KINDERGARTEN MANAGEMENT SYSTEM WITH PARENTAL CONTROL

BY

LIM JIA YI

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS) INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2025

COPYRIGHT STATEMENT

This Final Year Project proposal is submitted in partial fulfillment of the requirements for the degree of Bachelor of Information Systems (Honours) Information Systems Engineering at Universiti Tunku Abdul Rahman (UTAR). This Final Year Project proposal represents the work of the author, except where due acknowledgment has been made in the text. No part of this Final Year Project proposal may be reproduced, stored, or transmitted in any form or by any means, whether electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author or UTAR, in accordance with UTAR's Intellectual Property Policy.

ACKNOWLEDGEMENTS

I would like to sincerely thank and appreciate my supervisor, Mr Cheang Kah Wai for his guidance, motivation and continuous support throughout the development of this project. His feedback helped guide me in the right direction and improved the quality of my work.

I would also like to thank my friends for their support, advice and sharing ideas with me when I felt lost or stressed. They always motivated me to overcome challenges during the tough times. Finally, I must thank to my parents and my family for their love, support, and constant encouragement during my studies.

ABSTRACT

This project develops a kindergarten management system with parental control to enhance communications among parents, teachers, and administrators and to reduce administrative tasks. The proposed system will develop a mobile application for parents and a web application for administrators and teachers. With the help of this mobile and web applications, parents and teachers are provided with real-time communication, online payment security of tuition fees, and the ability for event reminders that enable them to keep their eagerness in the child's education. It allows administrators to update information about students, payments for courses, and scheduling of events. This project was developed using the Agile methodology. Technologies used include Flutter for mobile application and React for web application, Firebase for real-time data management and authentication, and Stripe for secure in-app payment processing. It reduces work involving administration, like communication, fee collection, and event scheduling, at the same time enhancing parental involvement in early childhood education. The solution was user-centric and can be used to increase kindergarten management operational efficiency while improving the experiences of teaching children and their families.

Area of Study: Educational Technology, Information Systems

Keywords: Kindergarten Management System, parental control, real-time communication, administrative automation, Flutter, React, Firebase, Stripe

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1
Introduction

1.1 Problem Statement and Motivation

Current kindergarten management systems mainly focus on administrative tasks such as attendance tracking and record-keeping but lack essential features that involve parents, such as real-time parent-teacher communication, in-app payment, and notifications for important updates. This causes communication delays, ineffective collection fees, and reduced parental engagement in children's early education. Without a real-time communication channel, important updates about a child's progress are often missed because traditional methods like letters and meetings have become impractical in today's fast-paced world due to parents are busy with work and other responsibilities [1][2]. Administrative staff also face extra burdens as they need to spend time reminding parents about payments and upcoming events, which are inefficient [3].

The motivation for this project was to develop a more advanced kindergarten management system that uses modern technology for seamless communication, automated administrative tasks, and flexible payment solutions. This system will help parents stay better connected and engaged in their child's development. It will also allow teachers and administrators to spend less time on repetitive tasks and more time focusing on education. This project aims to fix the problems in current systems and make a better and easier environment for everyone.

1.2 Project Scope

At the end of the project, the delivery will be a kindergarten management system that consists of a mobile application for parents and a web application for teachers and admins. The mobile app will allow parents to communicate with teachers in real-time, make tuition fee payments, receive reminders, track attendance, view announcements, and stay updated on their child's academic progress and upcoming events. This will help parents stay involved in their child's education and easily manage payments from their phones.

The web application will be designed for teachers and administrators to manage tasks like recording attendance, updating payment records, posting announcements, scheduling events, and monitoring student progress. This can save time, reduce manual work and improve

communication between parents, teachers and administrators which is more efficient and organized.

1.3 Project Objectives

The primary objective of the kindergarten management system is to provide an integrated platform that enhances the involvement of parents in the academic life of the students. This can be achieved by facilitating effective communication between the parents and the teachers, and by offering them secure means of payment for the services. To achieve this main objective, the system will **provide a communication channel** that enables instant updates on student progress that allows parents to raise concerns or ask questions directly. This will be done through the integration of real-time messaging and notification systems in the mobile application. Parents will receive immediate updates and alerts about their children's activities.

Next, the system will provide an **in-app payment with alternative payment method**, enabling parents to pay for tuition fees and other expenses directly through the app. This integrated payment system will support various methods such as credit or debit cards, bank transfers, and e-wallet. This will reduce the possibility of missed payments and offer a convenient, secure way for parents to manage their financial obligations.

It will also integrate a **automatic notifications** to send reminder to parents regarding upcoming events, such as parent-teacher meetings, field trips, and school holidays. This can ensure that parents will be informed in advance, thus enabling them to make prior preparations. This is important to maintain high levels of participation and involvement by parents in their child's education. By automating this process, missed communication can be reduced and parents can actively involve in school events.

1.4 Contributions

This project will make contribution to the field of early childhood education by introducing an innovative kindergarten management system designed to enhance parental involvement and streamline administrative processes. The system's ability to facilitate real-time messaging between parents and teachers ensures that critical updates about a child's progress are communicated instantly. This feature is vital in early childhood education, where timely information can significantly impact a child's developmental trajectory. Instant notifications keep parents continuously informed, fostering a more engaged and supportive learning environment. The system also features an integrated online payment functionality that allows parents to complete transactions directly within the app. By eliminating the need to sign in to a third-party app, this feature streamlines the payment process and enhances convenience for parents. Supporting multiple payment options within the app minimizes the risk of missed payments and improves financial management efficiency, making it faster and more convenient for parents to handle their payments.

The implementation of an automated notification system enhances parental engagement by ensuring that parents receive timely notifications about upcoming events such as parent-teacher meetings and school activities. This ensures parents are well-informed and prepared, improving participation rates and reducing missed communications. Additionally, the project contributes to administrative efficiency by integrating automated payment processing, event scheduling, and notifications within a single platform reduces time-consuming manual tasks. This allows educators to focus more on personalized teaching and student engagement, thereby improving overall educational quality

1.5 Project Background

The rapid development of technology has impacted various sectors, including education. Innovative technologies such as smartphones and the Internet provide schools with new tools to reach parents and keep them informed about their children's progress [4]. Modern young teachers and parents prefer to use internet applications as they provide efficient and convenient communication [5].

Effective communication between parents and teachers is important for children's learning achievements. Research shows that children whose parents are actively involved in their education tend to do better academically and have better social skills [6]. However, traditional methods such as paper notes, parent teacher meetings, and phone calls, are always constrained by time, availability, and accessibility. These methods cause delays in information sharing

which are very critical in early childhood education, as the parents need to receive continuous information regarding the progress and activities of the children [5].

This integration of modern technology into early childhood education facilitates real-time updates and increases engagement between parents, teachers, and students. For instance, schools that implemented digital communication platforms recorded high levels of parental involvement and satisfaction [6]. This is due to technology allowing teachers to spend more time on teaching and hence improving the quality of the education provided. Thus, kindergartens are increasingly adopting advanced management systems to meet parent expectations, reduce administrative workload, and enhance parent engagement [7].

An ideal kindergarten management system should include features such as in-app payment, attendance tracking, announcements and notification to remind parents about some actions or events, student data storage and access control. It generally consists of an extensive database storing and managing students' information, like personal data, attendance, and academic records. A type of central storage like this allows teachers and staff to draw upon and update student information to maintain accuracy and coherence in the records.

The market report notes that kindergartens adopting management software can automate fee management, which is a major factor in system adoption [8]. Automation of attendance tracking allows teachers to record and keep track of the attendance among students. Besides, billing and payment management is another key feature, which assists in making tuition fee collection organized and increases financial transaction convenience for parents and school administrators. This helps reduce late payments, improve cash flow, avoids lost or incorrect receipts and makes financial records more transparent to administrators and parents.

Furthermore, these systems often provide reporting capabilities, which are student performance reports, attendance, financial records, and many others to make accurate decisions with more transparency in mind. This proposal presents the development of a kindergarten management system with parental control and is aimed at enhancing communication, promoting involvement by parents, and facilitating many administrative and educational processes in early childhood educational establishments. In addition, modern technology relieves teachers from administrative tasks to focus on personalized teaching, which enhances the quality of education given. Implementing such a system aligns with the broader goal of fostering a collaborative and innovative educational environment that supports the holistic development of young children.

1.6 Report Organization

This report is organized into 7 chapters: Chapter 1 Introduction, Chapter 2: Literature Review, Chapter 3: Methodology/ Approach, Chapter 4: System Design, Chapter 5: System Implementation, Chapter 6: System Evaluation and Discussion and Chapter 7: Conclusion and Recommendation. The first chapter is about the introduction of this project which includes the problem statement and motivation, project scope, project objectives, contribution and background information. Chapter 2 is about the literature review carried out on several existing kindergarten management systems to evaluate the features of each system. The third chapter discusses the methodology used to develop the system, including user requirements, use case diagrams, and the approach taken for system development. Furthermore, Chapter 4 provides detailed system design diagrams, illustrating how the system is structured and its components interact. Chapter 5 explains the process of implementing the system and how it operates. In addition, Chapter 6 shows the testing result of the system and lastly Chapter 7 is the conclusion of whole project and future recommendations.

Chapter 2
Literature Review

2.1 Why Kindergarten Management Systems Are Popular

Educational technology (Ed tech) refers to the use of digital tools, software, hardware, and infrastructure in learning, teaching, and administration in the educational environments [9]. The primary goal of ed tech is to enhance the learning experience and make education more interactive, accessible, and personalized to meet the needs of students. By leveraging data analytics, real-time feedback, and personalized learning paths, educational technology aims to improve learning outcomes and streamline school operations.

Kindergarten management systems are type of application in Ed tech, designed to simplify the administrative and operational aspects of early childhood education. These systems include features such as fee collection, attendance tracking, and progress reporting. Like other ed tech tools, kindergarten management systems offer technology-based solutions that reduce the manual workload for administrators and teachers while improving parent engagement and enhancing the overall educational experience [10].

2.1.1 Administrative Efficiency

The automated kindergarten management system automates the frequent administrative tasks related to attendance tracking, processing fees, and creating progress reports. It decreases the load of administration and minimizes human errors, thus saving a lot of time for teachers and staff to put more concentration on students.

2.1.2 Centralized Data Management

Kindergartens deal with a large amount of data, including student records, health information, and attendance logs. Kindergarten management systems provide a secure, centralized platform for storing and accessing this data, which enhances the efficiency and accuracy of data handling.

2.1.3 Growing Demand for Digital Solutions

With an increasing reliance on digital tools across all sectors, kindergartens are no exception. Parents expect more convenient and digital methods for interacting with schools, and management systems address this need by offering easy, tech-driven solutions for administrative tasks.

2.2 Review Existing System

2.2.1 iKEY

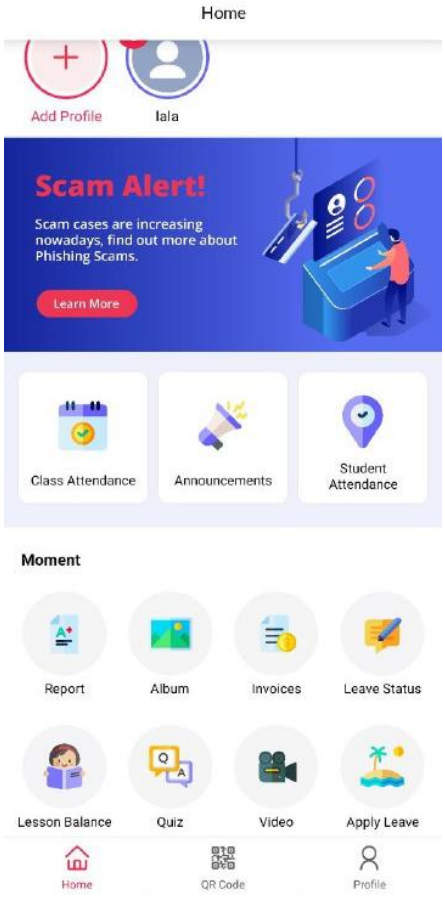


Figure 2.2.1: Homepage of iKEY

iKEY is a comprehensive kindergarten management system designed to enhance communication, safety, and administrative efficiency for both educators and parents. This system allows user to switch their children's profile without logout the app. It is importance as some parents have more than one child studying in a kindergarten.

One of its primary features is **Class Attendance**, it allows real-time tracking of students' daily attendance to notify parents when their child checks in or out of the

classroom. The **Announcement** feature is an in-app centralized hub for important school-related notifications, such as events, closures, or administrative reminders. **Student Attendance** offers more detailed reports, including insights into late arrivals, ensuring that both educators and parents stay informed.

The system also supports **Report** functionality, it generates detailed updates on a child's progress, activities, and behavior. The **Album** feature is a digital space where parents can view photos or videos of their child's school activities. For administrative needs, **Invoices** enable easy access to bill information, including tuition fees and other payments that parents must manage.

iKEY also includes **Lesson Balance**, which displays the number of remaining lessons or hours for specific programs or extracurricular activities. The **Quiz** feature supports educational assessments. It allows teachers to assign quizzes to monitor children's learning progress. For **leave management**, parents can request leave for their child, such as for illness or personal reasons, and track the status of these requests through Leave Status, keeping everything organized and transparent.

However, there is **lack of a real-time communication channel** between parents and teachers and **in-app announcements** could be improved. To solve this problem, developers can implement a **chat feature within the app** to allow direct communication between parents and teachers. This would enable quick discussions, and address concerns in real time to enhance a stronger connection between educators and families. Besides, to ensure that important announcements are not missed, **integrating push notifications** could be beneficial. This will notify parents instantly in case any new announcements are added so that no information of importance to them gets missed at the wrong time.

2.2.2 iParent App

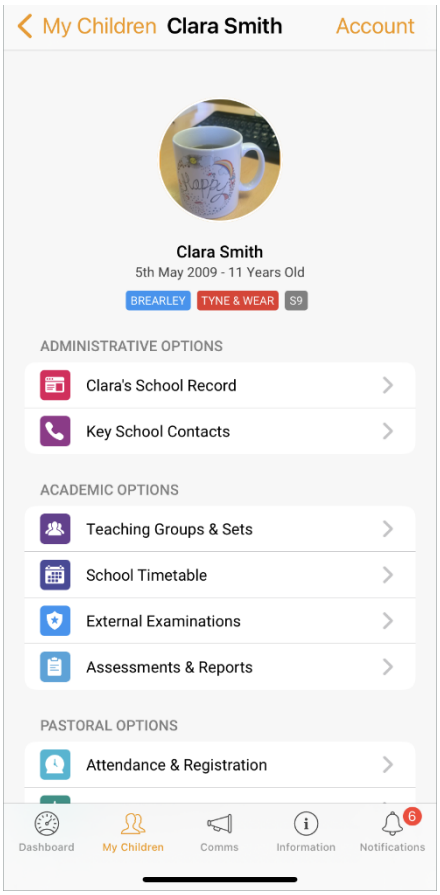


Figure 2.2.2: My Children page iParent App

iParent app is a system designed for parents to stay connected with their child's school activities. First, the app allows parents to choose the child profile they want to see if they have more than more children.

Administrative option features allow access to administrative details like student records and tutor information. **Timetable** allows parents to view their child's daily class schedule. The app also monitors **attendance**, providing updates on absences or late arrivals, and **generates reports** about academic performance, such as grades and assessments.

In addition, parents will receive **school notices and announcements** in app, ensuring they are up to date on events and any changes. The app facilitates **communication** between parents and teachers through email. Lastly, there is also **school event calendar that** keeps parents informed about extracurricular activities, encouraging active involvement in their child's school life.

There is some limitation in this application as it lacks an **integrated payment management system**. This absence makes fee payment inconvenient, as parents cannot

view invoices or track payment history as they need to handle these matters outside the app. Although communication between parents and teachers is available through email, it may not be the most efficient method as important updates could be missed due to an overflow of emails.

To address these issues, it is recommended that the app includes a real-time messaging feature for better communication and introduce an in-app payment system with invoice management and alternative payment methods, streamlining the payment process for parents.

2.2.3 AOneSchools

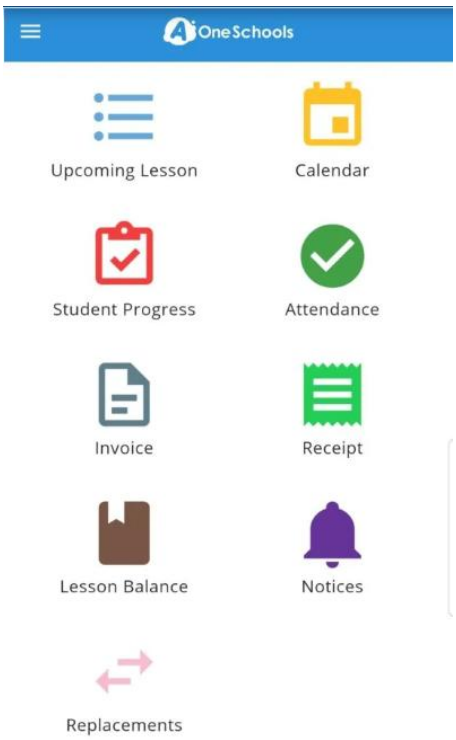


Figure 2.2.3: AOneSchools Home Page

AOneSchool is a management system designed for educational institutions, particularly enrichment centres, tuition centres, and schools. It offers a range of features aimed at improving administrative efficiency, communication, and overall school management.

The **Upcoming Lessons** feature acts as a **timetable** for students and parents, providing a clear view of scheduled classes. This can ensure that both students and parents are well-prepared, reducing the risk of missed classes. The **Calendar** feature offers a monthly view of all scheduled lessons, events, and important dates. This

allows parents to plan and stay updated on upcoming **important events**, such as exams, holidays, or parent-teacher meetings.

Parents can view detailed **student progress** reports within the app. These reports provide insights into their child's academic performance, participation in class, and development over time. The app also provides a **daily attendance tracking** feature, allowing teachers to mark attendance for each student.

Parents can view all paid and unpaid invoices, track their payment history, and download receipts for each transaction from the app but they cannot make payment directly from the app. **Lesson Balance** feature helps parents and students keep track of how many lessons or credits are remaining in their current program or course.

The **Notices** allows teachers or administrators to post announcements about upcoming events, holidays, or any changes to class schedules. Parents and students can check this section to stay updated on important information related to the school.

The **Replacement Request** feature enables students to request a replacement class if they have missed a scheduled session. This is particularly useful for students who may need to reschedule due to illness, personal reasons, or scheduling conflicts.

However, there is no direct communication channel that allow parents and teachers to communicate or discuss about their children. Besides, there is no in-app payment option for the parent which cause inconvenience during they make transaction and parents need to log in to the app regularly to see the announcement to prevent missed which are very inconvenient.

2.3 Comparison of Reviewed System

Feature	iKEY	iParent	AOneSchools	Proposed System
Multi-Profile	✓	✓	✗	✓
Attendance Tracking	✓	✓	✓	✓
Progress Reports	✓	✓	✓	✓
In-app Payment	✓	✗	✗	✓
Communication	✗	✓(Email)	✗	✓
In-app Announcement	✓	✓	✓	✓
Push notification	✗	✗	✗	✓
Leave Management	✓	✗	✓	✓

Table 2.3.1: Comparison of Reviewed System

2.4 Limitation of Existing Systems

Among the reviewed systems, it can be found that not all the systems have the in-app payment functionality which is important for the parents to have a seamless payment process. It is important to have an integrated payment system and provide alternative payment methods so it can save parents time and ensure the financial transactions are handled efficiently and securely.

Besides, although all the system has an announcement functionality but it still not enough as it needs parents log into the app regularly to check for announcements. It is better to have a push notification that can instantly alert parents about important announcements, events or changes.

Lastly, there is no direct communication channel for the parents and teachers to communicate effectively. Although iParent offers email communications but it may lead to missed messages due to flooded emails. Thus, a chat channel within the app can provide immediate communication for parents and teachers to have a quick discussion about the children.

Chapter 3
System Methodology/Approach

3.1 Agile Development Methodology

The system was developed using a systematic and iterative methodology that includes analysis, design, implementation, integration and testing phases. Requirement analysis was performed first to gather all necessary features through research and stakeholder needs. After analysing the requirements, the design phase was conducted to design the functional architecture, database structure and user interface layout.

Agile methodology was used to develop the Kindergarten Management System with Parental Control due to its flexibility and adaptability to changes. Agile enables the project to be divided into smaller, manageable sprints where specific features such as attendance tracking, tuition billing, and real-time messaging can be developed incrementally. Continuous testing, validation and user feedback during each sprint will ensure that the system evolves according to the needs of parents, teachers, and administrators, ultimately meeting the overall objectives of the project efficiently.

3.2 Iterative Development Process (Sprint)

Sprint No	Function	Deliverables
1	Student Registration Function	A module that allows admin to register student to the system using student and parent information. Parent accounts for mobile applications are automatically created during this process.
2	Billing Function	A billing system that allows admin to create bill, generates bill statement. Parents can view their outstanding payments directly through the app.
3	Payment Function	Integration with Stripe payment gateways that allows parents to make payments in-app. Automatic receipt generation is included.
4	Chat Room Function	Real-time messaging between parents and teachers with unread message indicators conversation history.
5	Announcement Function	A posting module that allows web administrators to create announcements on important events. Users can

		interact with announcements through likes and comments from both the web and mobile platforms.
6	Attendance Function	A student attendance tracking system where teachers can mark attendance and parents can view updates.
7	User Authentication	A login system using Firebase Authentication for parents, teachers and administrators.
8	Notification	A background notification service using Pushy that allows parents to receive updates such as bills, reports, chat messages, and announcements even when the app is not open.
9	Report Function	A report module that displays exam results and overall performance progress reports for parents to monitor their child's academic performance.
10	Leave Management	A leave application system where parents can submit leave and teachers are always notified when student is absent.

Table 3.2.1: Iterative Development Process (Sprint)

3.3 Gantt Chart

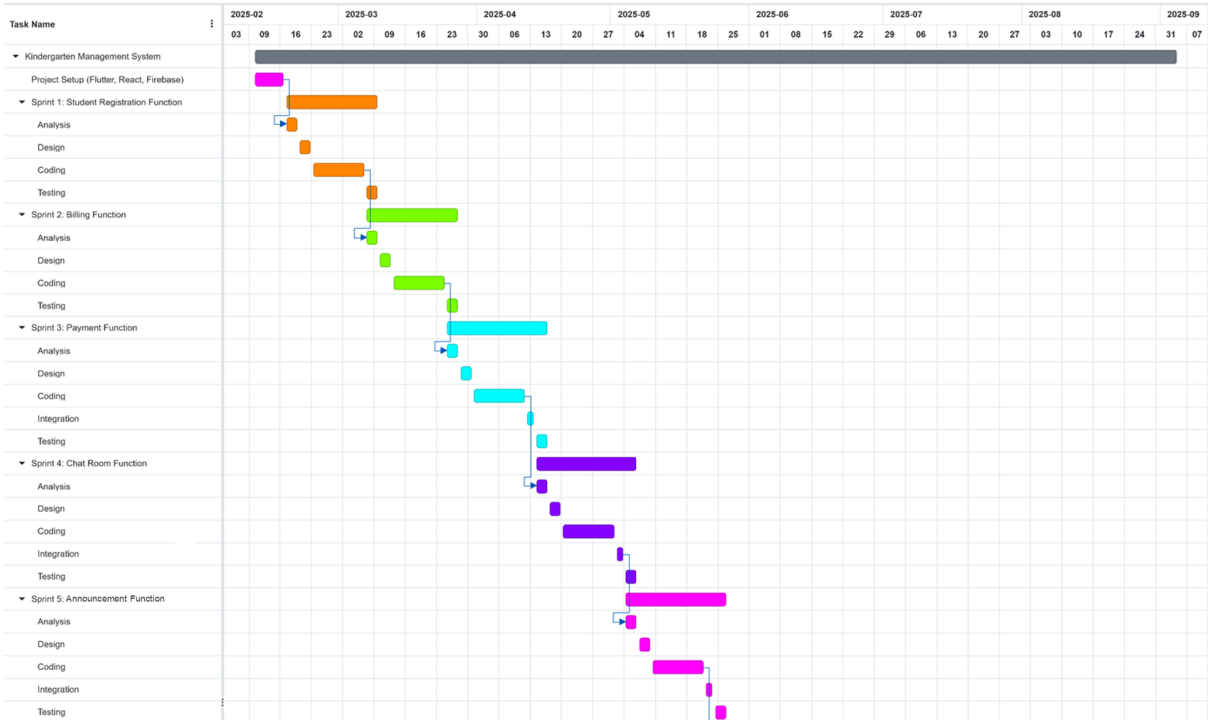


Figure 3.3.1: Gantt Chart

Figure 3.3.2: Gantt Chart (Continued)

3.4 User Requirements

3.4.1 Parent Requirements

1. **Receive Real-Time Notifications**

   Parents can get push notifications for new bills, bill reminders, exam reports, announcements and chat messages even when the app is closed.

2. **View and Pay Bills**

   Parents can access outstanding bills, view detailed statements and make secure payments through integrated Stripe payment gateways.

3. **View Payment History**

   Parents can access previous payment records and download receipts for reference.

4. **Communicate with Teachers**

   Parents can send and receive messages through a real-time chat system with support for text, images, and videos.

5. **Monitor Child's Attendance**

   Parents can view daily attendance records, including present, absent or on leave status marked by teachers.

6. **View Academic Reports**

   Parents can access exam results and progress reports in a structured and easy-to-read format.

7. **Apply for Leave**

   Parents can submit leave applications for their children to inform the teachers if they are absent.

8. **View Announcements**

   Parents can always stay updated on upcoming school events and interact on important announcements posted by the administration using like and comment features.

3.4.2 Teacher Requirements

1. **Record Student Attendance**

   Teachers can mark student attendance daily through the system, if student is absent, teacher can choose to enter the reason for absence.

2. **Update Student Performance in form of Report**

   Teachers can enter exam results and progress reports as an update so parents can monitor their child's development.

3. **Communicating with Parents**

   Teachers can send and receive real-time messages, share important information and attach images or videos when necessary.

4. **View Announcements**

   Teachers can be informed and interact on important announcements posted by the administration using like and comment features.

3.4.3 Administrator Requirements

1. **Manage Student Registration**

   Admin can register students in the system along with their parent information, which automatically creates parent accounts.

2. **Create and Manage Billing**

   Admin can create bills for students and generate statements for parents to view. They can manually update the payment status if the bill is payment outside of the app and send reminders to specific parents if there is bill that near to due date.

3. **Manage System Users**

   Admin can create and manage teacher accounts and assign class and subject for each teacher.

4. **Post Announcements and Events**

Admin can create announcements for important school events and share them with parents and teachers.

5. **Communicating with Parents**

   Admin can send and receive real-time messages, share important information and attach images or videos when necessary.

3.5 Use Case Diagram

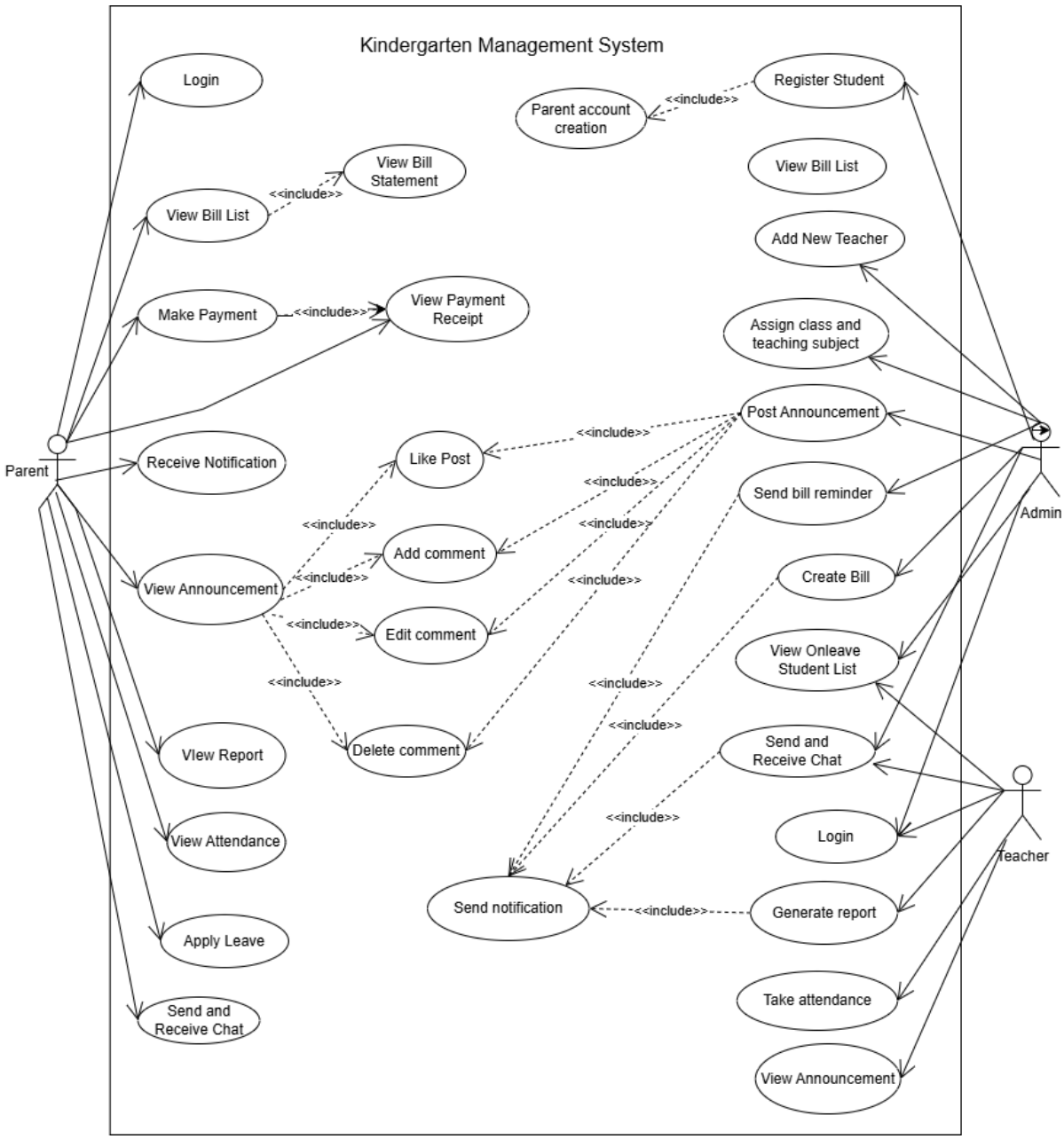


Figure 3.5.1: Use Case Diagram

The use case diagram represents the functional requirements of the Kindergarten Management System from the perspective of different users, which are parents, teachers and

administrators. Parents can perform multiple actions including billing operation, receiving notification and announcements, viewing report and attendance, leave application and communication via chat. Admin can handle tasks like student registration, teacher assignment, bill creation and reminder and announcement. Furthermore, teachers can update students' attendance, create reports and chat with parents.

3.6 Use Case Description

3.6.1 Admin, Teacher and Parent Use Case Description

Use Case Name	Login
Description	Allows users to access the system using valid credentials.
Pre-Condition	Users must be registered in the system
Basic Flow	1. User enters username and password. 2. User clicks Login, 3. System verifies credentials. 4. User is redirected to their respective dashboard.
Alternative Flow	- Forgot Password: 1. User clicks **Forgot Password**. 2. System guides user through the password reset process.
Exceptional Flow	- Invalid Credentials: 1. If the username or password is incorrect, system displays an error message. 2. User is allowed to re-enter login credentials.
Post-Condition	User is authenticated and logged into the system.

Table 3.6.1.1: Login Use Case Description

Use Case Name	Send and Receive Chat
Description	Allow users to send and receive message
Pre-Condition	Recipient must exist in the system.
Basic Flow	1. User selects a recipient. 2. User types a message in the chat input field. 3. User optionally attaches an image or video. 4. User clicks Send. 5. System send notification and delivers the message to the recipient.

	6. Recipient receives the message in real time and can reply.
Alternative Flow	- No Attachment: 1. User sends only a text message without any media.
Exceptional Flow	If message delivery fails, system will notify the user.
Post-Condition	Message is stored in the system and can be viewed in the chat history.

Table 3.6.1.2: Send and Receive Chat Use Case Description

Use Case Name	View Announcement
Description	Users can view announcements or upcoming events posted by the admin.
Pre-Condition(s)	Admin has posted announcements.
Basic Flow	1. User opens the announcement page. 2. System displays a list of announcements. 3. Users view the details.
Alternative Flow	-
Exceptional Flow	Announcement fails to load.
Post-Condition(s)	Users successfully view event information.

Table 3.6.1.3: View Announcement Use Case Description

Use Case Name	Like Announcement Post
Description	Users can like the announcements posted by admin.
Pre-Condition(s)	At least one announcement exists in the system.
Basic Flow	1. User opens the announcement page. 2. System displays a list of announcements. 3. User clicks the "Like" button for a selected announcement. 4. System records the like and updates the like count.
Alternative Flow	-
Exceptional Flow	If the like action fails, system displays an error message.
Post-Condition(s)	Like count for the announcement is increased by one and displayed to all users.

