

The Development of LITTLECARE: Childcare Management System Using Laravel Framework

Intan Nur Fatiha Saahari¹, Nur Ariffin Mohd Zin^{1*}

¹ Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: ariffin@uthm.edu.my
DOI: <https://doi.org/10.30880/aitcs.2025.06.02.096>

Article Info

Received: 18 July 2025

Accepted: 20 November 2025

Available online: 30 November 2025

Keywords

Childcare management system,
Management information system,
Monitoring camera.

Abstract

Manual handling of childcare operations at Taska Hikmah, Batu Pahat, often leads to inaccuracies, inefficient data management, and lack of real-time children monitoring. This study aims to develop and evaluate LITTLECARE: Childcare Management System to automate attendance tracking, daily event updates, billing processes and provide real-time monitoring for administrators, staff and parents. The system was designed through a structured development approach with the prototyping model, and it used Laravel as the backend framework, HTML, CSS, and JavaScript for the frontend, and MySQL for data storage. The system was evaluated through alpha beta testing. User Acceptance Testing evidence displayed high satisfaction among 90% of respondents, confirming the system success in solving the inefficiencies of childcare operation management. The system significantly increased data accuracy and overall efficiency, including suggestions for the future to integrate into mobile applications along with deriving AI-driven perceptions to allow better safety measures to enhance user experience and accessibility.

1. Introduction

Childcare is a place to support the growth and well-being of preschoolers by offering a secure environment, creative materials, physical and intellectual, and a cooperative parent-teacher connection [1]. In Malaysia, taska, which stands for Taman Asuhan dan Didikan Awal Kanak-Kanak, operates within the Childhood Care and Education industry. Taska is where children are taken care of and managed regularly by other individuals with licenses [2]. To develop and increase the number of high-quality children, having high-quality learning spaces and experiences is important, especially in this era. Hence, the Malaysian government initiated a step by providing subsidies for childcare in Malaysia by making it accessible and affordable. The subsidies aim to improve childcare for children to have safe, high-quality learning and education spaces while aiming to enhance the spirit inside children to learn and grow with confidence [3]. Developing a system for childcare management purposes is one of the strong options for efficient and effective management. A good childcare management system will enhance productivity and give parents trust and peace of mind.

Furthermore, the case study in this project is conducted at Taska Hikmah, located at Universiti Tun Hussein Onn Malaysia (UTHM), Parit Raja, Johor. Currently, the existing process of Taska Hikmah's management is still performing and collecting data manually using various software and applications. For registration, parents had the option to register using a printed form or Google Form, and the details will be uploaded and updated by staff in Excel and Google Drive. Parents are provided with a fee card, and payments are made by cash or online transfer, and proof is submitted via email or WhatsApp. WhatsApp group is maintained for daily updates. Safety measures to monitor children through CCTV can only be viewed in the manager's room.

Therefore, the problems encountered when they still use the current conventional method are inefficient in handling and distributing data and information. Manual processes lead to human error. Additionally, staff spend more time and effort searching through multiple platforms to locate specific documents or information. Also, there is a lack of real-time monitoring and updates among nursery staff, which causes delays during emergencies, causing concern among parents.

The project LITTLE CARE: Childcare Management System is proposed to address these problems by providing more efficient and effective childcare management. This system provides a better process for managing, handling, storing, and distributing the children's daily data and information while being a reliable system for the parents from the real-time camera that can view their kids virtually without concern about their children's safety.

This paper consists of five sections. Section 1 introduces the project background. Section 2 describes the related works of the project. Section 3 describes the project methodology used and illustrates the system analysis and design. Section 4 discusses the final project implementation and results. Lastly, section 5 will be the conclusion of this report.

2. Literature Review

2.1 Management Information System (MIS)

A Management Information System (MIS) is an information system that collects, processes, stores, and summarizes data from different parties and presents it in a form that helps management make better decisions [4]. In this project, data and information management are compulsory, and this project will be designed and integrated into all components of childcare management, including the children, staff, and parent management, attendance, billing, and reporting, into one system.

2.2 Web Technology System

In terms of technology, this project will apply a web technology approach. Web technology is defined as using markup language and multimedia packets to communicate between computers, making it easy to interact with online content such as web pages [5]. In this system, web technologies connect camera hardware and end users.

2.3 Client-Server Architecture

In the information technology (IT) context, the client is an individual or an organisation that uses the IT services. A server is a part of the computer system that provides clients access to data and services. The client-server architecture refers to part of the computer network architecture that serves the host, delivers, manages the resources, processes the client requests, and returns the request to clients [6]. This is used in the proposed system to ease the centralized control between clients (admin, staff, and parents). Client-server architecture helps to enhance effectiveness and efficiency in delivering resources while decreasing the staff workload.

2.4 Real-Time Camera Technology

A web camera is used for this project to clone or act like a CCTV. A webcam is a small device built for a laptop that captures video and audio data. Then, it is transmitted in real-time via the internet [7]. APIs are used for webcam integration, such as `getUserMedia` API, to permit to get webcam access for the browser [8]. The generic webcam integration is simpler and compatible with any browser while maintaining video quality. The proposed system uses the webcam functionality as a real-time camera that captures live video for administrators, staff, and parents to view for real-time monitoring.

2.5 Programming Framework

The Laravel platform was chosen as a framework for the proposed system. Laravel is an open-source PHP web application framework developed by Taylor Otwell, and provides a modern web with an expressive and elegant syntax [9]. The Laravel framework uses MVC architecture. The model of MVC manages application logic, the view of MVC handles and displays the user interface, and the controller of MVC controls the data flow [10]. Laravel framework is suitable for use in the proposed system as the modern features align with the system's needs and support a better management process of childcare.

2.6 Database Management System

MySQL database is chosen as the database for the proposed system. MySQL is an open-source relational database management system (RDBMS) that can manage a large amount of data and complex queries that not just store and retrieve but also modify the databases [11]. MySQL is a suitable database for the proposed system to ensure smooth operations since it provides scalability and ease of use and supports complex queries to handle large amounts of data in the system [12].

2.7 Comparison of Existing System

Taska Khairiyah Wafiq Management System is a web-based system focusing on managing children's development and admin efficiency [13]. The Permataku Monitoring System is a system that focuses on the digital logbook for children's progress and activity [14]. MyKidzDay is a web-based and cloud-based application that manages daily operations for childcare, preschools, homecare providers, before and after care, special needs programs, summer programs, and schools [15]. The three existing systems have been studied, including the proposed system, LITTLECARE: Childcare Management Systems, to explore their features, strengths, and weaknesses. These systems have been compared in detail in Table 1.

Table 1 System comparison

Features/System	Taska Khairiyah Wafiq Management System [13]	Permataku Monitoring System and Student Development of Taska Qaseh Umami [14]	MyKidzDay [15]	LITTLECARE: Childcare Management Systems
System type	Web-based	Web-based	Web-based, Cloud-based application	Web technology
Programming language	PHP, HTML, CSS	HTML, CSS	Unknown	PHP, CSS, JavaScript
Login and Registration module	Yes	Yes	Yes	Yes
Attendance module	No	No	Yes	Yes
Daily board module	No	Yes	No	Yes
Announcement module	No	No	No	Yes
Monitoring camera	No	No	No	Yes
Payment module	Yes	Yes	Yes	Online gateway
Report generation module	No	No	No	Yes
Notification	No	Notify parents when they are late for pickup and who are still pending payments.	Notify parents of upcoming due date payments.	Notify parents of their children's attendance and who are still pending payments.
Development progress	Yes	Yes	No	No

Based on Table 1, there are 3 childcare management that have been compared with the proposed system. All four childcare management systems work through web-based or websites, but only MyKidzDay uses both web-based and cloud-based applications. Every system allows parents to register their children, log in, and handle payments, but their operations differ. For example, in tracking children's attendance, MyKidzDay uses QR to update attendance, while LITTLECARE: Childcare Management Systems updates automatically on the system when the attendance option is selected. While Permataku and Taska Qaseh Umami focus on daily updates and late pick-up alerts. Taska Khairiyah Wafiq Management System is simpler, tracking development progress. The proposed system is different from others by having live cameras, report generation, and a news board for parents. All these systems aim to achieve organized and efficient childcare management for staff and parents.

