



School Alert Application: Parent-Teacher Engagement System

Tan Wan Ying¹, Rosziati Ibrahim^{1*}

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

DOI: <https://doi.org/10.30880/aitcs.2024.05.01.052>

Received 24 June 2023; Accepted 18 June 2024; Available online 30 August 2024

Abstract: The School Alert Android application is designed for busy parents of SJI International School to stay informed about their children and the school. It also provides a web-based system for teachers to manage the system. A simple Software Development Life Cycle (SDLC) model is implemented throughout software development. The system consists of ten modules, including login, registration, class management, user management, student information, attendance, academic performance, financial management, activities and messaging. Android Studio and Java are used for mobile development, while web technologies like HTML, PHP, and MySQL are used for the administrative website. The system then undergoes acceptance testing and functionality testing to ensure its effectiveness. The goal is to enhance parent-teacher engagement and improve student performance. Future enhancements may include notification feature of the latest news for all the system's module.

Keywords: School Alert, Android app, website

1. Introduction

A system of engagement focuses on real-time communication between people using IT services like collaboration tools and chat [1]. Thus, a system complemented with collaboration capabilities, facilitates communication between parents and teachers, which is crucial for the positive development of students in terms of their mental and physical growth. In today's Internet-driven society, communication technology has rapidly advanced, offering various alerting functions such as email notifications, text messaging, and mobile applications, providing effective channels for continuous communication between parents and teachers.

However, existing school management systems often lack the involvement of parents, as they only receive detailed information about their children's performance during scheduled parents' day meetings. This method becomes impractical for parents who are unable to attend these meetings due to work commitments. Consequently, parents seeking instant feedback on their child's behavior and attendance can only resort to messaging applications like WhatsApp or Telegram. Unfortunately, these methods are not efficient in terms of timely information updates, storage and accuracy.

*Corresponding author: rosziati@uthm.edu.my

| This is an open access article under the CC BY-NC-SA 4.0 license.

Research has shown that authoritative and authoritarian parenting styles positively impact students' performance in school [2]. Both parents and teachers play vital roles in students' academic success, providing formal and informal education as they spend significant time in school and at home. To address the limitations of existing systems, this project proposes the development of the 'School Alert' Android mobile application and an administrative web-based system for St. Joseph's Institution (SJI) International School. The aim is to provide parents with functions to access school and student-related information, such as attendance, school activities, fee structure, academic performance and chat messaging with teachers. Additionally, the proposed administrative web-based system will enable teachers to keep all system information up-to-date, including managing mobile user access, student information and overall system information available to mobile users. The project objectives include designing the UML specification and developing the School Alert mobile application for parents and administrative web-based system for teachers at St. Joseph's Institution International School, followed by acceptance testing and functionality testing.

The subsequent sections of this paper are organized in which Section 2 discusses related work on the methods and technologies used in this project, Section 3 explains the software development model employed, and Section 4 presents the results of the functionality and user acceptance testing for the system.

2. Related Work

This section explains about the method and technology used in the project. Some similar systems are also studied and compared with the proposed system.

2.1 Web Development

The HyperText Markup Language (HTML) is a markup language used to annotate text so that computers may modify it to create web pages. HTML also specifies how a browser should load, format and align text and graphics on a web page. Despite the ability limitation of HTML to define the appearance and style, the Cascading Style Sheets (CSS) is defined internally or externally of HTML file to enhance the display capabilities of HTML web page or entire website. JavaScript can also be implemented in HTML files to provide computational capability in web documents [3].

Hypertext Preprocesses (PHP) is a common open-source scripting language that can be used almost with all existing database and it also can be embedded into HTML. It is different from the client-side JavaScript in which the PHP code is executed on the server and the HTML is generated to the client [4]. Figure 1 depicts the development of a module in a project. The content of the website project is stored in a MySQL database, which then uses the PHP programming language to access the data using Structured Query Language (SQL) queries and send it to the browser for HTML display [5].

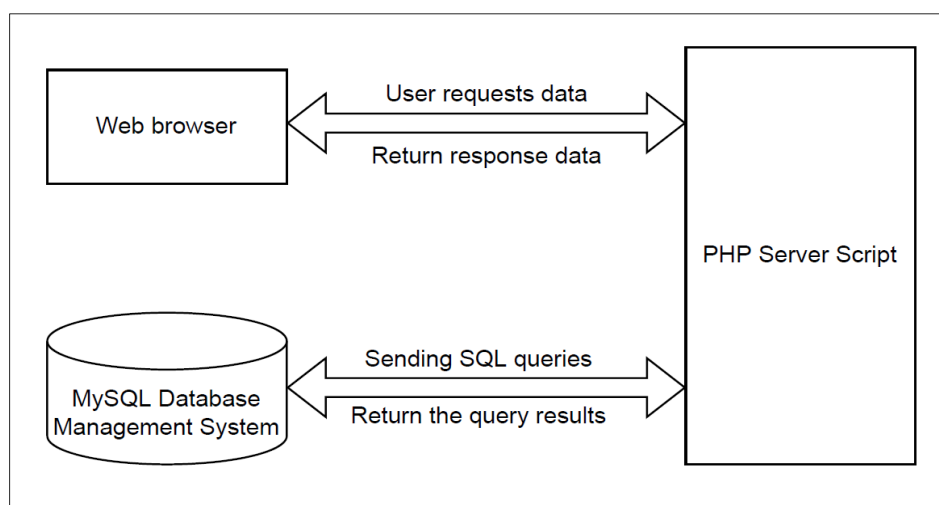


Figure 1: Web-based System Database [4]

2.2 Mobile Platform

Android is an open-source mobile operating system based on the Linux kernel, initially developed by Android Inc. and later acquired by Google in 2005. It provides a platform for developers to write managed code in the Java language, utilizing Google Java libraries to control the device. The Android SDK offers the necessary tools and frameworks for creating Java-based applications for Android-powered devices. Manufacturers can modify the system, including hardware and software layers to meet their specific requirements. The official IDE for Android development is Android Studio, built on JetBrains' IntelliJ IDEA software. It provides features such as the Android Gradle plugin, emulator, and integration with Firebase and cloud services. XML is used in Android Studio to define the UI layout, while Java serves as the controller to define class objects and file directories [6].

Java, known for its wide acceptance and versatility, is an object-oriented programming (OOP) language extensively used for developing applications across various platforms, including servers, desktop computers, and mobile devices. Java's key advantages lie in code reusability through software modules, extensibility through the addition of small collaborating modules, and improved maintainability due to the system being built with smaller, interconnected applications [7].

2.3 Database Management System (DBMS)

Database management system is a general-purpose software that ease the definition, manipulation, construction and sharing of databases among diverse users and applications. A database may be modified by doing operations like querying it to get particular data, updating it to account for changes in the entities involved and creating reports from the data. Hence, by sending Structured Query Language (SQL) queries or data requests to the DBMS, an application software can access the database and perform operations. Nonetheless, multiple users and applications could access the same database simultaneously when it is shared.