Table 3.6.1.4: Like Announcement Post Use Case Description

Use Case Name	Comment Announcement
Description	Users can comment on the announcements or upcoming events posted by the admin.
Pre-Condition(s)	At least one announcement exists in the system.
Basic Flow	1. User opens the announcement page. 2. System displays a list of announcements. 3. Users view the details and comment on the post. 4. System saves the comment and displays it under the announcement.
Alternative Flow	-
Exceptional Flow	If the announcement fails to load, system shows an error message.
Post-Condition(s)	Comments shown by every users.

Table 3.6.1.5: Comment Announcement Post Use Case Description

Use Case Name	Edit Comment
Description	Users can edit the comment they have posted under an announcement.
Pre-Condition(s)	The comment exists and belongs to the logged-in user.
Basic Flow	1. User opens the announcement page. 2. System displays a list of announcements. 3. User clicks "Edit" on the comment. 4. System allows the user to modify the text. 5. User clicks "Save". 6. System updates the comment with the latest version.
Alternative Flow	-
Exceptional Flow	If the comment cannot be updated, system shows an error message.
Post-Condition(s)	Comment content is updated and displayed with the edited version.

Table 3.6.1.6: Edit Comment Use Case Description

Use Case Name	Delete comment
Description	Users can delete a comment they have posted under an announcement.
Pre-Condition(s)	The comment exists and belongs to the logged-in user.
Basic Flow	1. User opens the announcement page. 2. System displays a list of announcements. 3. User clicks "Delete" on the comment. 4. User confirms deletion. 5. System removes the comment from the database and refreshes the comment list..
Alternative Flow	-
Exceptional Flow	If deletion fails, system shows an error message.
Post-Condition(s)	The comment is removed and no longer visible to users.

Table 3.6.1.7: Delete Comment Use Case Description

Use Case Name	Send and Receive Chat
Description	Allow users to send and receive message
Pre-Condition	Recipient must exist in the system.
Basic Flow	7. User selects a recipient. 8. User types a message in the chat input field. 9. User optionally attaches an image or video. 10. User clicks Send. 11. System send notification and delivers the message to the recipient. 12. Recipient receives the message in real time and can reply.
Alternative Flow	- No Attachment: 1. User sends only a text message without any media.
Exceptional Flow	If message delivery fails, system will notify the user.
Post-Condition	Message is stored in the system and can be viewed in the chat history.

Table 3.6.1.8: Send and Receive Chat Use Case Description

3.6.2 Admin Use Case Description

Use Case Name	Register Student
Description	Allows the admin to register a new student in the system, along with parent details. If the parent already exists, the system links the student to the existing parent account.
Pre-Condition(s)	Admin logged in.
Basic Flow	1. Administrator enters student details. 2. Administrator enters parent d. 3. System checks if parent already exists in the database. 4. If parent does not exist, system creates a new parent record. 5. System stores the new student record and links it with the parent record. 6. System confirms successful registration.
Alternative Flow	- Parent Already Exists 1. System detects existing parent record. 2. Admin can choose to use the existing parent details or cancel and re-enter correct parent data if it is a mistake. 3. System proceeds to link the student to the selected parent record.
Exceptional Flow	- Missing or invalid data. 1. System displays an error message indicating which field are invalid. 2. Admin is prompted to correct the data before proceeding.
Post-Condition(s)	- Student record is successfully created. - Parent record is created. - Student is linked to the parent account.

Table 3.6.2.1: Student Registration Use Case Description

Use Case Name	Create Bill
Description	Admin generates bills for specific students or an entire class. Parents are notified once the bill is created and the bill becomes available in the parent's account for viewing and payment.
Pre-Condition(s)	Student record exists.
Basic Flow	1. Admin selects a class from the class list.

	2. Admin select a student or whole class 3. Admin enter bill details. 4. Admin confirms the bill creation. 5. System saves the bill(s) to the database 6. System displays a confirmation message of successful bill creation.
Alternative Flow	-
Exceptional Flow	- Missing or invalid data. 1. System displays an error message indicating which field are invalid. 2. Admin is prompted to correct the data before proceeding.
Post-Condition(s)	- Bill(s) are stored in the system. - Parents receive a notification about the new bill. - Bills are available for parents to view in their accounts.

Table 3.6.2.2: Bill Creation Use Case Description

Use Case Name	Send Bill Reminder
Description	Admin sends reminders for unpaid bills that are due in 3 days or already overdue. Parents receive a notification so they can make payment.
Pre-Condition(s)	Unpaid bill exists.
Basic Flow	1. Admin click on the send reminder button. 2. System generates reminder notifications for selected bills. 3. System delivers notifications to the corresponding parents.
Alternative Flow	-
Exceptional Flow	- Notification fails: 1. System logs the failed delivery attempt.
Post-Condition(s)	Parents receive notification about unpaid bills.

Table 3.6.2.3: Bill Reminder Use Case Description

Use Case Name	Add New Teacher
Description	Allows admin to add a new teacher to the system and automatically create their user account for system access.
Pre-Condition(s)	Admin logged in.
Basic Flow	1. Admin clicks on the "Add" button.

	2. Admin enters teacher details. 3. System saves the teacher's record. 4. System automatically creates a user account for the teacher and assigns login credentials.
Alternative Flow	Duplicate Teacher Record: 1. If a teacher with the same email already exists, the system displays a warning message. 2. Admin enter the correct information.
Exceptional Flow	- Missing or invalid data. 1. System displays an error message. 2. Admin is prompted to correct the data before proceeding.
Post-Condition(s)	- New teacher record is successfully stored in the database. - Teacher account is created and ready for login.

Table 3.6.2.4: Add New Teacher Use Case Description

Use Case Name	Assign class and teaching subject
Description	Allows the administrator to assign one or more classes and teaching subjects to a selected teacher.
Pre-Condition(s)	- Teacher record must exist in the system. - Class records and subject list must already exist.
Basic Flow	1. Administrator selects a teacher from the teacher list. 2. Administrator chooses the class and subject to assign. 3. Administrator clicks "Save" to confirm the assignment. 4. System stores the class-subject assignment for the teacher.
Alternative Flow	Multiple Assignments: - Admin can assign multiple classes or subjects to the same teacher in a single action.
Exceptional Flow	Duplicate class and subject: - System displays an error message and prompts the administrator to choose a different combination.
Post-Condition(s)	Teacher is successfully assigned to the selected class and subject(s).

Table 3.6.2.5: Assign Class and Subject Use Case Description

Use Case Name	View On Leave Student List
Description	Allows the admin or teacher to view a list of students who are on leave for a selected date.
Pre-Condition(s)	Leave requests must have been submitted.
Basic Flow	1. Admin or teacher selects date from the calendar. 2. System displays a list of all students who have applied for leave on the selected date. 3. They can view additional details for each leave request including uploaded documents if uploaded.
Alternative Flow	Filter by Class - They can filter the leave list by class for easier viewing
Exceptional Flow	If no leave requests exist for the selected date, the system displays a message such as “No students On Leave.”
Post-Condition(s)	Admin or teacher successfully views the on-leave student list for the selected date.

Table 3.6.2.6: View On Leave Student List Use Case Description

Use Case Name	Post Announcement
Description	Admin can create, edit or delete announcements for important updates or events, which will be visible to parents and teachers in the system.
Pre-Condition(s)	Admin logged in.
Basic Flow	1. Administrator clicks the Add button to create a new announcement. 2. Administrator enters announcement details. 3. Administrator clicks Post. 4. System saves the announcement and publishes it to all users.
Alternative Flow	Edit or Delete Announcement a. Admin selects an existing announcement. b. Admin edits the content and clicks Save or deletes it entirely. c. System updates or removes the announcement.
Exceptional Flow	Posting Fails a. System displays an error b. Admin can correct the issue and retry.

Post-Condition(s)	- Announcement is published and visible to all users in their announcement list. - Parents receive a notification about the new announcement.

Table 3.6.2.7: Post Announcement Use Case Description

3.6.3 Teacher Use Case Description

Use Case Name	Generate report
Description	Allows teachers to generate exam results or progress reports for students.
Pre-Condition(s)	Student records must exist.
Basic Flow	1. Teacher selects a class. 2. Teacher selects a student from the list. 3. Teacher selects a report type. 4. System displays the default report form. 5. Teacher fills in required details. 6. Teacher enters comment. 7. Teacher submits the report. 8. System saves the report and sends a notification to the parent
Alternative Flow	- Exam Report Selected 1. The system displays the exam report form. 2. Teacher fills in exam details. - Other Reports 1. The system shows the report progress form. 2. Teacher fills at least one subsection.
Exceptional Flow	- Missing or invalid data. 1. System displays an error message indicating which field are invalid. 2. Teacher is prompted to correct the data before proceeding.
Post-Condition(s)	- The report is successfully stored in the database. - Parents are notified and can view the report in their app.

Table 3.6.3.1: Report Generation Use Case Description

Use Case Name	Take Attendance
Description	Allows teachers to mark student attendance for a selected class.
Pre-Condition(s)	Class and student records must exist in the system.
Basic Flow	1. Teacher selects the class from the list. 2. Teacher marks each student as present or absent. 3. Teacher clicks Save to confirm attendance. 4. System updates the attendance records and notifies parents
Alternative Flow	2a: If a student is marked absent, the teacher can optionally provide a reason.
Exceptional Flow	-
Post-Condition(s)	Attendance records are successfully saved and made available for parents to view.

Table 3.6.3.2: Attendance records Use Case Description

3.6.4 Parent Use Case Description

Use Case Name	View Bill Details
Description	Allows parents to view the details of their child's bills.
Pre-Condition(s)	Bills must exist in the system.
Basic Flow	1. Parent selects a bill from the billing page. 2. System displays bill details.
Alternative Flow	Parent downloads bill statement.
Exceptional Flow	-
Post-Condition(s)	Bill details are displayed on the screen for the parent.

Table 3.6.4.1: View Bill Details Use Case Description

Use Case Name	Make Payment
Description	Allows parents to make in-app payment.
Pre-Condition(s)	A bill must exist and a valid payment method must be available.
Basic Flow	1. Parent selects a bill to pay. 2. Parent selects payment method. 3. Parent confirms payment. 4. System processes the transaction. 5. Receipt is generated and saved.

Alternative Flow	2**a:** If card selected → parent enters card details. 2**b:** If GrabPay selected → parent is redirected to Grab app/payment page. - Transaction is aborted if parent cancels payment.
Exceptional Flow	Transaction fails - system displays error message and does not record payment.
Post-Condition(s)	Payment is successfully recorded and bill is marked as paid.

Table 3.6.4.2: Make Payment Use Case Description

Use Case Name	View Payment Receipt
Description	Allows parents to view their payment receipt after completing a transaction
Pre-Condition(s)	Payment has been completed.
Basic Flow	1. System automatically generates a receipt once the payment is successful. 2. Receipt is stored in the parent's account. 3. Parent selects "Receipt" button for the bill. 4. System displays the receipt details.
Alternative Flow	-
Exceptional Flow	-
Post-Condition(s)	Receipt remains available in the system for future reference.

Table 3.6.4.3: View Receipt Use Case Description

Use Case Name	Receive Notification
Description	Parents receive real-time notifications for bills, announcements, reminders, reports and chat messages.
Pre-Condition(s)	Notifications must be enabled in the app.
Basic Flow	1. Admin or teacher posts a bill, announcement, report or sends a message. 2. System sends push notification to parent's device. 3. Parent views notification.
Alternative Flow	Parent can view missed notifications later in the notification history.
Exceptional Flow	Notification not delivered due to network issues.

Post-Condition(s)	Notification is stored in parent's notification list for reference.

Table 3.6.4.4: Receive Notification Use Case Description

Use Case Name	View Report
Description	Parents can view their child's academic and progress reports.
Pre-Condition(s)	Reports must exist in the system.
Basic Flow	1. Parent selects "Reports." 2. Parent chooses a report type (exam results/progress). 3. System displays selected report.
Alternative Flow	-
Exceptional Flow	-
Post-Condition(s)	Report is displayed for parent's review.

Table 3.6.4.5: View Report Use Case Description

Use Case Name	View Attendance
Description	Parents view attendance records marked by teachers.
Pre-Condition(s)	Teacher must have recorded attendance.
Basic Flow	1. Parent selects child to view record. 2. System displays attendance details.
Alternative Flow	2a Parent can tap **"Recent"** to see last 5 days. 2b Parent can select **"Overview"** to view monthly summary.
Exceptional Flow	-
Post-Condition(s)	Attendance records are displayed for the selected child.

Table 3.6.4.6: View Attendance Use Case Description

Use Case Name	Apply Leave
Description	Allows parents to apply leave for their child.
Pre-Condition(s)	Parent logged in.
Basic Flow	1. Parent fills out leave form with date and reason. 2. Parent optionally uploads supporting document. 3. Parent submits leave application.
Alternative Flow	--
Exceptional Flow	

Post-Condition(s)	Leave application is stored in the system and attendance status is marked as “On Leave”.

Table 3.6.4.7: Apply Leave Use Case Description

Chapter 4

System Design

4.1 Architecture Diagram

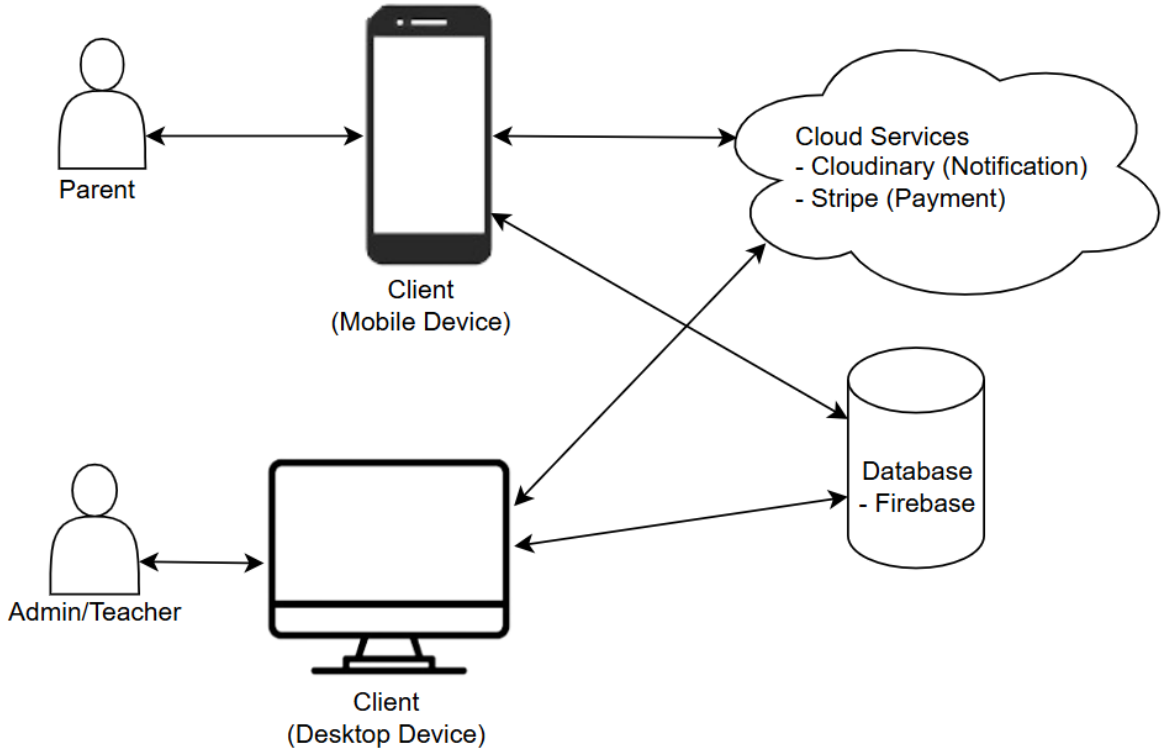


Figure 4.1.1: Architecture Diagram

The system architecture diagram illustrates the interaction between different users, clients, cloud services and the database in the Kindergarten Management System. Parents can access the system through a mobile client to interact with the system features, receive notifications and make online payments while admins and teachers use a desktop client to manage students, billing, announcements, and other administrative tasks.

Both mobile and desktop clients connect to the Firebase database to store user accounts, attendance, reports, billing records and other essential data. The system also integrates with cloud services such as Cloudinary to handle and deliver real-time notifications and Stripe to securely process online payments made by parents.

4.2 Class Diagram

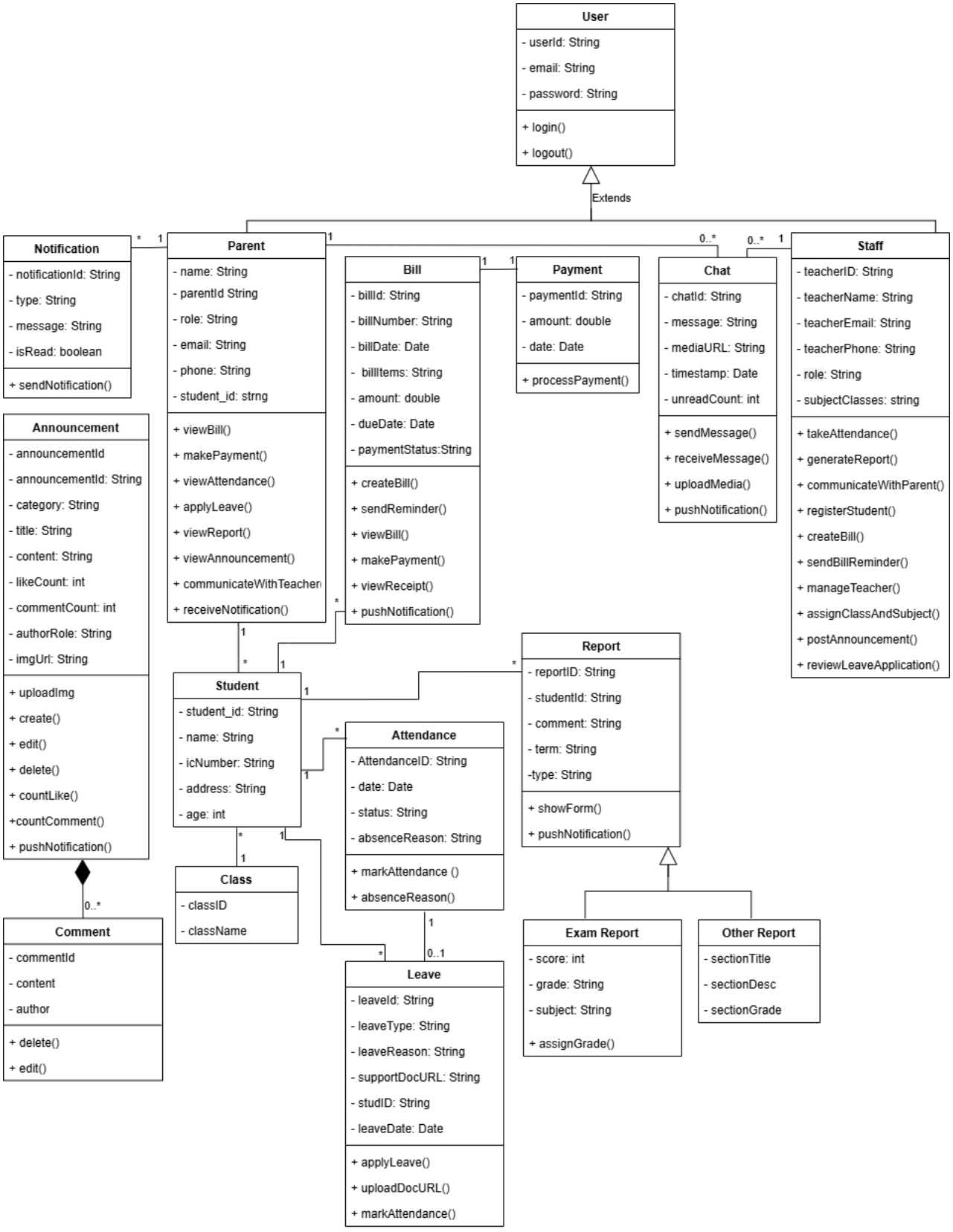


Figure 4.2.1: Class Diagram

The class diagram illustrates the main entities and their relationships in the Kindergarten Management System. The User class provides common attributes such as userId, email and password and is extended by Parent and Staff classes. Parent able to do actions like viewing announcements, applying leave, making payments, and communicating with teachers while Staff includes functions for taking attendance, generating reports, creating bills, posting announcements and viewing on leave student lists.

The Student class stores the details of the student and is linked to Parent and Class. Bill and Payment used to manage billing information, due dates, payment amounts and payment status. Besides, Attendance tracks student attendance which includes status and reason for absence. Report represents academic and other student report which has subclass like Exam Report and Other Report. Announcement handle the features such as like and comments while Comment linked to the announcement for interaction and feedback, Notification ensures that parents always stay updated. In addition, chat provides real-time messaging and media uploads for parent and staff. Finally, the Leave allows parents to submit leave applications with reason and document so they can update attendance accordingly.

4.3 Activity Diagram

4.3.1 Login

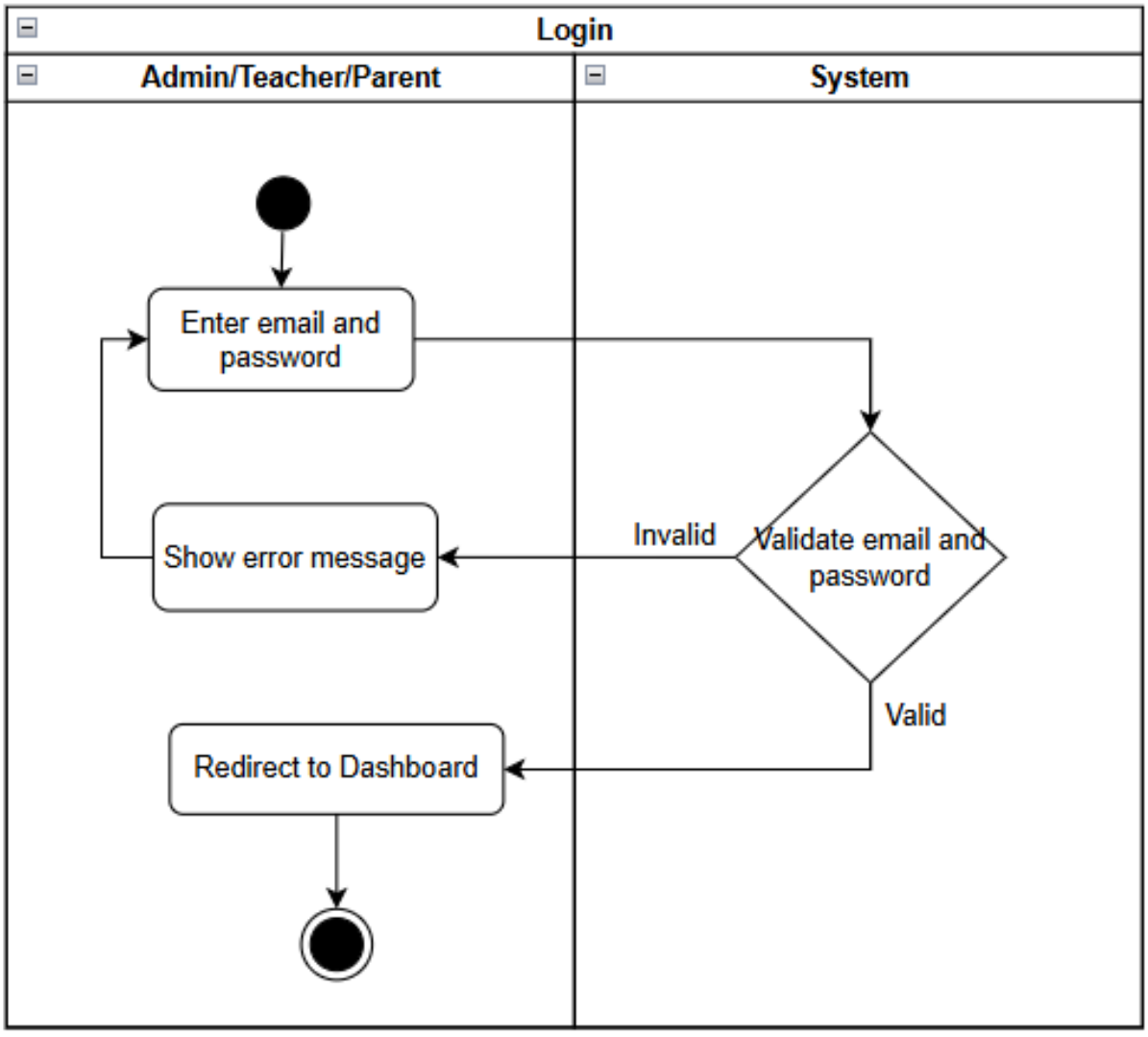


Figure 4.3.1: Login Activity Diagram

The login process starts when a user enters their email and password into the system. The system then validates the provided credentials. If the credentials are valid, the user can logs into the system. However, if the validation fails, the system notifies the user of the error and prompts them to re-enter their login information.

4.3.2 Student Registration

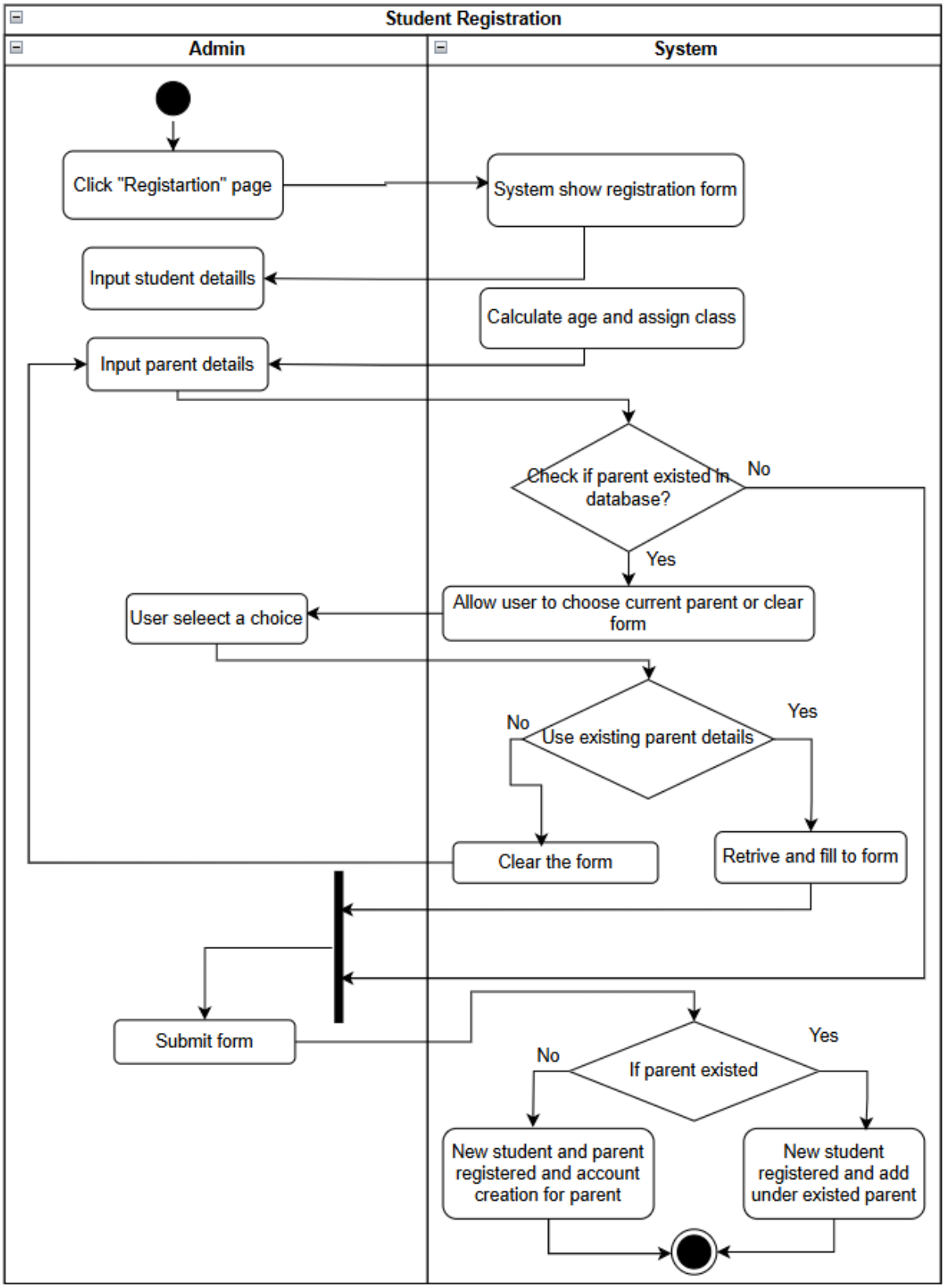


Figure 4.3.2: Student Registration Activity Diagram

The student registration process starts when user click into the Registration page and the system show the registration form. The user is required to enter the student's and the parent's details.

Student age will be calculated based on the Identity Card Number and the class will be assigned to the student using the age. Besides, the system will determine if the parent is existed in the database or not, if yes, the student record will be linked to the existing parent account. Otherwise, the system will create a new parent account and associate the student with the newly created account.

4.3.3 Add New Teacher

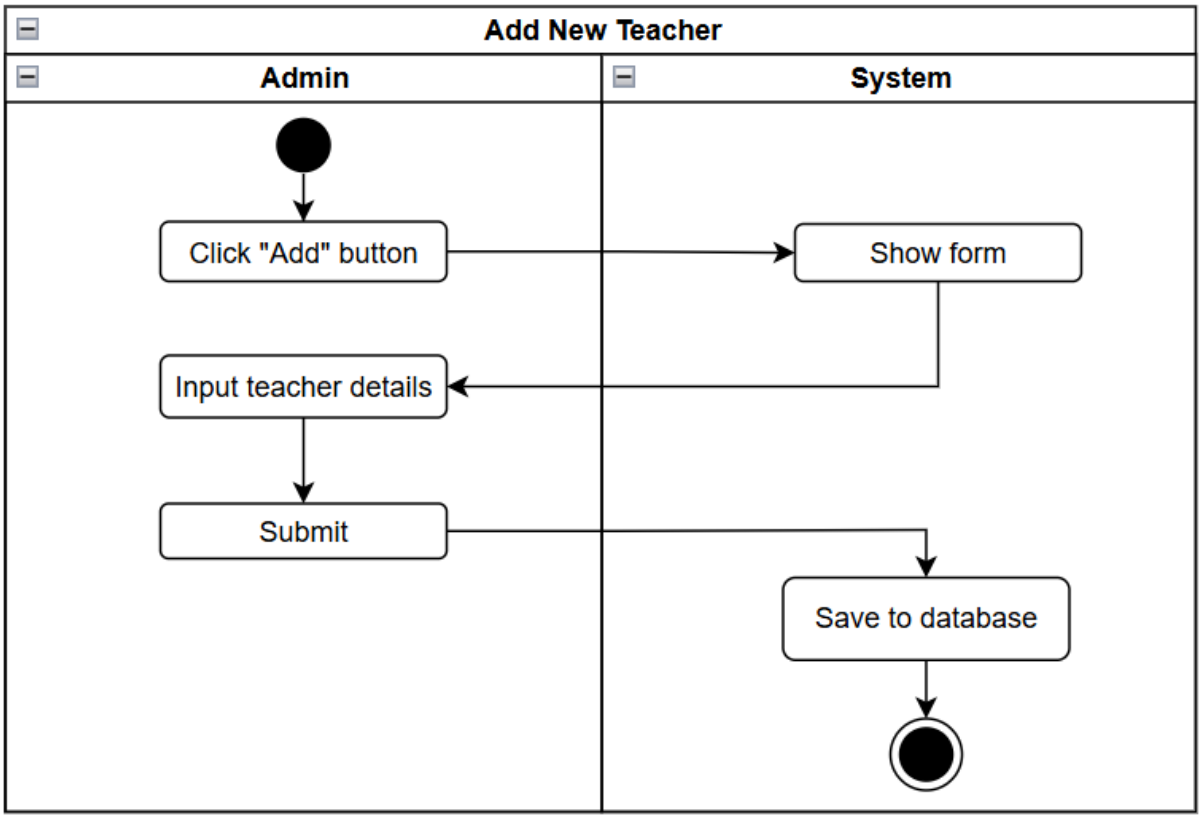


Figure 4.3.3: Add New Teacher Activity Diagram

The process begins when user clicks on the Add button, then the system displays the form to allow user to input teacher details. Once submitted the form, the system validates the information and stores the data in the database successfully.