3. Methodology

3.1 System Development Workflow

The software development life cycle used in the proposed system is the prototyping model, which divides complex problems into manageable tasks for effective application development [16]. The prototyping method excels quickly in getting everyone on the same page, getting feedback and refinement, and easily can detecting errors earlier. The main phases in this project are the planning phase, prototype design phase, prototype development phase, user evaluation phase, implementation phase, and testing phase. Table 2 shows the software development activities and their output in developing the proposed system.

Table 2 System development workflow

Phase	Task	Output
Planning	<input type="checkbox"/> Proposed the project	<input type="checkbox"/> Project proposal
	<input type="checkbox"/> Determine the project schedule, activities and output	<input type="checkbox"/> Gantt chart <input type="checkbox"/> DFD
	<input type="checkbox"/> Identify stakeholder	<input type="checkbox"/> ERD
	<input type="checkbox"/> Requirement gathering	<input type="checkbox"/> Flowchart
Prototype Design	<input type="checkbox"/> Determine requirements for module	<input type="checkbox"/> System architecture <input type="checkbox"/> User interface design
	<input type="checkbox"/> Design the system flow	<input type="checkbox"/> Database schema and data dictionaries
Prototype Development	<input type="checkbox"/> Determine prototype module function	<input type="checkbox"/> Initial prototype for each module
	<input type="checkbox"/> Prototype testing	<input type="checkbox"/> Refine prototype
User Evaluation	<input type="checkbox"/> Collect feedback from stakeholders for each module	<input type="checkbox"/> Refine prototype
Implementation	<input type="checkbox"/> full development of each module	<input type="checkbox"/> Develop a complete function system
Testing	<input type="checkbox"/> Developer and admin of Taska Hikmah testing	<input type="checkbox"/> Alpha testing <input type="checkbox"/> Beta testing
	<input type="checkbox"/> Staff and parents from Taska Hikmah testing	<input type="checkbox"/> Test cases
	<input type="checkbox"/> Determine errors and improvements	

3.2 System Requirements

The functional requirement is to identify the system's expected behavior to the targeted users. Table 3 shows the list of functional requirements in developing the proposed system.

Table 3 *Functional requirements*

No	Modules	Description
1	Register and Login	<ul style="list-style-type: none"> • Allow the administrator to manage account registration for staff, parents, and children. • Allow the administrator to receive and approve registration requests from parents. • Allow the administrator to create an account and send login credentials to staff and parents. • Allow the parents to register children into childcare by submitting a registration of children's information for approval before getting the login credentials. • Allow the existing users to log in with the ID and password given by the administrator. • Redirect the valid users to the dashboard when successful login.
2	Attendance Management	<ul style="list-style-type: none"> • Allow the administrator and staff to take children's attendance records. • Allow the administrator and staff to update children's attendance records. • Allow parents to view children's attendance list. • Allow parents to receive email notifications when children arrive and leave the childcare.
3	Children's Daily Board	<ul style="list-style-type: none"> • Allow the administrator and staff to create posts of children's daily activities. • Allow the administrator and staff to update the post. • Allow parents to view children's daily activity posts.
4	Kid Camera	<ul style="list-style-type: none"> • Allow the administrator to record the camera footage. • Allow the administrator to save the footage on the computer. • Allow all users to view the live footage.
5	Announcement	<ul style="list-style-type: none"> • Allow the administrator and staff to create memos and upcoming events. • Allow the administrator and staff to update memos and events. • Allow parents to view the memos and upcoming events. • Allow parents to receive email notifications from the childcare. • Administrators generate email notifications to parents about memos and upcoming events.
6	Billing and Payment Management	<ul style="list-style-type: none"> • Allow administrator to create payment details fee for parents. • Allow the administrator to update the parent's payment details. • Allow the administrator to generate email notifications to parents when payment is overdue. • Allow parents to make payments using a payment gateway. • Allow parents to receive email notifications when payment is overdue. • Allow all users to view the fee amount that needs to be paid. • Allow all users to view the parent's payment status (complete, pending, or overdue).
7	Report Generation	<ul style="list-style-type: none"> • Administrator generates a report list of children attendees and payment statements made by parents. • Allow the administrator to print or export reports in a PDF file.

The non-functional requirement is a software requirement that outlines how the software will function instead of what it will do [17]. Table 4 shows the list of non-functional requirements in the proposed system.

Table 4 Non-functional requirements.

No	Requirement	Description
1	Performance	<ul style="list-style-type: none"> The proposed system should always be usable, responding to each user's request and ensuring stability over time.
2	Reliability	<ul style="list-style-type: none"> The loading time required for a website is no more than 1 minute. The proposed system should stream and ensure that parents will have access to view from the cameras of the time during the childcare operating hours.
3	Security	<ul style="list-style-type: none"> To avoid any form of error, the proposed system must validate each input. To prevent unauthorized access, the proposed system should define the user's role and limit their access depending on that function. The proposed system should catch errors and warnings and properly display error messages without revealing technical specifics.
4	Compatibility	<ul style="list-style-type: none"> The proposed system must be compatible with any browser. Examples of browsers are Chrome and Microsoft Edge.
5	Usability	<ul style="list-style-type: none"> The proposed system should have an easy-to-use interface and operations.

3.3 System Analysis

In system analysis, context diagrams present the overview of the interaction between the system and its user. Fig. 1 shows the context diagram of the developed system. There are three external entities: the administrator, parent, and staff. The proposed system should be able to receive children's registration details, staff information details, login details, children's attendance details, children's daily activities details, announcement details, and payment details.

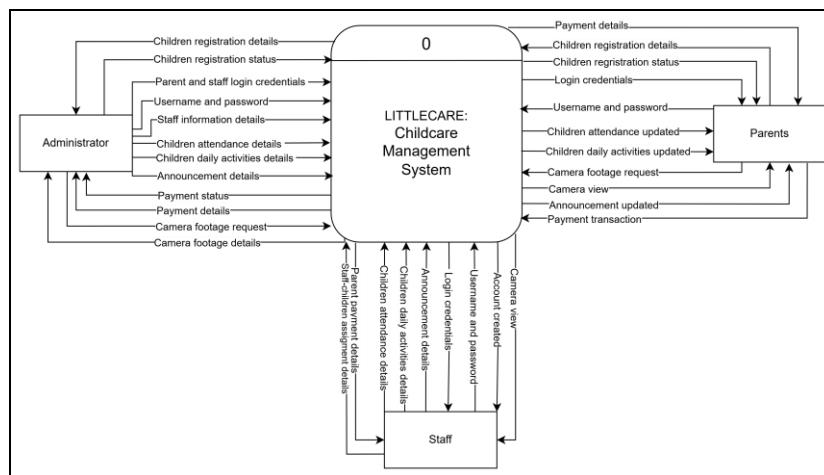


Fig. 1 Context diagram of proposed system

Moreover, the Data Flow Diagram (DFD) Level 1 of the proposed system is shown in Appendix A. DFD is a graphical representation of the flow of data or input from an entity through a process, which then generates output either to another entity or stored in data storage. There are seven processes, along with entities (administrator, staff, and parents), and 11 data stores in this proposed system. This DFD Level 1 shows how the

proposed system manages childcare management, including user login and registration, attendance management, children's daily board, kid camera, announcement, payment, and report generation.

Furthermore, an Entity Relationship Diagram (ERD) is a visual representation of how different entities relate to each other. Fig. 2 shows the ERD of the proposed system which contains 11 tables such as Users, Staff, Parents, Children, Enrolment, Staff_assignment, Attendance, Daily_activity, Camera_footage, Announcement, and Payment. This ERD shows the relationship between entities and their attributes.