One of the most popular DBMS is MySQL with its latest version of 8.0. It is a relational open-source DBMS developed by Oracle. It is most of the time being used with phpMyAdmin which provides a Graphical User Interface (GUI) functionality for creating and submitting queries as well as to administer the database [8]. Therefore, some of the significance of using DBMS as an approach to store the system data is the control of data redundancy by practicing data normalization, provide security and authorization subsystem in which the information can only be accessed by authorized user and enforcing referential integrity constraint.

2.3 Study of Existing Related Systems

This section describes the study of existing related systems. The three existing systems that will be discussed in this section are School2me, Illumine and Bloomz. Table 1 shows the comparison between the existing systems and the proposed School Alert application.

Table 1: Comparison of Existing Systems with the Proposed System

Features	School2me [9]	Illumine [10]	Bloomz [11]	SchoolAlert
System Type	Mobile application for both teacher and parents	Mobile application and web-based for both teacher and parents		Mobile application for parent and web-based for teacher
Notification/Reminder		Yes		
Attendance Module	Yes	No, only exist in teachers' module	No, only inform parents in	Yes

Features	School2me [9]	Illumine [10]	Bloomz [11]	SchoolAlert
			notification section	
Activities Module			Yes	
Financial Module	Yes, online payment gateway	Yes, pay in cash or check	No, only notify parents on the fee payment	Yes, upload receipt as a PDF file
Academic Performance Module	No	Yes	No, only receive report card in private message	Yes
Chatroom	Yes, private message only		Yes, group message and private message	Yes, private message only

All parent-teacher engagement systems have the basic functions of login and registration, user profile, activities module and chat room. The attendance module exists in School2me and School Alert application but this function does not exist in the parent module of Illumine and Bloomz apps. The financial module exists in all the system except Bloomz that only notifies about the payment in the notification section. Also, the academic performance module is not provided in School2me and Bloom software. In the aspect of chat message, all applications have the features.

3. Methodology

The methodology of simple Software Development Life Cycle (SDLC) has been selected to develop the School Alert Mobile Application and the web-based administrative system. The SDLC provides a comprehensive plan for step-by-step creating a software system. There are four important phases in SDLC model which are analysis, design, implementation and testing phase as shown in Figure 2.

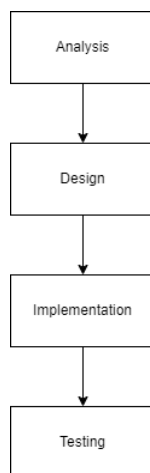


Figure 2: Simple SDLC

Nonetheless, Figure 3 shows the swimlane diagram of the new system. The system flow is visualized based on the project scope. The diagram uses two horizontal lane represents two different entities which are teacher as the admin and parent as the mobile user to show the interaction and collaboration between them. The swimlane diagram provides a clear and structured overview of the entire process, enabling stakeholders to understand the workflow, identify potential bottlenecks or inefficiencies in the system.

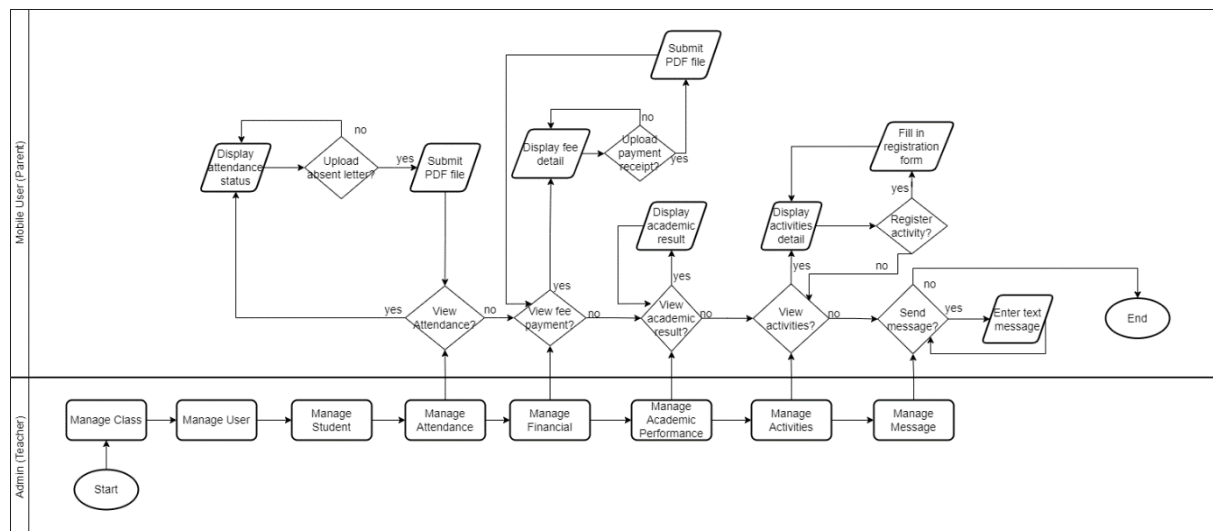


Figure 3: Swimlane Diagram of School Alert Application: Parent-Teacher Engagement System

3.1 Analysis and Design Phase

The analysis phase of the system development process involves understanding the user requirements gathered to gain a clear understanding of the system's behavior [12]. This phase focuses on comprehending the specific functionalities, as outlined in Table 2, that the system needs to fulfill, as well as addressing the non-functional requirements listed in Table 3.

Table 2: Functional Requirements of School Alert Application: Parent-Teacher Engagement System

Modules	Functionalities
Login	<ul style="list-style-type: none"> The system shall be able to validate identity of user and staff.
Register	<ul style="list-style-type: none"> The system should store the admin information in the database.
Manage Class	<ul style="list-style-type: none"> The system shall be able to display existing class enrolled by each teacher The system shall be able to register teacher on a new class
Manage User	<ul style="list-style-type: none"> The system shall be able to display user information The system shall be able to register new user information in database The system shall be able to edit or delete existing user information
Manage Student	<ul style="list-style-type: none"> The system shall be able to display student information The system shall be able to add new student information in database The system shall be able to edit or delete existing student information
Manage Attendance	<ul style="list-style-type: none"> The system shall be able to display attendance information The system shall be able to add new attendance in database The system shall be able to edit or delete existing attendance information The system shall be able to upload absent letter
Manage Activities	<ul style="list-style-type: none"> The system should display all existing school activities The system shall be able to create new activity The system shall be able to edit or delete activity The system shall be able to make registration for the activity

Modules	Functionalities
Manage Financial	<ul style="list-style-type: none"> • The system should display all financial statement • The system shall be able to create new financial statement • The system shall be able to edit or delete financial statement • The system shall be able to upload the payment document
Manage Academic Performance	<ul style="list-style-type: none"> • The system shall be able to display academic result of student • The system shall be able to add new student academic result • The system shall be able to edit or delete the financial statement
Manage Message	<ul style="list-style-type: none"> • The system should be able to send message between admin and user • The system should be able to store message information in database

Table 3: Non-Functional Requirements of School Alert Application: Parent-Teacher Engagement System

