KINDERGARTEN MANAGEMENT SYSTEM

By

Liang Kwan Yee

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS) BUSINESS INFORMATION SYSTEMS

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2025

COPYRIGHT STATEMENT

This Final Year Project report is submitted in partial fulfillment of the requirements for the degree of Bachelor of Information Systems (Honours) Business Information Systems at Universiti Tunku Abdul Rahman (UTAR). This Final Year Project report represents the work of the author, except where due acknowledgment has been made in the text. No part of this Final Year Project report 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 sincerely thank and appreciate my supervisor, Dr Nur Syahirah binti Roslan, who fully supported and believed in me. Her willingness to help and give suggestions for the project is greatly appreciated, and her generosity with her time is invaluable. I am genuinely grateful for her support and guidance.

In addition, I want to thank all my friends who have helped me and were willing to give me valuable advice on what they could contribute to my project. Finally, I must thank my family for the love, support, and constant encouragement they have given me throughout the course. With their help, I have been able to achieve such success. Their unwavering faith in me has always motivated and inspired me.

ABSTRACT

This project proposes the development of a comprehensive Kindergarten Management System to streamline administrative operations, enhance communication between teachers and parents, and improve lesson planning for educators. The system aims to address inefficiencies in current manual processes by offering an intuitive and user-friendly web-based solution. Key features include attendance tracking, billing management, lesson planning tools, a shared school calendar, and a communication platform for parents and teachers. The project leverages modern technologies such as PHP, HTML5, CSS, JavaScript, and MySQL to create a robust, scalable, and accessible system. By adopting principles from Extreme Programming (XP), the development process ensures flexibility, high code quality, and alignment with user needs. The proposed system will significantly reduce administrative workload, foster better collaboration between teachers and parents, and empower teachers with tools to create and manage lesson plans effectively. This project aims to deliver a reliable, efficient, and comprehensive solution tailored to the unique needs of kindergartens.

Area of Study: Kindergartens, Web-based System

Keywords: Kindergarten Management System, Educational Technology, Web-based System, Teacher-Parent Communication, Lesson Planning Tools, Early Childhood Education, User-friendly Interface

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

PHP	Personal Home Page
HTML 5	HyperText Markup Language
CSS	Cascading Style Sheets
SDLC	Software Development Life Cycle
XP	Extreme Programming

CHAPTER 1 INTRODUCTION

1.1 Project Background

Owing to advances in technology and the information-saturated society of today, the importance of preschool education in a child's development is widely acknowledged. As stated in [1], early childhood education is a significant public good, the foundation of lifelong learning, and a crucial component of the national education system. In today's fast-paced world, children who regularly attend a high-quality preschool program are better positioned to excel academically [2]. Early education in kindergarten encompasses all the personal, social, and structured experiences a child should acquire during their toddler years. Its goal is to protect, develop, and enrich human potential by imparting skills and accomplishments in physical, mental, and cultural domains, fostering identity and self-respect [3].

Additionally, to create a strong support system that prioritizes the well-being of children and families, the conventional style of managing the overwhelming workload in a kindergarten has made tasks challenging. Thus, building a community environment that fosters resources and services tailored to meet the needs of families and schools, alongside supporting children's overall personality development, the interaction between parents and teachers enables parents to recognize, understand, and address challenges in child-rearing collectively [4]. Moreover, family relationships are part of society and influenced by socioeconomic changes, making planning and promoting family-kindergarten engagement vital [6].Other than that, effectively leveraging information technology to adapt to new educational reforms has become a critical method to improve kindergarten educational quality and enhance operations [7].

As technology advances, teachers increasingly use social media or websites to communicate information about school events, functions, sports activities, and class projects to parents. Research indicates that families find digital communication more effective, immediate, and convenient than traditional methods [8]. Plus, parents' roles in supporting their children in school are now more closely linked to their use of technology. However, surprisingly, limited research has been conducted on parents' online school-related activities [8]. In fact, by utilizing the management system wisely, both management efficiency and operational costs can be improved. While the system

integrates various resources to achieve a paperless workflow, administrative burdens can be reduced, and data can be appropriately managed and modified throughout the process [7]. Therefore, the need for an effective and efficient kindergarten management system is crucial to act as an integrated platform to streamline various administrative and operational tasks as well as foster the collaboration and engagement of parents in children's early education.

Therefore, the need for an effective and efficient kindergarten management system is crucial to act as an integrated platform to streamline various administrative and operational tasks as well as foster the collaboration and engagement of parents in children's early education.

1.2 Problem Statement

1.2.1 Lack of effective communication and collaboration

Teachers frequently find it difficult to involve parents in their children's education in a pragmatic way [9]. The gap in effective communication and collaboration between the teachers and parents will indirectly influence the children's skill development, school outcomes, and their lifelong success [10]. According to [7], the lack of effective communication between parents and the childcare facility staff has caused the parents to become confused and feel disconnected from their children's daily activities in the kindergarten. Moreover, based on the analysis of [11-14], the current system mostly lacks the features of allowing teachers to schedule a meeting to share an announcement verbally.

1.2.2 Inefficient and error-prone data management

Excessive workload has been repeatedly found to be the main obstacle to efficient lesson planning. Due to their overwhelming responsibilities, teachers say they are unable to create thorough lesson plans [15]. The time-consuming process of making good plans causes a lot of stress and can result in shortcuts that lower the quality of instruction, especially for new teachers who need more thorough planning [16].

1.2.3 Inefficient and error-prone data management

In today's fast-paced digital environment, the traditional approaches like paper-based documentation and fragmented communication methods management process in the kindergarten have led to the inefficiency and ineffectiveness in the kindergarten operations. Studies [17-19] highlight that the rely on paper works and physical books to handle numerous amounts of students and staff data is not only time-consuming but also prone to errors. Therefore, this significantly led to a low productivity in handling data and increased difficulty in meeting the demand of precise data management.

To address these issues, there is an increasing need for a comprehensive kindergarten management system that utilizes the technology to streamline complex operations and enhance the communication between the parents and teachers as well as provide a user-friendly interface for all users to leverage things online.