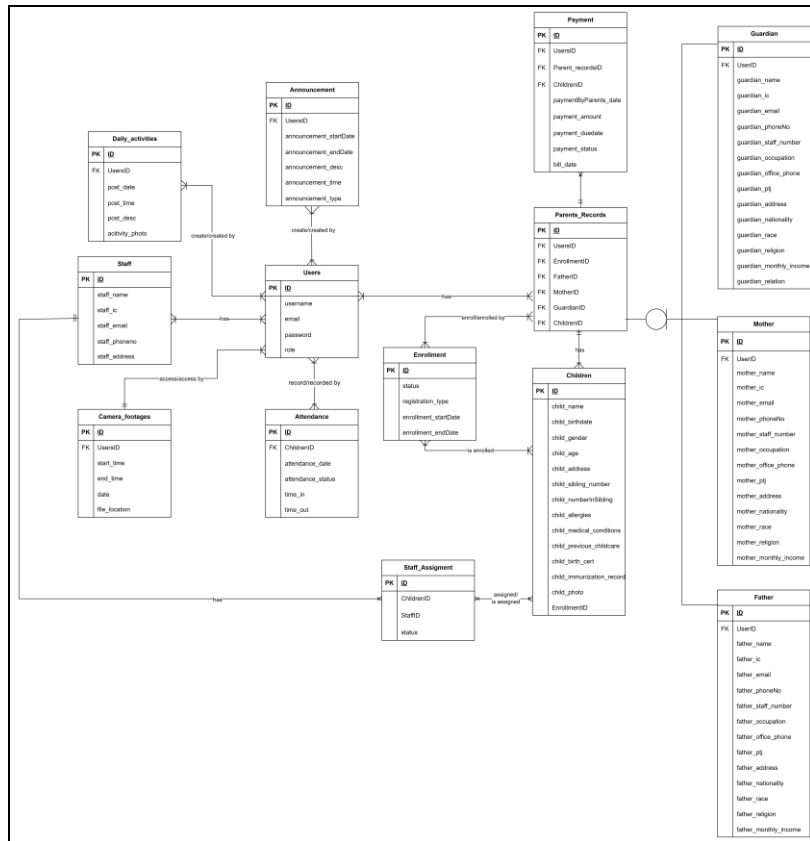


Fig. 2 ERD of the proposed system

Thus, the system flowchart diagram of parents, staff, and administrator activity in the proposed system is shown in Appendix B. This system flowchart shows the flow activity of respective users in using the proposed system.

3.4 System Design

The system architecture is a diagram that visualizes the communication and interaction between user and computer components [18]. Fig. 3 shows the system architecture of the proposed system, which visualizes the interaction between the user, frontend layer, middleware layer, backend layer, database, and external components.

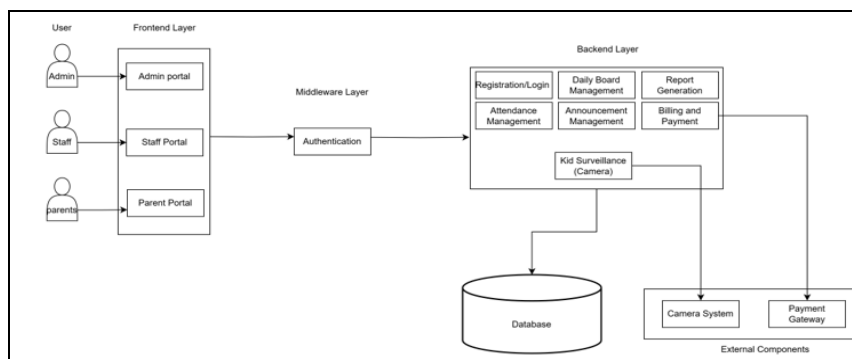


Fig. 3 System architecture of the proposed system

The following is the database schema designed and extracted from the entity relationship diagram. It will be created using MySQL database.

- i. Users (user_id, username, password, email, role)
- ii. Parent_records (parent_recordsID, enrollment_id, father_id, mother_id, guardian_id, child_id)
- iii. Mother (mother_id, user_id, mother_name, mother_ic, mother_email, mother_phoneNo, mother_staff_number, mother_occupation, mother_office_number, mother_ptj, mother_address, mother_nationality, mother_race, mother_religion, mother_monthly_income)
- iv. Fathers (father_id, user_id, father_name, father_ic, father_email, father_phoneNo, father_staff_number, father_occupation, father_office_number, father_ptj, father_address, father_nationality, father_race, father_religion, father_monthly_income)
- v. Guardians (guardian_id, user_id, guardian_name, guardian_ic, guardian_email, guardian_phoneNo, guardian_staff_number, guardian_occupation, guardian_office_number, guardian_ptj, guardian_address, guardian_nationality, guardian_race, guardian_religion, guardian_monthly_income, guardian_relation)
- vi. Childrens (children_id, enrolment_id, child_name, child_birthdate, child_gender, child_age, child_address, child_sibling_number, child_numberInSibling, child_allergies, child_previous_childcare, child_birth_cert, child_immunization_record, child_photo)
- vii. Enrollments (enrollment_id, status, registration_type)
- viii. Staff (staff_id, user_id, staff_name, staff_ic, staff_email, staff_phoneno, staff_address)
- ix. Staff_assignments (staff assignmentsID, children_id, primary staff id, status)
- x. Attendances (attendance_id, children_id, attendance_date, status, time_in, time_out, attendance_overtime)
- xi. Daily_activities (daily activitiesID, user_id, post_title, post_date, post_time, post_desc, acitivity_photo)
- xii. Camera_footages (camera footageID, user_id, start time, end_time, date, file_location)
- xiii. Announcements (announcement_id, announcement_title, announcement_date, announcement_location, announcement_time, activity_description, announcement_type)
- xiv. Payments (payment_id, user_id, parent_recordsID, children_id, paymentByParents_date, payment_duedate, payment_amount, payment_status, bill_date)

User interfaces have been designed as a guide for developing the proposed system. Fig. 4 and Fig. 5 show the basic interface in the proposed system. The UI is designed using Canva. Fig. 4(a) shows the user login interface where user input their credentials to get into the system. If the login is successful, the user gets access to their respective page. Otherwise, the system displays an error message. Fig. 4(b) shows the children's registration form. Parents just need to fill in the children and their details. Once submitted and the registration request is accepted and parents will get login credentials from the admin to access the childcare system.

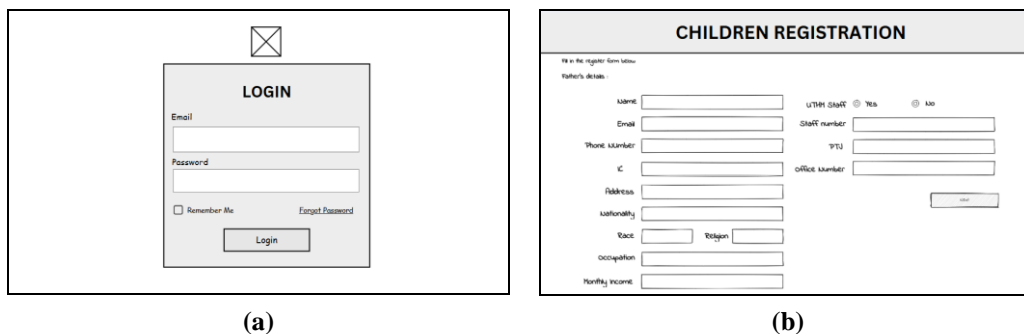
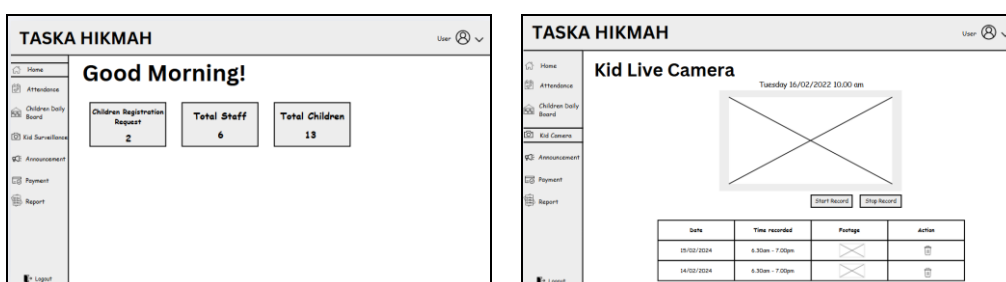


Fig. 4 User interface (a) Login interface; (b) Children registration

Fig. 5(a) shows the admin dashboard from the admin side. Shows the summary of total pending registration requests, total staff, and total children, including the seven modules on the sidebar that the admin can manage. Fig. 5(b) shows the interface of the kid camera where the admin can manage the live record footage.