Requirements	Description
Performance	<ul style="list-style-type: none"> • The interaction between the system and user of mobile application as well as between the system and admin of the administrative web-based system should not exceed 5 seconds
Operational	<ul style="list-style-type: none"> • The mobile devices and desktop computer should be operated with internet availability. • The mobile application and administrative web-based system should be able to read from the database and write to the equivalent database
Security	<ul style="list-style-type: none"> • The admin can login and register account using their registered staff ID and password • Only authorized mobile user can access the mobile application using valid user ID and password registered by admin
Availability	<ul style="list-style-type: none"> • The mobile application and administrative web-based system should be available for use to the mobile user and system admin all the time especially during working hour

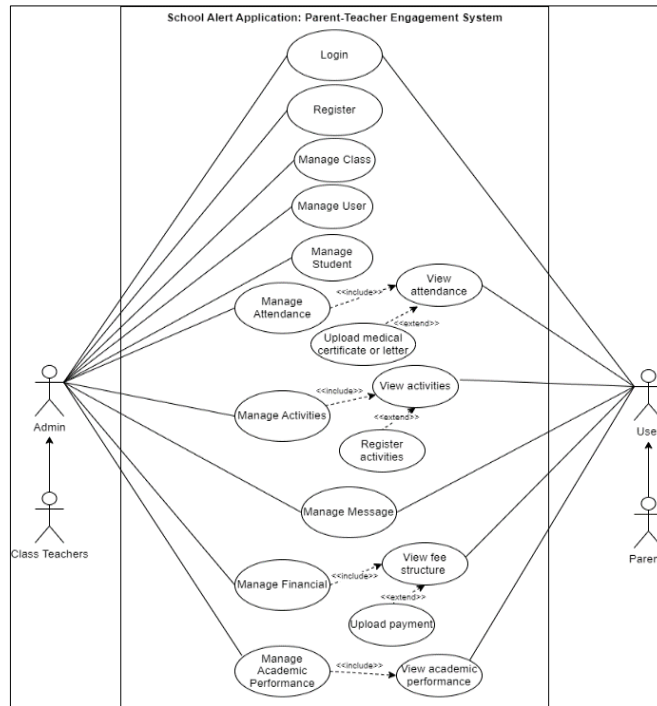


Figure 4: Use Case Diagram

Figure 4 showcases the key functions and interactions between the parent (mobile user) and the teacher (admin) of SJI International School in a use case diagram. This diagram is based on the functional requirements outlined in Table 2. The diagram illustrates that the admin has 10 management modules, while the mobile user has 6 modules. Use case diagrams are created during the analysis phase of the project to provide a clear overview of the system's functions and components.

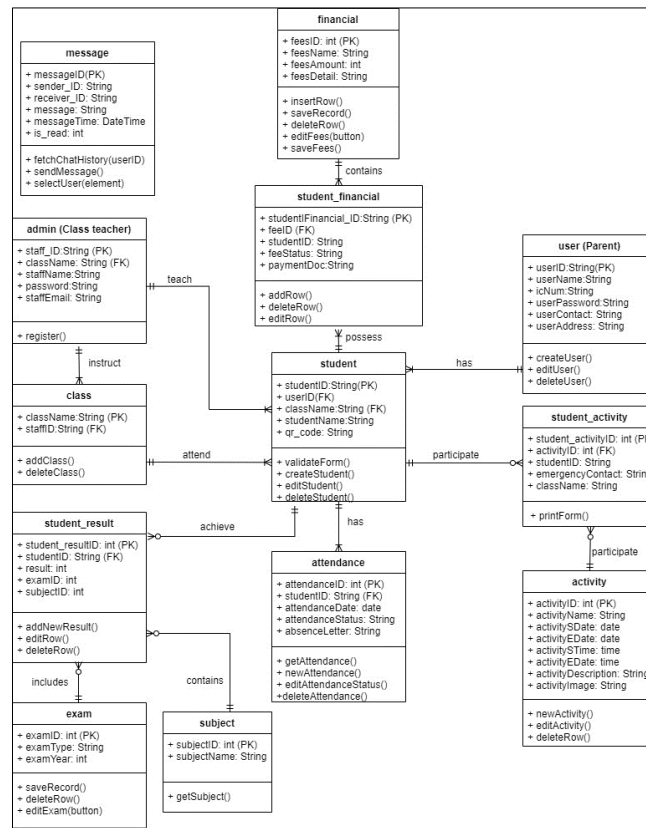


Figure 5: Class Diagram

In the design phase, the structure for School Alert application and the administrative web-based system is defined. The class diagram is developed to define the system design such as classes, attributes and their relationships. The class diagram serves as blueprints for the implementation phase, ensuring the system is developed according to the desired functionalities and specifications. There are a total of 13 classes as shown in Figure 5.

3.2 Implementation Phase

During the implementation phase, the actual system is created using Java for the mobile application, HTML and PHP for the web-based administrative system and MySQL for the database. This phase focuses on developing functional modules within the system and certain program code is provided to enhance clarity.

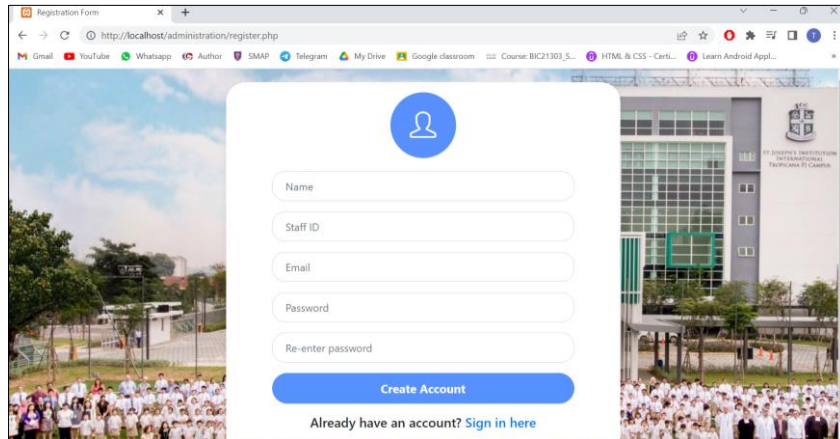


Figure 6: Admin Account Registration User Interface

Figure 6 shows the user interface of the admin account registration page, which includes input boxes for name, staff ID, email, password, re-enter password, and a register button to submit all the user input. The system checks for non-empty inputs and verifies if the passwords match. It also checks if the staff ID already exists in the database. If the ID is not found, the new ID is to add the admin database.