1.3 Motivation

The motivation for proposing the Kindergarten Management System arises from the increasing demand for efficient and streamlined management in early childhood education institutions. Early childhood is a critical period for a child's social, emotional, intellectual, and physical development [18]. However, the current conventional manual and fragmented operation process has slowed down the ability of teachers and administration to focus on delivering a quality and effective learning environment. Due to the fast-paced world nowadays, educators' and parents' overall productivity and satisfaction are significantly affected by several challenges such as time-consuming administrative tasks, lack of real-time communication between teachers and parents, and inefficient lesson planning [19]. Besides, it is challenging to ensure operational transparency and data accuracy without a centralized platform to handle student records, attendance, fee payments, and academic progress.

1.4 Research Objectives

1. To develop a communication platform for teachers and parents that can ease the process of conveying school and children's information.

   a. To implement real-time messaging and notification features that enable the sharing of school announcements, schedules, and event reminders for seamless communication between teachers and parents.
   b. To allow teachers to arrange meetings with parents to discuss and provide feedback on student's progress and participate in school-related discussion.
2. To develop a user-friendly environment that enables teachers to create and manage lesson plans effectively in the Kindergarten Management System
   a. To design a user-friendly interface that allows teachers to draft, edit, and save lesson plans efficiently.
   b. To ease the teaching process by allowing teachers to attach supplementary materials to planning lessons.
3. To develop a comprehensive and user-friendly kindergarten management system that can mitigate the operation in the kindergarten progress.
   a. To develop a web-friendly interface that enables administrators, teachers, and parents to access the system conveniently.
   b. To integrate analytics and reporting tools for monitoring students' overall performance in the kindergarten.

1.5 Project Scope and Direction

The scope of this project is to design and develop a comprehensive Kindergarten Management System that enhances operational efficiency, strengthens communication, and supports teaching activities in Malaysian kindergartens through a user-friendly interface. The system will automate administrative processes such as generating monthly bills for parents, maintaining attendance records, and managing daily activity reports. It will also improve communication by providing a shared school calendar, announcement tools, and features for scheduling parent-teacher meetings. In addition, the system will support teaching and learning by offering a dedicated lesson planning module that enables teachers to draft, edit, save, and attach supplementary materials to lesson plans efficiently. Designed in alignment with Malaysia's early childhood education guidelines, this system will integrate all essential functions into a single platform to streamline daily operations, foster collaboration between teachers and

parents, and enhance the overall educational experience for children. Also, the system is specifically designed to cater to kindergartens with an enrollment capacity of up to 100 students and a teaching staff of up to 20, ensuring optimal performance, scalability, and usability for small to medium-sized institutions.

1.6 Contributions

This project aims to provide a comprehensive solution that enhances the overall operation of the Kindergarten Management System by addressing the pain points of the current system through the following key improvements:

- Improved communication between parents and teachers

  A real-time communication platform is able to solve the gaps between parents and teachers. The instant notification feature can achieve effective communication between these 2 parties. Teachers and administrators can share updates and announcements with parents through dedicated portals or mobile notifications to foster seamless communication and hence results in strengthening trust and promoting active parents' engagement in their child's learning journey.

- Streamlined teachers' lesson planning process

  Implementing a lesson planning feature in the system such as pre-defined templates for creating teaching material can greatly tackle the problem of creating time-consuming lesson plans and lack of flexibility to modify in order to adapt to the dynamic classroom needs.

- Streamlined the administrative operation

  A centralized data management system is significant to strike this objective. Tasks like attendance recording, student enrollment, and fee invoicing are automated and stored in a centralized database to ensure the consistency of data and to ease the data retrieval process.

1.7 Report Organization

This report is organized into seven chapters, each addressing different aspects of the project. Chapter 1 introduces the project by outlining the background, problem statement, motivation, objectives, scope, contributions, and overall structure of the report. Chapter 2 presents the literature review, discussing existing kindergarten management systems such as Anak2U, LittleLives, Illumine, and Taidii, followed by their limitations and a comparison with the proposed solution. Chapter 3 describes the proposed method and approach, including system requirements, design diagrams, entity-relationship modeling, and project timelines, along with potential implementation issues and challenges. Chapter 4 details the preliminary work, covering system setup, architecture design, and initial module development such as login, dashboard, student and teacher management, attendance, lesson planning, and payment features. Chapter 5 explains the system implementation in detail, structured around the roles of administrators, teachers, and parents, with a focus on their respective tasks and system interactions. Chapter 6 provides the system evaluation and discussion, including module testing, results, project challenges, and an assessment of whether the project objectives have been achieved. Finally, Chapter 7 concludes the report with a summary of the work carried out and provides recommendations for future improvements. The report ends with references and appendices that include supporting materials such as the project poster.

CHAPTER 2 LITERATURE REVIEW

2.1 Previous works on Kindergarten Management System

Several websites are designed to integrate the Kindergarten Management System in a web-based form. In this chapter, Anak2U, LittleLives, Illumine, and Taidii are reviewed, and the strengths and weaknesses of the websites are discussed and compared.