4.3.4 Assign Class and Subject

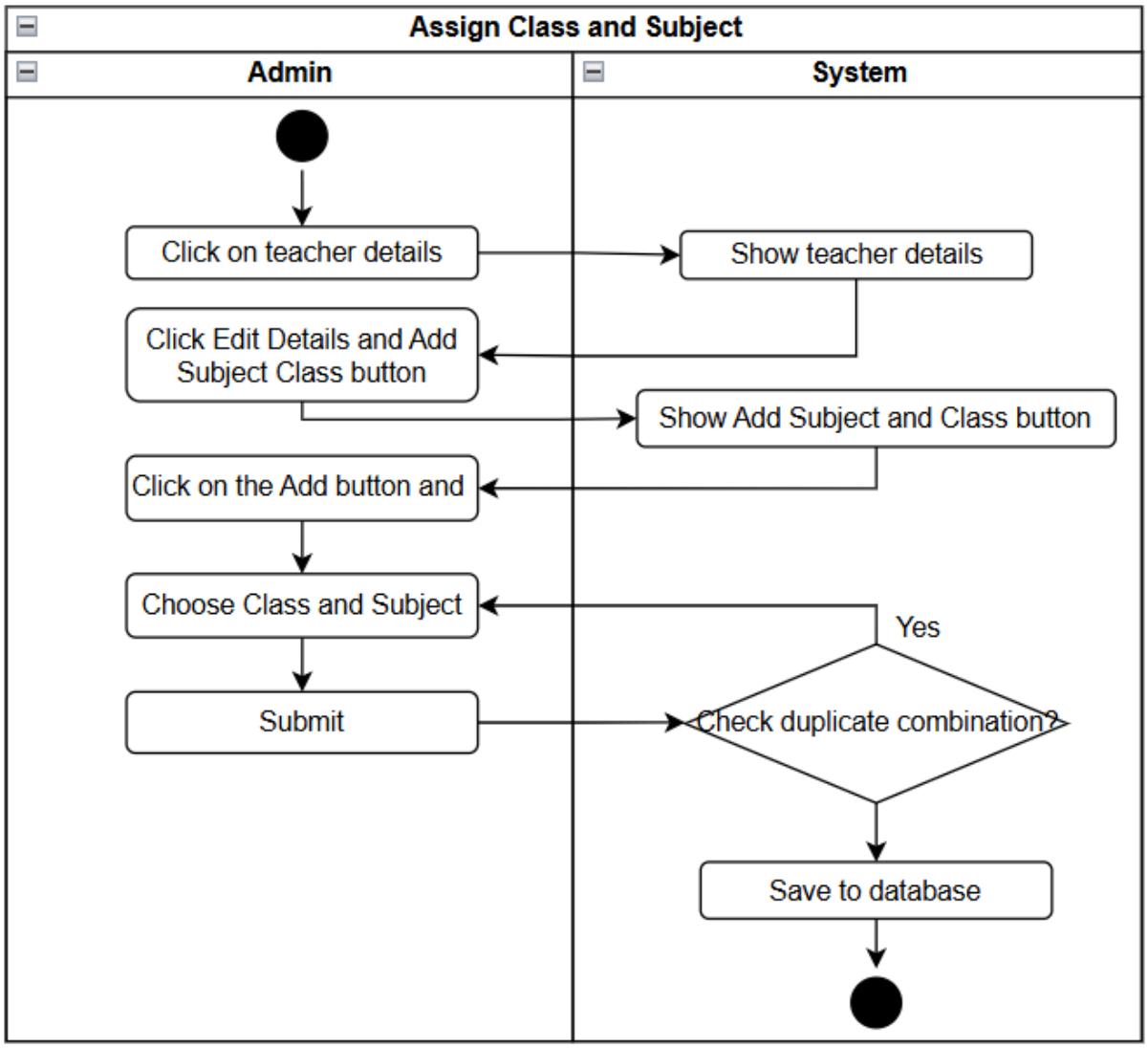


Figure 4.3.4: Assign Class and Subject Activity Diagram

The process starts when the user clicks on the teacher details. The system then displays a dialog showing the teacher's information. The user clicks on the Edit and Add Class button and selects the class and subject to assign. Upon submission, the system checks for any duplicate combinations. If a duplicate is found, the user will be prompted to choose again. Otherwise, the information is saved into the database successfully.

4.3.5 Create Bill

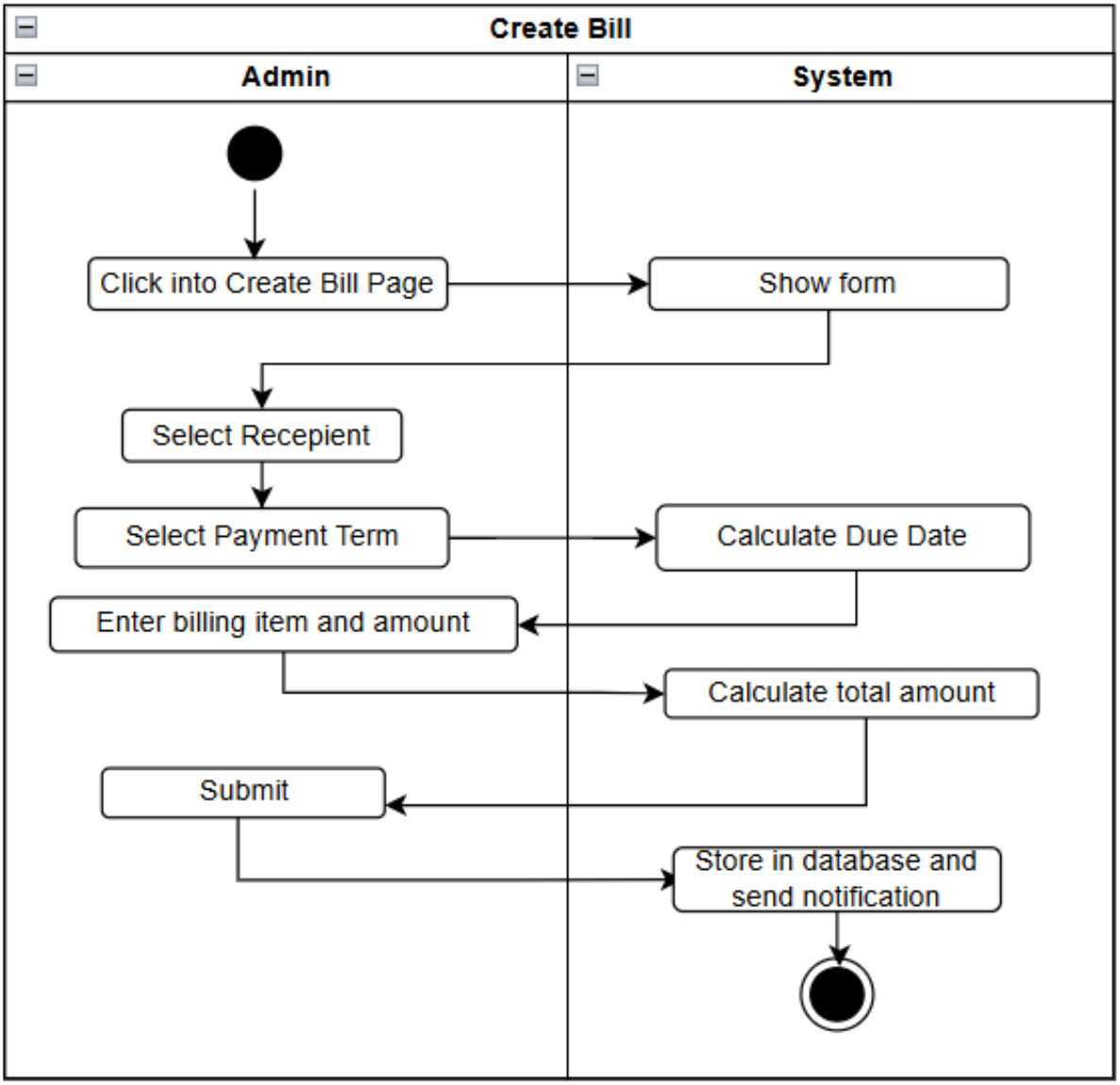


Figure 4.3.5: Create Bill Activity Diagram

The process begins when the user clicks the Create Bill page. The system displays a form where the user must select the recipient and choose a payment term. Based on the selected payment term, the system automatically calculates the due date. Next, the user enters the billing items and their respective amounts. The system calculates the total amount and displays it for confirmation. Once the user submits the bill, the system stores the billing information in the database and sends a push notification to the selected parents to inform them about the new bill.

4.3.6 Bill Reminder

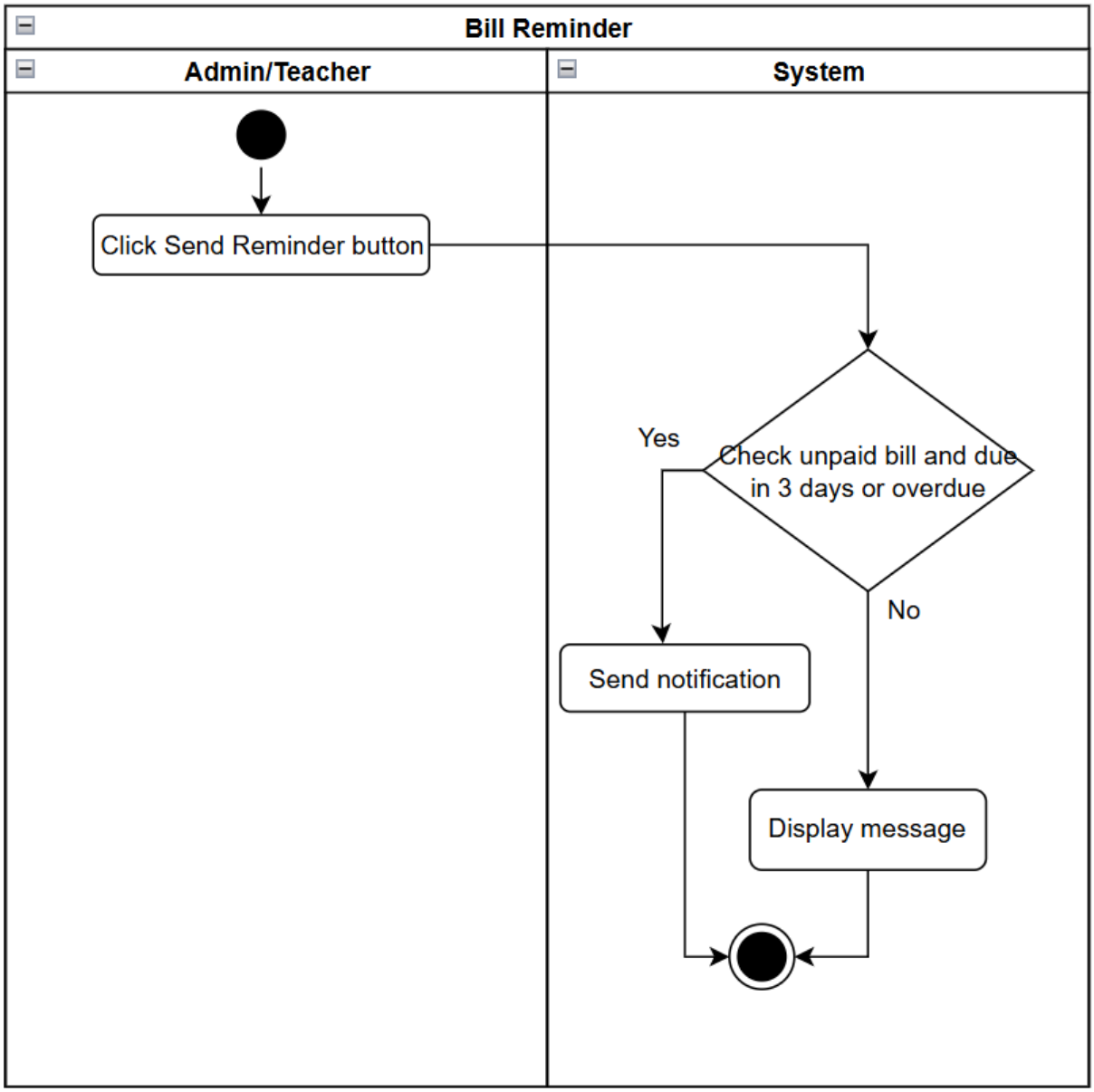


Figure 4.3.6: Bill Reminder Activity Diagram

The process begins when the user clicks the Send Reminder button. The system searches for all unpaid bills that are either due within the next 3 days or already overdue. If such bills are found, the system sends a push notification to the respective parents reminding them to complete the payment. If no bills meet the criteria, the system displays a message indicating that there are no unpaid bills due within the next 3 days.

4.3.7 View Bill and Make Payment

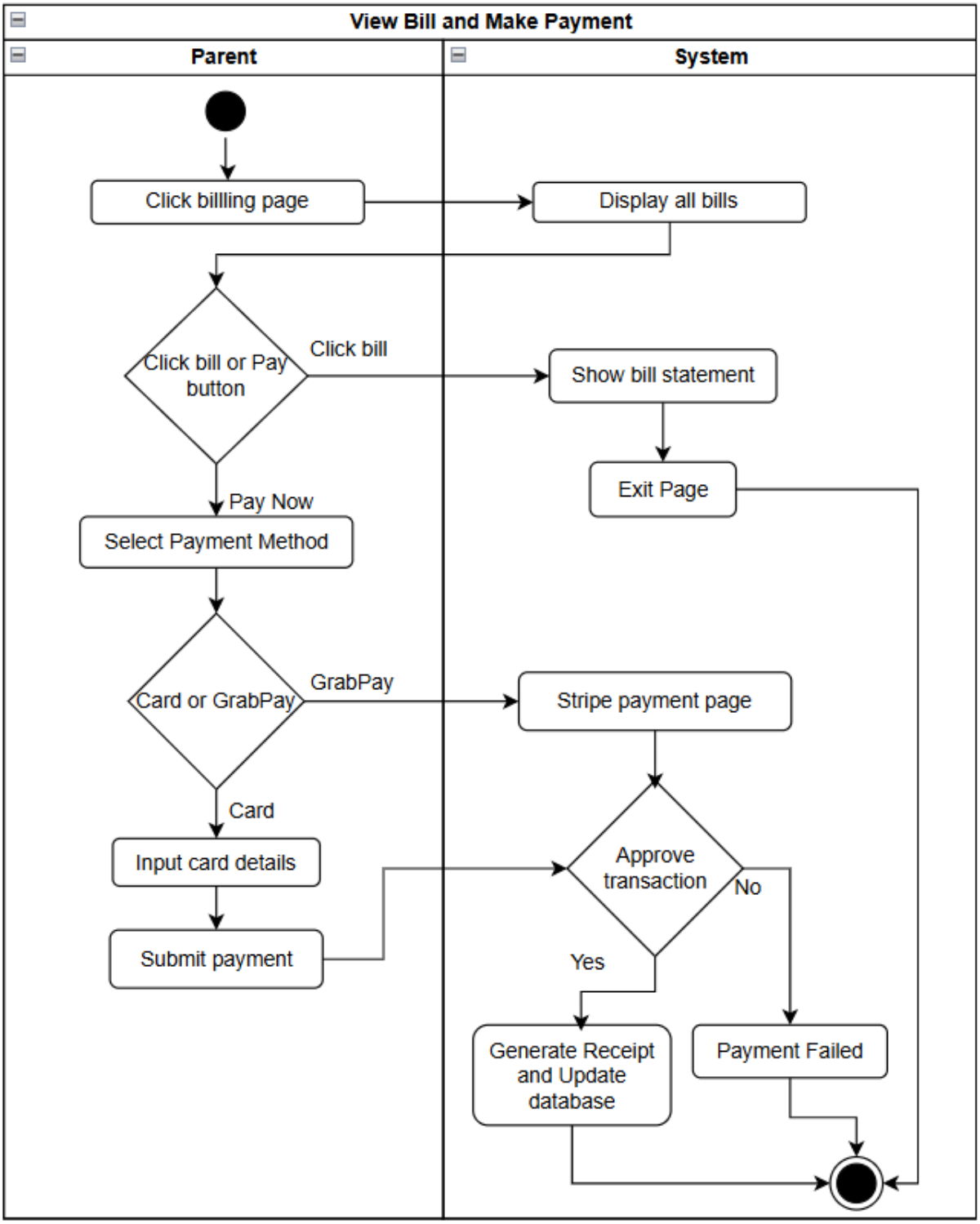


Figure 4.3.7: View Bill and Make Payment Activity Diagram

The process starts after user clicks into the billing page and the system lists out all bills. User can choose to see the bill statement or make payment. If view payment, the user will be directed to a bill statement page, else user proceeds with payment. User needs to select a payment method which are card that needs to enter card details and confirm payment while GrabPay

will be redirected to the payment authorize page and must approve the transaction. Then the system will check the payment status, if successful, the system generates a receipt and update the payment status in database. Otherwise, the system displays payment failure messages.

4.3.8 Take Attendance

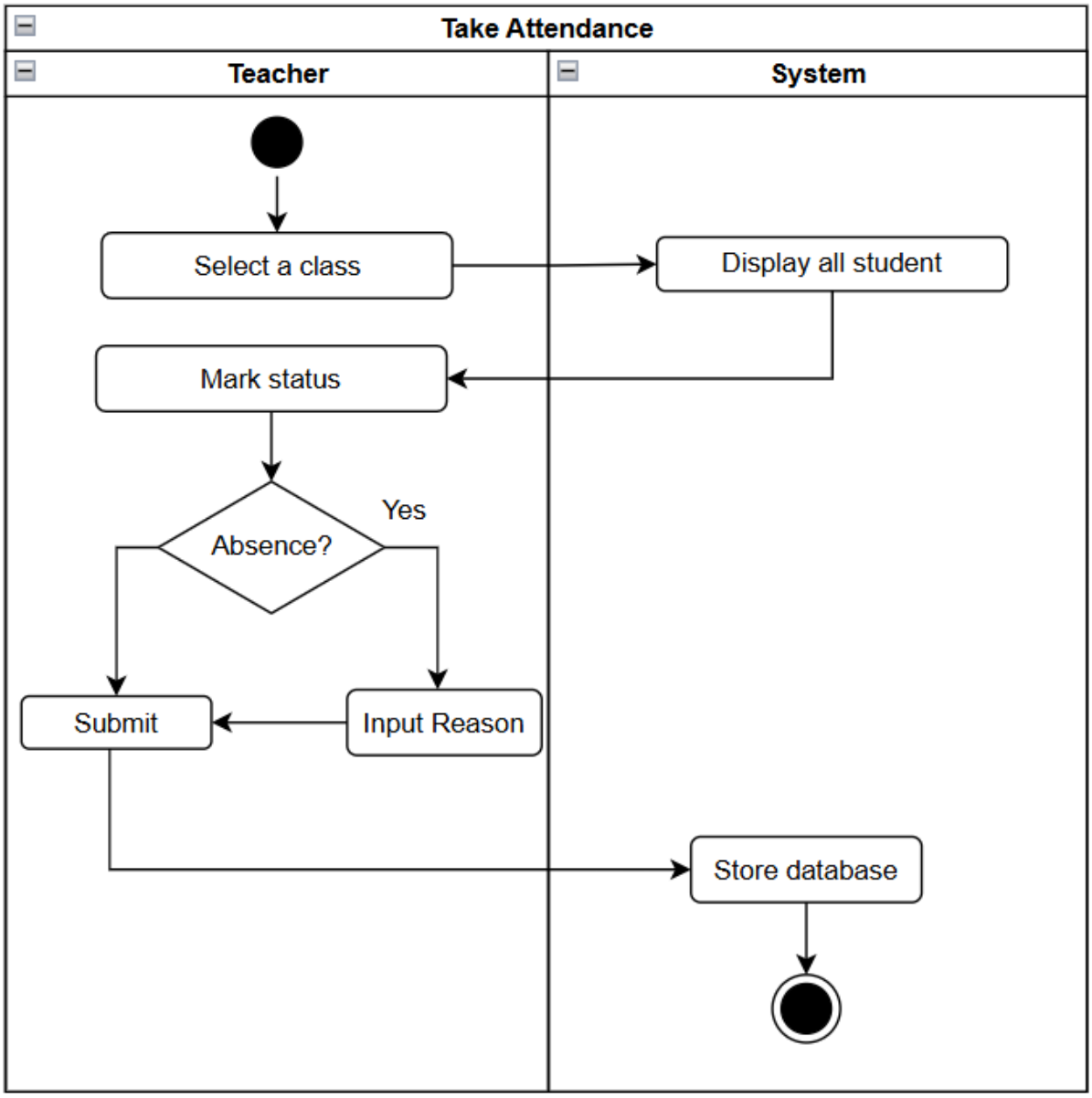


Figure 4.3.8: Take Attendance Activity Diagram

The process starts by selecting a class from the attendance page and the system lists out all students in the entire class. User marks the students' status as present or absence, if absence, user is allowed to enter the reason which is optional. After all attendance has been marked, user submit and system update the database.

4.3.9 View Attendance

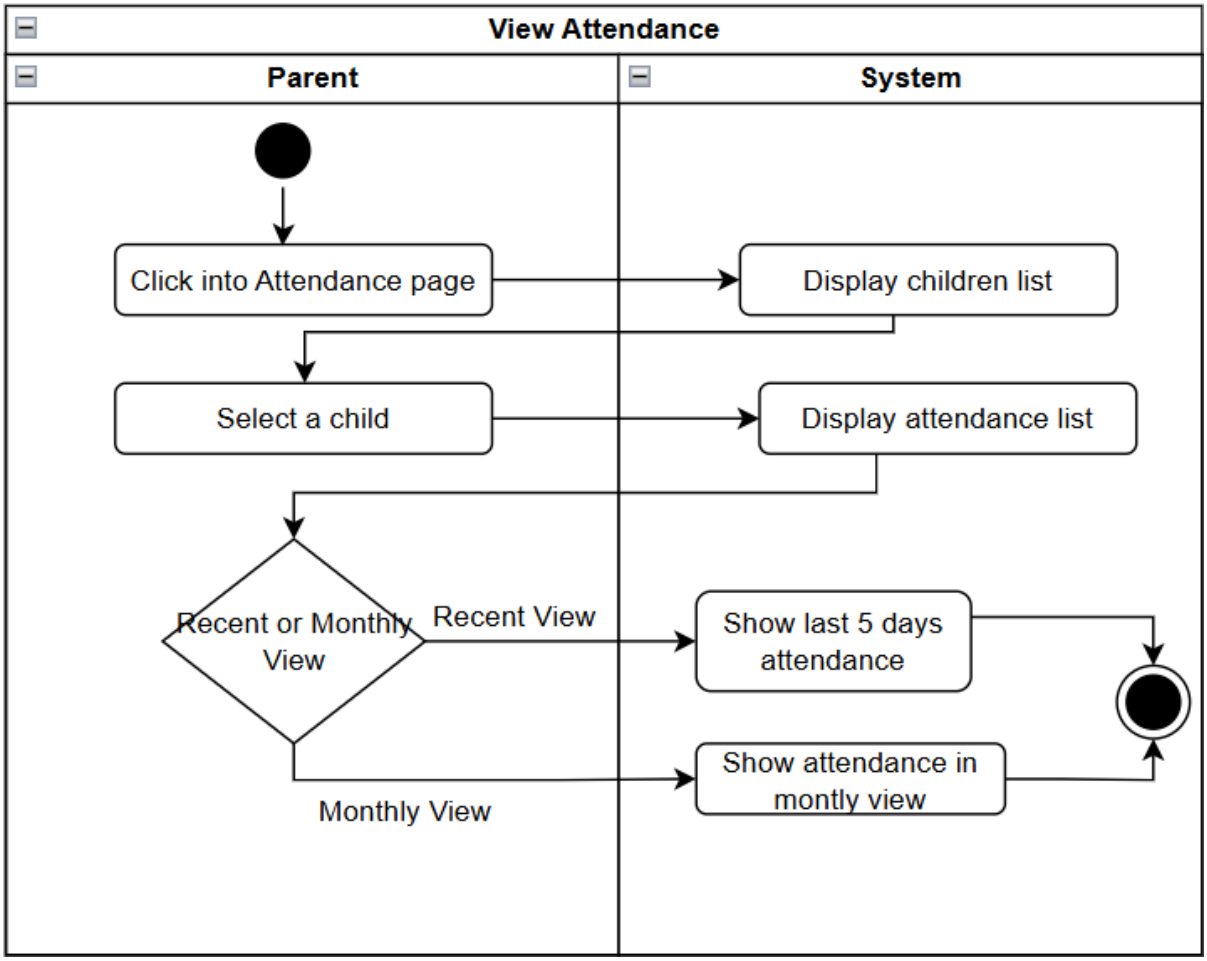


Figure 4.3.9: View Attendance Activity Diagram

The process starts when the user clicks on the Attendance page. The system displays a list of all children registered under the parent's account. The user selects a specific child and the system retrieves and displays the child's attendance records. The user can then choose between two viewing modes which are Recent View that displays the attendance records for the last five days and Monthly View that allows the user to select a specific month and the system shows the attendance records grouped by week for that month.

4.3.10 Apply Leave

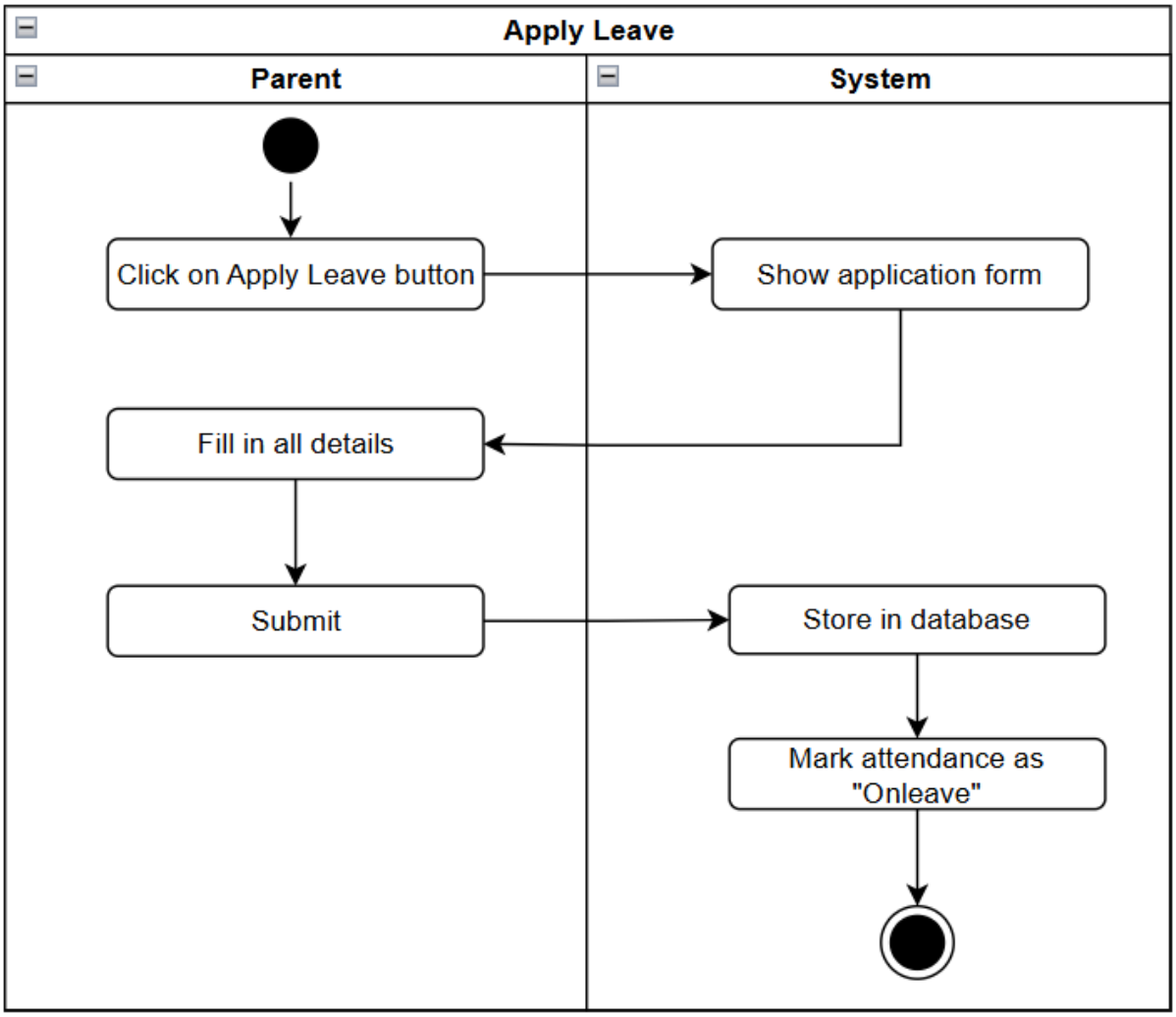


Figure 4.3.10: Apply Leave Activity Diagram

The process starts when the user clicks the Apply Leave button. The system displays the leave application form. The user is required to fill in all the necessary details, including the absence type, start and end dates and reason. Supporting documents can be uploaded optionally. Once the form is completed and submitted, the system stores the leave application in the database and automatically updates the attendance record by marking the selected days as On Leave.

4.3.11 View On leave List

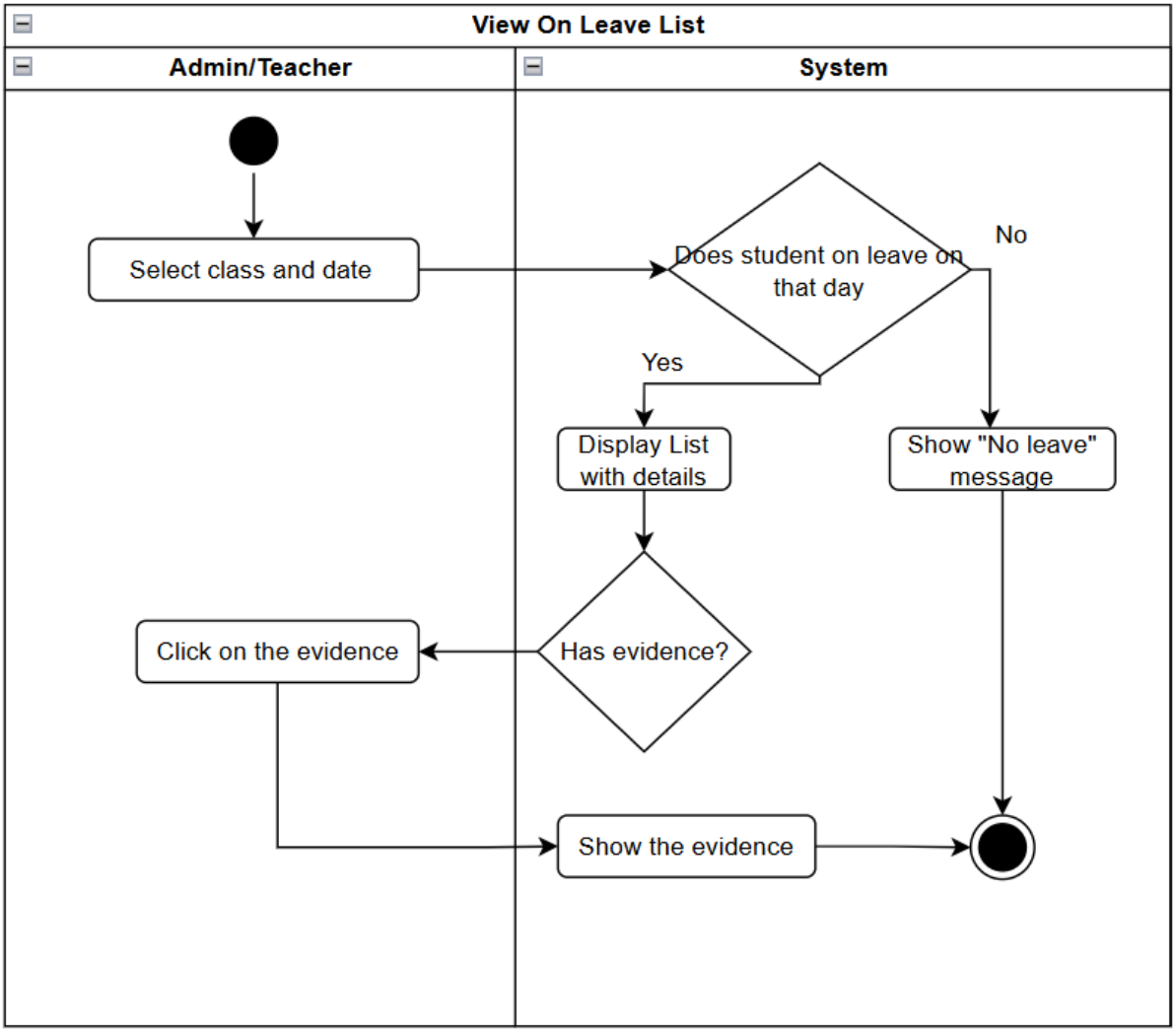


Figure 4.3.11: View On Leave List Activity Diagram

The process starts when the user is on the View Leave page. The user selects a specific date and class. The system then searches for students who are marked as On Leave on the selected date. If there are no students on leave, the system displays a message indicating that no students are on leave for the selected day. If there are students on leave, the system displays the list of students along with their leave details. If any supporting evidence was provided by the parent, the user can click on it and the system will display the evidence for review.