Figure 7: Mobile User Account Login User Interface

Figure 7 shows the user interface with text fields for user ID and password. This system executes user verification upon clicking the "LOG IN" button and output toast message whether the user had successfully login to the system or not.

```

1 <?php
2 include('database.php');
3 // Check if class name already exists
4 $teacherID = $_POST['teacherID'];
5 $className = $_POST['className'];
6 $sql = "INSERT INTO class (className,staffID) VALUES ('$className','$teacherID')";
7 $updateSql = "UPDATE admin SET className = '$className' WHERE staff_ID = '$teacherID'";
8 $result = mysqli_query($conn, $sql);
9 $updateResult = mysqli_query($conn, $updateSql);
10
11 if ($result) {
12     // The class name exists in the database
13     echo "<script>alert('New class added!');window.location.href='manageClass.php';</script>";
14 } else {
15     // The class name does not exist in the database
16     echo "<script>alert('Failed! Unexpected error occur.');

```

Figure 8: Admin Add New Class Code Segment

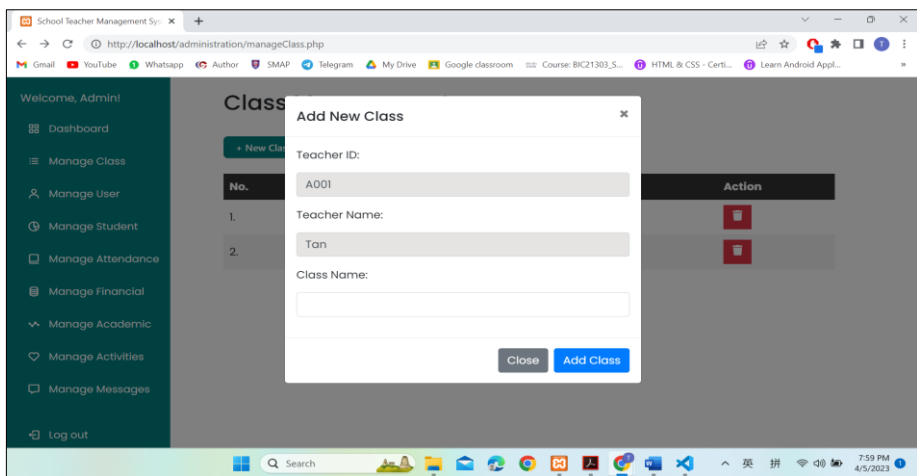


Figure 9: Admin Add New Class User Interface

Figure 8 features a PHP script that inserts a new class into the database. It retrieves values from a submitted form and constructs an SQL INSERT statement. The script executes the query using `mysqli_query()` and displays a success message using JavaScript. Figure 9 displays a modal pop-up form for adding a new class, which triggers the code in Figure 8 upon clicking the "Add Class" button.

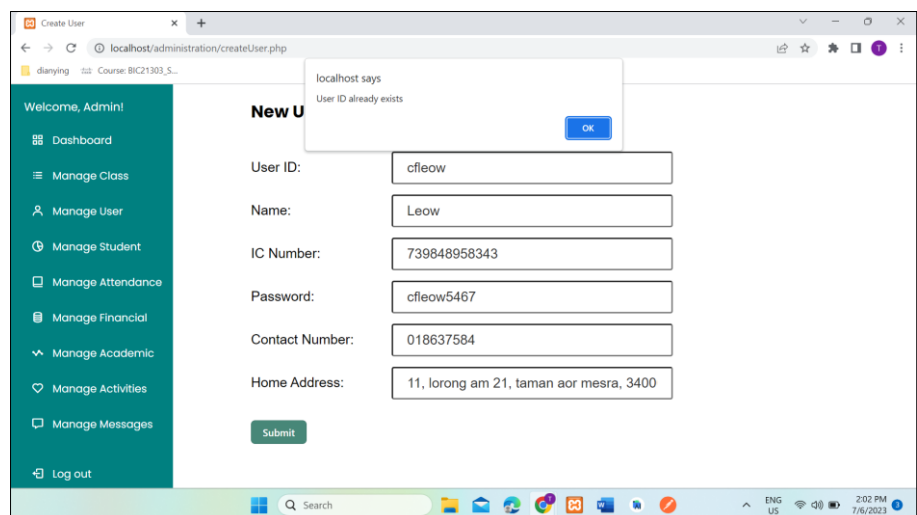


Figure 10: Admin Validate User Interface

Figure 10 illustrates the user interface for adding a new user. The system performs checking when the form is submitted to validate the user ID's existence and empty fields, triggering alert messages as necessary, else the system saves the new user information in database.

```

1 <?php
2 include('database.php');
3
4 if(isset($_POST["std_ID"])){
5
6     $std_ID = $_POST["std_ID"];
7     $std_name = $_POST["std_name"];
8     $std_class = $_POST["std_class"];
9     $std_user = $_POST["std_user"];
10
11     $sql = "UPDATE student SET studentName='$std_name',userID='$std_user' WHERE studentID='$std_ID'";
12
13     $result = mysqli_query($conn, $sql);
14
15     if (!$result) {
16         // Handle database error
17         die("Error: " . mysqli_error($conn));
18     } else {
19         // Update successful
20         echo "true";
21     }
22 }
23
24 mysqli_close($conn);
25 ?>

```

Figure 11: Admin Edit Student Code Segment

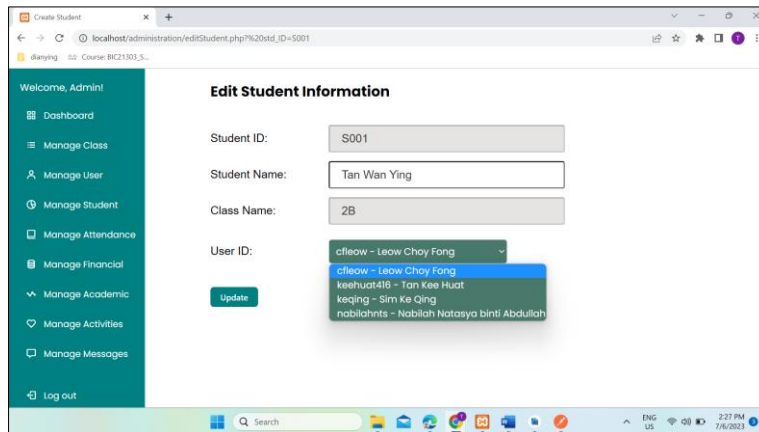


Figure 12: Admin Edit Student User Interface

The PHP code segment in Figure 11 updates a record in the "student" table of a database based on submitted information from an HTML form for editing students. If the "std_ID" parameter is present, the code retrieves the new values for the student's name and user ID. It constructs an SQL UPDATE statement to update the corresponding fields for the record with the provided student ID. The statement is executed using the "mysqli_query" function. Figure 12 shows the user interface for editing students, where the admin can modify the student's name and user ID using a dropdown list. The student ID and class name cannot be edited in this module. The "Update" button submits the edited information to the database by executing the code in Figure 11.

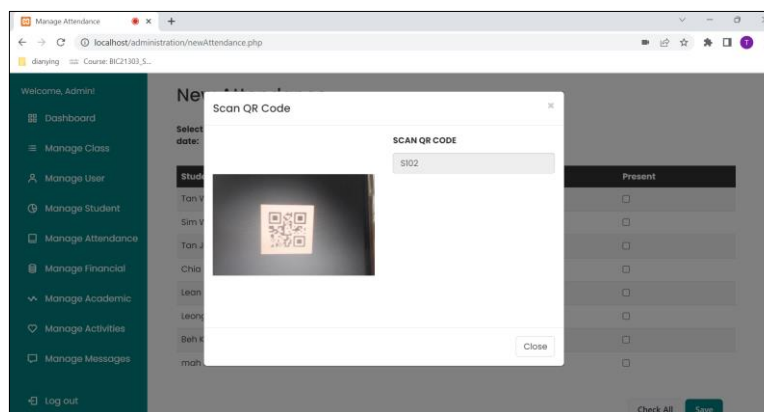


Figure 13: Scan QR Code User Interface

Figure 13 shows the user interface for scanning QR codes to mark student attendance. Admin selects the date and can manually check present students or use the QR code scanning feature. The scanned student ID is displayed in a modal window, and the corresponding checkbox is automatically checked upon scanning.