2.1.1 Anak2U

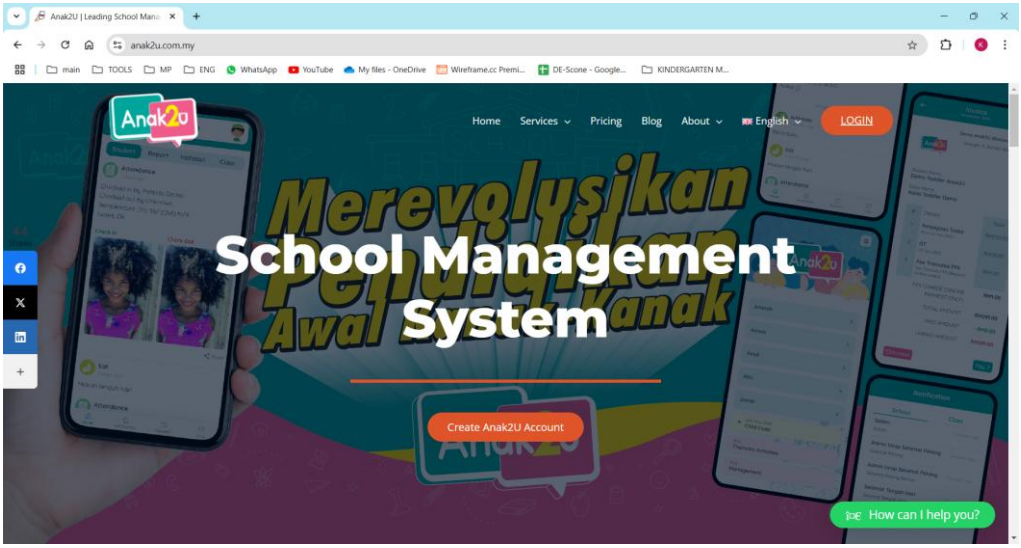


Figure 2.1 Anak2U Website

Anak2U is an end-to-end preschool management platform designed to help Early Education Childcare (EEC) owners manage and run their centers using a digital-centric strategy built-in 2018 [20]. Anak2U offers a range of features. One of the main features is invoice management where the system will automatically send the parent a monthly bill. This feature has greatly saved the parents from being reminded to pay the bill. It has directly eliminated the stress of managing invoices and thus can focus on running their institution. The Admin Portal also includes a feature to automate the generation and distribution of payslips for the teachers and staff and has significantly streamlined payroll management. This feature has reduced manual effort and human error by providing employees with quick and secure access to their salary information. Another important feature of Anak2U is the centralized announcement distribution,

whereby the announcement can reach all parents promptly without getting mixed with unrelated messages, ultimately improving communication efficiency between parents and teachers. The other main feature of this system is the health declaration form and attendance record, which can ensure the safety and well-being of everyone in the institution. This feature has fostered a safe and organized environment by integrating the health monitoring and attendance tracking module into a unified.

Strengths:

1. Provide bilinguals which are English and Bahasa Malaysia.
2. Provide customization capabilities to adapt to different school requirements to fit with various institution business models.
3. Provide live chat

Weakness:

1. Occasional app bugs, such as crashes and payment glitches.
2. No publicly accessible demo or tutorial without account setup.
3. Limited transparency in pricing information.

2.1.2 LittleLives

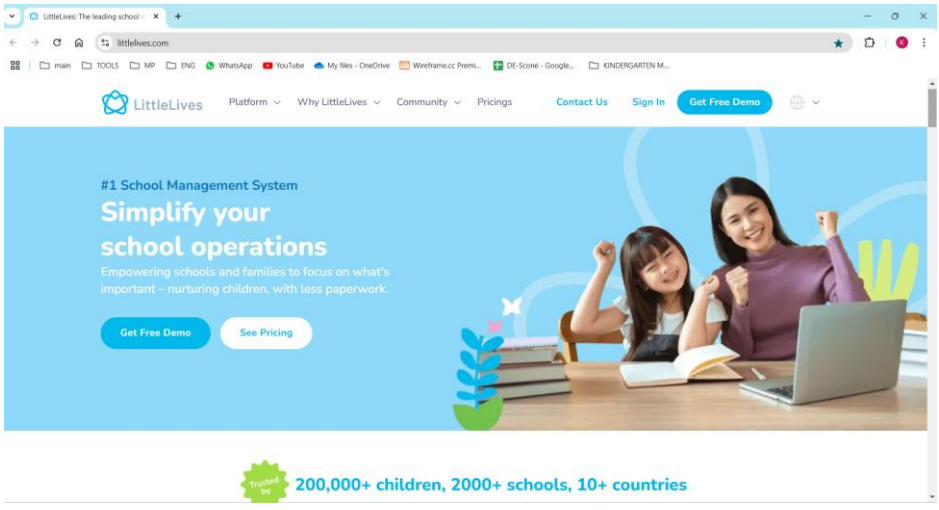


Figure 2.2 LittleLives Website

LittleLives is a platform established in 2011 in Singapore that allows the institution's owner to run their school operations smoothly and aims to streamline

the school management process [21]. It provides a centre operation to manage the entire school in order to stay updated on what happening in school by using Centre Log and Enrolment. In the school management system, one of the key features included is that it can record children's progress during the school's event or curriculum. Parents can monitor their children's school's daily status on the dashboard. All the school events, notices, and announcements are updated in an integrated school calendar. Parents are allowed to synchronize their phone calendars and stay updated on the latest school notification. Besides, LittleLives School Management System has designed a streamlined system to simplify the billing process. Fee reminders will be generated with a detailed fee breakdown for every transaction. This feature relieves the parents' burden of handling money, and the administration can concentrate on providing superior educational experiences to the children. Other than that, this system also provides an important feature that helps the institution visualize the key metrics across students, staff, and administration through an intuitive graphs and charts system to monitor and analyze data related to students, staff, parents, and daily administration. Thus, the childcare centers can use data analytics to simplify their operations to optimize the overall process therefore strengthen the connections of stakeholders.

Strengths:

1. Provides powerful analytics and reporting tools
2. Compliance and audit-readiness
3. Provides multiple languages

Weakness:

1. Need to log in with personal information to get a free demo
2. Does not provide detailed public pricing information

2.1.3 Illumine

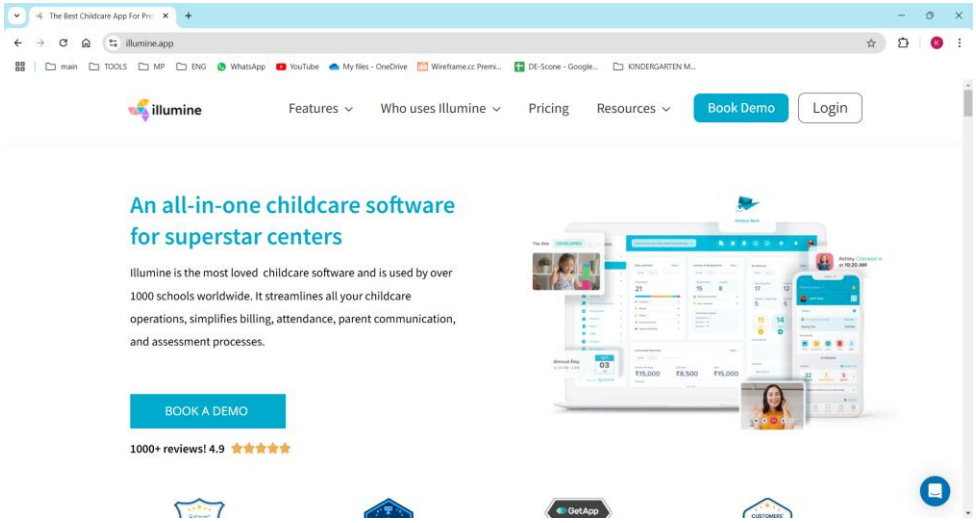


Figure 2.3 Illumine Website

The Illumine website is launched in 2019 which offers a comprehensive childcare management system designed to simplify preschool and daycare center operations. It is the most popular childcare software implemented by over 1000 schools worldwide [22]. Illumine's website provides preschool management software to streamline all preschool operations with various useful features. Student and staff attendance management is one of the key features in this system where both teachers and parents can conduct a contactless check-in and check-out of the child by scanning the QR code provided by the school. Besides, centralized communication is available to ease the interaction between the management, the staff, and the parents. Teachers can record the children's meal consumption in real-time on the system as well as notify parents of naps and diaper changes. Thus, parents are notified of their children's progress and able to collaborate through comments. The system can also keep track of the payments in the company by sending reminders and generating reports. On the other hand, teachers can create and share daily or weekly lesson plans with parents and follow up on the assignments and submissions with this system. Hence, the system can keep track of the child's development and provides parents with tailored feedback on each activity their child participates in at school.

Strengths:

1. Flexible and customizable plans according to requirements are available
2. Teachers can create and plan lessons
3. User-friendly interface

Weakness:

1. Occasional glitches, such as delayed notifications or app crashes.
2. Video upload limitations for long files.

2.1.4 Taidii

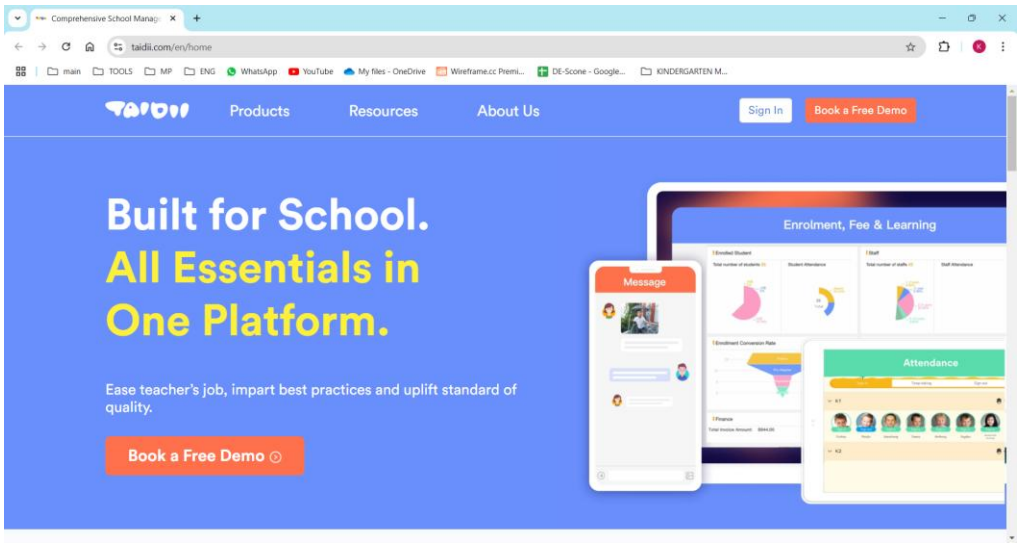


Figure 2.4 Taidii Website

Taidii is founded in 2013 and is an edu-tech company dedicated to building a comprehensive school management system and focuses on streamlining the administration, automating routine tasks, and enhancing parent-teacher communication [23]. This website includes several key features in the school management system. The Student Attendance System by Taidii uses advanced technology to strengthen attendance tracking which integrates Diibot's facial recognition and Bluetooth thermometers for seamless and automated attendance management. Besides, the staff attendance system by Taidii empowers administrators with a centralized platform to manage staff attendance efficiently. Furthermore, Taidii simplifies invoicing, and payment tracking through an efficient finance management system by generating useful

financial reports and using notifications to keep all the stakeholders informed. The system simplifies student admission by integrating custom digital inquiry forms for online submission to ensure seamless handling of inquiries to registrations and ultimately enhance communication and improve enrollment rates. In the aspect of parent communication, Taidii's suite of features enhances communication, engagement, and operational efficiency for schools and parents. By centralizing tools for interaction, event management, and child development tracking, it fosters a seamless and collaborative educational experience.

Strengths:

1. Clear and organized user interface
2. Integrates advanced features (facial recognition)

Weakness:

1. Does not provide multiple languages
2. Usability issues in the app, such as disappearing typed text in chats and password reset failures.

2.2 Limitation of Previous Studies

The reviewed Kindergarten Management System websites—Anak2U, LittleLives, Illumine, and Taidii—have several general limitations that affect their overall effectiveness. One major issue is the limited customization options, as these platforms offer predefined features that may not fully meet the unique needs of every kindergarten, requiring additional costs or technical support for customization. Besides, the reviewed websites do not provide the capability for the instructor or educational workers to schedule a face-to-face meeting with the parents leading to a lack of communication and summary updates on the students to their parents. Finally, the cost and accessibility of these systems can be prohibitive for smaller institutions, as the pricing models may not offer enough flexibility for schools on a tight budget. These limitations suggest the need for improvements to make kindergarten management systems more adaptable, user-friendly, secure, and accessible to a wider range of users.

2.3 Proposed Solution

The proposed Kindergarten Management System is designed as a comprehensive platform that directly addresses the limitations of current systems by improving communication, supporting teaching, and streamlining administration. To enhance communication between teachers and parents, the system integrates a **real-time messaging and notification** module that enables the sharing of school announcements, schedules, and event reminders. This ensures parents remain informed of important updates and allows teachers to arrange parent-teacher meetings conveniently within the platform, strengthening collaboration and trust.

To support teaching and learning activities, the system offers a dedicated **lesson planning module** with a user-friendly interface. Teachers can draft, edit, and save lesson plans efficiently, while the option to attach supplementary materials enhances the effectiveness of the teaching process. This feature reduces the burden of lesson planning, particularly for new teachers, and encourages consistent, high-quality teaching practices.