4.3.12 Post Announcement

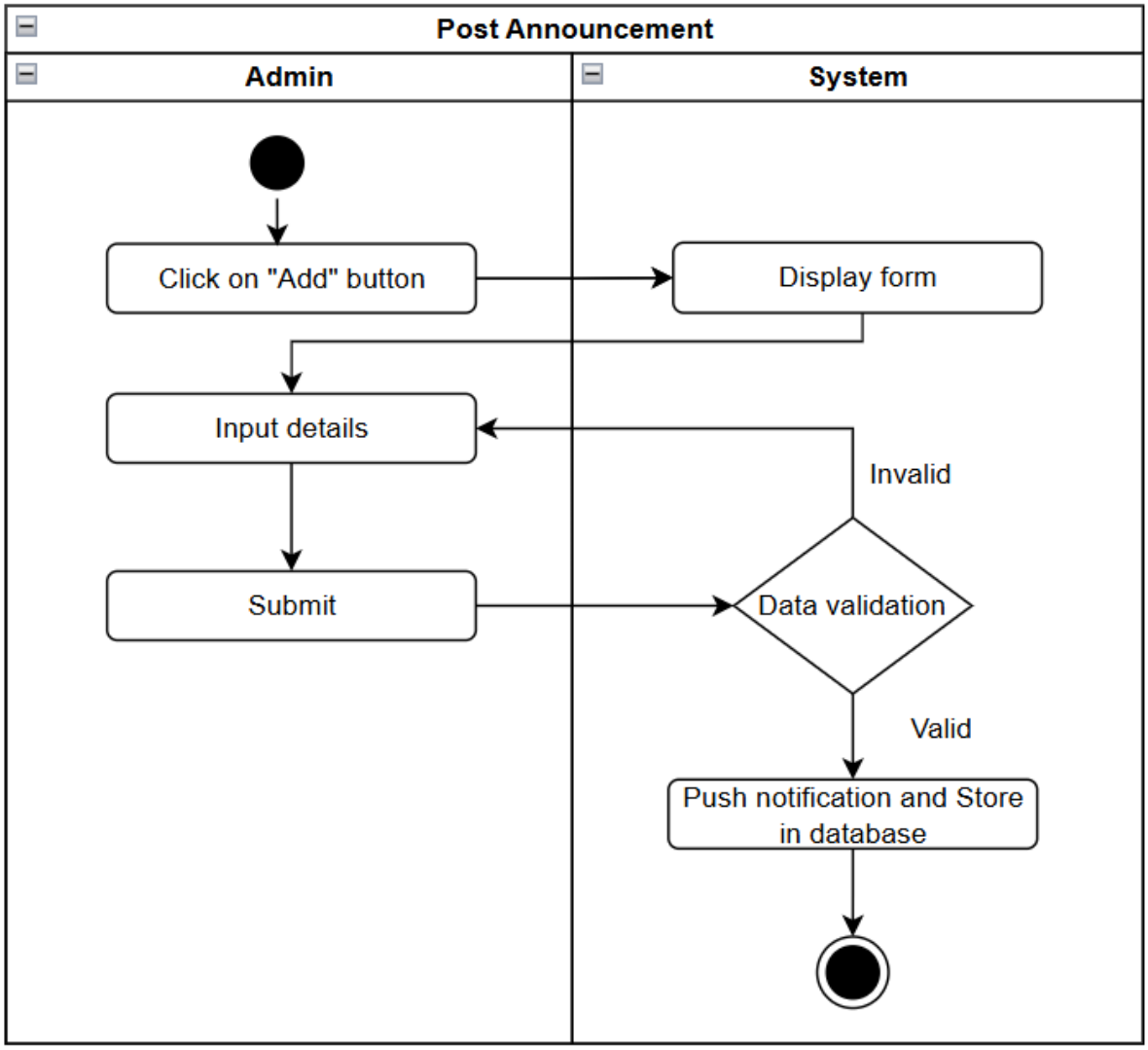


Figure 4.3.12 Post Announcement

The process starts when the user clicks the Add button. The system displays an announcement creation form where the user must input the details, including title, type, content and optionally upload an image. Once all details are filled in, the user clicks Submit. The system performs validation to ensure all required fields are correctly filled. If validation fails, the system prompts the user to correct the missing or invalid fields. If validation passes, the system saves the announcement data to the database and sends a push notification to all parents to notify them of the new announcement.

4.3.13 View Announcement

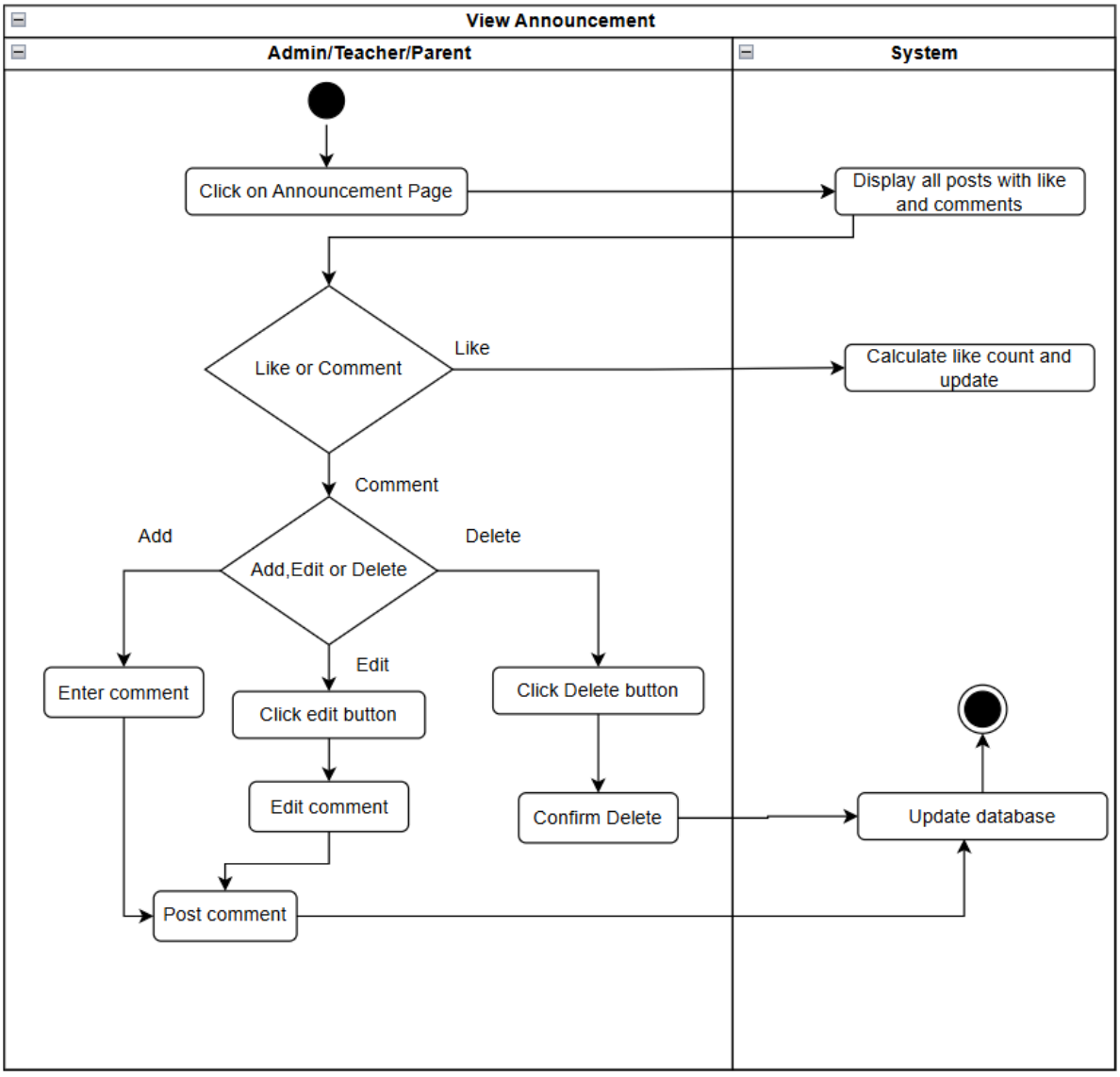


Figure 4.3.13 View Announcement Activity Diagram

The process starts when the user clicks on the Announcement page. The system displays all posts, including their like counts and comments. The user can perform the following actions which are Like a Post by clicking the like button then the system updates the like count in the database and reflects the change on the page. Add a Comment where the user types a comment and submits it and the system stores it in the database and displays it under the post. Besides, user can Edit a Comment by clicking the edit button that allows the user to modify their comment. Upon submission, the system updates the comment in the database and Delete a Comment by clicking the delete button prompts confirmation. Once confirmed, the system removes the comment from the database and updates the display.

4.3.14 Create Report

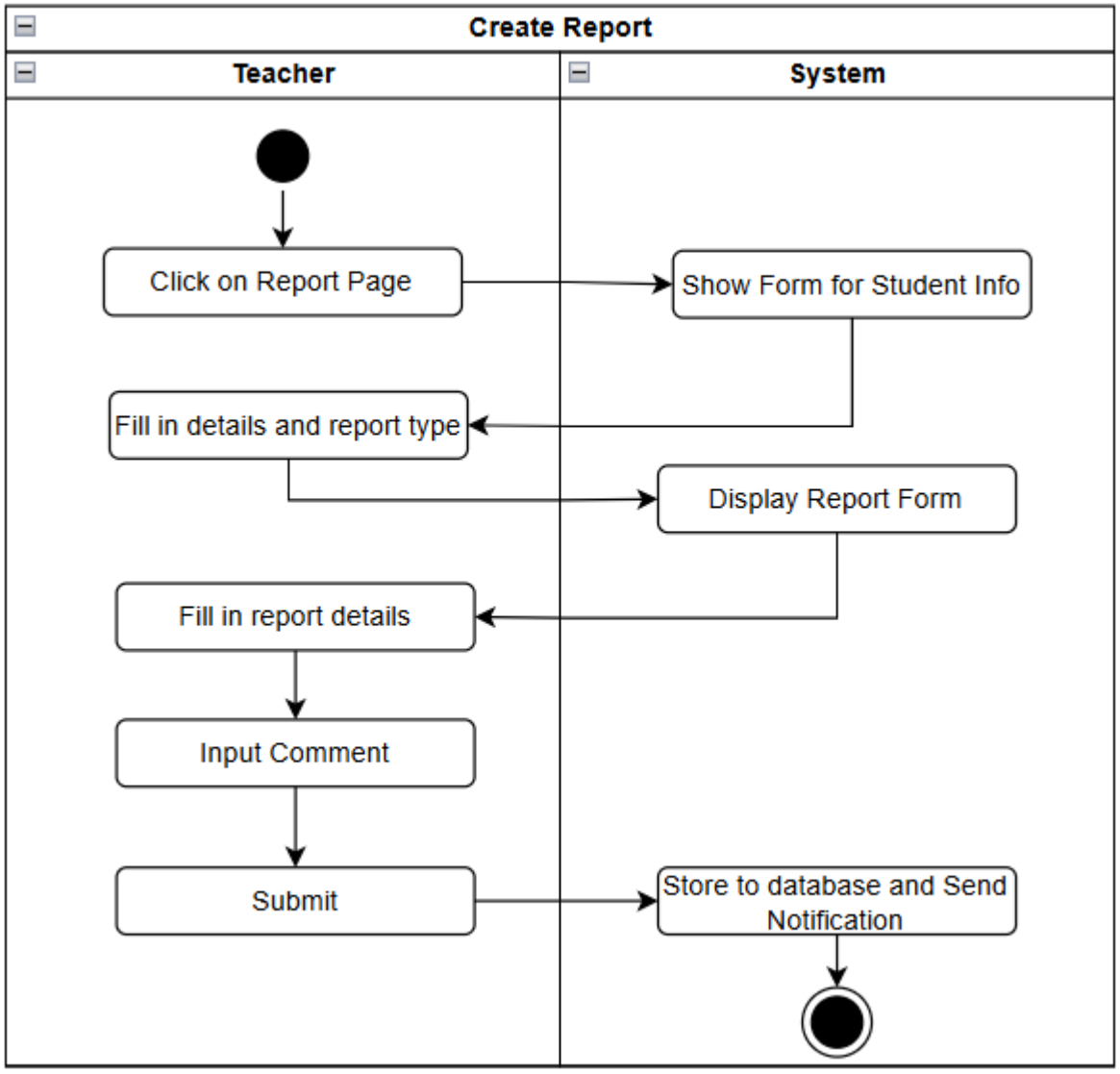


Figure 4.3.14 Create Report Activity Diagram

The process starts when the user clicks on the Report page and the system displays a form to input student information. The user enters the required details and selects a report type. Based on the selected report type, the system displays the corresponding form. The user fills in the report details, adds comments and submits the report. Once submitted, the system stores the data in the database and sends a push notification to the corresponding parent.

4.3.15 View Report

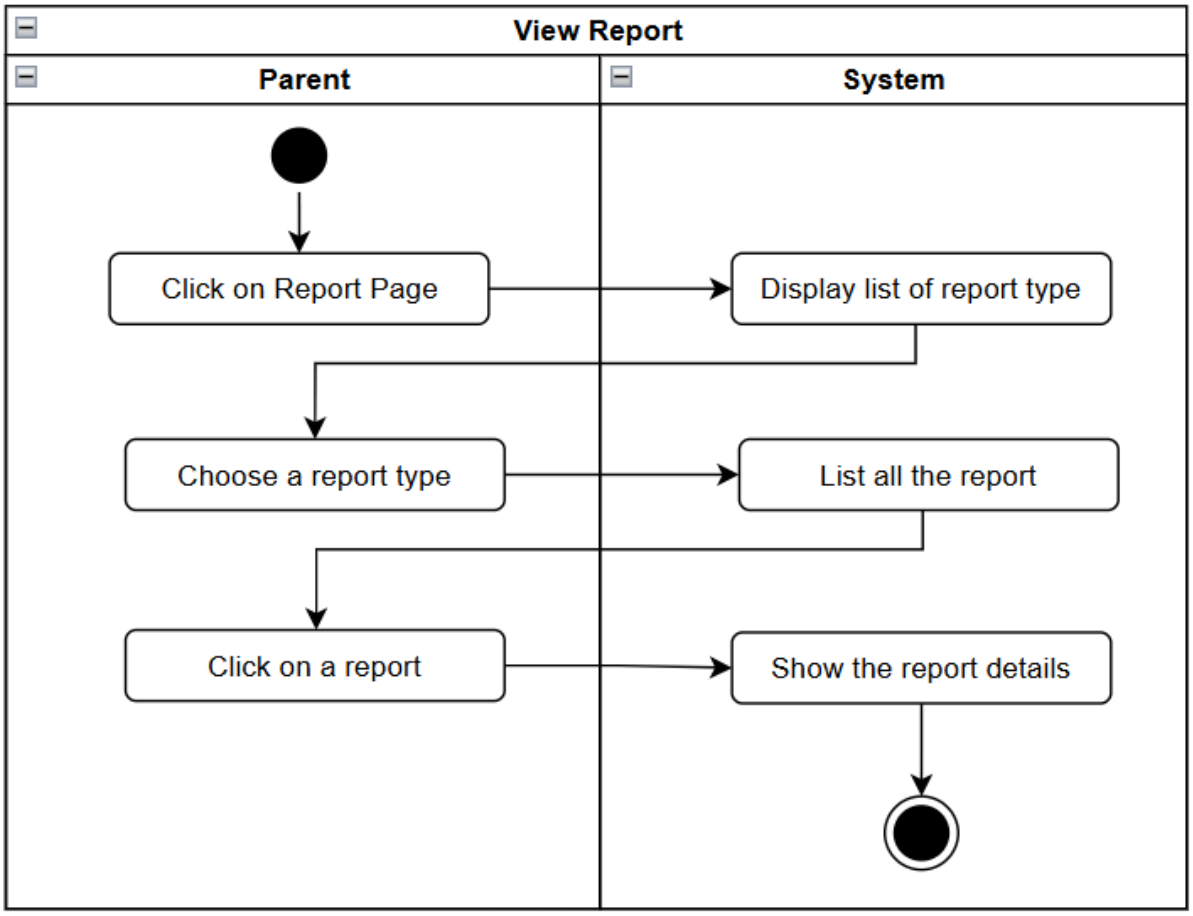


Figure 4.3.15: View Report Activity Diagram

The process starts when the user clicks on the Report page. The system displays a list of available report types. The user selects a report type and the system lists all reports related to the selected type. The user then clicks on a specific report and the system displays the detailed contents of that report.

4.3.16 Send and Receive Chat

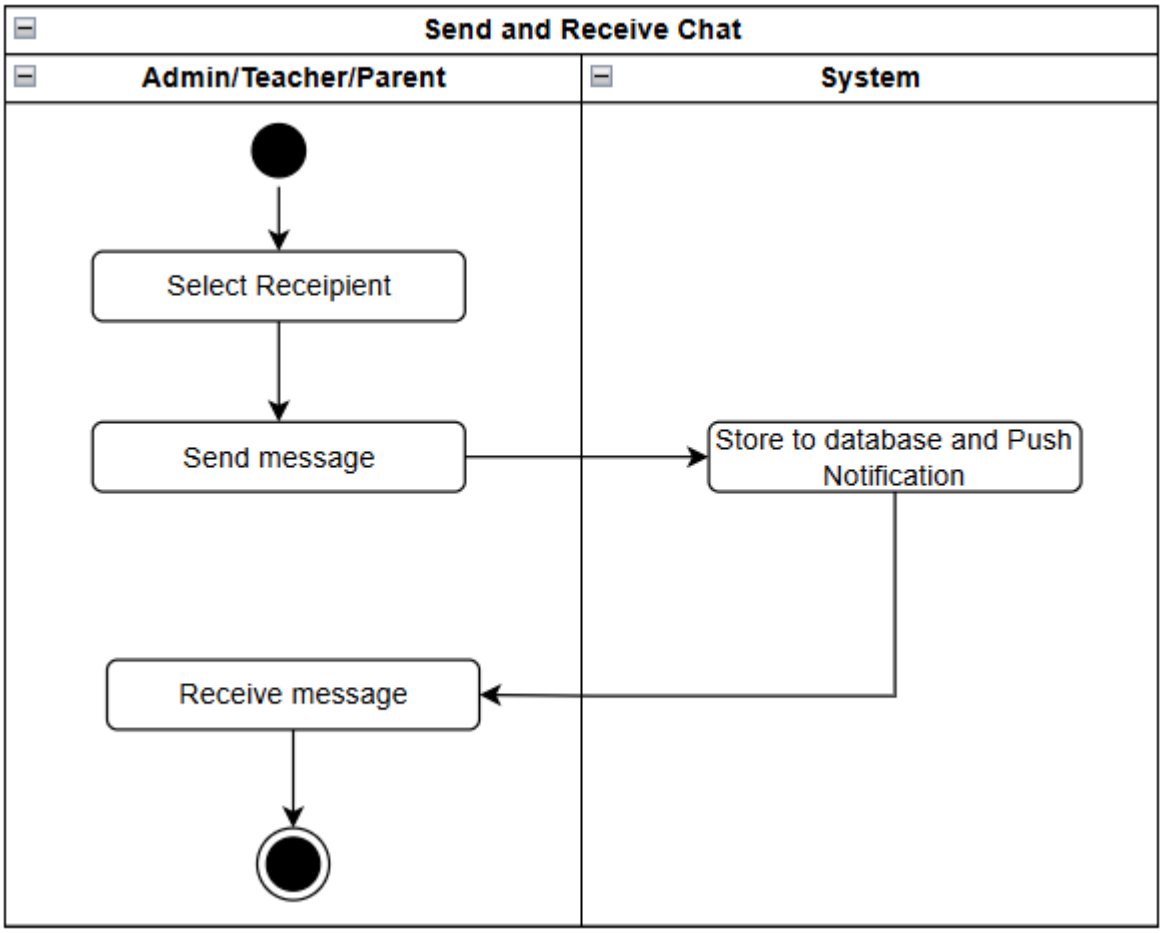


Figure 4.3.16: Send and Receive Chat Activity Diagram

The process starts when the user selects a recipient from the chat list. The user can send a message, which may include text, images or videos. Once the message is sent, the system pushes a notification to the recipient and stores the message in the database. The recipient then receives the message in real time.

4.3.17 Push Notification

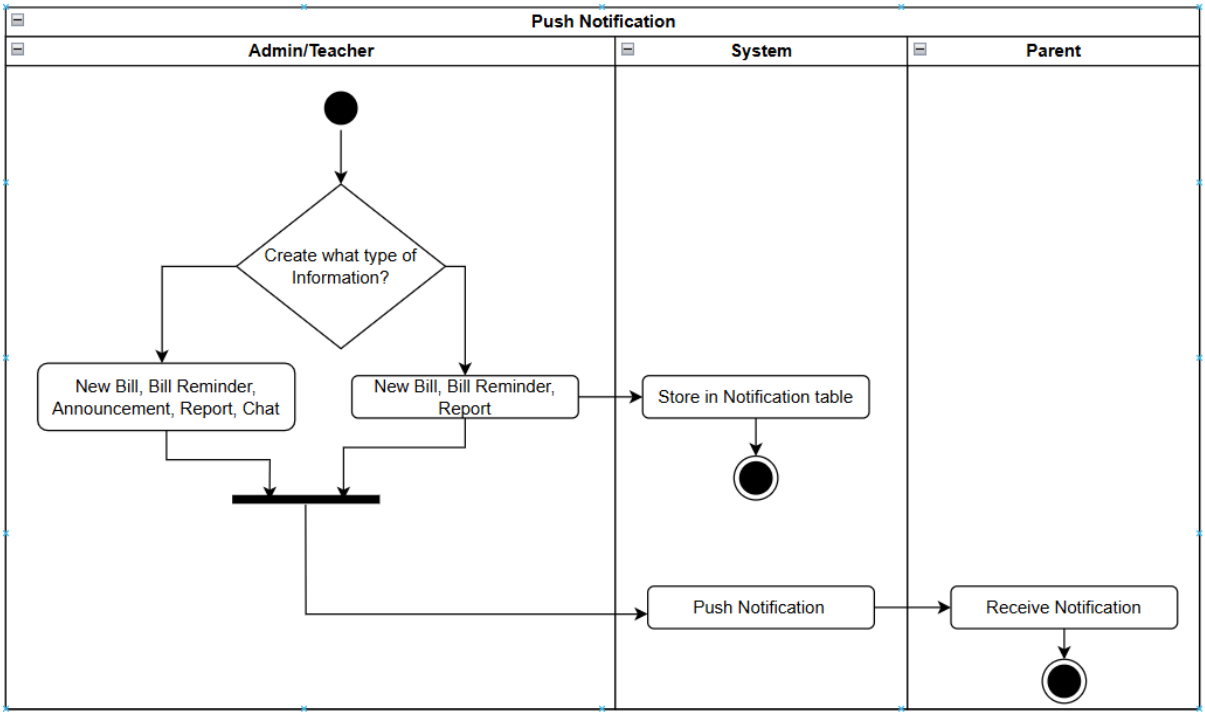


Figure 4.3.17: Push Notification Activity Diagram

The process starts after a user creates new information in the system. If a new bill, bill reminder, or report is created, the system pushes a notification to the targeted users and stores the notification in the database to allow users to track the notification history later. If a new announcement or chat message is created, the system pushes the notification to the recipient without storing it in the notification database.

4.3.18 View Notification

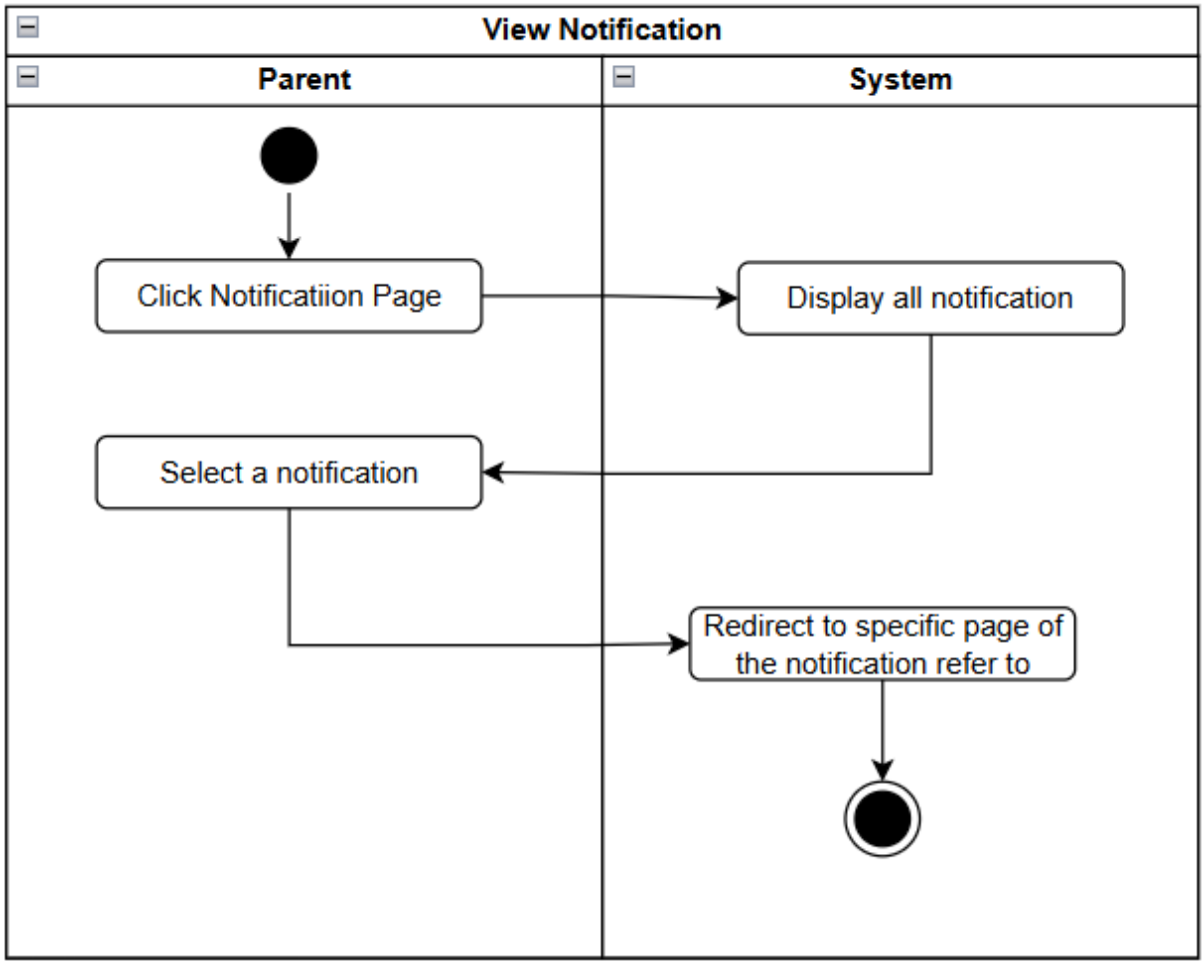


Figure 4.3.18: View Notification Activity Diagram

The process starts when the user clicks on the Notification Page. The system retrieves and displays all stored notifications for that user. The user can scroll through the list to view all available notifications. When the user selects a notification, the system redirects them to the corresponding page based on the reference of the notification.

Chapter 5
System Implementation

5.1 Hardware Tools

The hardware involved in this project includes a laptop and an Android mobile device. The laptop was used for the primary development tasks, including coding, system design, and testing of the kindergarten management system. The Android mobile device was used for testing and deploying the mobile application to ensure compatibility and functionality across different devices.

Description	Specifications
Model	Dell Inspiron 5502
Processor	Intel Core i7-1165G7
Operating System	Windows 11
Graphic	NVIDIA GeForce MX330 2GB
Memory	8GB DDR4 RAM
Storage	512GB SATA SSD

Table 5.1.1 Specifications of laptop

Description	Specifications
Model	Huawei Y9 Prime
Processor	HiSilicon Kirin 710F
Operating System	Android 10 with EMUI 10
Graphic	Mali-G51 MP4
Memory	4GB RAM
Storage	128GB

Table 5.1.2 Specifications of mobile device

5.2 Software Tools

A range of software tools were used to develop the mobile and web applications. Flutter and Dart was selected for mobile app development, while React supported by Node.js and npm was used for web application development. Visual Studio Code served as the main code editor that supports both Flutter and React projects. Firebase served as the backend service for real-time database management and user authentication. Furthermore, Stripe was integrated to handle secure payment transactions within the mobile application. Version control was managed by using Git to maintain code integrity and enable collaborative development. In addition, Cloudinary was implemented to handle image storage and delivery, while Pushy enabled real-time push notifications to keep users informed.

No.	Tool/Technology	Purpose
1	Flutter	Mobile application development for parents.
2	Dart SDK	Programming language used for Flutter app development.
3	Visual Studio Code	Code editor for developing mobile and web applications.
4	React	Web application development for administrators.
5	Node.js and npm	Runtime environment and package manager to run the React web app.
6	Firebase	Real-time database and user authentication
7	Stripe	Secure payment gateway integration for the mobile application.
8	Git	Version control to manage and track changes in source code.
9	Cloudinary	A service to store and deliver images
10	Pushy	Push notification to send real-time alerts to users' devices.

Table 5.2.1: List of Software Tools

5.3 Setting and Configuration

5.3.1. Development Environment Setup

```
name: kinder_app
description: "A new Flutter project."

publish_to: 'none' #
version: 1.0.0+1

environment:
  sdk: ^3.7.0

dependencies:
  flutter:
    sdk: flutter
  webview_flutter: ^4.4.2
  cupertino_icons: ^1.0.8
  firebase_core: ^3.12.1
  cloud_firestore: ^5.6.5
  firebase_auth: ^5.5.1
  intl: ^0.20.2
  pdf: ^3.11.3
  path_provider: ^2.1.5
  flutter_dotenv: ^5.1.0
  firebase_messaging: 15.2.4
  flutter_local_notifications: 19.0.0
  syncfusion_flutter_pdfviewer: 29.1.33
  open_filex: 4.7.0
  http: ^1.1.0
  url_launcher: ^6.1.14
  flutter_stripe:
  fl_chart: ^0.64.0
  image_picker: ^1.0.7
  cloudinary_public: ^0.23.1
  permission_handler: ^11.3.1
  device_info_plus: ^11.5.0
  collection: ^1.19.1
  pushy_flutter: ^2.0.38
```

Figure 5.3.1.1: Flutter Mobile App Configuration

Flutter was used to build mobile application for parents. Flutter SDK was installed and Visual Studio Code was used as the development environment. A Flutter project has been initialized and connected to Firebase.

```
{
  "name": "fyp-1",
  "private": true,
  "version": "0.0.0",
  "type": "module",
  ▷Debug
  "scripts": {
    "dev": "vite",
    "build": "tsc -b && vite build",
    "lint": "eslint .",
    "preview": "vite preview",
    "server": "node server/index.js"
  },
  "dependencies": {
    "@cloudinary/react": "^1.14.3",
    "@cloudinary/url-gen": "^1.21.0",
    "@emotion/react": "^11.14.0",
    "@emotion/styled": "^11.14.0",
    "@mui/icons-material": "^7.1.1",
    "@mui/material": "^7.1.1",
    "@mui/x-date-pickers": "^8.7.0",
    "@radix-ui/react-label": "^2.1.7",
    "@radix-ui/react-select": "^2.2.5",
    "@radix-ui/react-slot": "^1.2.3",
    "@radix-ui/react-tabs": "^1.1.12",
    "axios": "^1.11.0",
    "bootstrap": "^5.3.3",
    "bootstrap-icons": "^1.13.1",
    "cors": "^2.8.5",
    "dayjs": "^1.11.13",
    "dotenv": "^17.2.1",
    "express": "^5.1.0",
    "firebase": "^11.9.0",
    "firebase-admin": "^13.4.0",
    "lucide-react": "^0.475.0",
    "pushy": "^4.0.2",
    "react": "^19.0.0",
    "react-dom": "^19.0.0",
    "react-router-dom": "^7.2.0",
    "recharts": "^3.0.2"
  },
  "devDependencies": {
    "@eslint/js": "^9.19.0",
    "@types/express": "^5.0.3",
    "@types/node": "^24.3.0",
    "@types/react": "^19.0.8",
    "@types/react-dom": "^19.0.3",
    "@vitejs/plugin-react": "^4.3.4",
    "eslint": "^9.19.0",
    "eslint-plugin-react-hooks": "^5.0.0",
    "eslint-plugin-react-refresh": "^0.4.18",
    "globals": "^15.14.0",
    "typescript": "~5.7.2",
    "typescript-eslint": "^8.22.0",
    "vite": "^6.1.0"
  }
```

Figure 5.3.1.2: React Web App Configuration

React was used to build web application for administrators and teachers. The project was initiated using the create-react-app template for rapid setup. Firebase was integrated to manage real-time data handling. This web platform was a control panel for admin and teachers to interact with and update records in Firebase which were reflected in real time across the mobile application.

5.3.2 IDE Configuration

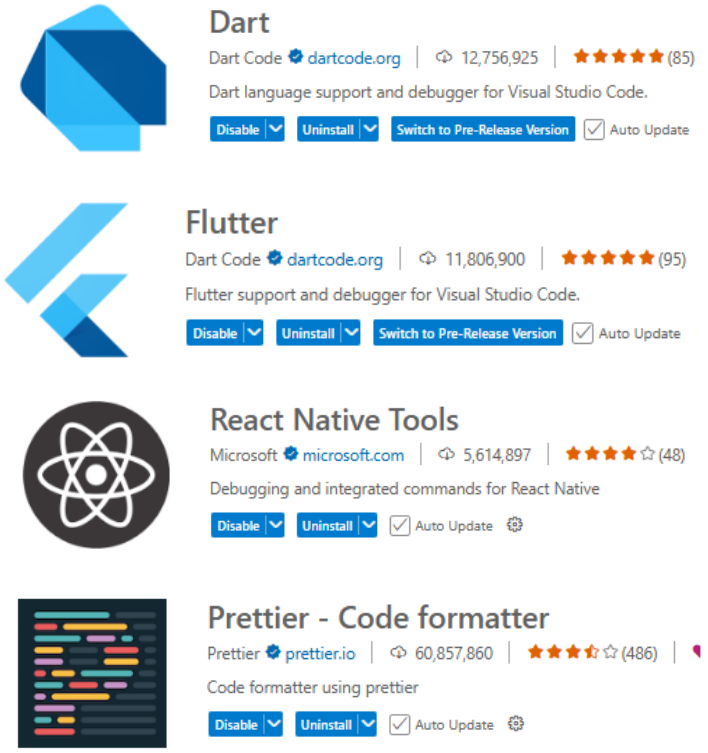


Figure 5.3.2.1: VS Code configuration

Several configuration settings were applied in Visual Studio Code to ensure smooth development. The Flutter and Dart plugins were enabled to support mobile app development and essential extensions such as Flutter, Dart, React Developer Tools and Prettier were installed to enhance productivity and maintain clean code formatting.

