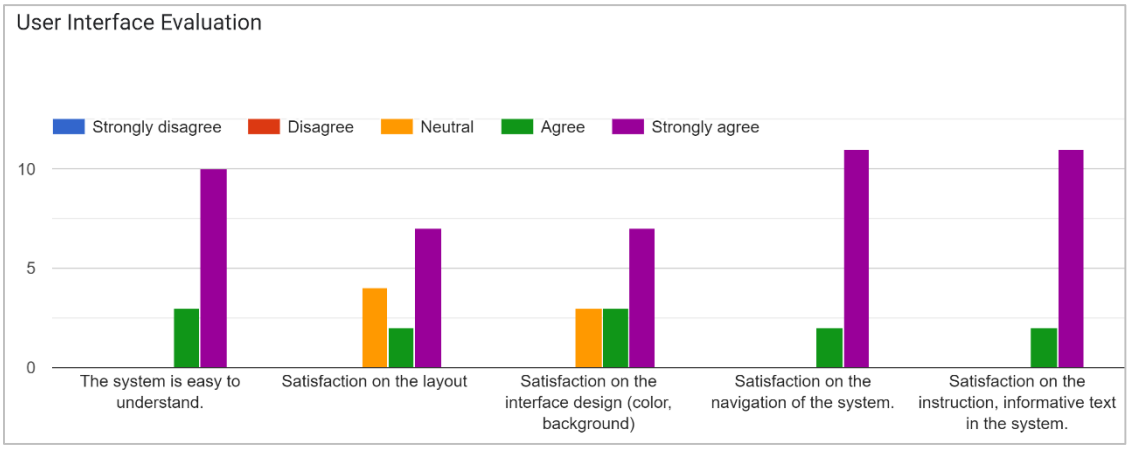


Fig. 16 *Result of user interface evaluation from student*

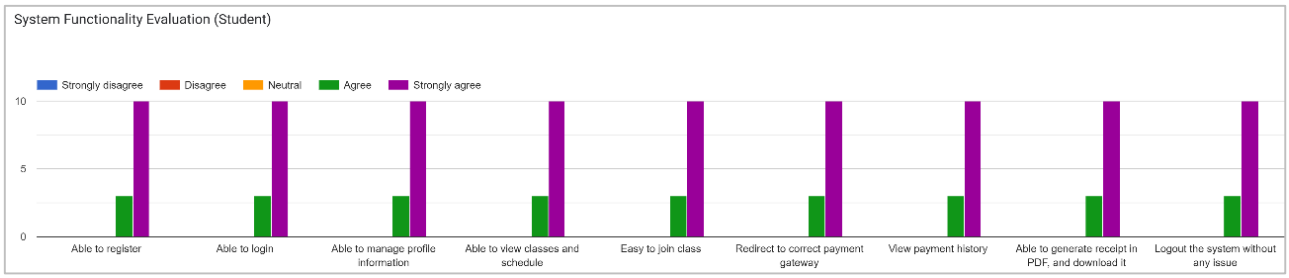


Fig. 17 *Result of system functionality evaluation from student*

5.2.4 Administrator UAT

Table 6 presents the results and feedback from an administrator's evaluation of the user interface. The administrator expressed satisfaction with the ease of use, layout, interface design, navigation, and clarity of instructions and informative text. Additionally, Tables 6 display the outcomes of the system features evaluations. The results from Tables 6 and 7 are visualized in charts shown in Fig. 18 and Fig. 19, which illustrate the user interface evaluation results from the administrator.

Table 6 Result of user interface evaluation from administrator

No.	Features	Ranking					Total
		1	2	3	4	5	
1.	The system is easy to understand.	-	-	-	1	1	2
2.	Satisfaction with the layout	-	-	-	1	1	2
3.	Satisfaction with the interface design (color, background)	-	-	-	-	2	2
4.	Satisfaction with the navigation of the system.	-	-	-	-	2	2
5.	Satisfaction with the instruction, and informative text in the system.	-	-	-	1	1	2

Table 7 Result of user interface evaluation from administrator

No.	Features	Ranking					Total
		1	2	3	4	5	
1.	Dashboard calculates the correct total number of end users (student, and administrator), and classes	-	-	-	1	1	2
2.	Able to register new users (administrator and student)	-	-	-	1	2	2
3.	Able to manage classes (create, view, update, and delete)	-	-	-	-	1	2
4.	Able to manage users (create, view, update, and delete)	-	-	-	-	2	2
5.	Able to generate QR code for the attendance	-	-	-	-	1	2
6.	Logout the system without any issue	-	-	-	-	2	2