On the administrative side, the system automates critical processes such as fee invoicing, and attendance management. By centralizing data in a secure database, the system ensures accuracy, reduces redundancy, and enables efficient data retrieval. Additionally, analytics and reporting tools are embedded to allow administrators to monitor student performance and generate insights for better decision-making. The scope of the system is tailored to Malaysian kindergartens, supporting up to 100 students and 20 teachers, ensuring usability, scalability, and alignment with local educational needs.

By combining these features into one integrated platform, the proposed solution directly meets the objectives and contributes to addressing the main pain points: closing communication gaps, reducing administrative burdens, and providing teachers with better tools to manage lesson planning. Ultimately, this system enhances efficiency, promotes parental engagement, and creates a supportive environment for early childhood education.

2.4 Comparison Between Similar Systems

	ANAK2U	LITTLELIVES	ILLUMINE	TAIDII	PROPOSED KINDERGARTEN MANAGEMENT SYSTEM
Generate Monthly Bills to parent	/	/	/	/	/
Send out announcements to parents instantly	/	X	/	X	/
Attendance Record	/	X	/	/	/
Shared school calendar		/	X	/	/
Plan and create lessons	/	X	/	X	/
Scheduling Parent-Teacher Meeting	X	X	X	X	/
Daily Activity Reports	/	/	/	/	/

Table 2.1 Comparison Between Similar System

CHAPTER 3 SYSTEM METHODOLOGY

The methodology outlines the systematic approach undertaken to design, develop, and implement the Kindergarten Management System. It functions as a methodical framework to guarantee that the project meets its goals successfully and efficiently.

3.1 System Requirement

3.1.1 Hardware Requirements

The hardware involved in this project is a computer, which plays a crucial role in the development and maintenance of the Kindergarten Management System. The computer can execute the browsers, integrated development environments (IDEs), and debugging tools required for writing and testing the code in order to construct and maintain the website. Its fast performance makes it easy to navigate through lengthy, detailed manuals and paperwork, ensuring that problems may be found and fixed promptly. Furthermore, the computer's multitasking capabilities make it easier to manage several jobs at once, including database integration, UI/UX design, and coding.

By leveraging the computer's capabilities, a powerful, easy-to-use, and responsive website that forms the basis of the Kindergarten Management System and satisfies the administrative, teachers, and parental communication requirements.

Description	Specifications
Model	Acer Aspire A515-56
Processor	11th Gen Intel(R) Core (TM) i5-1135G7 @ 2.40GHz
Operating System	Windows 11
Graphic	Intel® Iris ® Xe Graphics
Memory	8GB DDR4 RAM
Storage	475.67GB

Table 3.1 Specification of laptop

3.1.2 Software Requirements

To develop a successful and operational Kindergarten Management System, the required software tools and platforms are extremely significant to ensure the functionality and usability of the whole system. The software specifications are divided into the applications and the programming languages to be used for developing this system.

1. PHP Programming Language

Figure 3.1 PHP

PHP (Personal Home Page) is used to create the Kindergarten Management System. PHP is a server-side programming language known as Hypertext Preprocessor [24]. It is a widely used open-source general-purpose scripting language that can be integrated into HTML and is particularly well-suited for web development. It is an open-source server-side scripting language that is primarily used for creating online applications. Besides, its simple syntax enables quick development with straightforward debugging during the programming phase [25]. One of the reasons for implementing PHP in the development is it enables the creation of dynamic web pages that interact with databases and provide customized content for users. Besides, PHP integrates smoothly with MySQL, which is one of the most popular relational database systems, and allows for efficient storage and retrieval of data, such as student attendance, billing records, and lesson plans. Thus, PHP is essential for developing a scalable and user-friendly website for the Kindergarten Management System [24].

2. HTML5

Figure 3.2 HTML

HTML5 (HyperText Markup Language) is the foundational language for creating the structure of web pages and it makes it essential for developing a kindergarten management system. It is significant for building the structure and content of a website due to the dynamic features of HTML5 Web Components allow developers to create and define their own HTML elements, also known as custom elements [25]. Moreover, HTML4 works seamlessly across all modern web browsers and hence ensures the Kindergarten Management System is accessible on different devices, such as desktops, tablets, and smartphones without compatibility problems [26].

3. CSS

Figure 3.3 CSS

CSS (Cascading Style Sheets) is responsible for the design and visual presentation of a website which is important to create an appealing and user-friendly Kindergarten Management System [27]. It describes how the HTML elements are to be displayed on the webpage screen such as the colors, font styles, and spacing. CSS offers a vast array of design options that can enhance the overall user experience and increase the visual appeal of the website for visitors. Other than that, CSS provides the convenience for developers to create and apply consistent style across all pages of the system to ensure a uniform experience for users [28]. CSS also allows the separation of HTML content from

its visual representation enabling the developers to update the design without altering the HTML structure.

4. MySQL

Figure 3.4 MySQL

MySQL is a powerful, open-source relational database management system widely used in web applications. It is essential for the Kindergarten Management System to store, manage, and retrieve data efficiently. This is because the Kindergarten Management System needs to store a large amount of data such as staff information, attendance logs, lesson plans and daily activity reports, and so on. MySQL provides a structured and scalable solution to handle this data securely and reliably [29]. One of the important advantages is it can integrate seamlessly with PHP allow for smooth data handling and enable dynamic content updates in the database. MySQL's relational database capabilities also allow the programmer to define relationships between the data tables such as the linking between students and parents resulting in a consistent and integrity of data [29].