5.3.3 Firebase Configuration

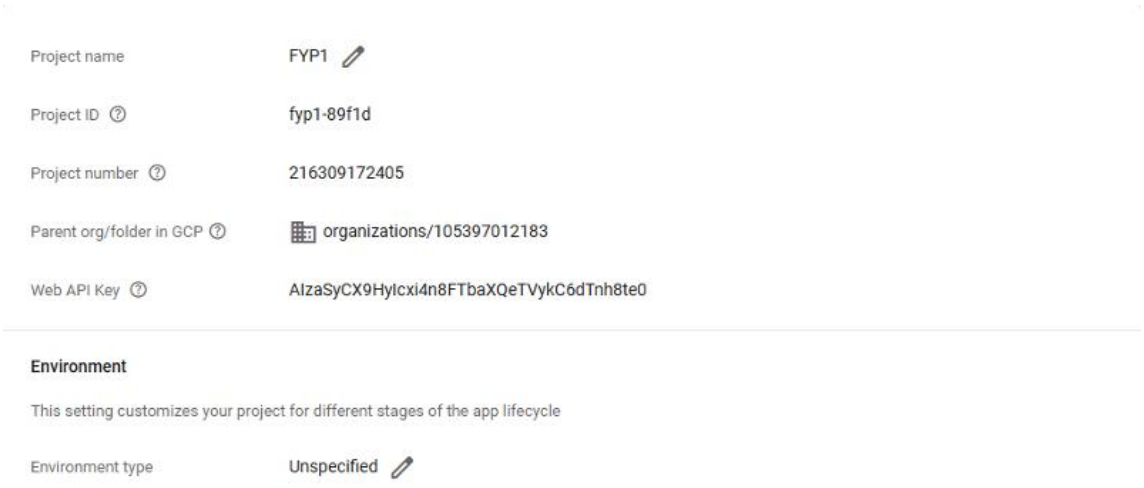


Figure 5.3.3.1: Details of Firebase Project Configuration

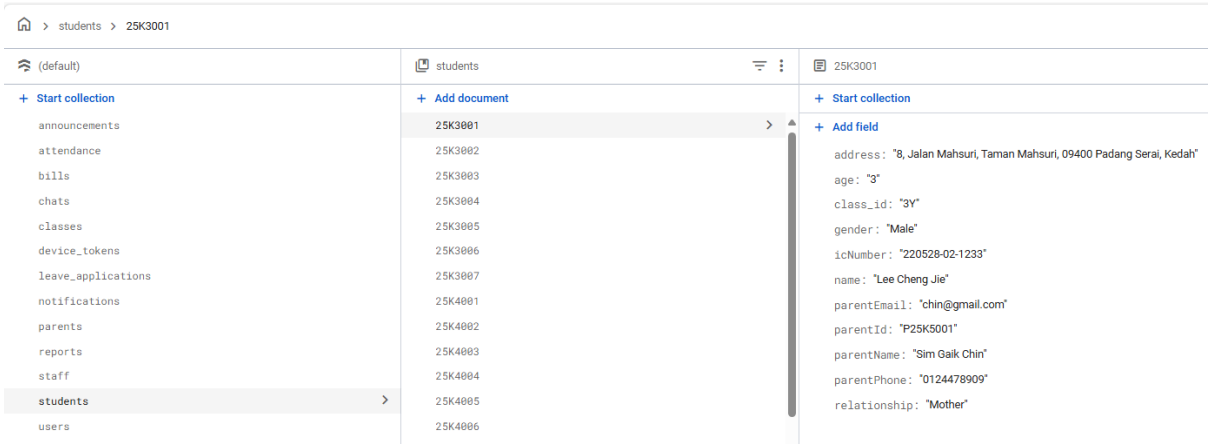


Figure 5.3.3.2: Students collection in Firestore Database

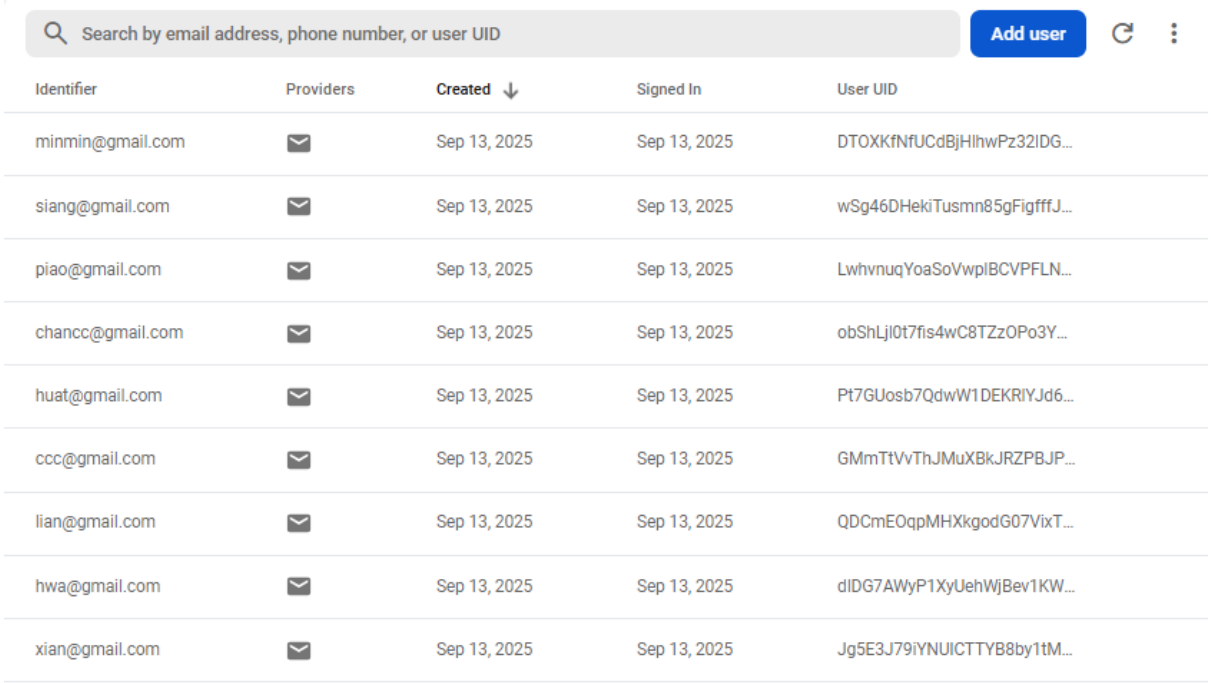


Figure 5.3.3.3: Firebase Authentication

Firebase was used as the backend service for database management and authentication. A Firebase project was created through the Firebase Console. The Firestore Databse was enabled to store and manage real-time data such as student records, billing information, and payment status. Collections were created for key modules such as users, attendance, payments and reports. For authentication, email/password authentication was enabled to allow secure access for parents, administrators and teachers. The Firebase SDK was successfully integrated into both the Flutter and React projects to enable seamless communication between frontend and backend.

5.3.4. Stripe Setup

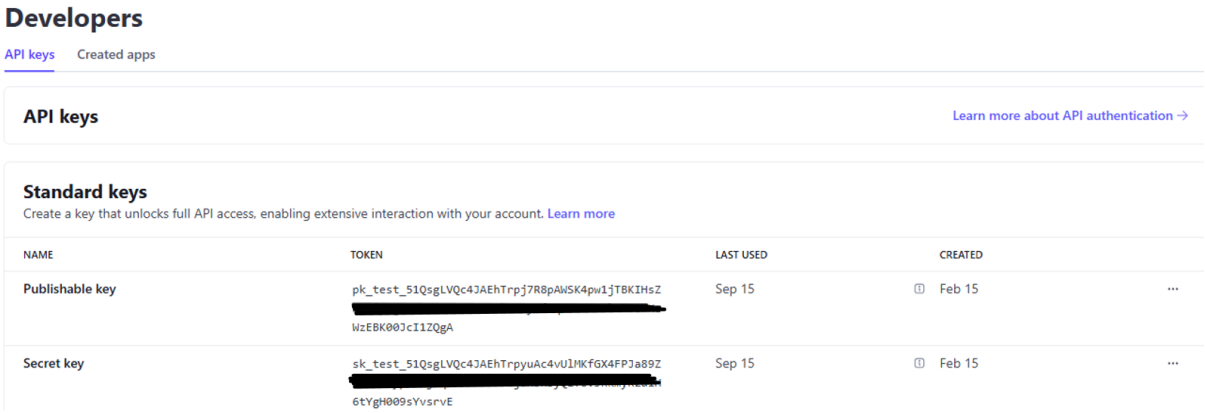


Figure 5.3.4.1: Stripe API Keys in Sandbox

```
var request = http.Request(
  'POST',
  Uri.parse("https://api.stripe.com/v1/payment_intents"),
);
request.headers.addAll({
  "Authorization": "Bearer ${dotenv.env['STRIPE_SECRET_KEY']}",
  "Content-Type": "application/x-www-form-urlencoded",
});
```

Figure 3.3.4.2: Stripe API Keys Integration in Flutter

Stripe account was registered to enable online payment processing. The publishable and secret keys were securely integrated into the Flutter. During the development phase, the payment flow was implemented and tested in Stripe's test mode to ensure seamless functionality and error handling.

5.3.5. Cloudinary Setup

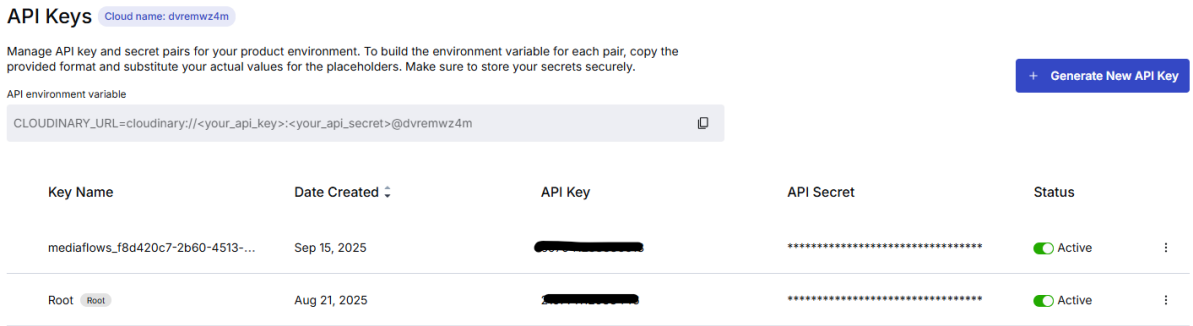


Figure 5.3.5.1: Cloudinary API Key

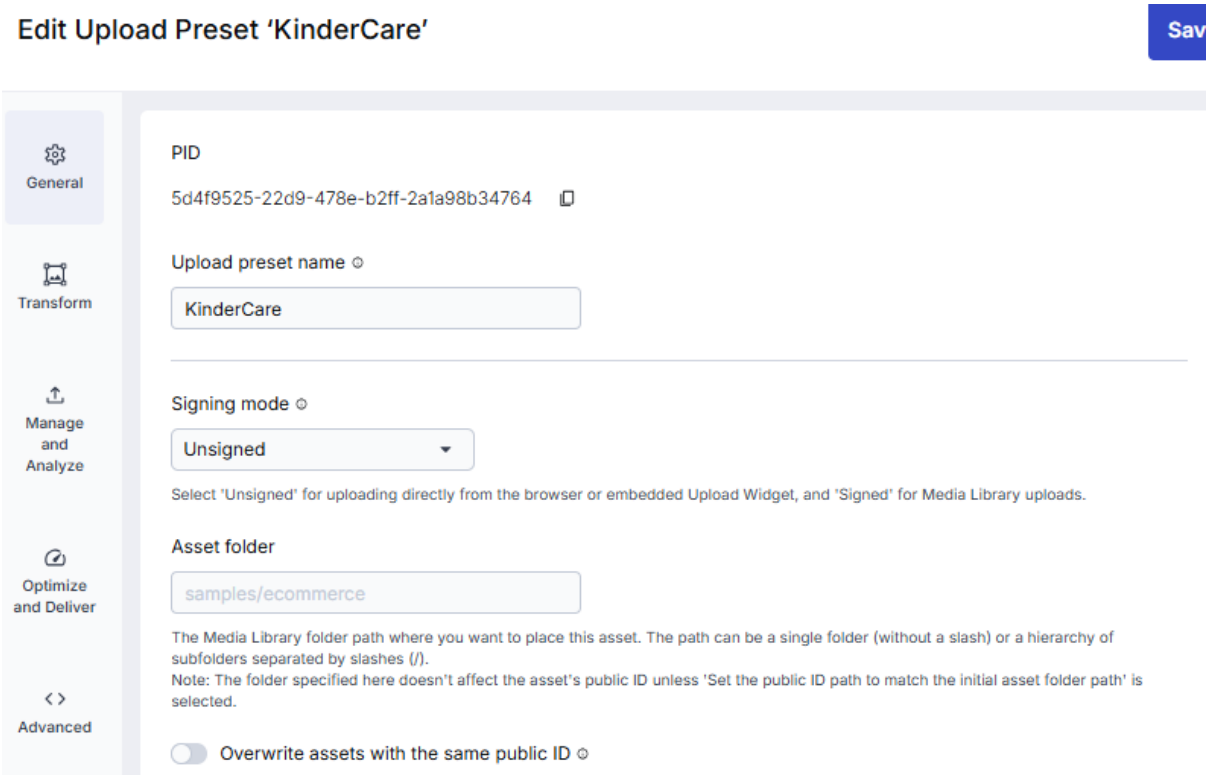


Figure 5.3.5.2: Cloudinary Upload Preset Configuration

Cloudinary account was created and configured to manage image uploads and delivery. API keys were generated and image upload presets were created in the Cloudinary dashboard. The Cloudinary API was then integrated into both mobile and web applications to allow users to upload images and fetch them in optimized formats.

5.3.6 Pushy Notification Setup

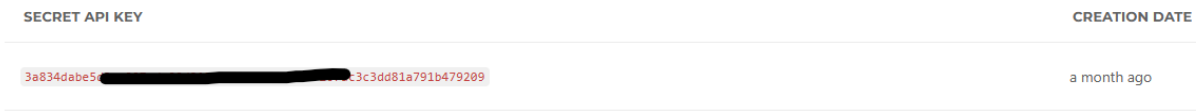


Figure 3.3.6.1: Pushy API Key

Figure 5.3.6.1: Pushy Active Device Token

Push notifications were implemented using Pushy to keep users informed in real time. The Pushy SDK was integrated into the Flutter mobile application to enable device registration and receive notifications. API keys were configured on the backend to ensure secure communication between the server and Pushy's notification service.

5.4 System Operation

5.4.1 Web Application Operation

Login

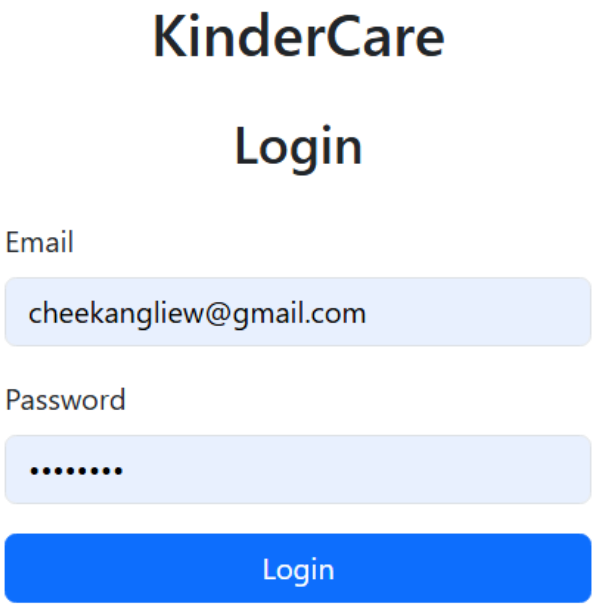


Figure 5.4.1.1: Login Page

The login page allows users to login using email and password. This ensures that only registered users can access the application. The account is created by admin during the registration of new staff.

Dashboard and Navigation Bar

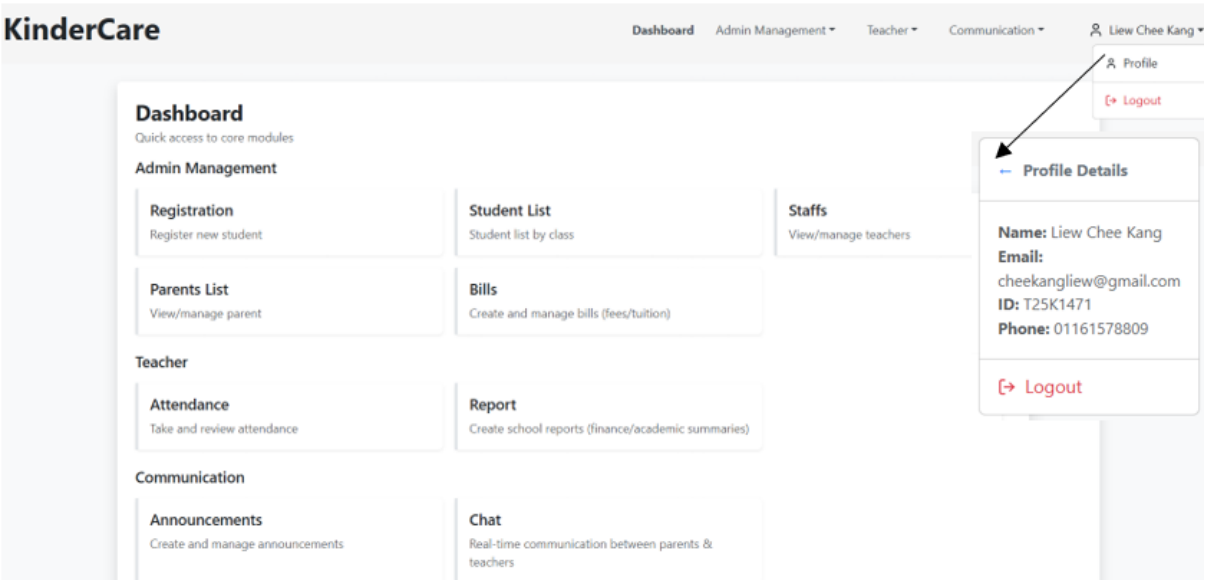


Figure 5.4.1.2: Dashboard and Navigation Bar

The dashboard and navigation bar allow the users to easily access all main pages of the system. At the top right corner, the current user's name is displayed for quick identification. When clicked, a dropdown menu appears with options **Profile** and **Logout**. When **Profile** is selected, it displays the user's details, including name, email, id and phone number, while selecting **Logout** will end the user session and redirects back to the login page.

Registration

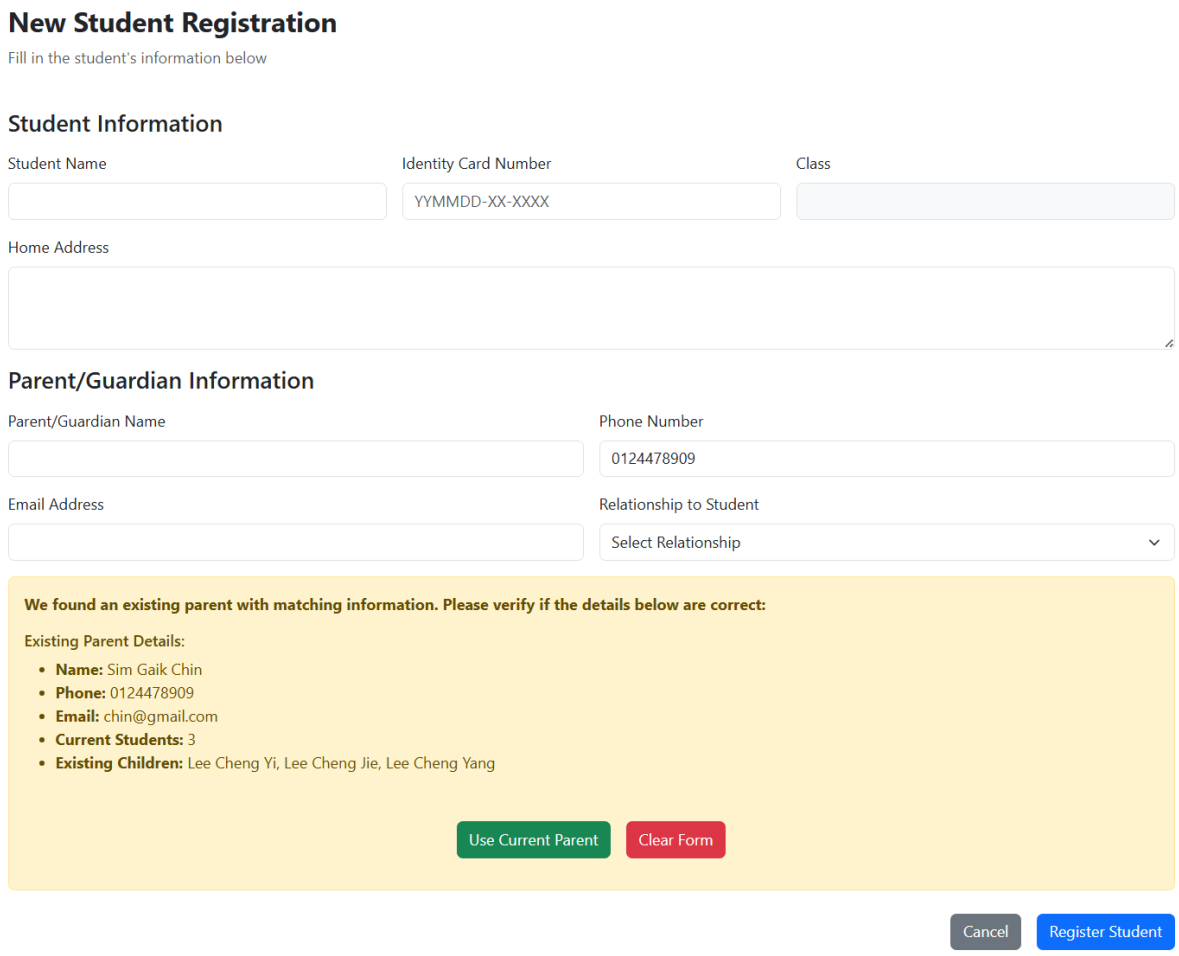


Figure 5.4.1.3: Student Registration Page

The student registration allows the r to register new students into the system. The form requires all student and parent details to be completed before submission. The system automatically assigns the class based on the student's IC number to reduce manual errors.

If the system detects that the parent's details already exist in the database, an alert is displayed to prompt the administrator to verify whether there is a typing error or if the existing parent record should be reused. If the admin chooses to reuse the existing record, the parent information fields are automatically filled. Otherwise, the form will be cleared and the admin needs to re-enter the information. When the registration is successful, the system automatically

generates a unique parent ID and creates a new parent account using the provided email address.

Student List

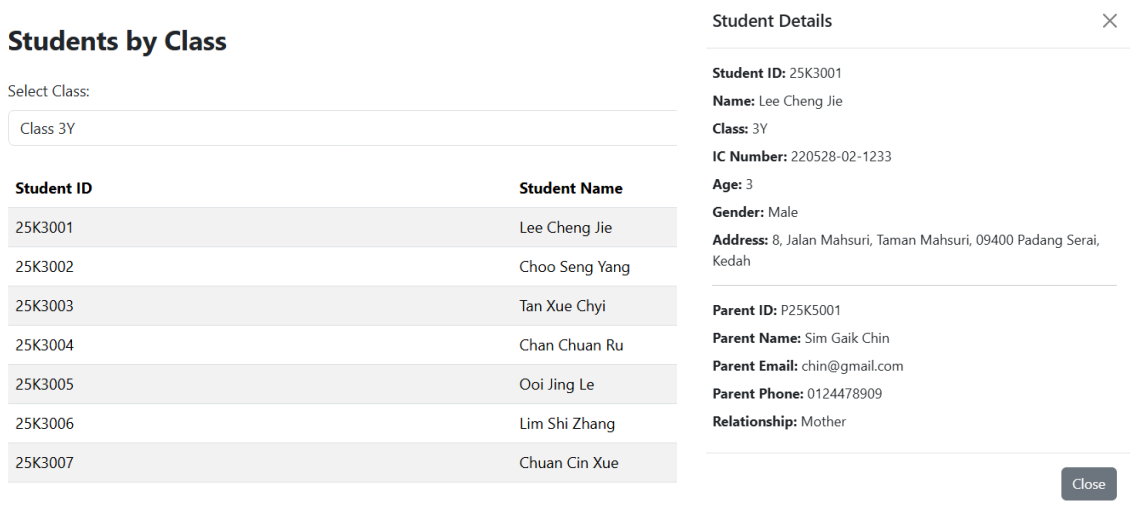


Figure 5.4.1.4: Student List Page

The Student List page allows admin to view all registered students in the system. Students are grouped by their assigned classes for easier navigation. When a student is selected, a dialog box appears and displays all the students and parents' information.

Parent List

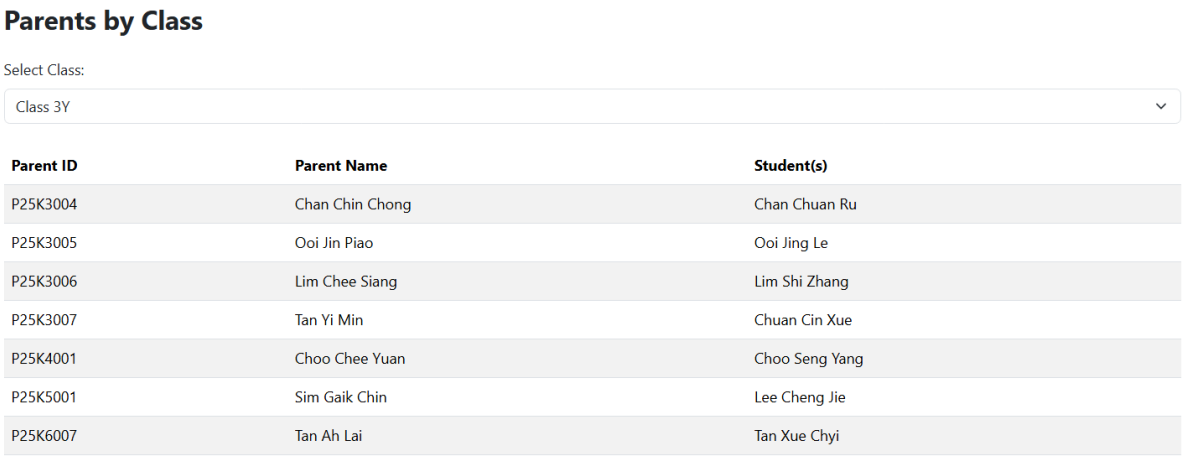


Figure 5.4.1.5: Student List Page

The Parent List page allows admin to view all parents' information in the system. Parents are grouped by their children assigned classes for easier navigation. When a parent is selected, a dialog box appears and displays all the parents' information.

Staff List

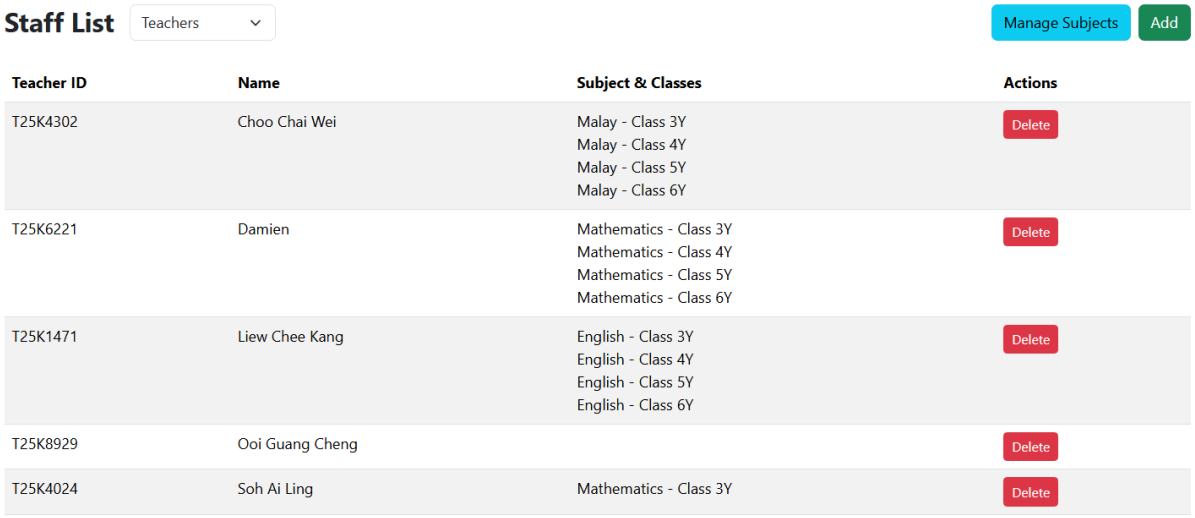


Figure 5.4.1.6: Staff List Page

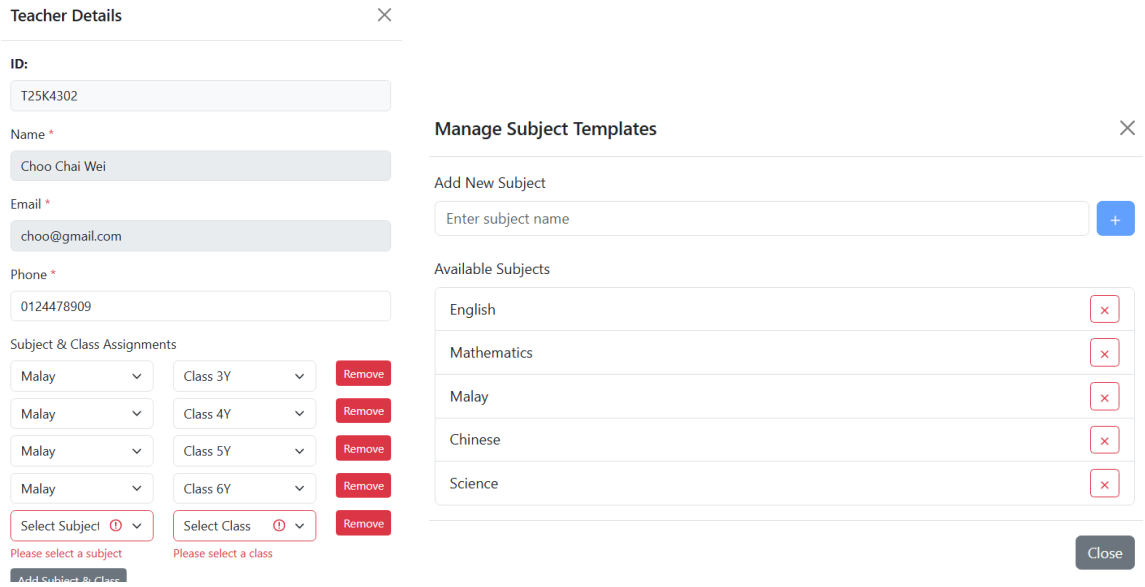


Figure 5.4.1.7: Subject Assignment

The staff list allows the admin to view and manage all the teachers in the system. Admin can add new staff, edit existing staff information or delete staff records when necessary. Additionally, admins can assign teachers to specific subjects and classes under the teacher's details dialog. There is also a manage subjects' template that allows admin to add or remove available subjects. This ensures that subject names are consistent across the system and eliminates the need for manual typing when assigning subjects to teachers.