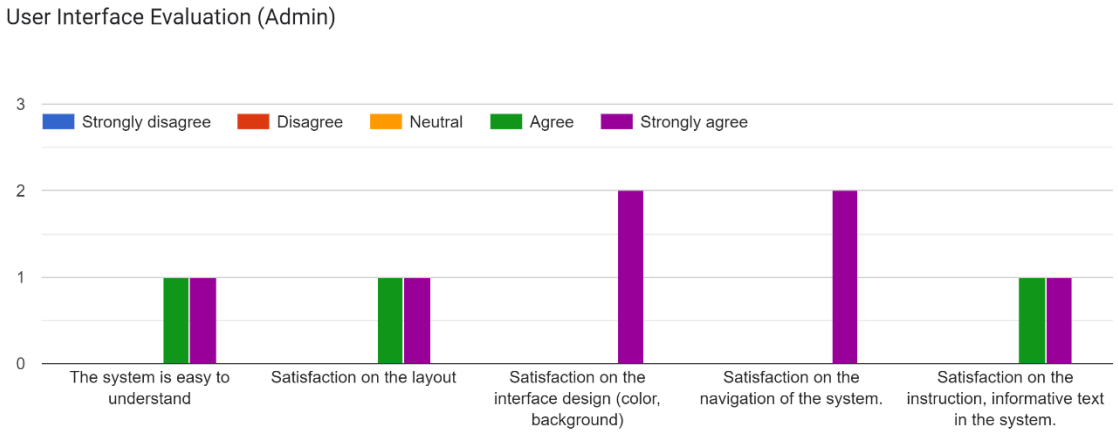


Fig. 17 *Result of user interface evaluation from administrator*

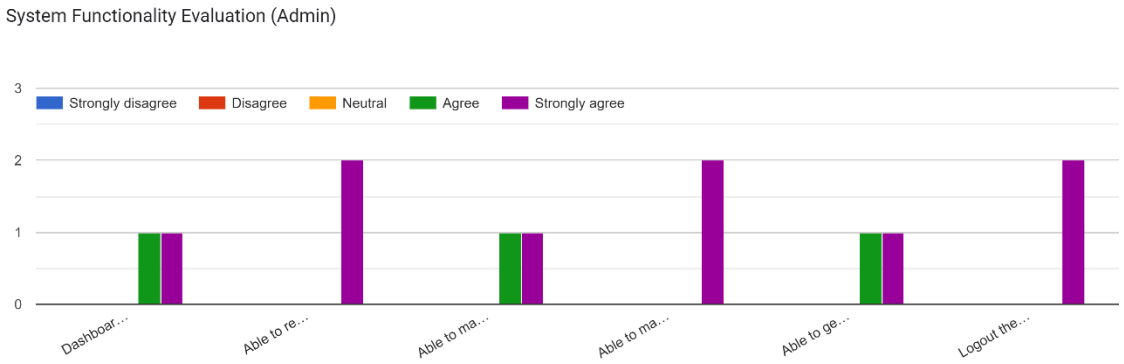


Fig. 18 *Result of system functionality evaluation from administrator*

6. Conclusion

In conclusion the E-Mengaji System has several advantages such as easing the management of classes held by Yayasan Restu. The automated system will streamline administrative tasks, reducing manual data entry and paperwork. This newfound efficiency will free up valuable time for administrators, allowing them to focus on more strategic aspects of Quranic education. Other than that, by minimizing the risk of human errors in record-keeping, the system will ensure data accuracy. The user-friendly system, coupled with improved services, is poised to boost satisfaction levels among administrators, teachers, and students, leading to a more harmonious and productive educational ecosystem. However, there are some limitations of this system such as the user need to register for the same class every month to keep accessing them. Hence, some recommendations for improvement in the future can be made such as improving the system usability, user interfaces design and expanding its use through the development of mobile application that can be applied to this project. E-Mengaji system hopefully can be a huge help to people who want to enjoy the ease of learning through online platform.

Acknowledgment

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

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

This journal requires that all authors take public responsibility for the content of the work submitted for review. The contributions of all authors must be described in the following manner:

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Yumni Abd Aziz, Hairulnizam Mahdin;* ***data collection:*** *Yumni Abd Aziz, Hairulnizam Mahdin;* ***analysis and interpretation of results:*** *Yumni Abd Aziz, Hairulnizam Mahdin;* ***draft manuscript preparation:*** *Yumni Abd Aziz, Hairulnizam Mahdin. All authors reviewed the results and approved the final version of the manuscript.*