3.2 System Design Diagram

3.2.1 Site Map

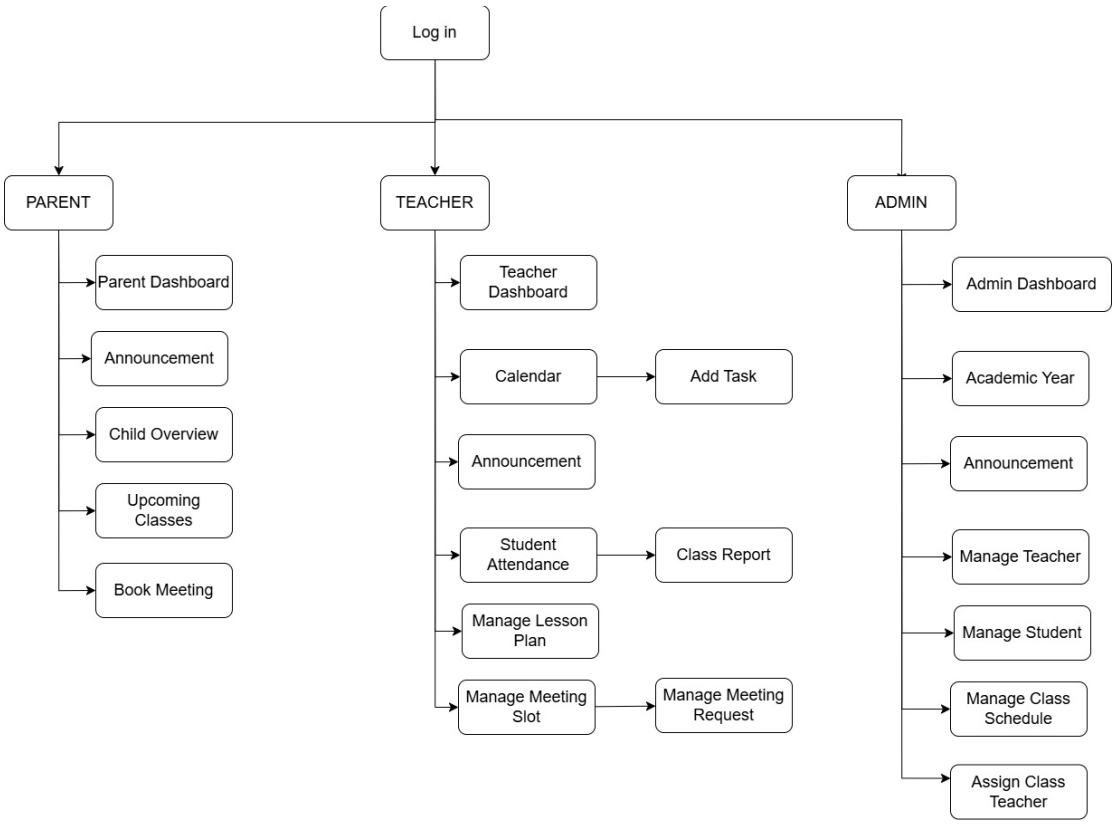


Figure 3.6 Site Map

A site map is a structured or visual illustration of a website or application that shows its pages, features, and navigation. It is frequently used to give a clear picture of how various system components are arranged and connected during the design and development phases. The site map for this kindergarten management system shows the hierarchy according to the three primary user roles of administrator, teacher, and parent. After logging in, users are taken to the dashboards that correspond to their roles. The Admin Dashboard and tools for managing users and classes are accessible through the Admin Page. Under "Manage Users," administrators can add new teachers and students as well as their access lists. Under "Manage Classes," administrators can add subjects with a subject list, assign teachers to classes, and establish new classes with a class list. Users are taken to the dashboards that correspond to their roles after logging in.

The purpose of the Teacher Page is to assist educators in handling their teaching tasks. They can develop lesson plans, control their time slot for parent meetings, and view the classes they have been assigned from the Teacher Dashboard. Parents' requests for meetings can also be handled directly by teachers via the system.The Parent Site allows parents to interact with teachers and keep an eye on their child's academic development. Furthermore, parents can view the list of upcoming classes of their child' attendance record as well as the upcoming lesson plans and classes through the Parent Dashboard. In order to facilitate communication between the home and the school, parents can also schedule meetings with teachers.

Overall, this site map provides an user-friendly, role-based navigation structure that makes sure any kind of user can locate and control the functions that are applicable and essential to them.

3.2.2 Use Case Diagram

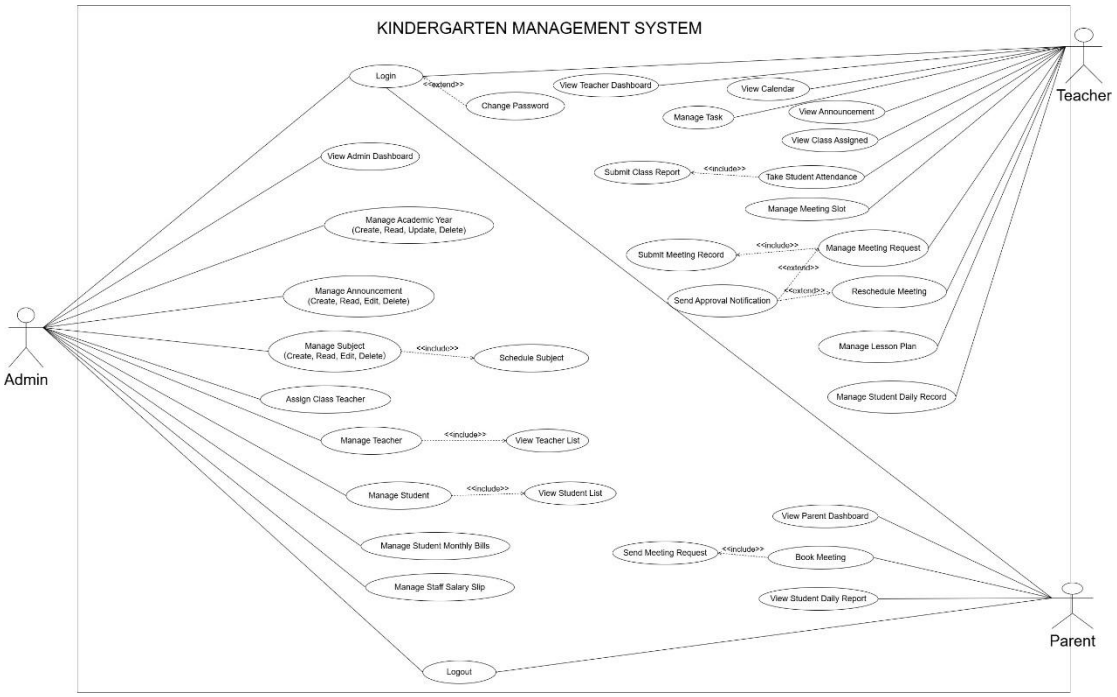


Figure 3.7 Use Case Diagram

A use case diagram visually represents a system's functionality from a user's perspective. In the use case diagram shown above, the system involves 3 main user roles: administrators, teachers, and parents. For administrators, the system enables comprehensive academic year management through CRUD operations. Continue with full control over the announcements and user management, which includes adding

teachers and students and viewing the user lists. The teacher interacts through their dashboard to handle meeting requests, while parents are limited to booking meetings with teachers. The diagram clearly shows the role-based access boundaries whereby the admin has the broadest system privileges. While it effectively covers core administrative and communication functions, the fundamental educational components like attendance tracking, lesson planning,, and payment processing have suggested that these may be handled in separate diagrams or represent future development phases. This diagram serves as a valuable foundation for the system development to clarify the structures and workflow requirements during the development.