Bill Creation

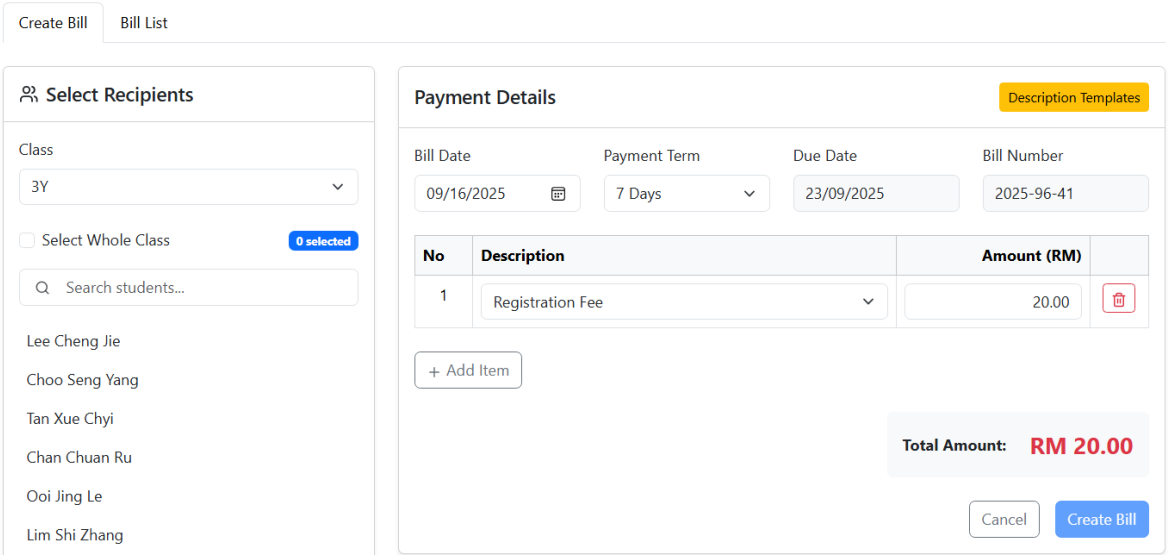


Figure 5.4.1.8: Bill Creation Page

The Bill Creation page allows admin to create and issue new bills for students. The recipient will be listed out by class and admin can select specific students or apply the bill to an entire class with a single action. After that, admin needs to fill in all the payment details. The bill date is automatically set to the current date, while the due date is calculated based on the payment term defined. The description of each payment can be selected from a dropdown list of payment items to reduce manual typing and ensure uniform naming conventions. After successful bill creation, the system automatically triggers **a push notification** to all selected recipients to inform them about the new bill.

Bill List

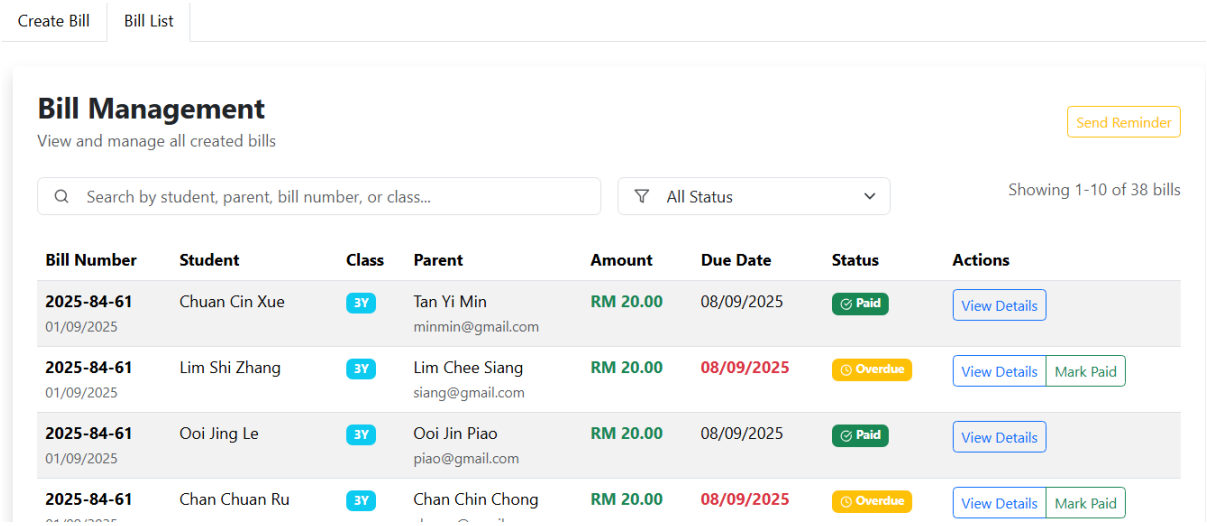


Figure 5.4.1.9: Bill List Page

The Bill List page displays all bills created by the admins to provide a centralized view of payment records. Each bill includes key information such as bill number, amount, issue date, due date, and payment status. Admin can manually mark the payment status if the bill is paid outside of the mobile application. Admin can quickly filter the list by All, Paid, Unpaid, or Overdue status to identify outstanding payments. This makes it easy to monitor payment progress and follow up on overdue accounts. A Send Reminder button is available to trigger push notifications for unpaid bills. When clicked, the system automatically identifies all unpaid bills that are due within the next three days and sends a payment reminder notification to the respective parents.

Attendance Overview

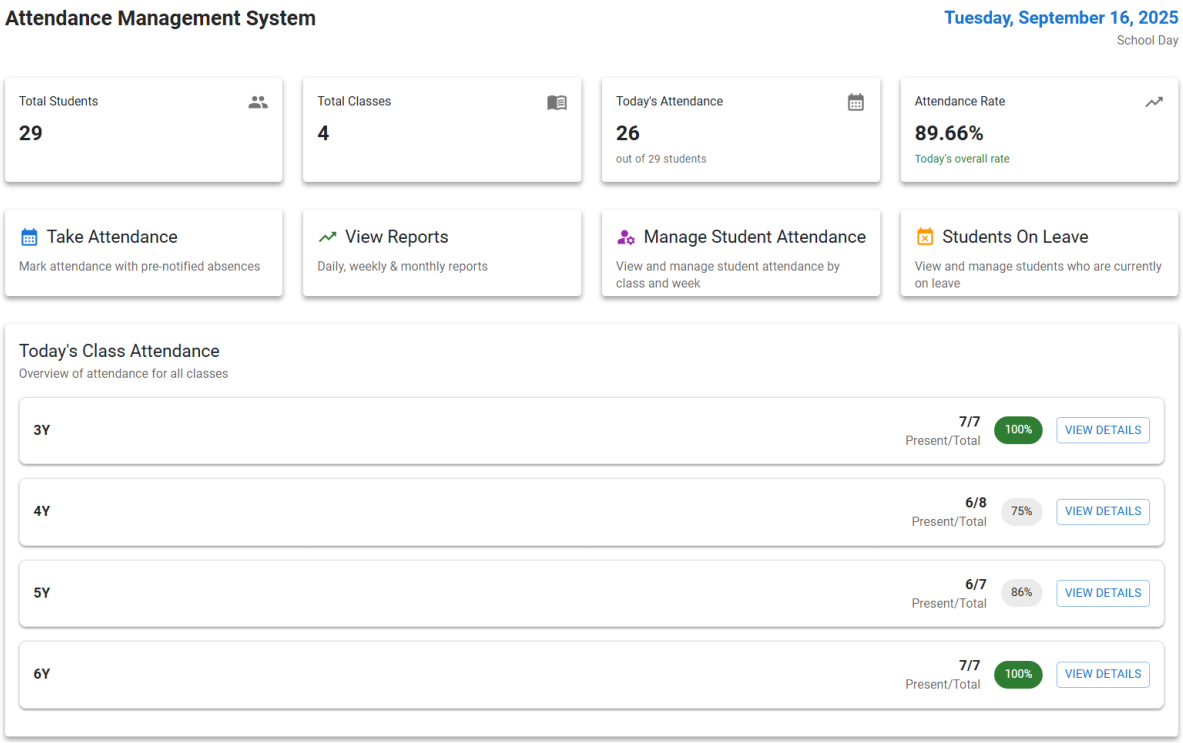


Figure 5.4.1.10: Attendance Overview Page

The Attendance Overview page displays the status and percentage of student attendance for the current day. From this page, teachers can navigate to related features such as Take Attendance, View Attendance Reports, View Student Attendance, and View Student On Leave Status

Take Attendance

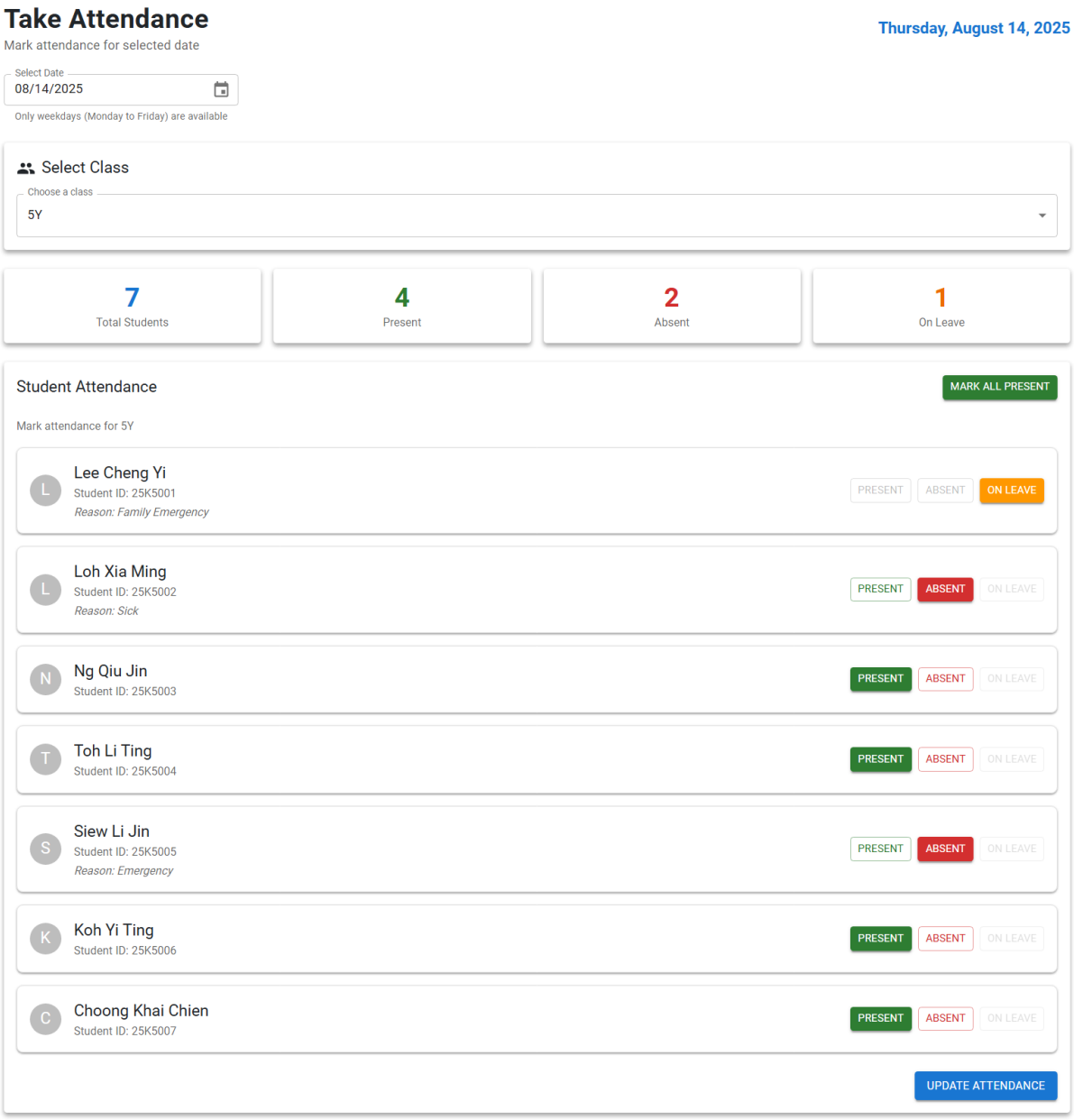


Figure 5.4.1.11: Take Attendance Page

The Take Attendance page allows teachers to record students' daily attendance. The date is automatically set to current date and teachers must first select a class to display the list of students. The "On Leave" status is pre-marked by the system for students who have applied for leave on that day and cannot be modified by the teacher. Teachers can mark each student as Present or Absent.

There is a "Mark All Present" button which allow teacher to mark all students as present with a single click. When a student is marked as Absent, the teacher has the option to enter a reason for the absence. Both leave reasons and absence reasons are displayed on the page for

reference. Teachers can update attendance records if any errors are found. Once all markings are completed, they must click Save or Update Attendance to store the data.

Attendance Report

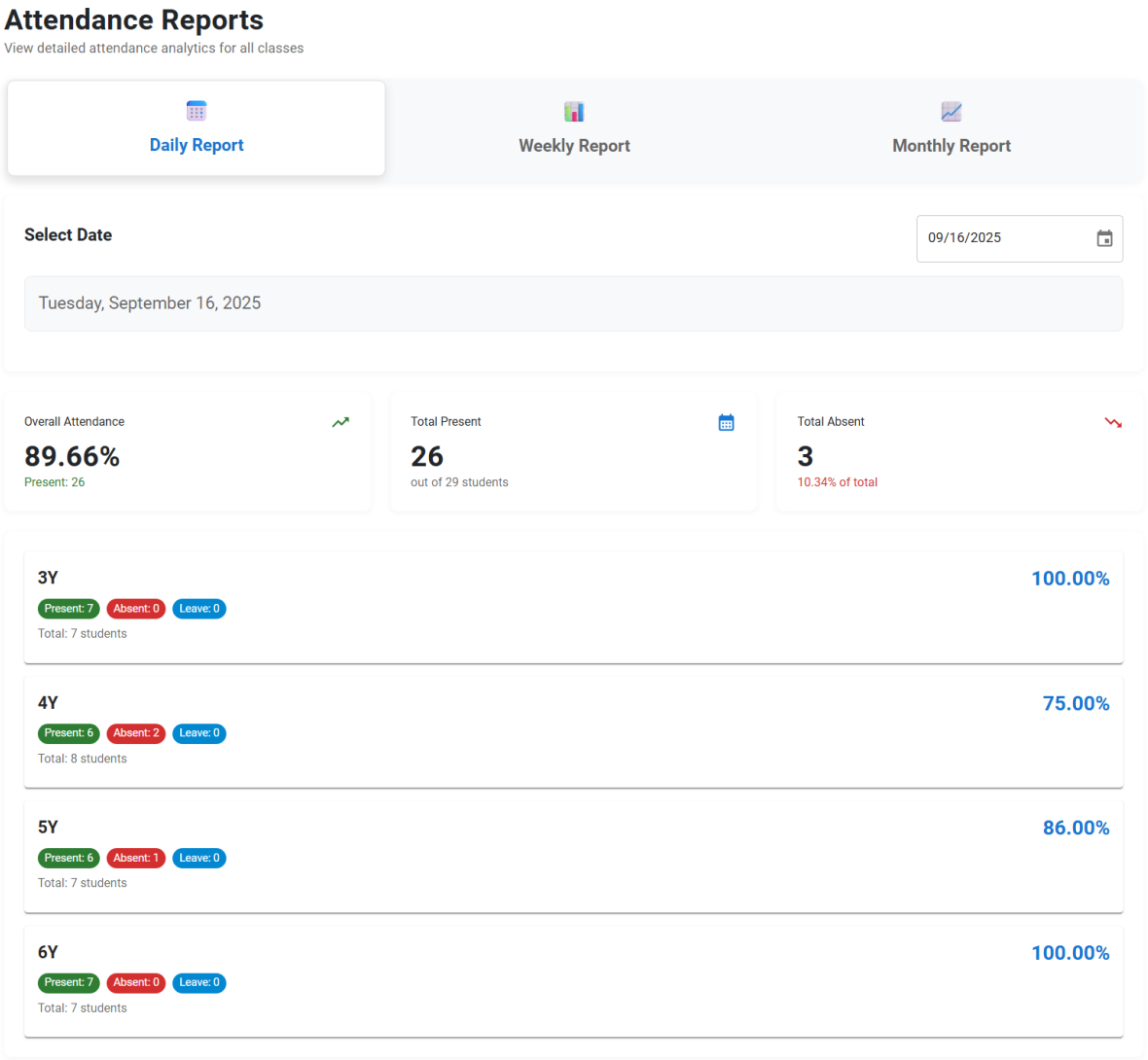


Figure 5.4.1.12: Attendance Report Page

The Attendance Report page allows teachers to review past attendance records in a structured and organized way. Reports can be generated based on Daily, Weekly, or Monthly views, enabling teachers to monitor attendance trends over time.

View Student Attendance

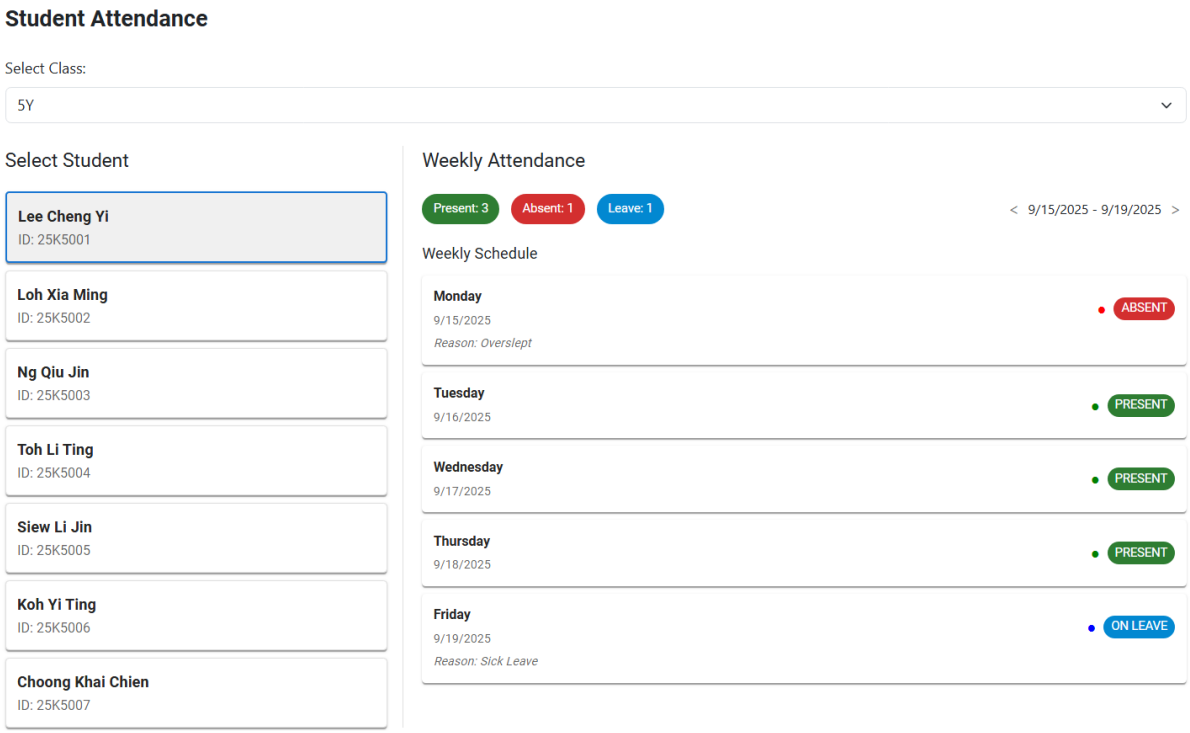


Figure 5.4.1.13: View Student Attendance Page

The View Attendance page allows teachers to view an individual student's attendance record for a selected week. Teachers must first select a class, then choose a student from the list. It will show daily attendance status (Present, Absent, On Leave) for each day of the selected week and weekly summary that shows the total number of Present, Absent, and On Leave days. Teachers can switch between weeks using the navigation controls to review attendance for previous or upcoming weeks.

View On Leave Student

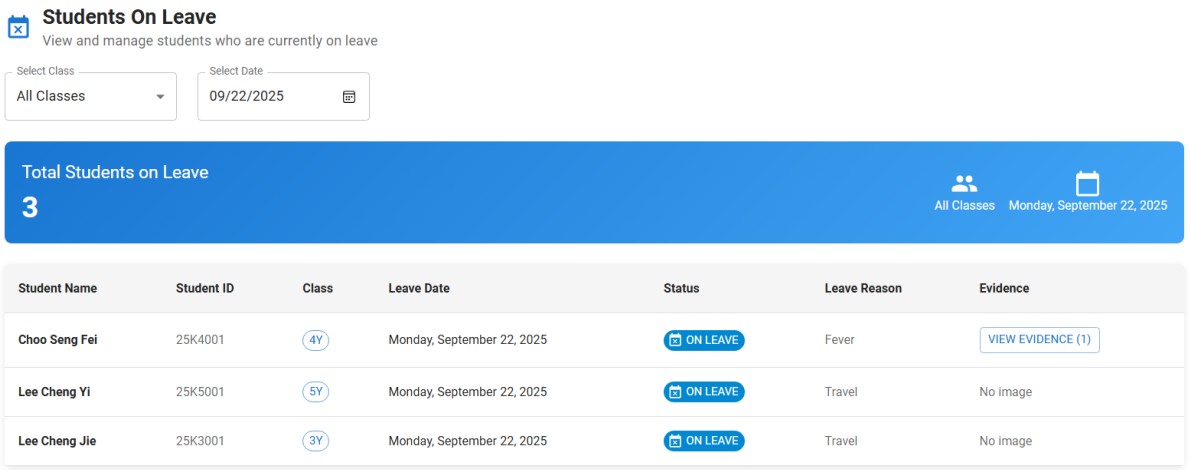


Figure 5.4.1.14: View On Leave Student Page

Report

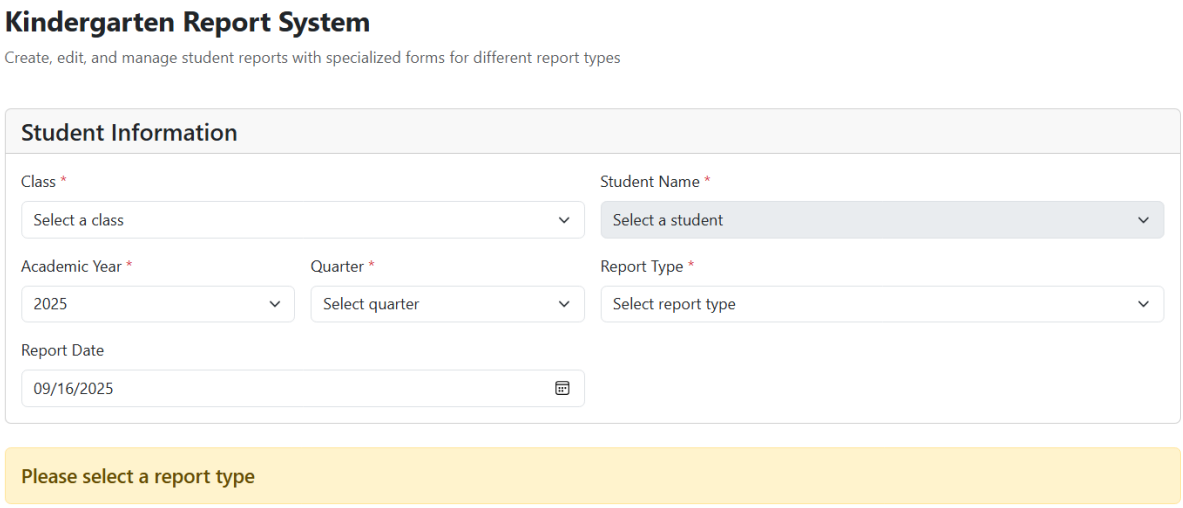


Figure 5.4.1.15: Report Page

The report system allows teachers to generate detailed reports for a specific student. Teachers must first select a class, then choose a student from the list. Once a student is selected, the teacher chooses the Report Type which can Exam Report, Academic Development Report, Social Development Report, or Physical Development Report. The system will display a form based on the selected report type, ensuring teachers fill in only the relevant information. After completing all required fields, teachers must provide additional comments before creating the report. Once the report is successfully created, the system automatically sends a notification to the selected parent to inform them that a new report is available.

Exam Report

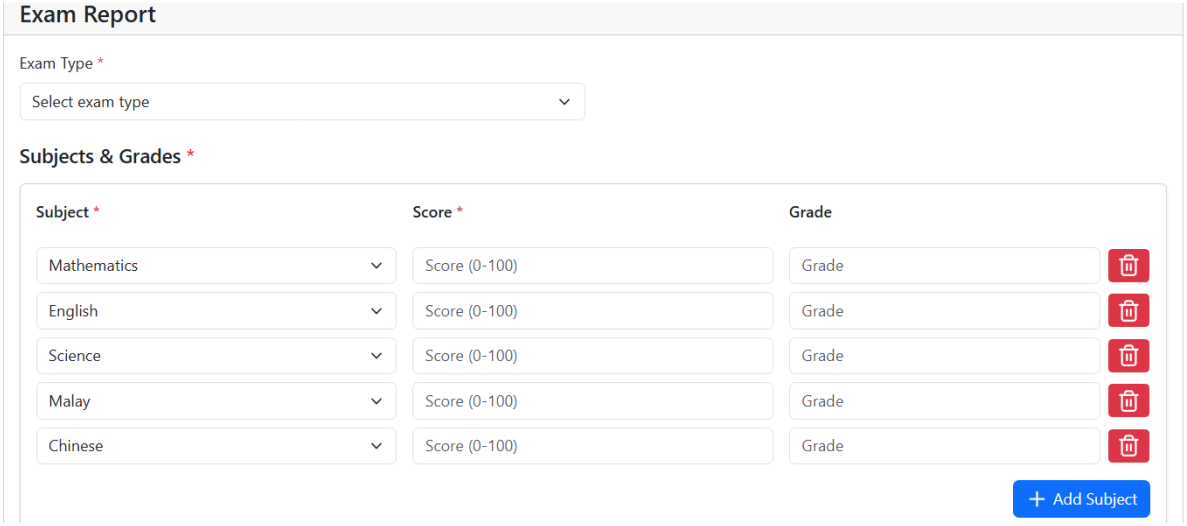
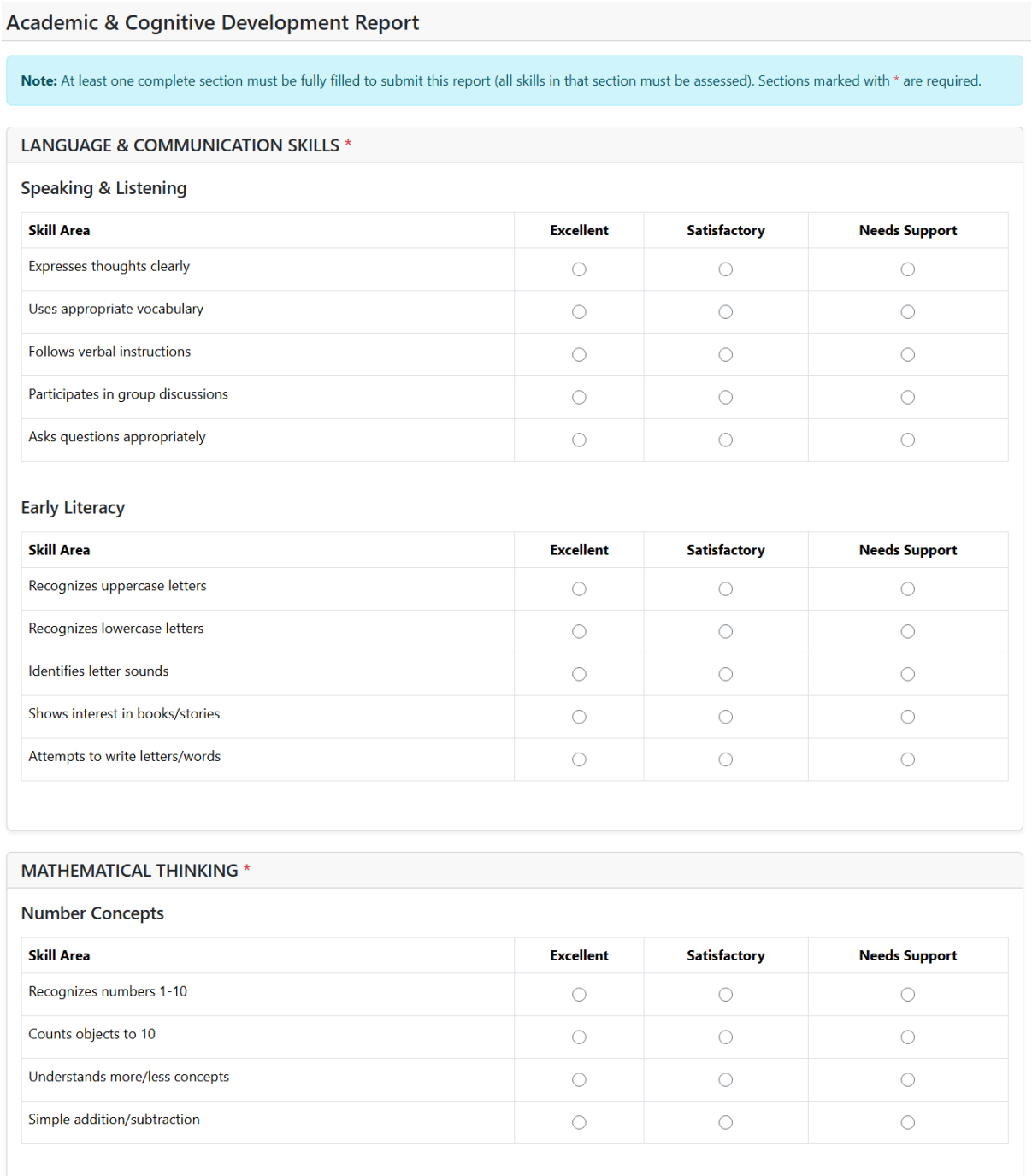


Figure 5.4.1.16: Exam Report Form

The Exam Report Form allows teachers to select the exam type and enter the scores for each subject and the grade will be assigned based on the score given. Several subjects are pre-filled by the system for convenience, but teachers have the flexibility to modify them. They can remove unnecessary subjects or change them using the provided dropdown options.

Other Reports



Academic & Cognitive Development Report

Note: At least one complete section must be fully filled to submit this report (all skills in that section must be assessed). Sections marked with * are required.

LANGUAGE & COMMUNICATION SKILLS *

Speaking & Listening

Skill Area	Excellent	Satisfactory	Needs Support
Expresses thoughts clearly	○	○	○
Uses appropriate vocabulary	○	○	○
Follows verbal instructions	○	○	○
Participates in group discussions	○	○	○
Asks questions appropriately	○	○	○

Early Literacy

Skill Area	Excellent	Satisfactory	Needs Support
Recognizes uppercase letters	○	○	○
Recognizes lowercase letters	○	○	○
Identifies letter sounds	○	○	○
Shows interest in books/stories	○	○	○
Attempts to write letters/words	○	○	○

MATHEMATICAL THINKING *

Number Concepts

Skill Area	Excellent	Satisfactory	Needs Support
Recognizes numbers 1-10	○	○	○
Counts objects to 10	○	○	○
Understands more/less concepts	○	○	○
Simple addition/subtraction	○	○	○

Figure 5.4.1.17: Other Reports Form

The other reports contain more fields to be completed, so teachers are required to fill in an entire subsection before submitting the form. For each skill area, they can choose one of three options which are Excellent, Satisfactory or Needs Support as the student's evaluation.

Announcement Posts

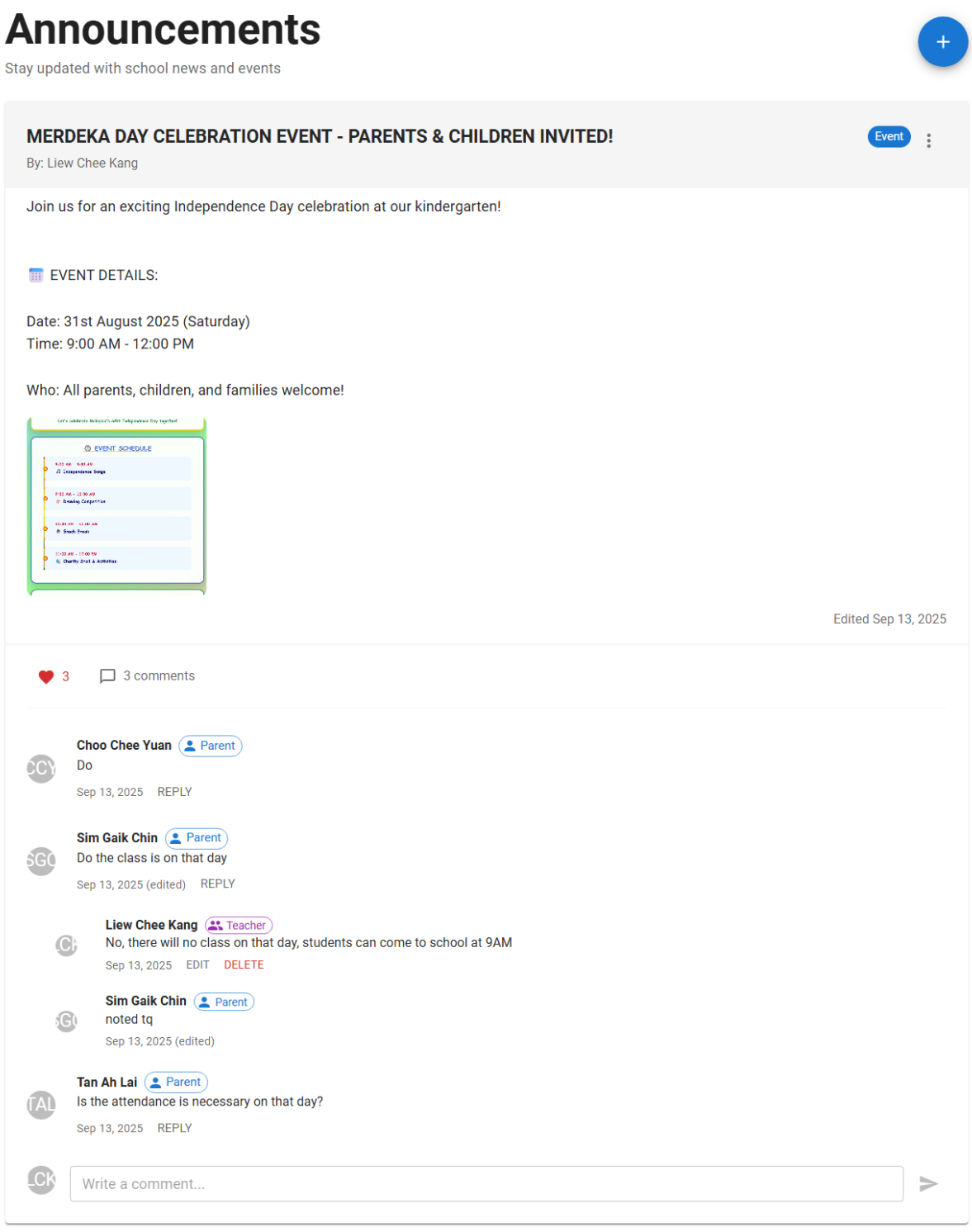


Figure 5.4.1.18: Announcement Posts Page

This page displays all announcements posted by the admin. Admins can edit or delete announcements when needed. Each announcement includes features for likes and comments, with counts shown for both. Users can view comments, reply to specific ones and engage in discussions. Besides, users have control over their own comments, with the option to edit or delete them.