(a)

(b)

Fig. 5 User interface (a) Admin dashboard; (b) Admin side kid camera

4. Result and Discussion

The purpose of this section is to discuss the system implementation process and system testing for the proposed system.

4.1 Result

4.1.1 Implementation of Module

For the registration and login module, Fig. 6 illustrates the login form interface, which features a secure authentication form containing fields for email and password. Additionally, there is a "Remember Me" option that allows users to save their credentials for quicker access during future logins. The login module also includes input validation to prevent unauthorized access or mismatched entries.

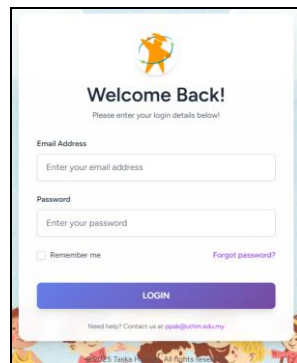


Fig. 6 User interface login form

Fig. 7(a) shows the parent-children registration form. This form is a must in order to register their children, and parents could choose either as parents or as guardians. Fig. 7(b) demonstrates the parent registration request interface that is received by the admin. The interface displays the pending and approved status of parent-child relationships. This process ensures a smooth and hassle-free experience for the admin, who can manage multiple registrations efficiently. Hence, the source code of parent-child registration is shown in Appendix C to support the system implementation details on the registration flow.

(a)

#	PARENT/GUARDIAN	CHILD	TYPE	STATUS	ACTIONS
1	Ahmad bin Ali & Syafiqah binti Hairi	Ashraf bin Ahmad	Parents	Approved	
2	Nur Asyila Binti Rahman	Nur Aina Binti Zulkifli	Guardian	Approved	
3	Intan Nur Fatma Binti Saahari	Azizah binti Sarib	Guardian	Approved	

(b)

Fig. 7 User interface (a) Parent-children registration form; (b) Parent-children registration request

Fig. 8 shows a code segment to create children's attendance for the time-in section. This code receives and validates input for status, time_in, time_out, attendance_date, and children_id for multiple children. These data will then be inserted into the respective column in the attendance table.

```

foreach ($validated['status'] as $childId => $status) {
    $attendance = Attendance::updateOrCreate(
        [
            'children_id' => $childId,
            'attendance_date' => $attendanceDate,
        ],
        [
            'attendance_status' => $status,
            'time_in' => $status === 'attend' ? ($validated['time_in'][$childId] ?? now()->format('H:i:s')) : null,
            'time_out' => null, // Always set time_out to null
        ]
    );
}

```

Fig. 8 Code Segment for children's attendance

Fig. 9 shows a code segment to stream the camera in real time. This code uses JavaScript's MediaRecorder API for streaming. This code will check if the device supports getUserMedia in order to access the camera. If it turns true, it starts the camera stream and shows the live video feed in the video element if it is supported. Else if camera access is not accessible or supported, it will display the proper error messages.

```

// Check for webcam support and initialize
if (navigator.mediaDevices && navigator.mediaDevices.getUserMedia) {
    navigator.mediaDevices.getUserMedia({ video: true, audio: true })
        .then(function (stream) {
            video.srcObject = stream;
            mediaRecorder = new MediaRecorder(stream);

            // Collect recorded chunks
            mediaRecorder.ondataavailable = function (event) {
                if (event.data.size > 0) {
                    recordedChunks.push(event.data);
                }
            };
        });
}

```

Fig. 9 Code Segment to stream live camera

Fig. 10(a) shows the side admin for the attendance module. The interface includes summary cards showing the total number of children, total attendance, and total absences for the day. This module will list the detailed attendance of registered children. The date filter, attendance status, time in, and time out demonstrated an efficient operation while giving a colourful visual interface for better readability. While Fig. 10(b) shows the kid camera module where the interface is the live camera shows real-time streaming capability video of the environment. The footage management, like "start record" and "stop record", only enables the admin to handle it for safety purposes.

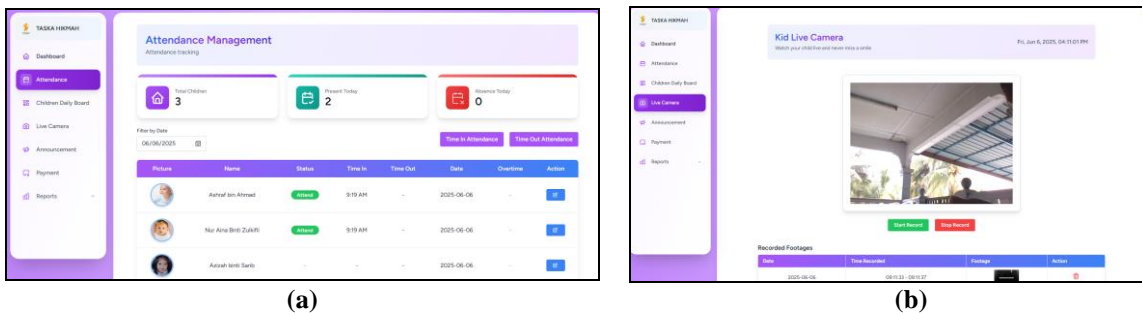


Fig. 10 User interface (a) Attendance module; (b) Kid camera module

Fig. 11 shows a code segment to create children payment fees. This will first validate the data obtained from the form input against its requirement into \$validatedData. Then, parents payment records will be created by assigning \$validatedData into the Payment table using Payment::create[].

```

// Create the payment with initial pending status
$payment = Payment::create([
    'user_id' => null,
    'child_id' => $validatedData['child_id'],
    'parent_id' => $validatedData['parent_id'],
    'payment_amount' => $validatedData['payment_amount'],
    'payment_due_date' => $validatedData['due_date'],
    'payment_status' => 'pending',
    'bill_date' => Carbon::now()->toDateString(),
]);

```

Fig. 11 Code Segment to create childcare payment

Fig. 12 shows a code segment to generate an attendance report. This code uses deep database gathering to look at extra hours over days, weeks, and months using raw SQL with keys like SUM(), ABS(), WEEK(), and DATE_FORMAT(). This data uses Laravel's collection mapping to make arrays for making chart visualization.

```
// Child Attendance Trends by Daily, Weekly, Monthly
$childAttendanceRaw = Attendance::selectRaw("
    children_id,
    attendance_status,
    attendance_date,
    DATE_FORMAT(attendance_date, 'Xd-Mn-Y') as daily,
    DATE_FORMAT(attendance_date, 'YV-Ku') as weekly,
    DATE_FORMAT(attendance_date, 'Mn-Y') as monthly
")
->where('attendance_status', 'attend')
->with('child')
->get();

$groupedTrends = ['daily' => [], 'weekly' => [], 'monthly' => []];

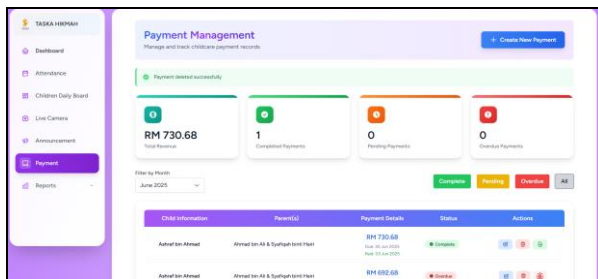
foreach (['daily', 'weekly', 'monthly'] as $period) {
    $step = [];
    foreach ($childAttendanceRaw->groupBy($period) as $label => $entries) {
        $childrenGroup = [];
        foreach ($entries->groupBy('child_name') as $childName => $records) {
            $childrenGroup[$childName] = $records->count();
        }
        $step[] = [
            'label' => $label,
            'children' => $childrenGroup
        ];
    }
    $groupedTrends[$period] = $step;
}

// Get all children name
$children = \App\Models\Child::pluck('child_name')->toArray();

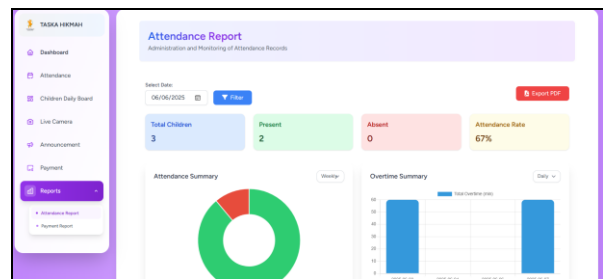
// Prepare periods (daily, weekly, monthly)
$periods = [
    'daily' => Attendance::selectRaw("DATE_FORMAT(attendance_date, 'YV-Ku-Kd') as period")->distinct()->pluck('period')->toArray(),
    'weekly' => Attendance::selectRaw("DATE_FORMAT(attendance_date, 'YV-Ku') as period")->distinct()->pluck('period')->toArray(),
    'monthly' => Attendance::selectRaw("DATE_FORMAT(attendance_date, 'Mn-Y') as period")->distinct()->pluck('period')->toArray(),
];
```