3.2.3 Block Diagram

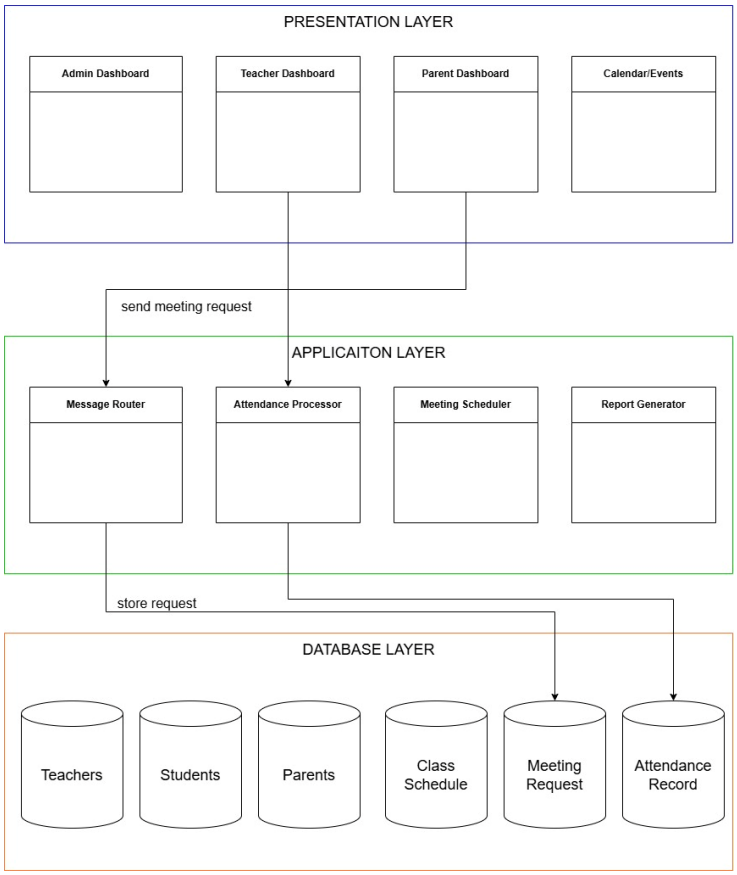


Figure 3.8 Block Diagram

The block diagram is divided into three separate levels that function as a single unit and provides an ordered summary of the architecture of the Kindergarten Management System. The Presentation Layer is the user interface at the top, with specific portals for parents, teachers, and administrators as well as a shared calendar

system for scheduling. From tracking student progress to requesting meetings, this layer manages all direct user interactions.

The middle application layers containing important processing components like the Message Router for handling communications, Attendance Processor for handling attendance data, Meeting Scheduler for scheduling appointments, and Report Generator for creating academic evaluations. Before data enters the database, these modules make sure that business logic is executed correctly and that data is validated. The Database Layer serves as the system's backbone, safely storing all necessary data in well-structured tables, such as class schedules, teacher profiles, student records, parent information, and attendance logs.
Through this architecture, data flows dynamically in both directions: system replies and retrieved information flow upward back to the user interfaces, while user inputs go downward from display to database levels for storage.

Significant benefits of this three-tier design include centralized data access for enhanced security, flexible scalability for future improvements, and explicit responsibility separation for simpler maintenance. All stakeholders can clearly see module boundaries for implementation, administrators can follow user processes, and non-technical staff can comprehend the overall system structure thanks to the color-coded depiction with named linkages.