Chat

Figure 5.4.1.19: Chat Page

This page displays the chat list and detailed conversation view. Users can start a new conversation by clicking the ‘New Conversation’ button. The chat list shows unread message counts with the most recent message for quick reference. Each chat entry displays the parent’s name together with the child’s name for clear identification.

5.4.2 Mobile Application Operation

Login

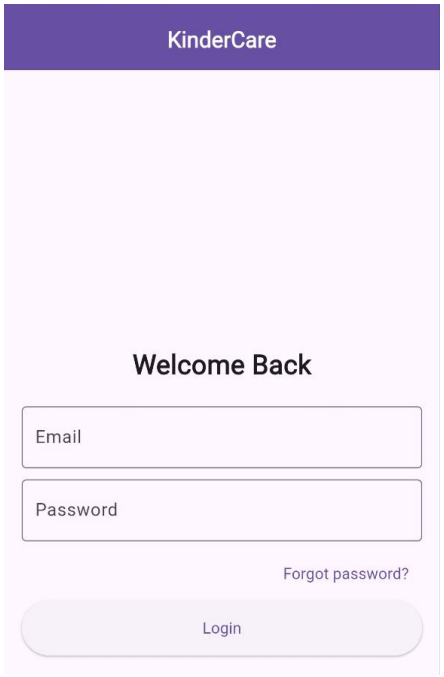


Figure 5.4.2.1: Login Page

The Login Page allows users to access the application by entering their registered email address and password. Parent accounts are automatically created by the admin during the student registration process so that only authorized users can log in to the system.

Side Navigation Bar

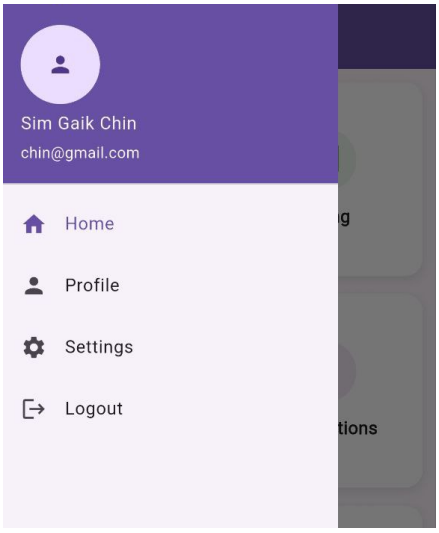


Figure 5.4.2.2: Side Navigation Bar

The Side Navigation Bar displays the current username and email for identification. It provides quick access to key pages which are Home that redirects the user to the dashboard,

Profile that opens the profile page where users can view their personal details and Logout which allows the user to log out of the application.

Profile

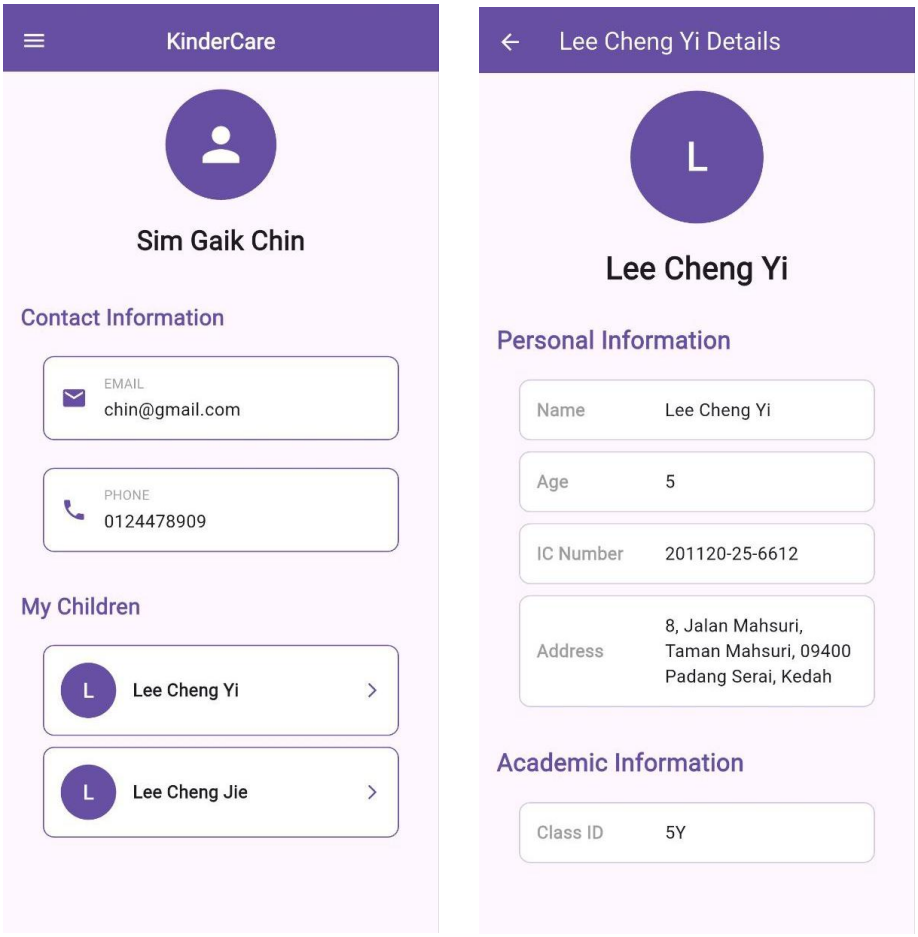


Figure 5.4.2.3: Profile Page and Student Details

The Profile Page displays the parent's personal information and provides a list of all children registered under their name. When the user clicks on a child's name, it will display detailed information about that child.

Dashboard

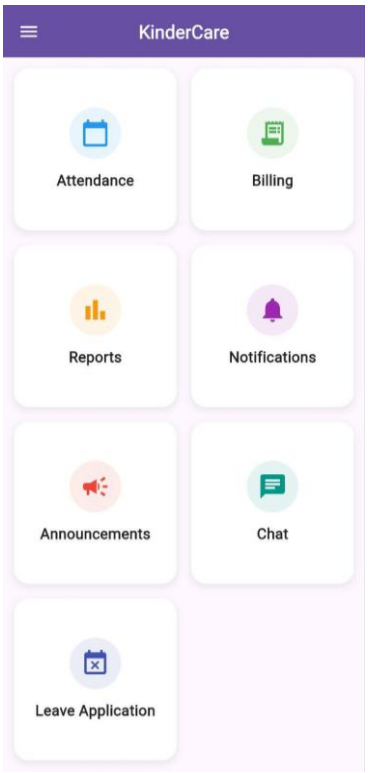


Figure 5.4.2.4: Home Page

The Home Page serves as a dashboard that shows all available features, including Attendance, Billing, Reports, Notifications, Announcements, Chat and Leave Application. Users can click on any of these to be redirected to the corresponding page.

Attendance

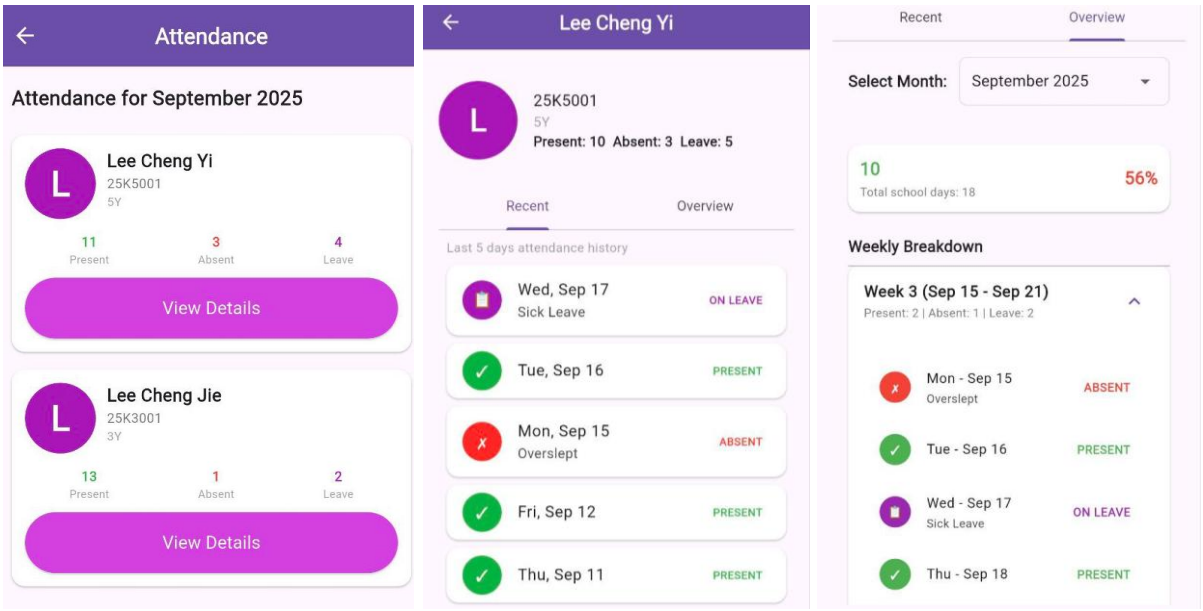


Figure 5.4.2.5: Attendance Page

The Attendance Page shows the attendance records of students registered under the parent's account. Once entering the page, the user is shown a list of all their children and must select one to view detailed attendance information. The detailed attendance view is divided into two tabs which are Recent that displays the attendance history for the last five days and Overview that allows the user to select a specific month, then the system displays attendance data organized by week. For each day, the attendance status (Present, Absent or On Leave) is clearly shown, along with the reason for any absence or leave if provided.

Billing

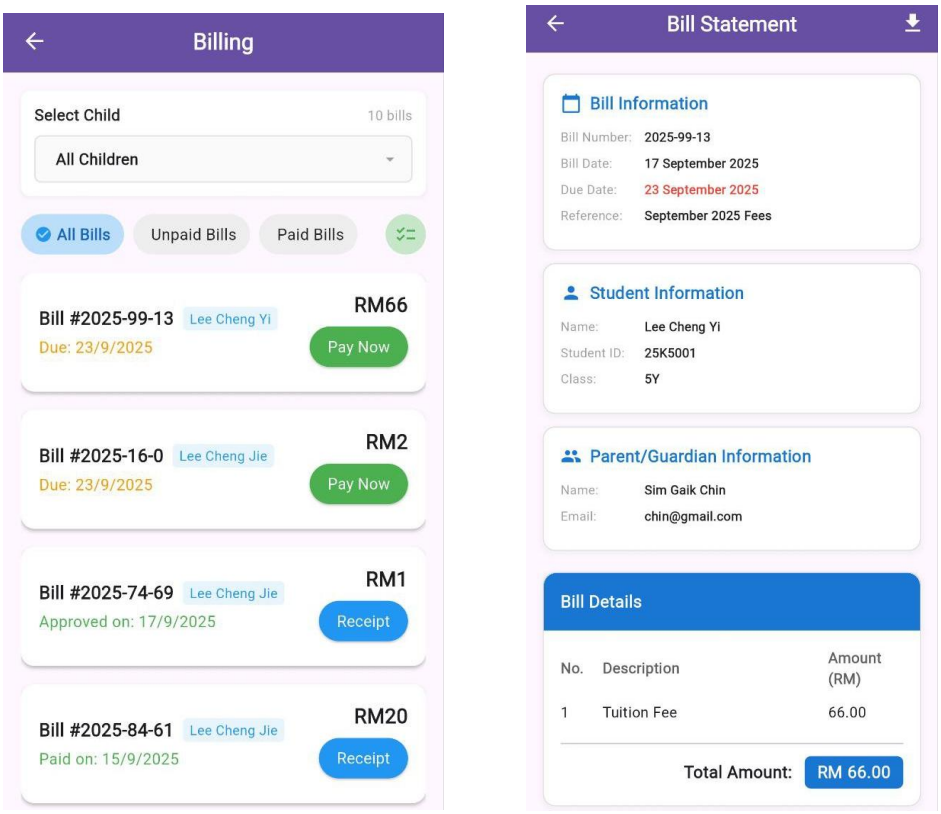


Figure 5.4.2.6: Billing Page

The Billing Page displays all bills associated with the user. If the user has more than one child, a child selector is provided at the top of the page to filter bills by child. Users can also filter bills by All Bills, Unpaid Bills or Paid Bills for easier tracking. Each bill entry includes information such as the bill name, student name, amount and due date. Users can click on a bill to view a detailed bill statement showing all payment information and Pay Now to make a payment securely through the integrated payment gateway. After successful payment, user can click on the Receipt button to view the receipt.

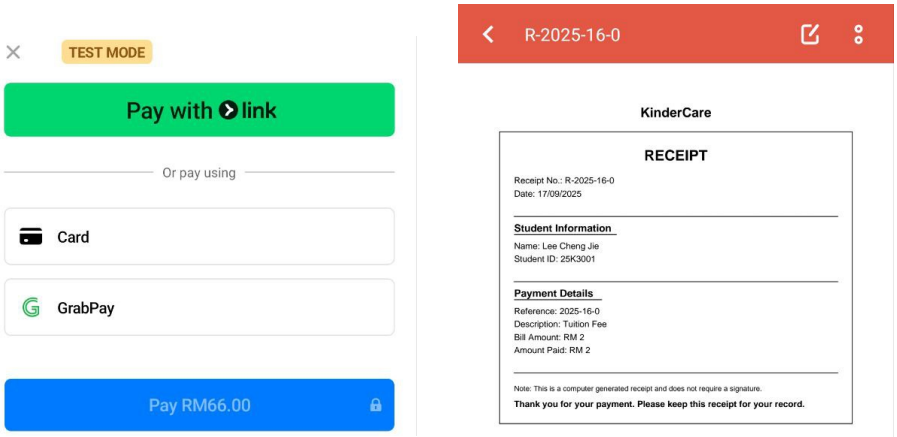


Figure 5.4.2.7: Payment Methods and Receipt

Once click on the Pay Now button, user is allowed to choose a payment method to make in app payment. After successful payment, the payment status will be updated and a receipt will be generated.

Reports

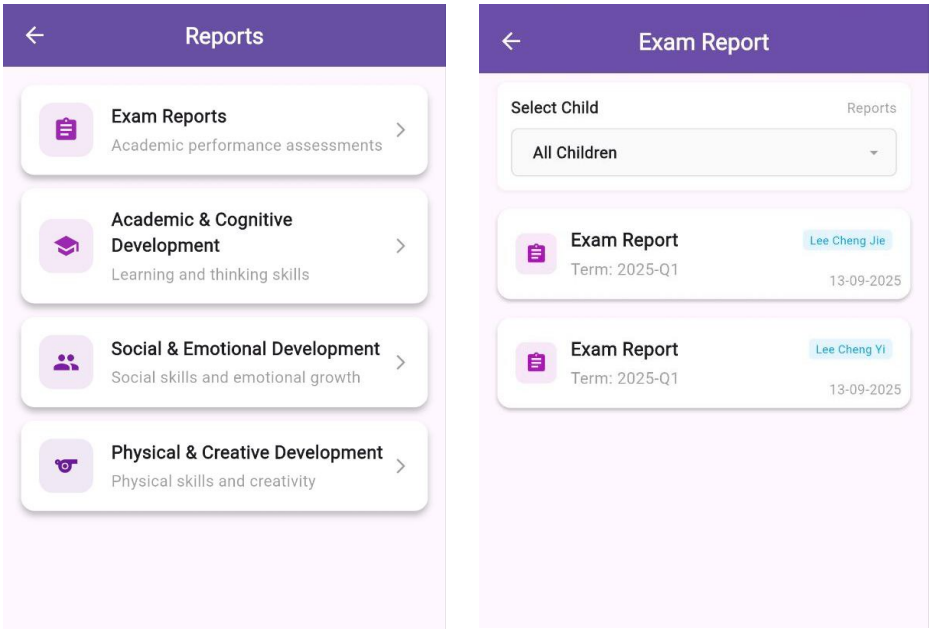


Figure 5.4.2.8: Report Type and List

Once entering the Report Page, there are four types of report that will be listed which are Exam Report, Academic & Cognitive Development Report, Social & Emotional Development Report, and Physical & Creative Development Report The user must first select a report type to view all reports under that category. After entering the selected report type, it will list out all the report. A child selector is available to filter the reports for a specific child if the user has multiple children. Each report shows the students' name and report title. By clicking on a report, the user can view the full report details.

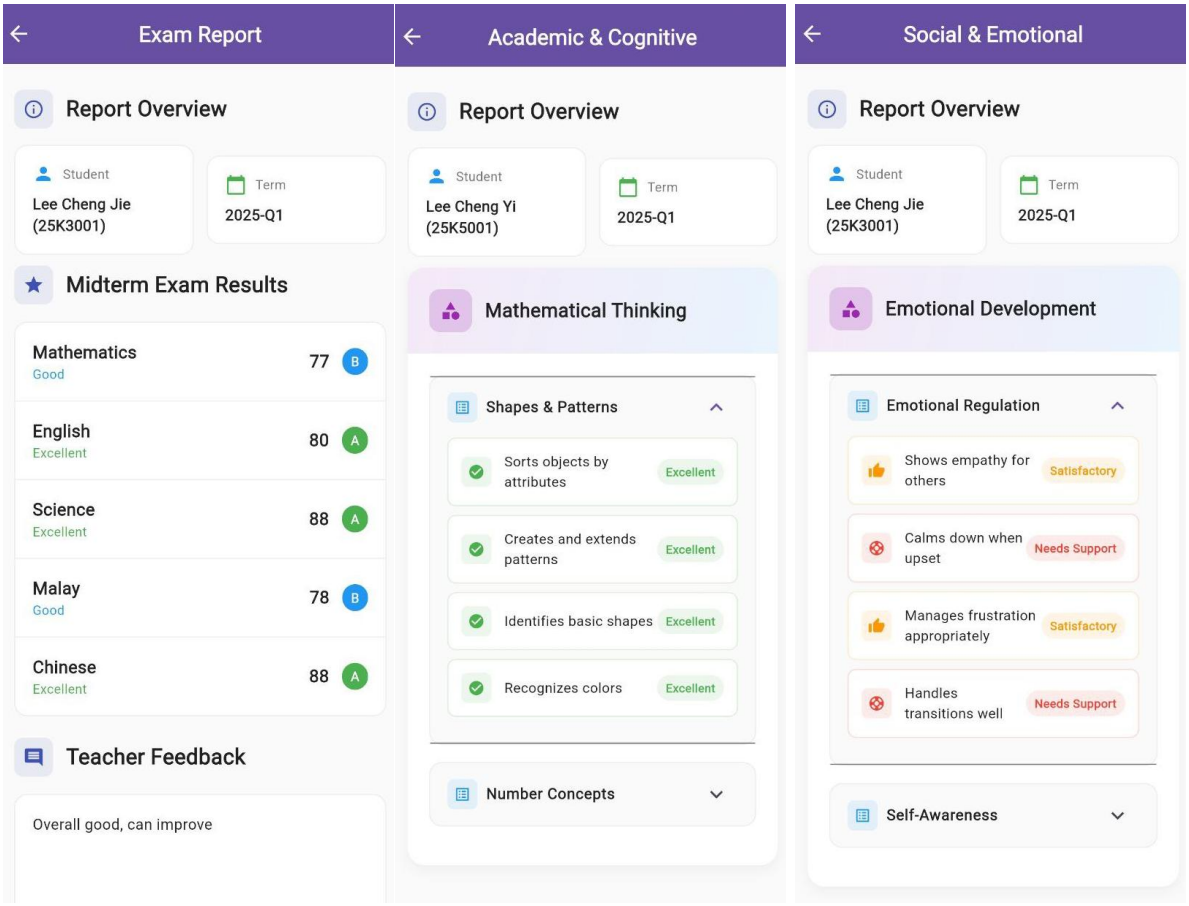


Figure 5.4.2.9: Example of Report Details

Each report displays the student's name and report term at the top, followed by the main report content. The content depends on the type of report where the Exam Report lists all subjects along with their corresponding scores and grades while the Other Reports (Academic, Social, Physical) are organized into categories and subsections. Each subsection shows specific skills or behaviors with their performance status clearly labeled which are Excellent, Satisfactory and Needs Support. At the bottom of every report, a Teacher Feedback section is included for additional feedback or remarks.

Announcements

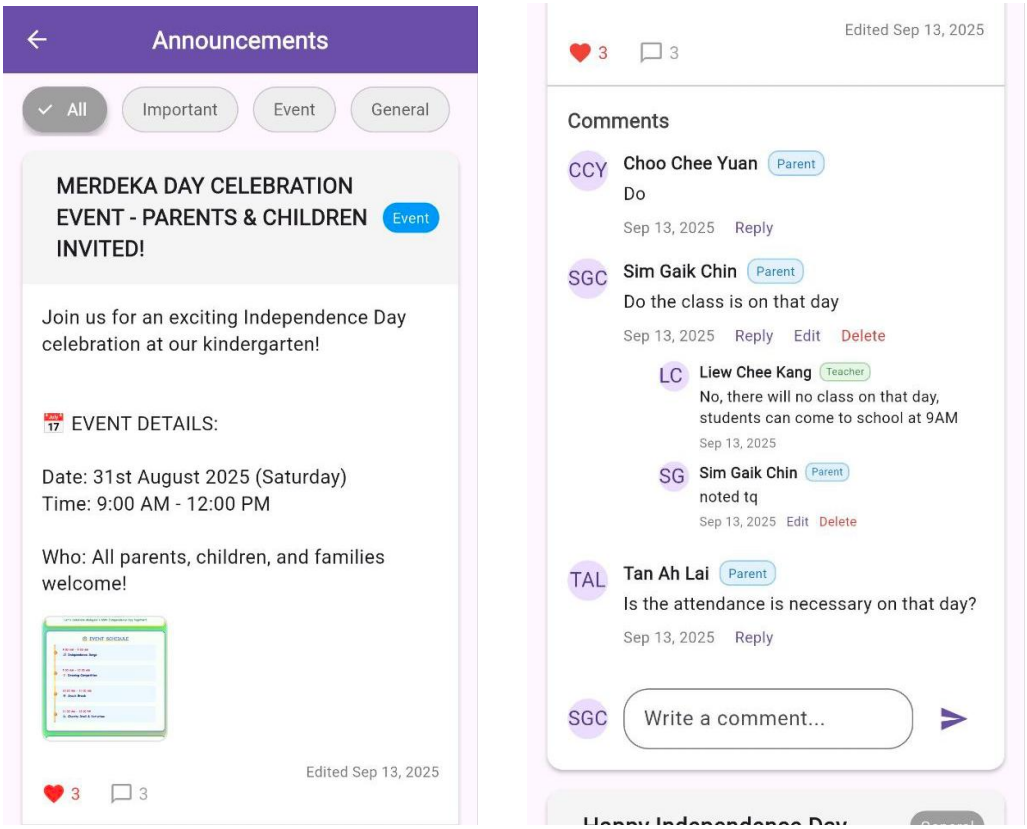


Figure 5.4.2.10: Announcements Page

The Announcement Page displays all posts created by the admin that contain important updates or events that parents need to be aware of. Announcements can be filtered by categories such as All, Important, Event and General for easier navigation. Each post allows users to like and comment and the like and comment counts are displayed below the post. Comments can receive replies and each commenter's name and role such as Teacher, Admin or Parent is clearly shown. Comment owners have the option to edit or delete their own comments.

Leave Application

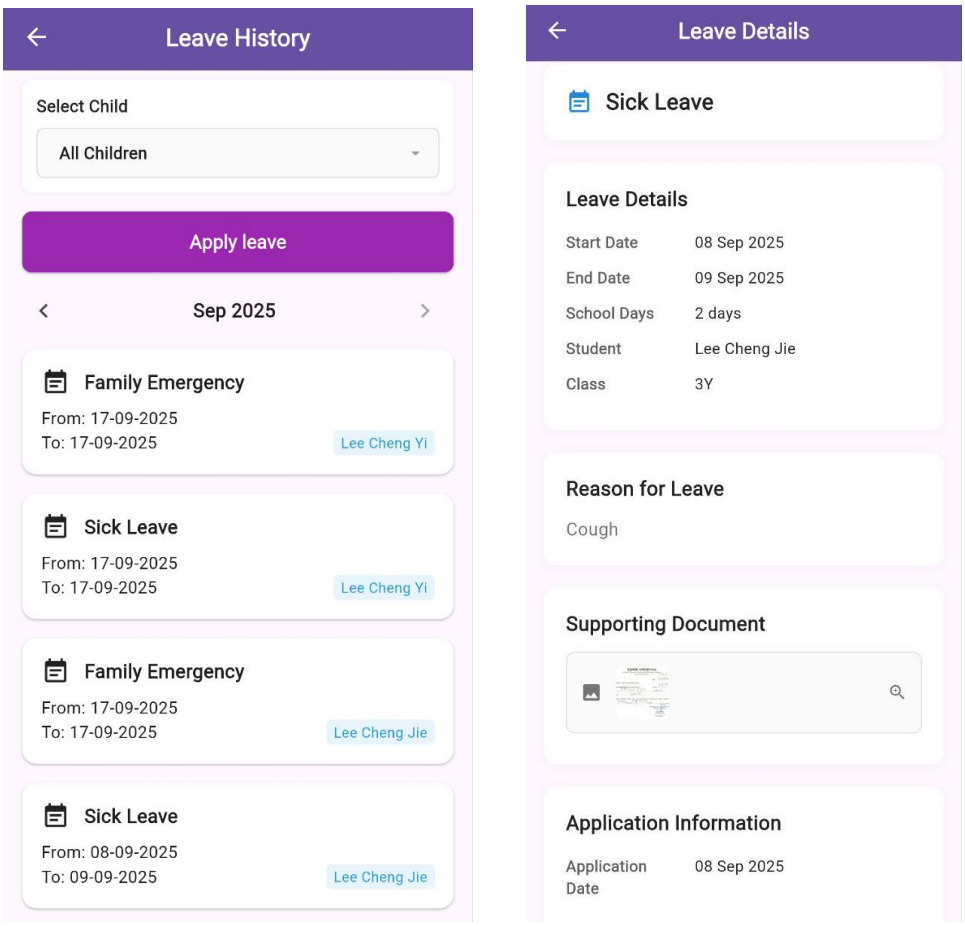


Figure 5.4.2.11: Leave List and Details

The first page of the Leave section displays the leave history of the child. If the user has more than one child, a child selector is available to filter the leave history by child. An Apply Leave button is provided to allow users to submit a new leave application. At the bottom of the page, leaves are grouped and displayed by month. When a user clicks on a leave, the system shows the leave details, including the leave type, reason, supporting document but it is optional based on whether it was uploaded during application and the application date. If a supporting document was uploaded, the user can click on it to preview.

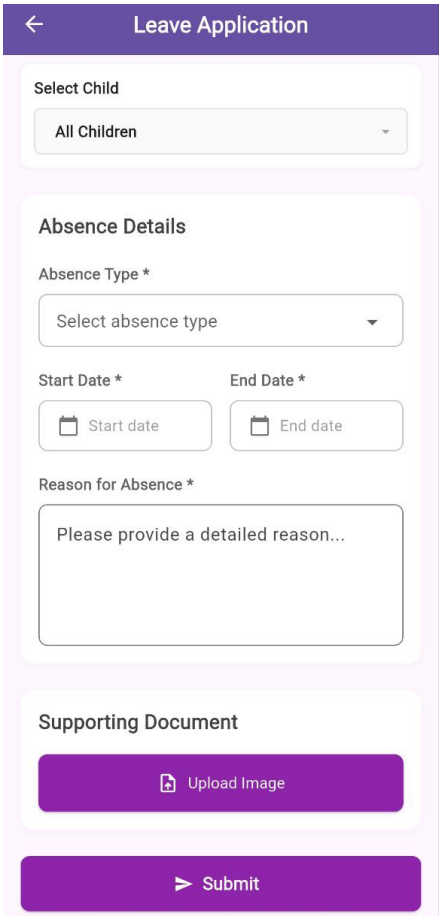


Figure 5.4.2.12: Leave Application

This page provides the leave application form. The user must first select the child for whom they are applying leave. Then, they need to fill in the absence details, including Absence Type which can be chosen from a dropdown menu, Start Date, End Date and the Reason for the leave while uploading a supporting document is optional. Once all required fields are completed, the user can submit the leave application. After submission, the attendance for the selected dates will be automatically marked as "On Leave."

Chat

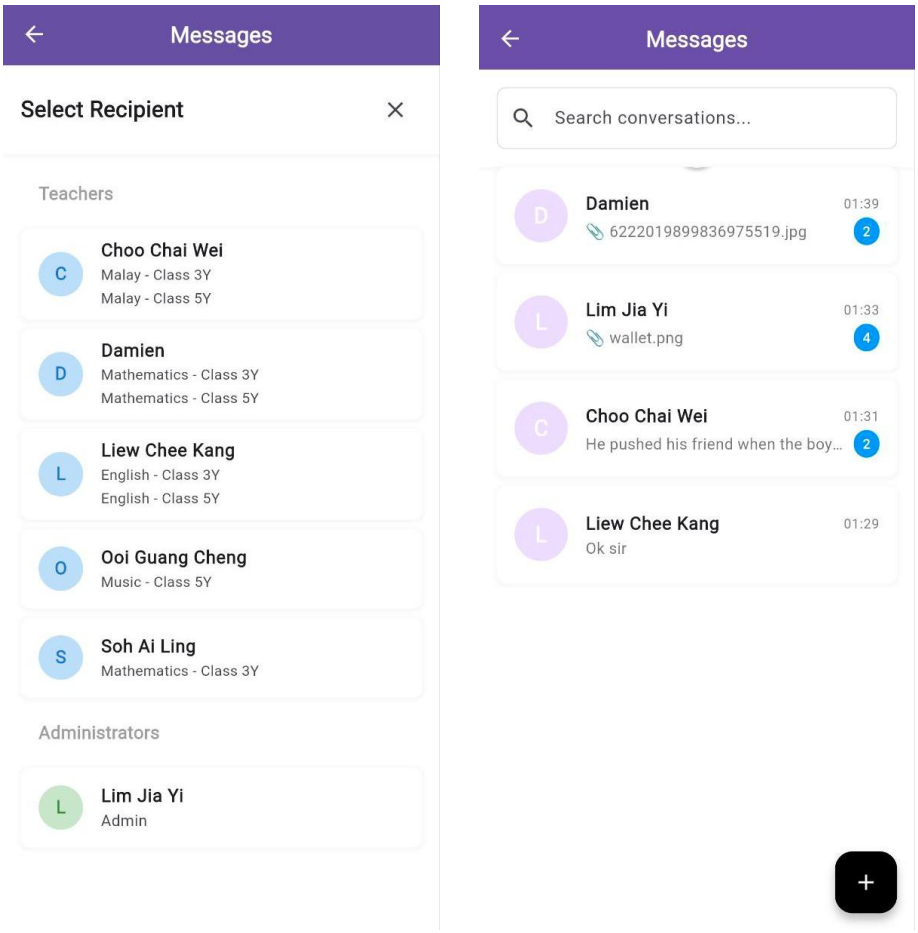


Figure 5.4.2.13: Chat

Once the user enters the chat page, all current conversations will be listed. To create a new chat, the user can tap the "+" button, which will display a list of all administrators and teachers who have taught the user's children. A search bar is available to help users quickly find existing conversations.

For unread messages, a notification badge (small blue circle with a number) will appear to indicate the unread message count. When the user opens the conversation, the badge will disappear.