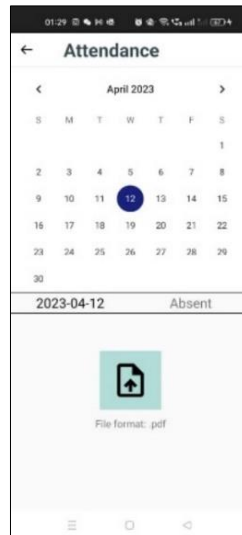


Figure 14: Mobile User Attendance User Interface

Figure 14 depicts the user interface of the attendance function in the mobile application, including date selection and displaying the attendance status for a specific student. The interface also features an "Upload PDF file" button for users to upload absence letters.

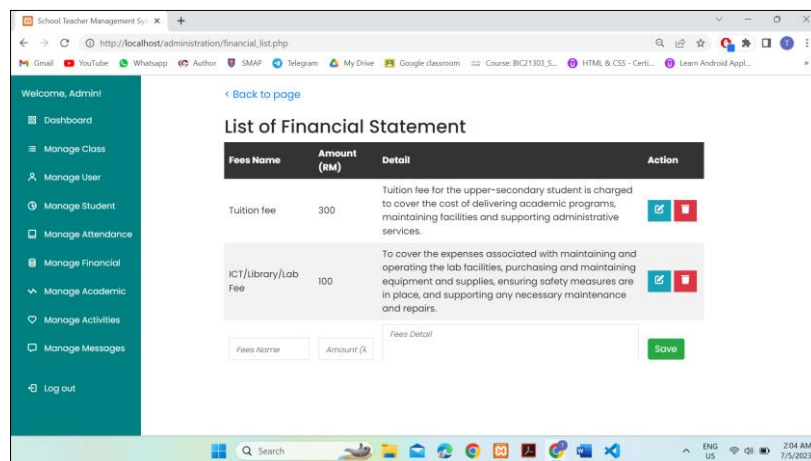


Figure 15: Admin Add New Financial User Interface

Figure 15 displays the admin interface for managing the list of financial statements. It includes columns for fee name, fee amount, details and corresponding actions. The "Add New" button generates new text fields and upon validation, the data is saved in the database. The edit and delete icon buttons perform server-side actions to update the database.

```

168 @Override
169 public View getView(final int position, View convertView, ViewGroup container) {
170     LayoutInflater inflater = LayoutInflater.from(container.getContext());
171     View view = convertView;
172     if (view == null) {
173         view = inflater.inflate(R.layout.fee, container, attachToRoot: false);
174     }
175     Log.d("ListViewIAdapter", "msg: getView() called for position: " + position);
176     LinearLayout lin4 = view.findViewById(R.id.lin4);
177     TextView fee_txt = view.findViewById(R.id.fee_txt);
178     TextView fee_amount = view.findViewById(R.id.fee_amount);
179     fee_txt.setText(_data.get(position).get("feesName").toString());
180     fee_amount.setText(_data.get(position).get("feesAmount").toString());
181     _clickAnimation(lin4);
182     lin4.setOnClickListener(new View.OnClickListener() {
183         @Override
184         public void onClick(View view) {
185             _clickAnimation(view);
186             Log.d("ListViewIAdapter", "msg: Clicked on item at position: " + position);
187             go.setClass(getApplicationContext(), PayActivity.class);
188             go.putExtra("name", _data.get(position).get("feesName").toString());
189             go.putExtra("amount", _data.get(position).get("feesAmount").toString());
190             go.putExtra("details", _data.get(position).get("feesDetail").toString());
191             startActivity(go);
192         }
193     });
194 }

```

Figure 16: Get Fee Payment Details Code Segment

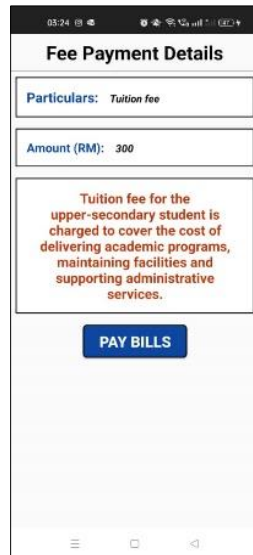


Figure 17: Fee Payment Details User Interface

Figure 16 contains code segment for retrieving fee payment details in the fee structure module. It configures views in a ListView adapter's getView() method, displaying fee information based on the data list. Clicking on an item passes relevant information to another activity. Figure 17 shows the user interface of the fee payment details page, accessed by clicking on a fee. It displays fee details and includes a "Pay Bills" button for uploading payment PDF files.

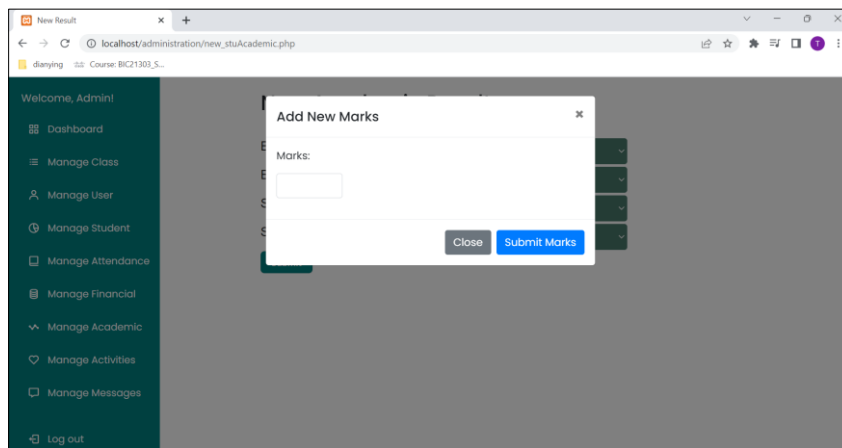


Figure 18: Admin Add New Marks User Interface

Figure 18 shows a modal pop-up form for inserting new marks for a student. The form is accessed from the "add new academic performance" page by selecting exam details and student information if the information selected to add new mark does not exist in database. Then, clicking the "Submit Marks" button will insert the data into the database.

```

377     for (int i = 0; i < tableData.size(); i++) {
378         TableRowData rowData = tableData.get(i);
379
380         TableRow dataRow = new TableRow(context: AcaPerformActivity.this);
381
382         TextView subjectTextView = new TextView(context: AcaPerformActivity.this);
383         subjectTextView.setText(rowData.getSubject());
384         subjectTextView.setPadding(left: 16, top: 16, right: 16, bottom: 16);
385         subjectTextView.setTextColor(Color.BLACK); // Set text color
386         dataRow.addView(subjectTextView, new TableRow.LayoutParams(0, TableRow.LayoutParams.WRAP_CONTENT, 1f));
387
388         TextView markTextView = new TextView(context: AcaPerformActivity.this);
389         markTextView.setText(rowData.getMark());
390         markTextView.setPadding(left: 16, top: 16, right: 16, bottom: 16);
391         markTextView.setTextColor(Color.BLACK); // Set text color
392         dataRow.addView(markTextView, new TableRow.LayoutParams(0, TableRow.LayoutParams.WRAP_CONTENT, 1f));
393
394         // Add the data row to the TableLayout
395         tableLayout.addView(dataRow);
396
397         // Add separator between rows
398         if (i < tableData.size() - 1) {
399             View separator = new View(context: AcaPerformActivity.this);