Fig. 12 Code Segment to generate attendance report

Fig. 13(a) shows the administrative payment module, which provides the interface for financial oversight and payment management, such as creating payment fees, status tracking (yellow for pending, green for complete, and red for overdue), and button actions. If the payment is overdue, the overdue button automatically displays, and the admin is able to send an email directly to parents. Fig. 13(b) demonstrates the attendance report generation interface. This page contains detailed attendance information for the selected day, a summary card, a bar chart, and a doughnut chart. Also includes a PDF button to export the report for administrator use.



(a)



(b)

Fig. 13 User interface (a) Admin side payment module; (b) Attendance report generation

Fig. 14 shows the admin dashboard interface, which consists summary card on total children registration requests, total staff, and total registered children in the childcare. This page also includes the login history for user history login tracking.

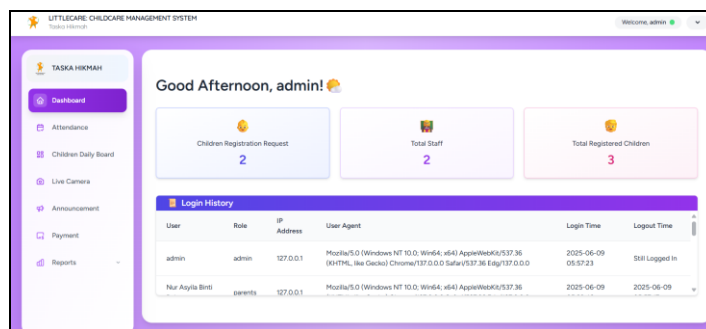


Fig. 14 Admin dashboard interface

Fig. 15 shows the staff dashboard interface, which consists of summary card on total children in the childcare and the total children that assigned to the staff. This page also includes the details of the staff assigned children.

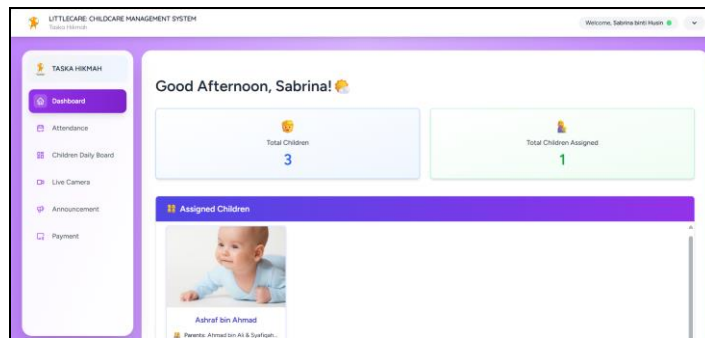


Fig. 15 Staff dashboard Interface

Fig. 16 shows the parents dashboard interface, which it consists of summary card on total children of the parents, and this page also includes the details of their registered children and an overview of every module for fast updates.

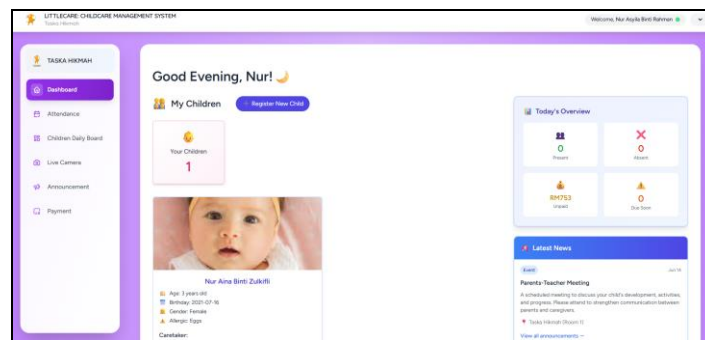


Fig. 16 Parents dashboard Interface

4.2 Testing

The testing case will be carried out to assess the functionality of each module. A User Acceptance Test (UAT) method is utilized to perform testing. The test case will be divided into 3 users, which are the administrator, the staff, and the parents.

4.2.1 Test Case for Administrator

Table 5 shows the list of test cases for the admin in testing the proposed system. All modules are passing as expected results.

Table 5 Test Case of the Proposed System for Administrator

Test Case ID	Description	Expected Result	Actual	Pass/Fail
M1-1	Register the admin account	Admin can register an account	Admin successfully created for an account	Pass
M1-2	Login as admin	Admin can login	Admin successfully logged in	Pass
M1-4	Approve parent registrations	Admin can approve registrations	Admin successfully approved registrations	Pass

Table 5: (cont)

Test Case ID	Description	Expected Result	Actual	Pass/Fail
M2-1	Take the children's attendance	Admin can mark attendance	Admin successfully marked attendance	Pass
M2-3	Update attendance records	Admin can update records	Admin successfully updated records	Pass
M3-1	Create daily activity post	Admin can create post	Admin successfully created post	Pass
M3-3	Update daily activity post	Admin can update post	Admin successfully updated post	Pass
M4-1	Record camera footage	Admin can record footage	Admin successfully recorded footage	Pass
M4-2	Save footage to computer	Admin can save footage	Admin successfully saved footage	Pass
M5-1	Create announcement	Admin can create announcement	Admin successfully created announcement	Pass
M5-2	Edit announcement	Admin can edit announcement	Admin successfully edited announcement	Pass
M6-1	Create payment record	Admin can create payment record	Admin successfully created payment record	Pass
M6-7	Send payment reminders	Admin can send reminders	Admin successfully sent reminders	Pass
M7-1	Generate attendance report	Admin can generate report	Admin successfully generated report	Pass
M7-2	Generate payment report	Admin can generate report	Admin successfully generated report	Pass

4.2.2 Test Case for Staff

Table 6 shows the list of test cases for staff in testing the proposed system. All modules are passing as expected results.

Table 6 Test Case of the Proposed System for Staff

Test Case ID	Description	Expected Result	Actual	Pass/Fail
M1-8	Login with credentials	Staff can login	Staff successfully logged in	Pass
M2-2	Take children's attendance	Staff can mark attendance	Staff successfully marked attendance	Pass
M2-4	Update attendance records	Staff can update records	Staff successfully updated records	Pass
M3-2	Create daily activity post	Staff can create post	Staff successfully created post	Pass
M3-4	Update daily activity post	Staff can update post	Staff successfully updated post	Pass
M4-4	View live camera feed	Staff can view live feed	Staff successfully viewed live feed	Pass
M5-4	Create announcement	Staff can create announcement	Staff successfully created announcement	Pass
M6-3	View parents' payment details	Staff can view payment details	Staff successfully viewed payment details	Pass

4.2.3 Test Case for Parents

Table 7 shows the list of test cases for parents in testing the proposed system. All modules are passing as expected results.