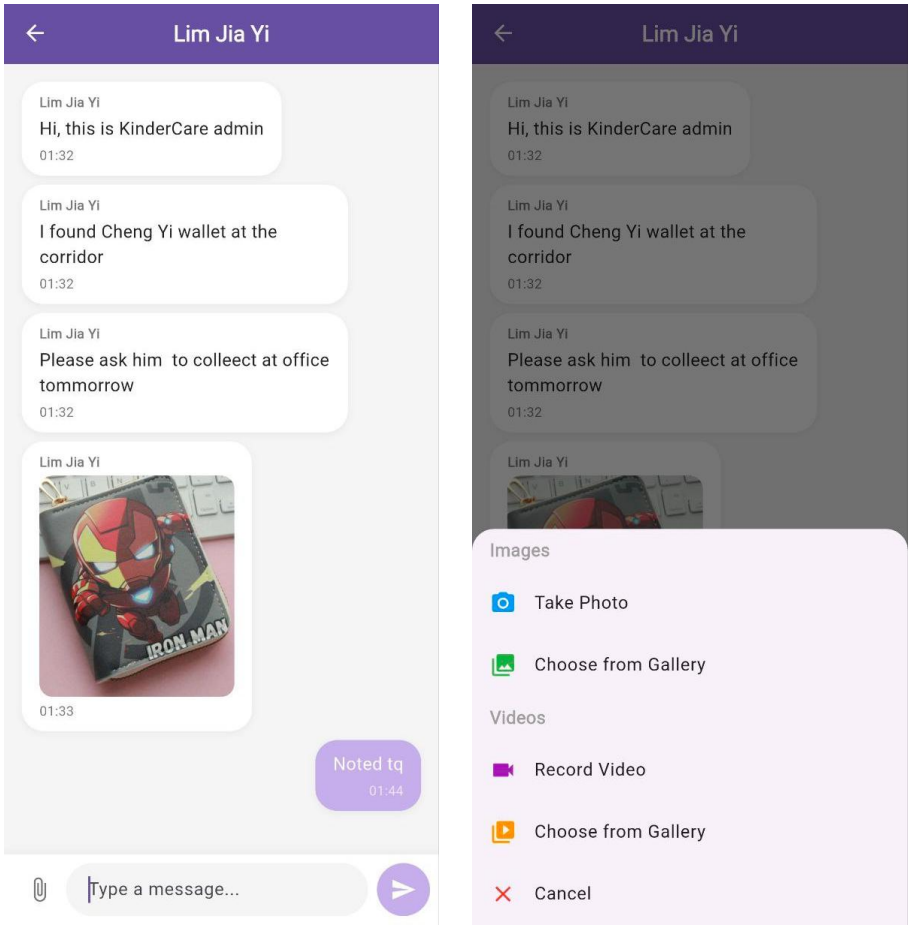


Figure 5.4.2.14: Conversation Details

This page displays the conversation between the user and the admin or teacher. At the top of the screen, the recipient's name is shown as the chat title. The conversation messages are displayed in the middle section while at the bottom of the screen is the message input bar, where the user can type and send text messages. In addition, users have the option to attach and send images or videos, either by taking a photo or video directly or selecting one from their gallery.

Notifications

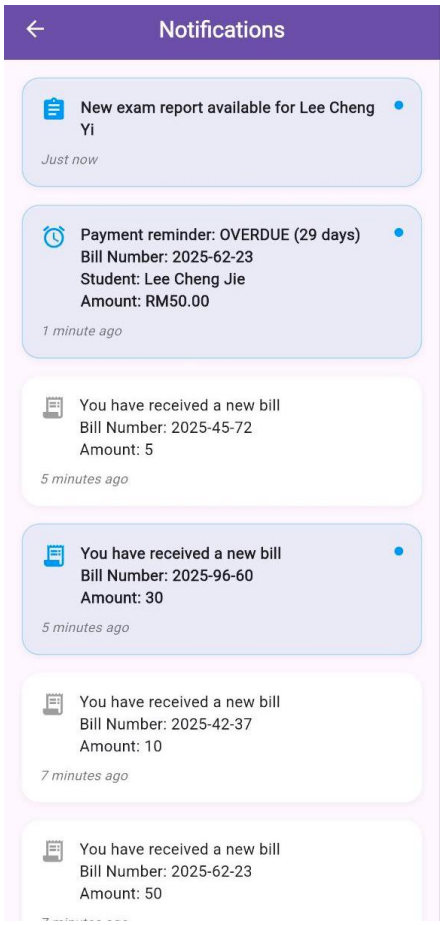


Figure 5.4.2.15: Notification List

The Notification Page displays important reminders created by the admin which includes new bills, bill payment reminders and report notifications. Notifications that have not yet been opened are marked with an unread badge to draw the user's attention. When the user taps on a notification, the unread badge is removed and the user is redirected to the corresponding page. For example, tapping on an exam report notification will navigate the user directly to the detailed report page.

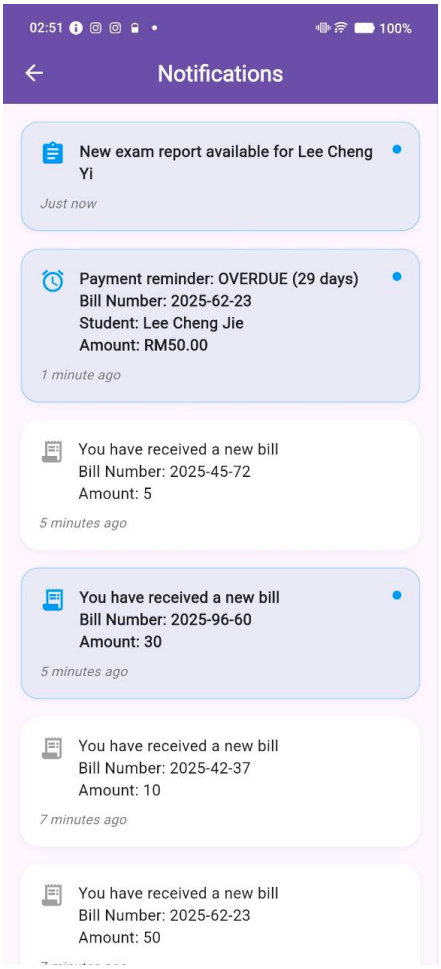


Figure 5.4.2.16: System Notification

The system notifications are sent via the Pushy Service whenever the administrator creates an important event. These notifications includes new bills, bill reminders, new reports, new chat messages and new announcements. This ensures that users stay informed even when they are not actively using the mobile application. By tapping on a notification, users are redirected to the corresponding detail page which allows them to quickly view and respond to the update.

5.5 Implementation of Issues and Challenges

There was an issue encountered when integrating Stripe for online payments. During testing with GrabPay, the sandbox environment successfully processed payments, but the redirect flow back to the application failed. Although the Stripe Dashboard confirmed that sandbox payments were processed, the failure of the redirect flow made it difficult to fully verify post-payment confirmation and database updates.

To resolve this, the app was enhanced to query Stripe's API whenever payment sheet exceptions occur or redirect flows fail. The system stores the Payment Intent ID to retrieve the actual payment status. Based on the verified response from Stripe's API, the payment status in

the database is updated accordingly, ensuring reliable and accurate transaction tracking even when redirects fail.

In addition, there were limitations with Firebase services. Initially, the project was designed to use Firebase Cloud Messaging for push notifications and Firebase Storage for image handling. However, during implementation, it was discovered that these features required billing verification and plan upgrades. Since payment verification could not be completed, both services became impractical for the project's requirements.

Thus, Pushy was used for push notifications. Pushy provided reliable delivery with a free usage tier, easy integration with Flutter and React and did not require additional billing setup. For image handling, Cloudinary was chosen as the replacement for Firebase Storage. Cloudinary offered free plan, built-in tools for uploading and storing media and developer-friendly APIs that simplified image management.

Chapter 6
System Evaluation and Discussion

6.1 System Testing and Performance Metrics

System testing was carried out through functional testing to ensure that all features of the kindergarten management system performed as expected. Functional testing was chosen because the main objective was to validate that each module met its specified requirements and operated correctly within the overall system.

6.1.1 Key Modules for Testing

The functional testing covered several key modules of the kindergarten management system. For Parent and Student Registration, the system was tested to ensure it could correctly detect duplicate parent records, auto-fill existing information and create new accounts when necessary. In Student and Staff Management, admin was able to add, edit and delete staff records, as well as assign classes or subjects without errors. The Billing Module was tested to confirm that bills could be created and assigned to students or entire classes, with notifications pushed correctly while the in-app payment function was also validated to ensure successful payments and status updates.

The Attendance Module was checked to verify that teachers could take daily attendance, mark students as present or absent and that "On Leave" statuses were automatically applied for approved leave requests. For Report generation, testing confirmed that teachers could generate exam reports, academic development reports and other report types and parents able to view the report. Finally, Announcements, Notifications and Leave Applications were validated to ensure that announcements displayed correctly by category, notifications were delivered reliably through Pushy and leave applications were properly submitted, stored and reflected in attendance records.

6.1.2 Performance Metrics

Since functional testing was the primary focus, the performance metric used was requirement coverage where the percentage of requirements successfully tested and passed. Test cases were derived from the system's functional requirements and a pass rate of 100% was achieved for all critical functions after bug fixing and retesting.

6.2 Testing Setup

Test Environment:

1. Frontend: Flutter (Mobile), React (Web)
2. Backend: Firebase Firestore
3. Push Notifications: Pushy
4. Image Handling: Cloudinary
5. Payment Gateway: Stripe (Test Mode)

Devices Used:

1. Android smartphone
2. Desktop browser

Test Procedure:

1. App is installed in smartphones.
2. Internet connection is available
3. Load the mobile and web app
4. Login to the system
5. Test each module

6.3 Test Result for Web Application

6.3.1 Register New Student

Test Case 1: Class assigned based on Identity Card Number

ID	Description	Expected Result	Actual Result
1	230331-02-0555	3Y	Same
2	220728-02-0820	3Y	Same
3	210521-07-0618	4Y	Same
4	201022-02-5561	5Y	Same
5	190115-05-7789	6Y	Same
6	180319-02-0523	N/A	Same

Table 6.3.1.1: Test Case for Class Assigned

Test Case 2: Checking for Existing Parent

ID	Description	Expected Result	Actual Result
1	Different phone number and email	Nothing shown, can submit registration	Same
2	Same phone number but different email	Show the existing parent details, "Use Current Parent" and "Clear Form" buttons	Same
3	Different phone number but same email	Show the existing parent details, "Use Current Parent" and "Clear Form" buttons	Same
4	Click "Use Current Parent"	The existing parent details are filled	Same
5	Click "Clear Form"	The parent form is cleared	Same
6	Success register student with new parent	New account created for parent to login	Same
7	Success register student with existing parent	Student is appended under the existing parent	Same

Table 6.3.1.2: Test Case for Checking Parent Details

6.3.2 Bill

Test Case 1: Test Case for Create Bill

ID	Description	Expected Result	Actual Result
1	No student selected and no billing details filled.	Cannot create bill	Same
2	Student(s) selected but no billing details	Cannot create bill	Same
3	Billing details filled but no student selected	Cannot create bill	Same
4	Student(s) selected and billing details filled	Bill created	Same
5	Bill created	Parents receive notification	

Table 6.3.2.1: Test Case for Create Bill

Test Case 2: Test Case for Calculate Due Date

ID	Description	Expected Result	Actual Result
1	Billing Date:12/9, Term: 3 Days	Due Date: 15/09	Same
2	Billing Date: 12/09, Term: 7 Days	Due Date: 19/09	Same
3	Billing Date: 12/09, Term: 14 Days	Due Date: 26/09	Same
4	Billing Date: 12/09, Term: 30 Days	Due Date: 12/10	Same

Table 6.3.2.2: Test Case for Calculate Due Date

Test Case 3: Test Case for Bill Description

ID	Description	Expected Result	Actual Result
1	Add new description through Template Dialog	Success added	Same
2	Delete description through Template Dialog	Success deleted	Same
3	Add new description through Description field	Success added	Same

Table 6.3.2.3: Test Case for Bill Description

Test Case 4: Test Case for Listing Bills

ID	Description	Expected Result	Actual Result
1	Use filter “All status”	All bills are listed	Same
2	Use filter “Paid”	All paid bills are listed	Same
3	Use filter “Unpaid”	All unpaid and overdue bills are listed	Same
4	Use filter “Overdue”	All overdue bills are listed	Same
5	Unpaid and Overdue bills	A “Mark Paid” button shown	Same
6	Click “Mark Paid” button	Status change to “Paid”	Same
7	Click reminder button	Will send notifications to the unpaid user	Same

Table 6.3.2.4: Test Case for Listing Bill

6.3.3 Chat

Test Case 1: Chat Layout

ID	Description	Expected Result	Actual Result
1	Start a chat with a parent with only one child	Show the parent and the child’s name	
2	Start a chat with a part with multiple children, chat for first child	Show the parent and child’s name	
3	Start a chat with a part with multiple children, chat for second child	Show the parent and the 2 children name	
4	Start a chat with a part with multiple children, chat for third child	Show the parent and the 3 children’s name	

Table 6.3.3.1: Test Case for Chat Layout

Test Case 2: Sending Message to Parent

ID	Description	Expected Result	Actual Result
1	Nothing is typed in message bar	Unable to click submit button	Same
2	Message bar is filled	Submit button enabled	Same
3	Send message to parent	Mobile app receive notification	Same

Table 6.3.3.2: Test Case for Sending Message

Test Case 3: Receiving Message from Parent

ID	Description	Expected Result	Actual Result
1	Unseen Message	Show icon with unread message count	Same
2	Click into the message	The icon disappears	Same

Table 6.3.3.3: Test Case for Receiving Message

6.3.4 Attendance

Test Case 1: Take Attendance

ID	Description	Expected Result	Actual Result
1	If no student has been marked for attendance	Cannot save attendance	Same
2	Some students only have been marked	Attendance can be saved	Same
3	If only some students have been marked	Can save attendance	Same
4	Student applied leave	The “On Leave” status is prefilled and the reason is stated	Same
5	Mark as absent	Show dialog to write specific reason	Same
6	Click on “Provide Reason” button	Can enter and submit the reason for absent	Same
7	Direct click the “Mark Absent” button	Submit the absent status without reason	Same
8	Absent reason provided	Show the specific reason	Same
9	No absent reason provided	Show “No reason provided”	Same

Table 6.3.4.1: Test Case for Take Attendance

Test Case 2: Update Attendance

ID	Description	Expected Result	Actual Result
1	If no student attendance is modified	Cannot update attendance	Same
2	If student attendance is modified	Attendance can be updated	Same

Table 6.3.4.2: Test Case for Update Attendance

Test Case 3: Student Leave Evidence

ID	Description	Expected Result	Actual Result
1	Student Leave with Evidence	A button to preview the image	Same
2	Student leave without evidence	No button to preview the image	Same

Table 6.3.4.3: Test Case of Student Leave Evidence

6.3.5 Report

Test Case 1: Create report

ID	Description	Expected Result	Actual Result
1	No report type chosen	No report form shown	Same
2	Select a report type	Form appeared	Same
3	Report created	Parents receive notification	

Table 6.3.5.1: Test Case of Create Report

Test Case 2: Exam Report

ID	Input (Score)	Description	Expected Result	Actual Result
1	35	Enter score <40	Grade F	Same
2	45	Enter score >=40 and <50	Grade E	Same
3	55	Enter score >=50 and <60	Grade D	Same
4	65	Enter score >=60 and <70	Grade C	Same
5	75	Enter score >=70 and <80	Grade B	Same
6	85	Enter score >=80	Grade A	Same

Table 6.3.5.2: Test Case of Exam Report

Test Case 3: Other Reports

ID	Description	Expected Result	Actual Result
1	Did not fill any column	Cannot submit report	Same
2	Fill only a few columns in a section	Cannot submit report	Same
3	Completely fill all columns in a section	Can submit report	Same

Table 6.3.5.3: Test Case of Other Reports

6.3.6 Announcement

Test Case 1: Create or edit Announcement

ID	Description	Expected Result	Actual Result
1	Post empty form	Unable to post	Same
2	Post without filling all fields	Unable to post	Same
3	Fills all fields except image	Able to post	Same
4	Fills all fields include image	Able to post	Same
5	Did not edit the post	Show the posted datetime	Same
6	Edited post	Show the edited datetime	Same
7	Post within 60 minutes	Show time in minute	Same
8	Post after 60 minutes	Show time in hour	Same
9	Post after 24 hours	Show time in yesterday	Same
10	Post after 48 hours	Show time in date	Same

Table 6.3.6.1: Test Case for creating and editing Announcement

Test Case 2: Like and Comment

ID	Action / Description	Expected Result	Actual Result
1	Click on first Like	Like count becomes **1**	Same
2	Click on second Like	Like count becomes **2**	Same
3	Unlike (remove like)	Like count decreases by **1**	Same
4	View post from different account	Can see the correct like count	Same

5	Add a comment	Comment appears under post, comment count increases by **1,** role is stated	Same
6	Reply to a comment	Reply appears under the comment, comment count unchanged	Same
7	Edit a comment or reply	Edited content displayed with "edited" beside timestamp	Same
8	Delete a comment or reply	Comment or reply removed from the post	Same

Table 6.3.6.2: Test Case for Like and Comment

6.3.7 Manage Teacher

Test Case 1: Assign Subject and Class

ID	Action / Description	Expected Result	Actual Result
1	Assign a teacher a different subject and class	Able to assign	Same
2	Assign a teacher same subject but different class	Able to assign	Same
3	Assign a teacher different subject but same class	Able to assign	Same
4	Assign a teacher a same subject and class	Unable to assign	Same

Table 6.3.7.1: Test Case for Like and Comment

6.4 Testing Result for Mobile Application

6.4.1 Attendance

Test Case 1: Test Case for View Attendance

ID	Action / Description	Expected Result	Actual Result
1	User has 1 child	Show one student when the parent has one child	Same
2	User has more than one child	Show multiple students when the parent has multiple children	Same
3	Click on Recent tab	Show last 7 days attendance	Same
4	Click on Overview tab	Show attendance categorized by weekly	Same
5	Student absence	Show the reason if the student absent	Same
6	Student on leave	Show the reason if the student on leave	Same

Table 6.4.1.1: Test Case for View Attendance

6.4.2 Billing

Test Case 1: Listing Bills

ID	Action / Description	Expected Result	Actual Result
1	User has 1 child	No child selector	Same
2	User has more than one child	Has a child selector	Same
3	Child selector is “All”	Show all the children’s bills.	Same
4	A child name is selected	Show all the entire child’s bills	Same
5	Filter: All Bills	Show all the bills.	Same
6	Filter: Unpaid Bills	Show all unpaid bills.	Same
7	Filter: Paid Bills	Show all paid bills.	Same
8	Click the entire bill	Show the entire bill’s details	Same

Table 6.4.2.1: Test Case for Listing Bills

Test Case 2: Payment

ID	Action / Description	Expected Result	Actual Result
1	Click “Pay Now” button	Allow to choose payment method	Same
2	Pay by card	Allow to input card details	Same
3	Pay by GrabPay	Redirected to Stripe test payment page	Same
4	Success payment	Status change to “Paid” and	Same
5	Paid bills	Receipt generated	Same

Table 6.4.2.2: Test Case for Payment

6.4.3 Report

Test Case 1: Test Case for showing Report

ID	Action / Description	Expected Result	Actual Result
1	User has 1 child	No child selector	Same
2	User has more than one child	Has a child selector	Same
3	Child selector is “All”	Show all the children’s report.	Same
4	A child name is selected	Show all the entire child’s report	Same

Table 6.4.3.1: Test Case for showing Report

6.4.4 Notifications

Test Case 1: Test Case for Notification

ID	Action / Description	Expected Result	Actual Result
1	Click on report notification	Directed to the entire report page	Same
2	Click on the bill notification	Directed to the entire bill page	Same
3	Click on the announcement notification	Directed to the announcement page	Same
4	Click on the chat notification	Directed to the entire chat	Same

Table 6.4.4.1: Test Case for Notification

6.4.5 Announcement

Test Case 1: Test Case for Announcement

ID	Action / Description	Expected Result	Actual Result
1	Click on first Like	Like count becomes **1**	Same
2	Click on second Like	Like count becomes **2**	Same
3	Unlike (remove like)	Like count decreases by **1**	Same
4	View post from different account	Can see the correct like count	Same
5	Add a comment	Comment appears under post, comment count increases by **1,** role is stated	Same
6	Reply to a comment	Reply appears under the comment, comment count unchanged	Same
7	Edit a comment or reply	Edited content displayed	Same
8	Delete a comment or reply	Comment or reply removed from the post	Same

Table 6.4.5.1: Test Case for Announcement

6.4.6 Chat

Test Case 1: Test Case for Chat

ID	Action / Description	Expected Result	Actual Result
1	Add new chat	Show all admins and teacher that teaches the children	Same
2	If the parent has more than 1 child	The top of the chat screen will show the children's name	Same
3	Attach image	Can choose camera or gallery	Same
4	Unread message	Show the count for unread messages	Same

Table 6.4.6.1: Test Case for Chat

6.4.7 Leave Application

Test Case 1: Test Case for Leave Application

ID	Action / Description	Expected Result	Actual Result
1	User has 1 child	No child selector	Same
2	User has more than one child	Has a child selector	Same
3	Child selector is "All"	Show all the children's leaves.	Same
4	Image is uploaded	Can apply leave	Same
5	Same start and end date	Apply leave for 1 day	Same
6	Start day: 15; End date:16	Apply leave for 2 days	Same
7	No image uploaded	Can apply leave	Same
8	All fields are filled	Can apply leave	Same
9	Empty field(s)	Cannot apply leave	Same

Table 6.4.7.1: Test Case for Leave Application

6.5 Project Challenges

When reviewing the existing systems, it was difficult to gain access to the existing kindergarten management systems because these platforms involve sensitive information such as parents, staff and children's personal data. Most systems required verified accounts to explore their full functionality which was impossible to obtain as an external researcher. As a result, the review process relied on available documentation, user manuals and feature

descriptions rather than direct hands-on experience. This limitation may have restricted the depth of understanding regarding how the systems are used in real-world.

Besides, Stripe was used as the online payments gateway in the system. As the system was in development mode, it only redirects users to Stripe sandbox testing page instead of actual banking applications or GrabPay. This caused inconsistencies compared to real production environment. Sometimes, the simulator did not complete the process correctly and users were not redirected back to the application even though the sandbox displayed a successful or failed payment outcome. Thus, the system needs to manually direct users back to the system interface when they clicked the "Authorize" or "Fail Payment" buttons on the Stripe testing page.

In the testing stage, the system was tested by small groups of friends instead of actual kindergarten staff or parents. This limitation may not fully capture the real needs, behaviors and expectations of the intended users. In addition, the number of available mobile devices for testing was limited and needed to be borrowed from family members and friends. Thus, testing was conducted with smaller datasets and fewer simultaneous users.

6.6 Objectives Evaluation

The primary objective was to provide an integrated platform that enhances parental involvement in the academic life of students. This is achieved as the kindergarten management system has successfully integrated multiple functionalities within a single platform which consists of a web application and a parent mobile application. These applications are connected to the same backend services such as Firebase, Cloudinary and Pushy. With the system, parents are able to access important information such as announcements, fee updates and communication with teachers through mobile applications.

The system includes a real time messaging feature that can facilitate effective communication between teachers and parents. It allows parents to raise concern or ask questions instantly and teachers can provide timely responses. Push notifications for messaging are also integrated to ensure parents do not miss important updates about their child's progress or activities. This communication channel strengthens the communication between parents and teachers.

Next, Stripe is integrated as the secure payment gateway which allows parents to pay fees directly through the mobile application. The system supports multiple payment methods such

as credit or debit card and GrabPay. Integrated payment service can reduce the risk of missed payments and provide convenience for parents and also automates the checking process for administrators.

The system also integrates a reminder and notification feature for parents. When the administrator creates a bill or posts an announcement regarding important events such as meetings, field trips or school holidays, all parents will receive notifications so that no critical information is missed. In addition, the administrator can send targeted reminders to parents who have not completed their payments. Since notifications are delivered in real-time, parents and students will have sufficient time to prepare and participate actively in school events. This functionality improves attendance, enhances parental engagement and minimizes missed communications.

In conclusion, all objectives of the kindergarten management system were successfully achieved. It provides a timely communication channel, secure in-app payments and real-time notifications that can ensure parents are always informed, engaged and actively connected with their children's education.

6.7 Concluding Remark

The kindergarten management system has been successfully tested and has achieved the stated objectives as all features are fully implemented. Although several challenges were encountered during development and testing stage, they were effectively managed with appropriate solutions.

Overall, the system performs well across its functionalities. It enables parents to remain actively engaged in their children's education through timely communication, secure payment options, event reminders and other integrated features. On the other hand, administrators also benefit from improved efficiency in their work, particularly in areas such as billing reminders, attendance tracking, and report management.

Chapter 7
Conclusion and Recommendation

7.1 Conclusion

The kindergarten management system has successfully addressed the limitations of traditional systems by integrating modern technology to improve communication, automate administrative tasks and strengthen parental involvement in early childhood education. Unlike existing systems which only focus on basic administrative functions like attendance and record-keeping, this developed system introduces real time communication, in app payment, instant notification and other features that keep parents connected and engaged.

The mobile application provides parents with a convenient platform to access messaging, payments, progress tracking, attendance, announcements leave applications and reminders. Meanwhile, the web platform allows teachers and administrators to manage attendance, report, announcements and payments more efficiently.

The key innovation of the system is the use of automated notification as reminders that can ensure parents never miss important updates about school events or their child's progress. In addition, the in-app payment helps in reducing missed payments, streamlines financial transactions and reduces the administrative burden on staff.

In conclusion, this system not only enhances parental involvement and improves administrative efficiency but also allows educators to focus more on personalized teaching.

7.2 Recommendation

There are some recommendations that can be made in future to improve the effectiveness and usability of the kindergarten management system. At the current stage, system testing was carried out with friends who had no direct experience with kindergarten management processes. Therefore, future testing should involve real stakeholders such as kindergarten staff, parents and administrators. This would allow the system to align more closely with real-world needs.

Besides, it is essential to enhance security and privacy measures since the system handles sensitive data. This can be done by implementing mechanisms such as data encryption, role-based access control and two-factor authentication that can help to ensure data protection.

Furthermore, the system can be enhanced by adding e-learning features such as online quizzes, shared learning materials and homework tracking. With these features, the system would go beyond administration and communication to provide direct support for children's learning. This would make the system more complete, benefiting parents and teachers while also supporting the academic growth of the students.

REFERENCES

[1] F. OZMEN, C. AKUZUM, M. ZINCIRLI, and G. SELCUK, “The communication barriers between teachers and parents in primary schools,” *Eurasian Journal of Educational Research*, vol. 16, no. 66, pp. 1–30, Dec. 2016. doi:10.14689/ejer.2016.66.2

[2] Tyson, Sharon D. *Exploring parent-teacher communication through technology: A case study*. Diss. Northcentral University, 2020.

[3] M. Şengül and M. Özkan, “Kindergarten management problems and solutions in Turkey: The experiences of principals,” *International Journal of Progressive Education*, vol. 17, no. 5, pp. 134–151, Sep. 2021. doi:10.29329/ijpe.2021.375.10

[4] A. Abu Ziden, M. F. Abdul Rahman, and T. Woon Ching, “Exploring the use of mobile instant messaging for parent-teacher communication,” *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 14, no. 04, p. 152, Mar. 2020. doi:10.3991/ijim.v14i04.12403

[5] S. J. McKnight, "Parent–Teacher Online Communication and Third- Through Fifth-Grade Student Academic Achievement" (2020). *Walden Dissertations and Doctoral Studies*. 9975. https://scholarworks.waldenu.edu/dissertations/9975

[6] Dahlan Abdullah, Poetri Al-Viany Maqfirah, Arifin, Nanda Saputra, and Gamar Al Haddar, “The role of parent-teacher communication for improving children’s achievement,” Cendikia : Media Jurnal Ilmiah pendidikan, https://www.iocscience.org/ejournal/index.php/Cendikia (accessed Aug. 31, 2024).

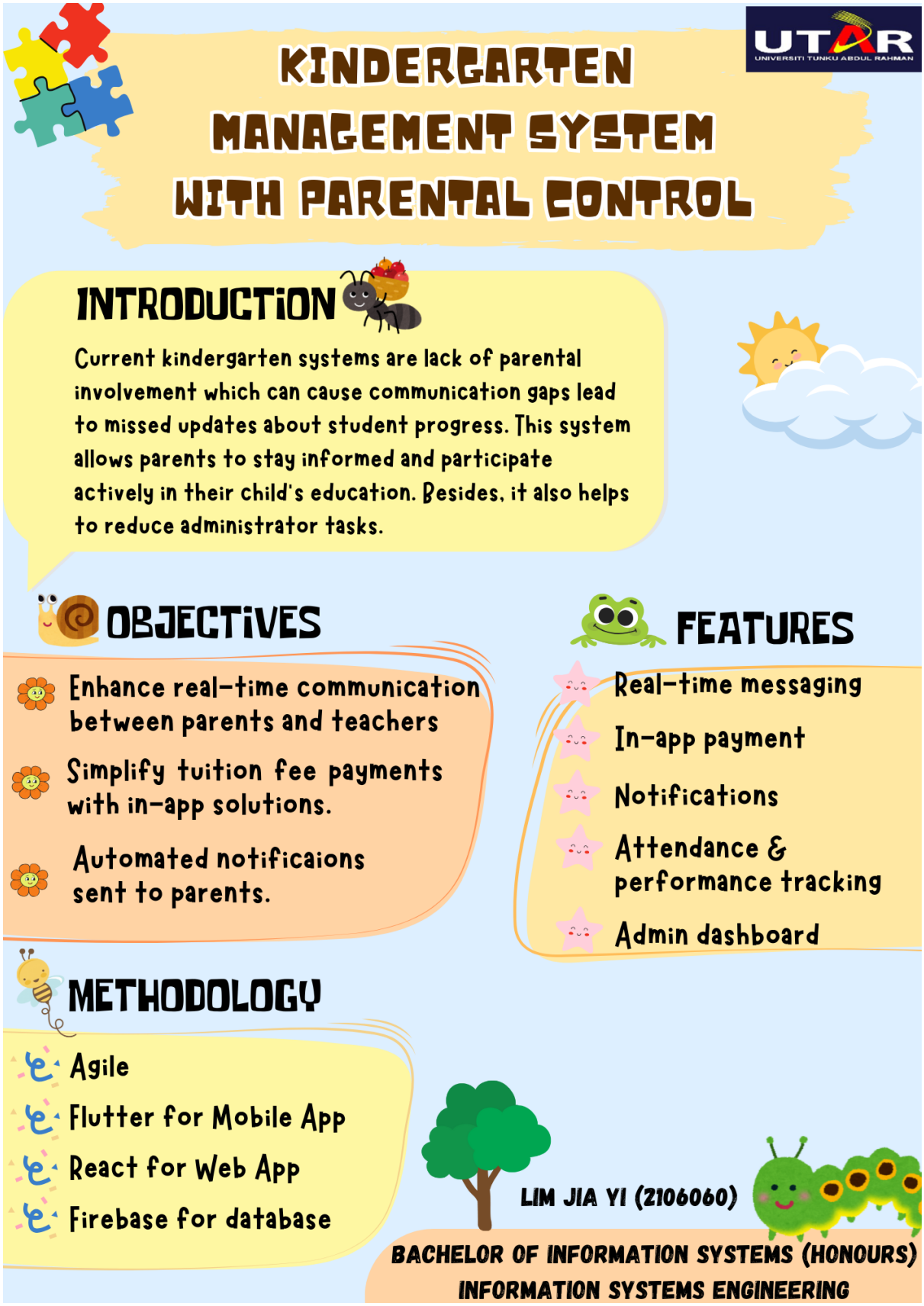
[7] Dataintelo and R. Sharma, “Kindergarten Management Software Market Report | Global Forecast From 2025 To 2033,” *Dataintelo.com*, Jul. 26, 2023. https://dataintelo.com/report/global-kindergarten-management-software-market? (accessed Sep. 20, 2025).

[8] Verified Market Reports, “Kindergarten Management Software Market 2025 in the United States | Demand, Size & Trends Report,” *Verified Market Reports*, Jun. 23, 2025. https://www.verifiedmarketreports.com/product/kindergarten-management-software-market (accessed Sep. 20, 2025).

[9] A. Shimalla, “What is Educational Technology (ed tech), and why should schools invest in it?,” Technology Solutions That Drive Education, https://edtechmagazine.com/k12/article/2024/03/what-educational-technology-ed-tech-and-why-should-schools-invest-it-perfcon (accessed Sep. 9, 2024).

[10] “Kindergarten Management Software: Modern School Solution - TUỔI THƠ,” *Tuoitho.edu.vn*, Feb. 17, 2025. https://tuoitho.edu.vn/en/kindergarten-management-software-modern-school-solution/ (accessed Sep. 6, 2025).

POSTER


KINDERGARTEN MANAGEMENT SYSTEM WITH PARENTAL CONTROL
UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
INTRODUCTION
Current kindergarten systems are lack of parental involvement which can cause communication gaps lead to missed updates about student progress. This system allows parents to stay informed and participate actively in their child's education. Besides, it also helps to reduce administrator tasks.
OBJECTIVES
Enhance real-time communication between parents and teachers
Simplify tuition fee payments with in-app solutions.
Automated notificaions sent to parents.
FEATURES
Real-time messaging
In-app payment
Notifications
Attendance & performance tracking
Admin dashboard
METHODOLOGY
Agile
Flutter for Mobile App
React for Web App
Firebase for database
LIM JIA YI (2106060)
BACHELOR OF INFORMATION SYSTEMS (HONOURS)
INFORMATION SYSTEMS ENGINEERING