```

Figure 19: Generate Result Table Code Segment

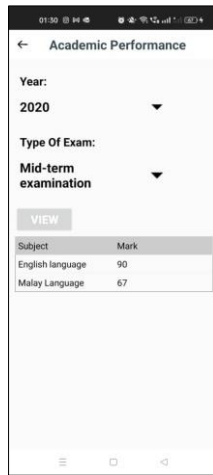


Figure 20: Mobile User Academic Performance User Interface

Figure 19 contains code segment for generating a result table using data from a web view. The generateTable method creates a header row with "Subject" and "Mark" columns, then iterates through the tableData list to create rows with subject and mark data. Figure 20 shows the user interface of the academic performance page in the mobile app, where users can select the year and type of exam from dropdown menus. Clicking the "VIEW" button displays the table with subject and mark data generated by the code in Figure 19.

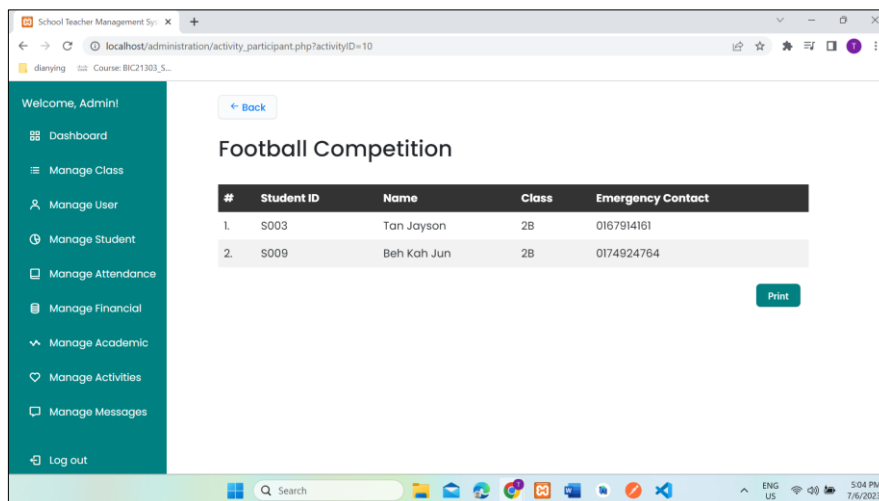


Figure 21: Admin View Activity Participant User Interface

Figure 21 shows the user interface with a table list of activity participants and a "Print" button. Clicking the button prompts a print dialog and allow admin to print the participant list.

```

180 RequestQueue requestQueue = Volley.newRequestQueue( context: this);
181 StringRequest stringRequest = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {
182     @Override
183     public void onResponse(String response) {
184         Log.d( tag: "response", response);
185         if (response.equalsIgnoreCase( anotherString: "success")) {
186             Toast.makeText( getApplicationContext(), text: "Registered Successfully", Toast.LENGTH_SHORT).show();
187         } else {
188             Toast.makeText( getApplicationContext(), text: "Registration failed", Toast.LENGTH_SHORT).show();
189         }
190     }
191 }, new Response.ErrorListener() {
192     @Override
193     public void onErrorResponse(VolleyError error) {
194         Log.d( tag: "error", msg: ""+error);
195     }
196 }) {
197     @Override
198     protected Map<String, String> getParams() throws AuthFailureError {
199         Map<String, String> params = new HashMap<>();
200         params.put("studentID", studentID);
201         params.put("className", className);
202         params.put("contact", contact);

```

Figure 22: Activity Registration Form Submission Code Segment



Figure 23: Activity Registration Form User Interface

Figure 22 contains code segment that creates a URL for server-side script handling database insertion. It uses RequestQueue and StringRequest to make a POST request to the specified URL. The response listener handles the server's response, displaying success or failure messages. The getParams method provides the parameters for the request. Figure 23 shows the user interface of an activity registration form where the class name and emergency contact are entered. Clicking the "Submit" button executes the code in Figure 22, storing the participant's information in the database.

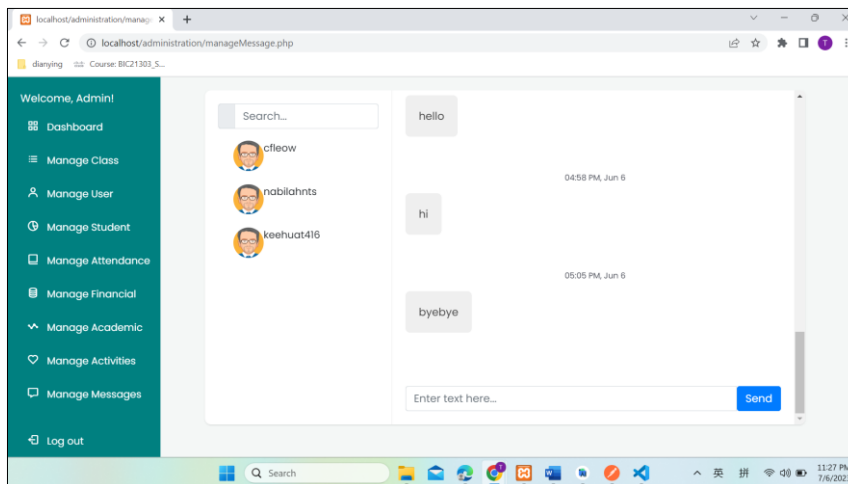


Figure 24: Admin Manage Message User Interface

Figure 24 shows the "Manage Message" user interface, allowing admin to search for users and chat with them after selecting the specific user icons. The admin can input and send messages using the text field and "Send" button, which saves the message in database.



Figure 25: Mobile User Message Center User Interface

Figure 25 shows the user interface of the message center for mobile users, where messages are aligned to the right or left depending on the sender. The interface includes a text field for typing messages and a "SEND" button for saving the message into database.

4. Results and Discussion

The results and discussion section discuss on the testing phase of the project. This phase presents system’s functionality testing using the planned test cases as well as the analysis of the results from the user acceptance testing that involve the target audiences which is the parent and teacher of SJI International School. Both of the users interact with the system and determine whether everything works as expected. The system has been tested to 10 respondents of parent who access the School Alert Android application and 10 respondents of teacher who access the web-based administrative system with the use of questionnaire in the form of Google Form.

4.1 Functionality Testing

Based on Table 4, the overall result of the test cases shows that there are a total of 41 test cases developed from 10 modules that had been conducted to test the School Alert application and web-based administrative system. The system has passed successfully.