Table 7 Test Case of the Proposed System for Parents

Test Case ID	Description	Expected Result	Actual	Pass/Fail
M1-7	Submit registration information	Parents can register children	Parents successfully submitted registration information	Pass
M1-9	Login with credentials	Parents can login	Parents successfully logged in	Pass
M2-5	View child's attendance list	Parents can view attendance	Parents successfully viewed attendance	Pass
M2-6	Receive child arrival notification	Parents receive arrival email	Parents successfully received child arrival email	Pass
M3-7	View child's daily activities	Parents can view activities	Parents successfully viewed activities	Pass
M4-5	View live camera feed	Parents can view live feed	Parents successfully viewed live feed	Pass
M5-7	View announcements	Parents can view announcements	Parents successfully viewed announcements	Pass
M6-4	View payment records	Parents can view payment records	Parents successfully viewed payment records	Pass
M6-5	Make online payment	Parents can make payment	Parents successfully made payment	Pass
M6-6	Receive and download payment receipt	Parents receive receipt	Parents successfully received and able to download payment receipt	Pass

4.2.4 User Acceptance Testing

User Acceptance Testing (UAT) has been conducted using a questionnaire survey from a Google Form. The proof that the proposed system has been tested is shown in Appendix D. There is a total of 31 respondents from the target users that related to childcare. Based on the results, the proposed system has received positive feedback. The detailed UAT results in Appendix D show that childcare staff and parents primarily aged 21-40 showed that 83.9% of them are often familiar with using digital systems, and 80.6% of them have no experience in using any childcare management system. Therefore, the testing of the proposed system illustrated a consistently high satisfaction among users, with over 90% of users rating 4-5 stars for the system interface design, registration process, real-time monitoring capabilities, attendance process, payment streamlined process, daily board, and announcement feature. 100% of respondents would suggest this proposed system compared to a manual system. This UAT confirms that the proposed system is very helpful, especially in solving the inefficient operations.

5. Conclusion

In conclusion, this project has achieved its objectives and successfully developed the system within the time. The LITTLECARE Childcare Management System has been developed for Taska Hikmah, allowing for improved management of childcare operations, including the administration of children, parents, and staff. Moreover, parents can ease their minds and be less worried about their children as the live camera helps parents to watch their children in live footage. While integrating the modules into one system can reduce the hustle and enhance the communication and operation between parents and staff. Overall, LITTLECARE addresses the childcare challenges in childcare management. Lastly, future developments could enhance the limitations of this project by advancing the features of surveillance, such as adding AI-driven insight for better safety measures in the future. Thus, by prioritizing security and efficiency, the system would grow bigger in the future for a better experience.

Acknowledgement

The author would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, for its support and a big thanks to my supervisor, lecturer, panel, family and friends for the support throughout the journey of this project.

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:

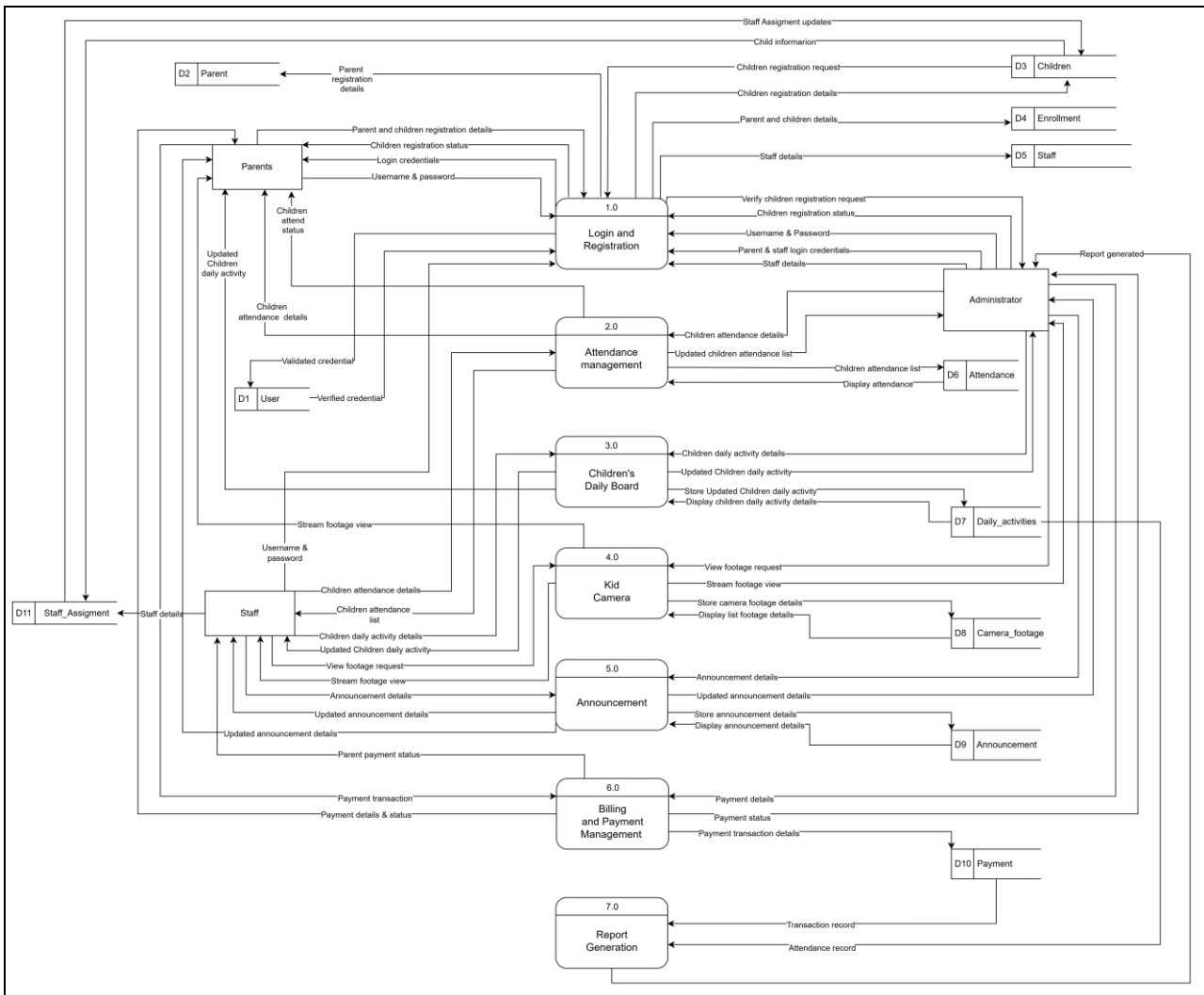
The authors confirm contribution to the paper as follows: **study conception and design:** Intan Nur Fatiha, Nur Ariffin Mohd Zin; **data collection:** Intan Nur Fatiha, Nur Ariffin Mohd Zin; **analysis and interpretation of results:** Intan Nur Fatiha, Nur Ariffin Mohd Zin.

References

- [1] Indarwati, F., and Firmansyah, G., "Promoting child's growth and development in childcare centers," *Bali Medical Journal*, vol. 10, no. 3, 2022. doi: 10.15562/bmj.v10i3.2881.
- [2] Kong, K., *Early childhood education in Malaysia*, in *International Handbook on Education in South East Asia*, Springer Nature Singapore, 2023, pp. 1-32.
- [3] Kosmo, "RM30.07 juta untuk kemudahan Tabika Taska Perpaduan," *Kosmo*, Mar. 27, 2024. [Online]. Available: <https://www.kosmo.com.my/2024/03/27/rm30-07-juta-untuk-kemudahan-tabika-taska-perpaduan/>.
- [4] Rainer, R. K., Prince, B., Watson, H. J., Chircu, A. M., and Marabelli, M., *Management Information Systems: Moving Business Forward*. John Wiley & Sons, 2019.
- [5] Bhardwaj, S., "A brief study on web technology," *International Journal of Innovative Research in Computer Science & Technology*, 2021. doi: 10.55524/ijircst.2021.9.6.61.
- [6] Nyabuto, G. M., Mony, V., and Mbugua, S., "Architectural review of client-server models," *International Journal of Scientific Research & Engineering Trends*, vol. 10, no. 1, pp. 139-143, 2023.
- [7] Wibowo, F. W., and Ardiansyah, M. A., "Low cost real time monitoring system and storing image data using motion detection," *International Journal of Applied Engineering Research*, vol. 11, no. 8, pp. 5419-5424, 2016.