The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

References

[1] Solahudin, D., & Fakhruroji, M. (2019). Internet and Islamic learning practices in Indonesia: Social media, religious populism, and religious authority. Religions, 11(1), 19.
[2] Sanudin, R., Sahdan, M.Z., MOhd Tawil, S.N., Ruslan S.H. 2019. Assessment using e-learning Platform in UTHM. 1st International Malaysian Educational Technology Convention. 149-156.
[3] Alqahtani, Abdullah. (2019). The Use of Edmodo: Its Impact on Learning and Students' Attitudes Toward It. Journal of Information Technology Education: Research. 18. 319-330. 10.28945/4389
[4] Sefriani, R., Sepriana, R., Wijaya, I., & Radyuli, P. (2021). Blended Learning with Edmodo: The Effectiveness of Statistical Learning during the COVID-19 Pandemic. International Journal of Evaluation and Research in Education, 10(1), 293-299.
[5] Johnna, D., & La Paz, I. C. (2019). Using Edmodo in blended learning: Its benefits and drawbacks. Asian Journal of Multidisciplinary Studies, 2(1).
[6] Ketut Sudarsana, I., Bagus Made Anggara Putra, I., Nyoman Temon Astawa, I., & Wayan Lali Yogantara, I. (2019, March). The use of Google Classroom in the learning process. In Journal of Physics: Conference Series (Vol. 1175, p. 012165). IOP Publishing.
[7] Rahmawati, B. F. (2020, May). Learning by Google Classroom in students' perception. In Journal of Physics: Conference Series (Vol. 1539, No. 1, p. 012048). IOP Publishing.
[8] Kraleva, R., Sabani, M., & Kralev, V. (2019). An analysis of some learning management systems. International Journal on Advanced Science, Engineering and Information Technology, 9(4), 1190-1198.
[9] Davis, B. (2021). What is methodological approach in research? Retrieved from MVOrganizing: https://www.mvorganizing.org/what-is-methodological-approach-in-research/
[10] Al-Saqqa, S., Sawalha, S., & AbdelNabi, H. (2020). Agile software development: Methodologies and trends. International Journal of Interactive Mobile Technologies, 14(11).
[11] Subih, M. A., Malik, B. H., Mazhar, I., Sabir, U., Wakeel, T., Wajid, A., ... & Suleman, M. (2019). Comparison of agile method and scrum method with software quality affecting factors. International Journal of Advanced Computer Science and Applications, 10(5).
[12] Dennis, A., Wixom, B. H., & Roth, R. M. (2012). *System Analysis And Design* (5th ed.). Hoboken, New Jersey: John Wiley & Sons. Retrieved from http://www.uoitc.edu.iq/images/documents/informatics-institute/Competitive_exam/Systemanalysisanddesign.pdf
[13] Kiktev, N. A., Kutyrev, A., Khort, D., & Kalivoshko, O. (2021). Web Application for an Information System for Diagnosing the Quality of Electricity Consumers Using Cloud Technologies. In *IT&I* (pp. 176-185).

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 520-537

Appendix A:

Gantt Chart: The Gantt chart illustrates the sequential arrangement of phases of requirements, design, implementation, testing, and maintenance within the Agile Method. Each phase is represented as a distinct block, showcasing its iterative progression. The chart includes task durations, dependencies, and the specific start and end dates for each phase. This visual representation accentuates the method's structured and non-iterative nature, emphasizing the strict order of progression. It also underscores the challenges associated with making adjustments once a phase has started.

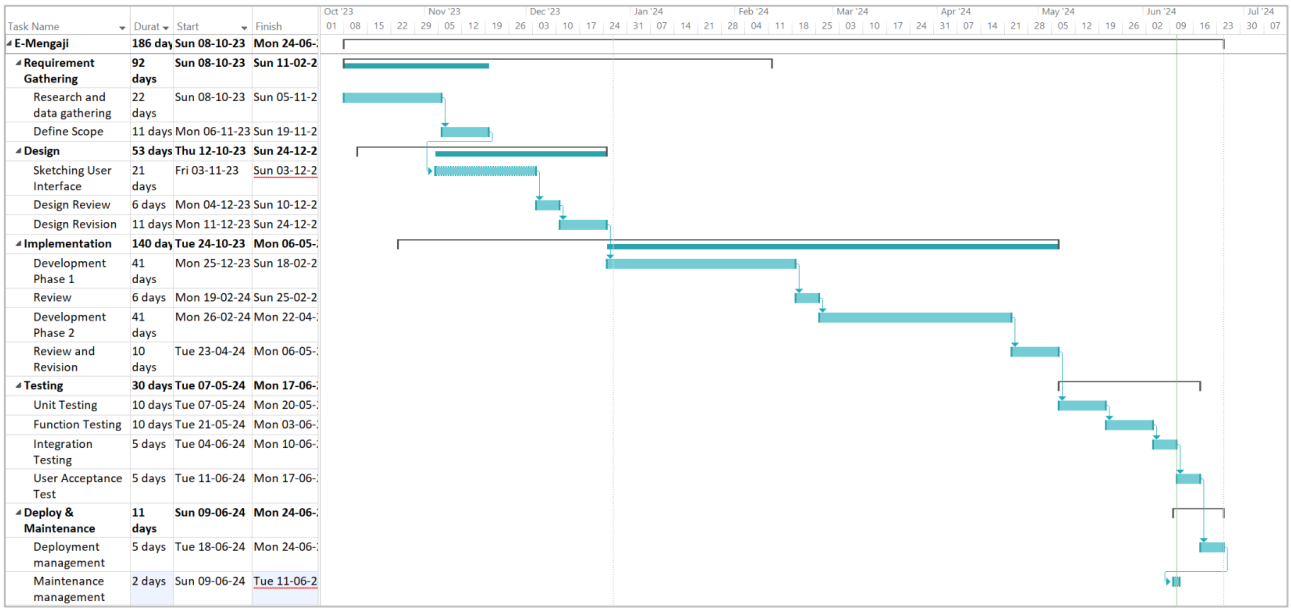


Fig. 19 *Gantt Chart*