Table 4: Overall Test Case Result

Test Case ID	Total Test Cases	Total Success	Total Fail
TC_100	2	2	0
TC_200	2	2	0
TC_300	3	3	0
TC_400	4	4	0
TC_500	4	4	0
TC_600	6	6	0
TC_700	5	5	0
TC_800	5	5	0
TC_900	6	6	0
TC_1000	4	4	0

4.2 User Acceptance Testing

User acceptance testing for the School Alert Mobile Application and web-based administrative system involved evaluating functionality and user experience on a scale of 1 to 5, from strongly disagree to strongly agree. The results were presented using a bar graph.

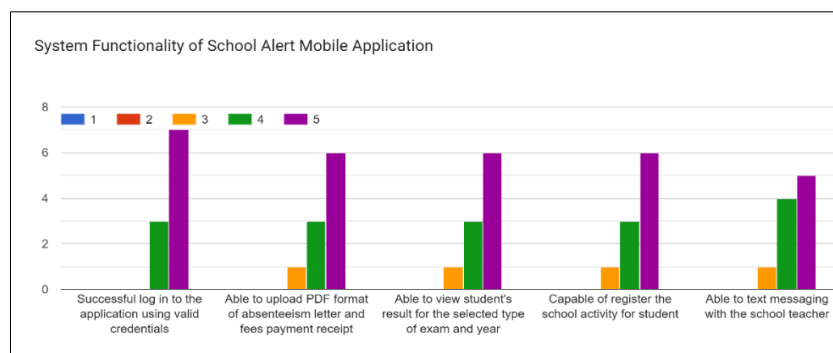


Figure 26: Bar Graph for School Alert Application Functionality Result

Table 5: School Alert Application Functionality Result

No.	Question	Scale					Total
		1	2	3	4	5	
1	Successful log in to the application using valid credentials	0	0	0	3	7	10
2	Able to upload PDF format of absenteeism letter and fees payment receipt	0	0	1	3	6	10
3	Able to view student's result for the selected type of exam and year	0	0	1	3	6	10
4	Capable of register the school activity for student	0	0	1	3	6	10
5	Able to text messaging with the school teacher	0	0	1	4	5	10

Based on Figure 26 and Table 5, the evaluation result of the School Alert application functionality shows a high level of satisfaction among the respondents. The majority agreed or strongly agreed on the successful login feature, uploading PDFs, viewing student's results, activity registration and text messaging with school teachers in the School Alert Mobile application.

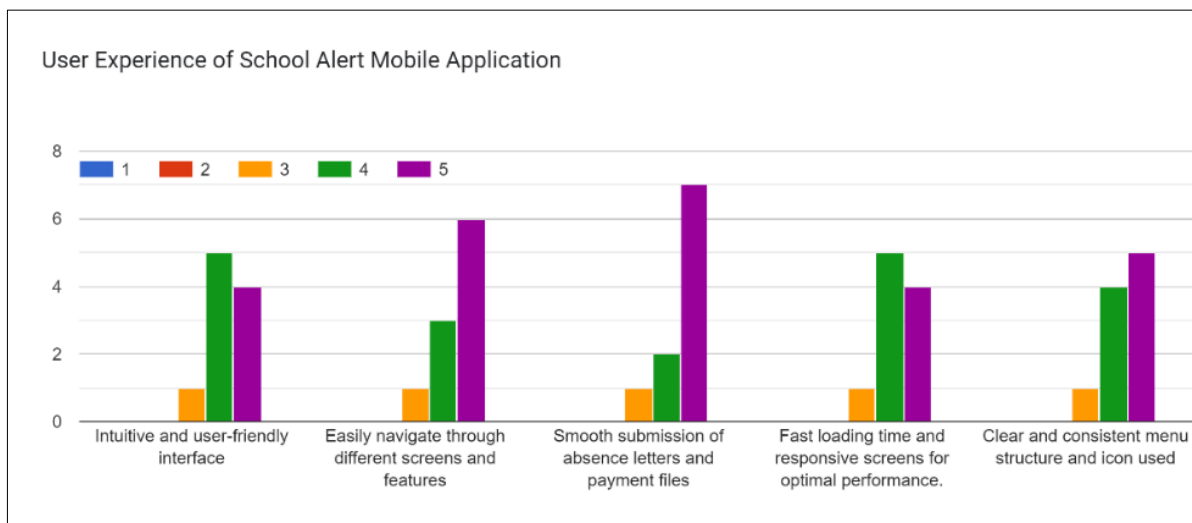


Figure 27: Bar Graph for School Alert Application User Experience Result

Table 6: School Alert Application User Experience Result

No.	Question	Scale					Total
		1	2	3	4	5	
1	Intuitive and user-friendly interface	0	0	1	5	4	10
2	Easily navigate through different screens and features	0	0	1	3	6	10
3	Smooth submission of absence letters and payment files	0	0	1	2	7	10

No.	Question	Scale					Total
		1	2	3	4	5	
4	Fast loading time and responsive screens for optimal performance	0	0	1	5	4	10
5	Clear and consistent menu structure and icon used	0	0	1	4	5	10

Based on Figure 27 and Table 6, the evaluation result of the School Alert application user experience indicates a high level of satisfaction among the respondents. The user experience feedback highlighted positive perceptions of the intuitive interface, fast loading times, responsive screens, easy navigation and clear menu structure in the School Alert Mobile application.

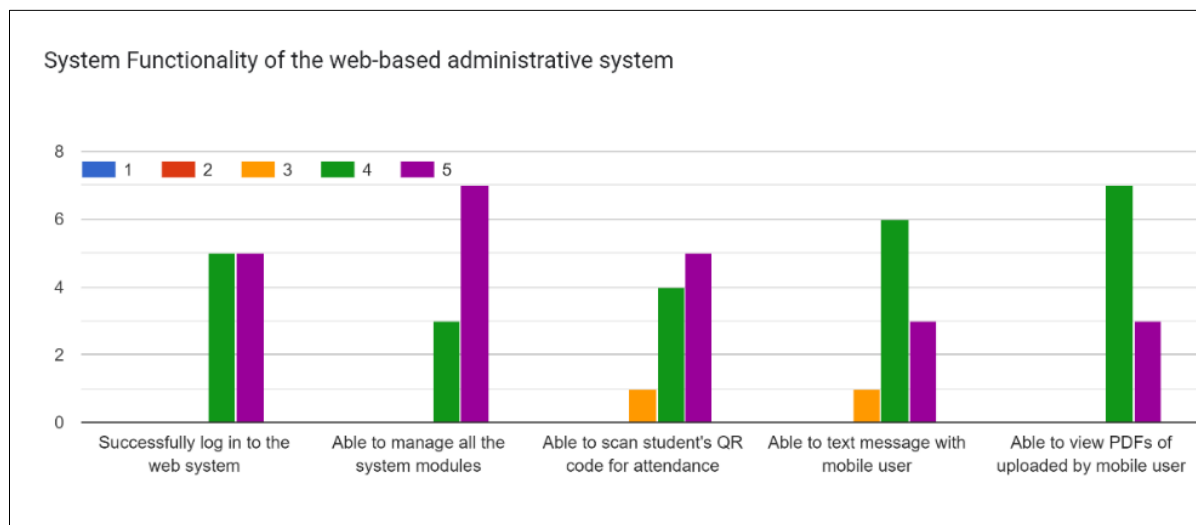


Figure 28: Bar Graph for the Administrative Web System Functionality Result

Table 7: Administrative Web System Functionality Result

No.	Question	Scale					Total
		1	2	3	4	5	
1	Successfully log in to the web system	0	0	0	5	5	10
2	Able to manage class, mobile user, student, attendance, financial, academic, activities	0	0	0	3	7	10
3	Able to scan student's QR code for attendance management.	0	0	1	4	5	10
4	Able to text message with mobile user	0	0	1	6	3	10
5	Able to view PDFs of uploaded absent letters and payment receipts.	0	0	0	7	3	10