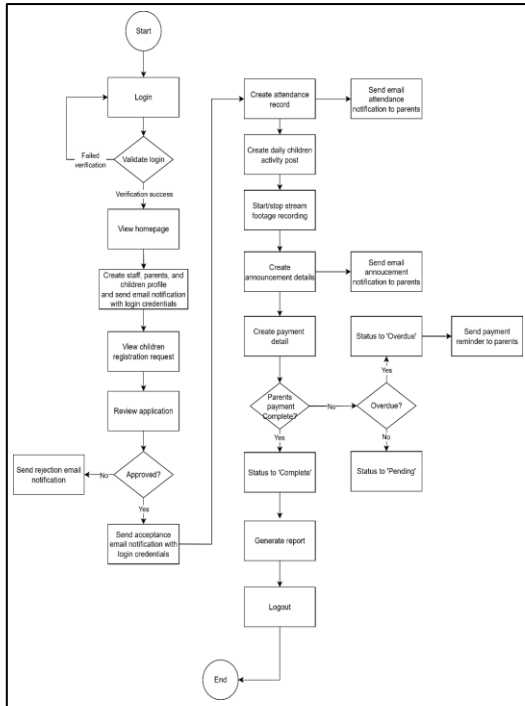
- [8] Talukder, A. K., "The next generation web: Technologies and services," in *Big Data Analytics: 8th International Conference, BDA 2020*, Springer International Publishing, 2020, pp. 209–229.
- [9] Setyawan, K. R. V., Rizal, M. F., Widodo, S., and Hikmawan, R., "Design of Continuous Web APP: Guidance and Counseling Management Information System at SMKN 1 Purwakarta using Laravel Framework," *International Journal of Software Engineering and Computer Science (IJSECS)*, vol. 3, no. 3, pp. 410–423, 2023.
- [10] Tang, E., "Investigating the effect of the entity relationship view on model-view-controller (MVC) web development," Ph.D. dissertation, 2020. [Online]. Available: <https://www.proquest.com/dissertations-theses/investigating-effect-entity-relationship-view-on/docview/2466049132/se-2>.
- [11] Wahyudi, J., Asbari, M., Sasono, I., Pramono, T., and Novitasari, D., "Database management in MySQL," 2022.
- [12] Jose, D. B. K., *Travel Management System*, Ph.D. dissertation, Davangere University, 2024.
- [13] Sithambranathan, H., and Nawari, R. M., "Web-based development of Taska Khairiyah Wafiq Management System," *Applied Information Technology and Computer Science*, vol. 4, no. 2, pp. 1688–1707, 2023.
- [14] Salleh, N. A. M., and Hassan, N., "Permataku Monitoring System and Student Development of Taska Qaseh Umami: Sistem Pemantauan Permataku dan Perkembangan Pelajar Taska Qaseh Umami," *Applied Information Technology and Computer Science*, vol. 3, no. 1, pp. 586–598, 2022.
- [15] myKidzDay, *Best Center Management Software, Free Childcare App, Daily Sheets and Attendance App to Enhance Parent Communication, Build Enrollment & Go Paperless!* Accessed Nov. 12, 2024. [Online]. Available: <https://www.mykidzday.com/>.
- [16] Saeed, S., Jhanjhi, N. Z., Naqvi, M., and Humayun, M., "Analysis of software development methodologies," *International Journal of Computing and Digital Systems*, vol. 8, no. 5, pp. 446–460, 2019.
- [17] Eckhardt, J., Vogelsang, A., and Fernández, D. M., "Are 'non-functional' requirements really non-functional? An investigation of non-functional requirements in practice," in *Proceedings of the 38th International Conference on Software Engineering*, 2016, pp. 832–842.
- [18] Weillkiens, T., Lamm, J. G., Roth, S., and Walker, M., *Model-based System Architecture*. John Wiley & Sons, 2022.

Appendix A: System's DFD Level 1

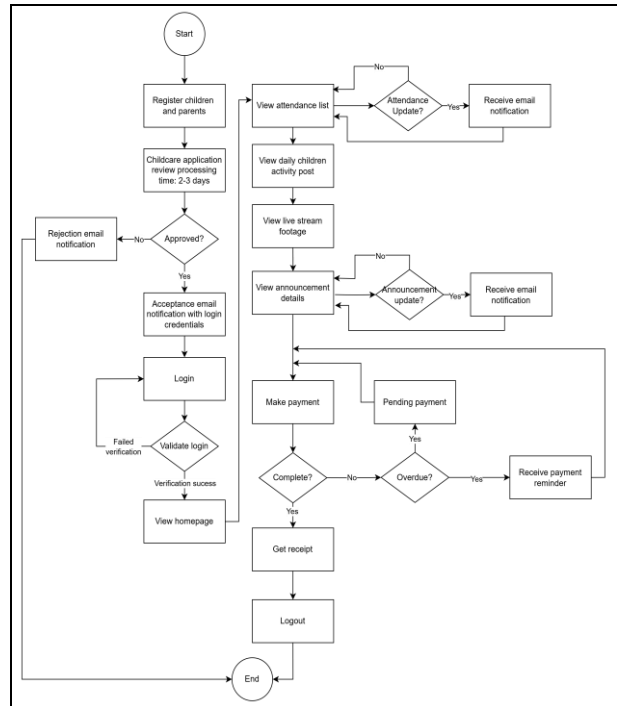


DFD Level 1 of Proposed System

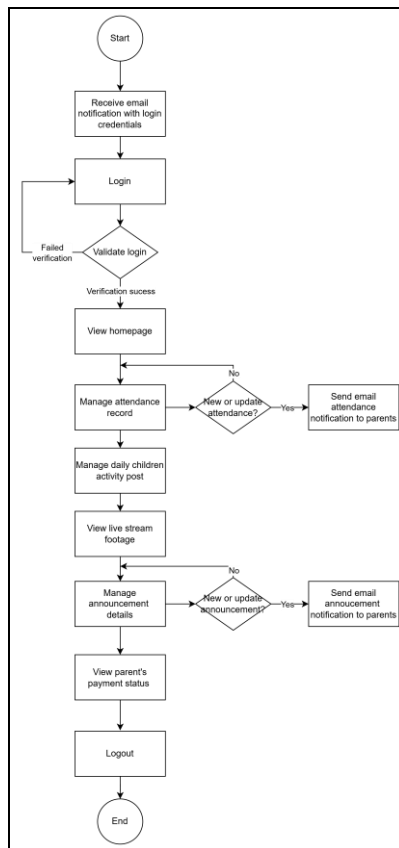
Appendix B: System's Flowchart



System's flowchart on the admin side



System's flowchart on the parents' side



System's flowchart on the staff side

Appendix C: Parent-Children Registration Source Code

```
// Father & Mother
if ($registration_type === 'parents') {
    $father = Father::where('father_ic', $request->father_ic)->first();
    if (!$father) {
        $father = Father::create([
            'father_name' => $request->father_name,
            'father_email' => $request->father_email,
            'father_phoneno' => $request->father_phoneno,
            'father_ic' => $request->father_ic,
            'father_address' => $request->father_address,
            'father_nationality' => $request->father_nationality,
            'father_race' => $request->father_race,
            'father_religion' => $request->father_religion,
            'father_occupation' => $request->father_occupation,
            'father_monthly_income' => $request->father_income,
            'father_staff_number' => $request->father_staff_number,
            'father_psj' => $request->father_psj,
            'father_office_number' => $request->father_office_number,
            'user_id' => null,
        ]);
    }

    $mother = Mother::where('mother_ic', $request->mother_ic)->first();
    if (!$mother) {
        $mother = Mother::create([
            'mother_name' => $request->mother_name,
            'mother_email' => $request->mother_email,
            'mother_phoneno' => $request->mother_phoneno,
            'mother_ic' => $request->mother_ic,
            'mother_address' => $request->mother_address,
            'mother_nationality' => $request->mother_nationality,
            'mother_race' => $request->mother_race,
            'mother_religion' => $request->mother_religion,
            'mother_occupation' => $request->mother_occupation,
            'mother_monthly_income' => $request->mother_income,
            'mother_staff_number' => $request->mother_staff_number,
            'mother_psj' => $request->mother_psj,
            'mother_office_number' => $request->mother_office_number,
            'user_id' => null,
        ]);
    }
}

// Guardian
if ($registration_type === 'guardian') {
    $guardian = Guardian::where('guardian_ic', $request->guardian_ic)->first();
    if (!$guardian) {
        $guardian = Guardian::create([
            'guardian_name' => $request->guardian_name,
        ]);
    }
}
```