The evaluation results of the administrative system's functionality, as depicted in Figure 28 and summarized in Table 7, indicates a high level of satisfaction among the respondents. The majority of participants agreed or strongly agreed with the successful login feature, management ability, QR code scanning feature, view PDFs upload by mobile user and text messaging with mobile user who is the

parent using the administrative web system. Overall, the results suggest a high level of satisfaction with the administrative system's functionalities.

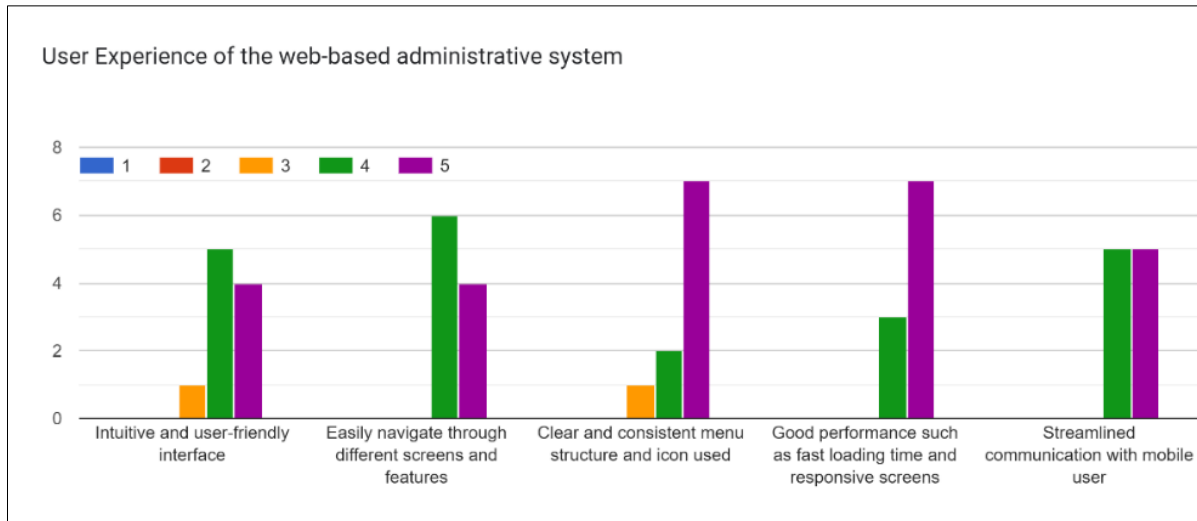


Figure 29: Bar Graph for Administrative System User Experience Result

Table 8: Administrative System User Experience Result

No.	Question	Scale					Total
		1	2	3	4	5	
1	Intuitive and user-friendly interface	0	0	1	5	4	10
2	Easily navigate through different screens and features	0	0	0	6	4	10
3	Clear and consistent menu structure and icon used	0	0	1	2	7	10
4	Good performance such as fast loading time and responsive screens	0	0	0	3	7	10
5	Streamlined communication with mobile user	0	0	0	5	5	10

Referring to Figure 29 and Table 8, the evaluation results of the administrative web system's user experience indicate that a significant proportion of respondents agreed or strongly agreed on various aspects, including user-friendly interface, easy navigation through different screens, clear and consistent menu structure and icon, good system performance and smooth communication with mobile user. Overall, the results suggest a high level of satisfaction with the administrative system's user experience.

5. Conclusion

In conclusion, the School Alert application and the administrative web-based system bring advantages such as QR code attendance scanning, messaging features and easy access for both parents and teachers in getting information. However, there are also limitations, including the need for active checking of information and restricted access for certain mobile operating systems. Recommendations for improvement include implementing notification features for all modules, expanding mobile application compatibility and adding a subject management module to the web-based system. Overall, the developed School Alert solution holds potential for implementation in various educational settings, offering enhanced monitoring and management capabilities for parents and educators.

Acknowledgment

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

References

- [1] G. Moore, "A sea change in enterprise IT: Systems of engagement and the future of enterprise IT," AIIM White Paper. Silver Spring, MD: AIIM, 2011. [Online]. Available: https://cdn2.hubspot.net/hub/332414/file-2314261840-pdf/Inbound_Assets/Systems-of-Engagement.pdf. [Accessed Dec 1, 2023]
- [2] Mian Mian Seet, M. Rabbani, S. Hosseinian, Ravindran Latha, and S. B. V. Rajan Mohan, "Parenting Styles and Academic Achievement among Malaysian Students: Mediating Role of Parental Involvement," *Journal of Cognitive Sciences and Human Development*, vol. 8, no. 1, pp. 122–138, Mar. 2022, doi: <https://doi.org/10.33736/jcshd.3951.2022>.
- [3] R. Connolly and R. Hoar, *Fundamentals of Web Development*. England: Pearson, 2017.
- [4] J. Duckett, *PHP & MySQL: Server-Side Web Development*. Wiley, 2022.
- [5] X. Chen and J. Zhang, "The Applications PHP, HTML and MYSQL in Development of Website – Query Function," *ICMLCA 2021; 2nd International Conference on Machine Learning and Computer Application*, pp. 1-4, 2021.
- [6] P. Khanna and A. Singh, "Google Android Operating System: A Review," *International Journal of Computer Applications*, vol. 147, no. 4, pp. 26–29, Aug. 2016, doi: [10.5120/ijca2016911008](https://doi.org/10.5120/ijca2016911008).
- [7] J. Lewis and W. Loftus, *Java Software Solutions: Foundation of Program Design*. England: Pearson, 2008.
- [8] A. Silberschatz, *Database system concepts*. New York: Mcgraw-Hill Education, 2020.
- [9] (2017). School2me (Version 3.3.151) [Mobile app]. Retrieved from Google Play Store. <https://play.google.com/store/apps/details?id=com.school2me.user>.
- [10] (2019). Illumine (Version 424s) [Mobile app]. Retrieved from Google Play Store. <https://play.google.com/store/apps/details?id=com.illumine.app>.
- [11] (2015). Bloomz (Version 4.72) [Mobile app]. Retrieved from Google Play Store. <https://play.google.com/store/apps/details?id=net.bloomz>.
- [12] A. Dennis, Barbara Haley Wixom, and R. M. Roth, *Systems analysis and design*. Hoboken, Nj: Wiley, 2021.