Parents Registration Source Code

```
$child = Child::create([
    'enrollment_id' => $registration->id,
    'child_name' => $request->child_name,
    'child_birthdate' => $request->child_birth_date,
    'child_gender' => $request->child_gender,
    'child_age' => $request->child_age,
    'child_address' => $request->child_address,
    'child_sibling_number' => $request->child_siblings_count,
    'child_numberInSibiling' => $request->child_position,
    'child_allergies' => $request->child_allergies,
    'child_medical_conditions' => $request->child_medical_conditions,
    'child_previous_childcare' => $request->child_previous_childcare,
    'child_birth_cert' => $birthCertPath,
    'child_immunization_record' => $immunizationPath,
    'child_photo' => $childPhotoPath
]);
```

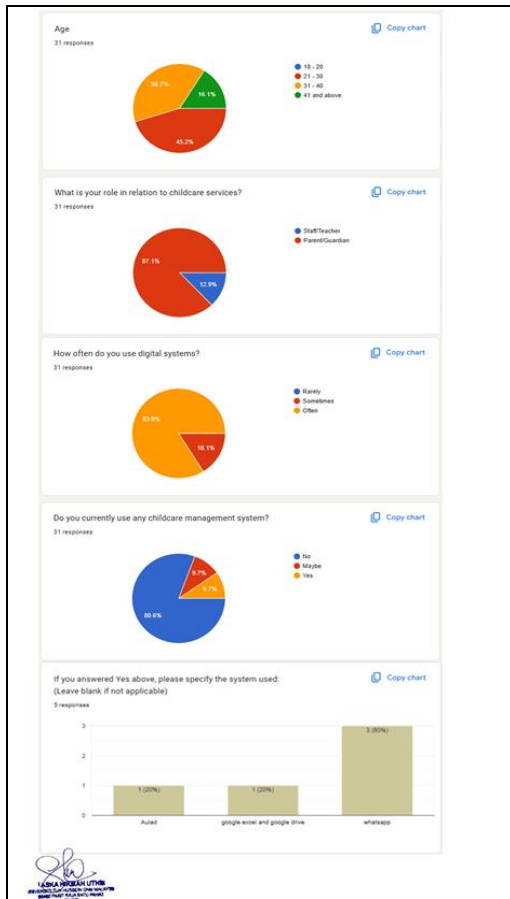
Children Registration Source Code

```
// Create user accounts and update parent tables with user_id
if ($validated['registration_type'] === 'parents') {
    if (empty($validated['father_email'])) {
        $fatherUser = User::where('email', $validated['father_email'])->first();
        if (!$fatherUser) {
            $fatherUser = User::create([
                'name' => $validated['father_name'],
                'email' => $validated['father_email'],
                'password' => Hash::make($validated['father_ic']),
                'role' => $validated['role'],
            ]);
        } else {
            $fatherUser->password = Hash::make($validated['father_ic']);
            $fatherUser->save();
        }
        // Update father table
        if ($father) {
            $father->user_id = $fatherUser->id;
            $father->save();
        }
        $emails[$validated['father_email']] = $validated['father_name'];
    }

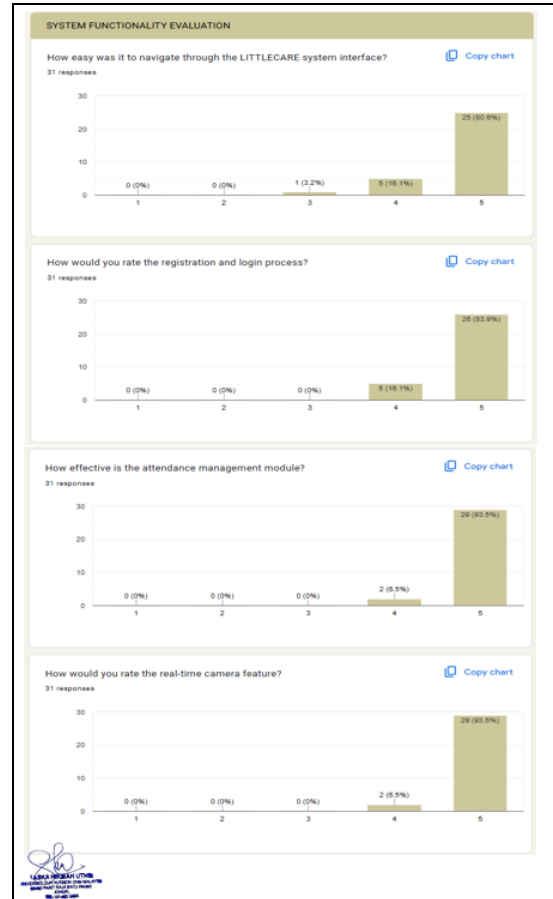
    if (empty($validated['mother_email'])) {
        $motherUser = User::firstOrCreate(
            ['email' => $validated['mother_email']],
            [
                'name' => $validated['mother_name'],
                'password' => Hash::make($validated['mother_ic']),
                'role' => $validated['role'],
            ]
        );
        if ($mother) {
            $mother->user_id = $motherUser->id;
            $mother->save();
        }
        $emails[$validated['mother_email']] = $validated['mother_name'];
    }
} elseif ($validated['registration_type'] === 'guardian') {
```

Create Account for Parents Source Code

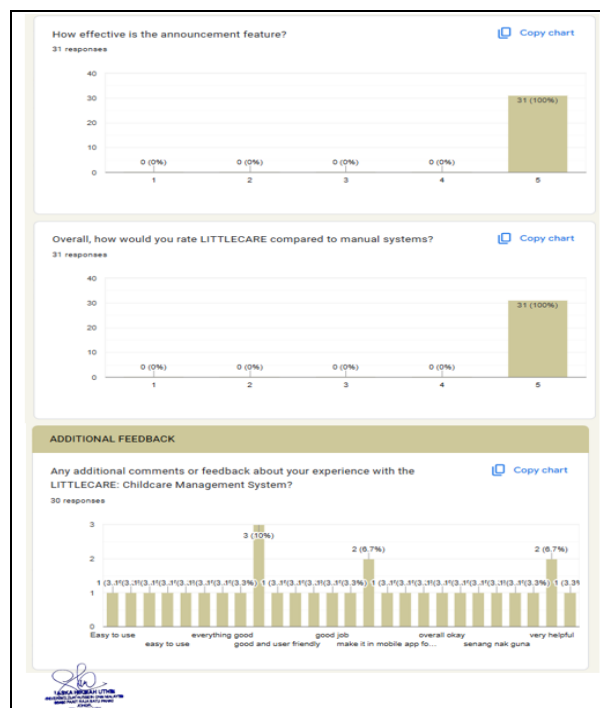
Appendix D: User Acceptance Testing (UAT) Results



Demographic profile



Module functionality rating



User evaluation feedback of proposed system performance