3.2.4 Entity Relationship Diagram

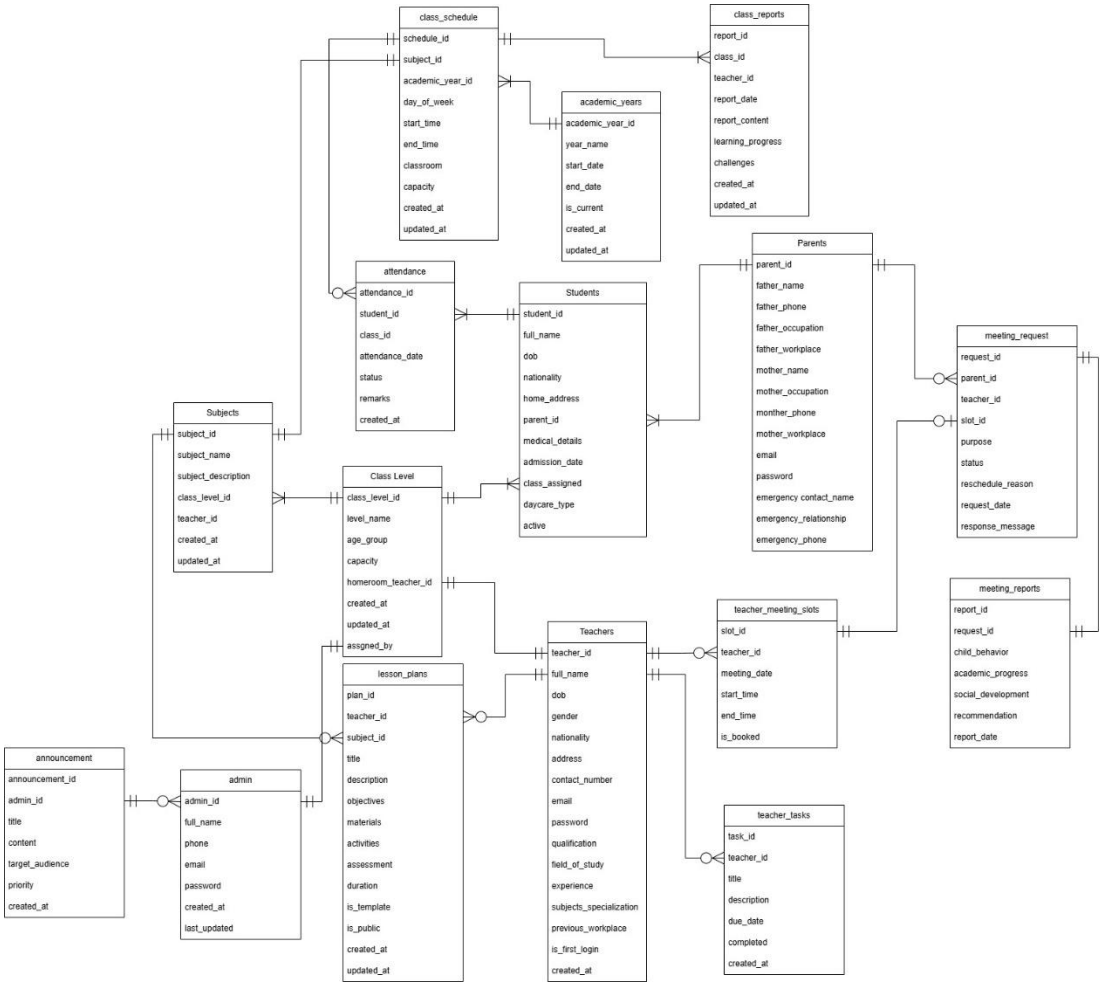


Figure 3.9 Entity Relationship Diagram (ERD)

The Entity Relationship Diagram (ERD) above illustrates the database structure for the KinderEase system, which manages a kindergarten system. The system revolves around several core entities and their relationships. The class_schedule table is responsible for the timetable management. It tracks the subject schedule with day, time, classroom, and capacity, which links to an academic year. The academic_years table defines the academic period, and the class_level organizes students into groups by age and capacity.

The student data in KinderEase is stored in the students table, which includes personal details, medical information, and links to their parents**,** where guardian and emergency contact information is maintained. The attendance records are tracked in the attendance table. The teachers are managed through the teachers table, which contains

professional qualifications and specialization details. Teachers are able to create their tasks and manage their own meeting slots comprehensively. Plus, teachers can also create their own lesson plan related to the subjects assigned to them and mark the lesson plan as public or private. The subjects are defined in the subjects table, linking to class levels and teachers. The student

record is managed by the attendance table with the class report documenting student progress and challenges are handled by the teachers.

Besides, the communication between parents and teachers is facilitated through meeting_request, which records the meeting purposes and statuses. The system also includes an announcement feature form admin communications and admin table for administrator accounts. This ERD reflects a robust structure for managing academic operations, student-teacher interactions, and institutional administration efficiently.

3.3 Timeline

3.3.1 Timeline FYP1

Activity		Period													
		W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
INTRODUCTION	Project Background														
	Problem Statement														
	Research Objectives														
	Project Scope														
	Impact, Significance, and Contribution														
LITERATURE REVIEWS	Review on Similar Website														
	Compare Similar Systems														
	System Design														

PROPOSED APPROACH	Timeline														
	Future Development Plan														
CONCLUSION	Design Prototype														
	Finalize FYP1 Report														
	Test Prototype														
	FYP1 Demonstration														

Table 3.2 FYP1 Timeline

3.3.2 Timeline FYP2

Activity		Period													
		W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
DEVELOPMENT PHASE	Develop each module														
	Fix Bugs														
IMPLEMENTATION PHASE	Develop to Target user														
	Collect Feedback														
	Fix Bugs and enhance system														
EVALUATION PHASE	Evaluate the system														
	Further enhancement														
	Submission of system														
	FYP2 Demonstration														

Table 3.3 FYP2 Timeline

3.4 Implementation issues and challenges

The implementation of the Kindergarten Management System may encounter several key challenges. First, data migration poses a significant issue, as transferring existing records of students, parents, and staff into the new system can risk data loss, duplication, or inconsistencies if not handled carefully. In addition, system integration is another challenge, since the platform may need to connect with external tools such as payment gateways, accounting software, or communication channels, where compatibility and interoperability issues could arise. Data security and privacy also remain a major concern, as the system handles sensitive information about children, parents, and financial transactions. Strict measures such as encryption, access control, and compliance with data protection laws must be implemented to safeguard against breaches. Lastly, scalability and performance are critical for long-term success. As the number of students and staff grows, the system must be able to handle increased loads efficiently without compromising speed or reliability, ensuring consistent usability over time.

CHAPTER 4 Preliminary Work

4.1 Setting up

4.1.1 Web Browser

Users will need a modern, updated web browser to access the system effectively. It is recommended to use the la

- Google Chrome
- Mozilla Firefox
- Apple Safari
- Microsoft Edge

Using an up-to-date browser ensures better website compatibility, security, and overall performance

4.1.2 Local Web Server

The Kindergarten Management System will be hosted on a local server environment for development and testing purposes.

We recommend using XAMPP which bundles:

- Apache (web server)
- MySQL (database management system)
- PHP (server-side scripting language).

4.1.3 Source Code

The source code for the Kindergarten Management System must be downloaded from the designated repository. It includes all necessary files such as HTML, CSS, JavaScript, and PHP scripts required to run and maintain the system.

4.1.4 Database Management Tool

For managing the system's database, it is recommended to use phpMyAdmin, a user-friendly, web-based interface for handling MySQL databases.

phpMyAdmin typically comes bundled with XAMPP and can be accessed by visiting:

http://localhost/ after starting the local server.

4.1.5 Setting and Configuration

Users and developers must properly configure the software components in order to access and utilize the Kindergarten Management System. This section offers a thorough how-to for configuring and setting up the required parts for a seamless experience.

4.2 System Architecture Diagram

4.2.1 Login Account

Use Case Description

Use Case Name: Login	**ID:** 1	**Importance Level:** High
Primary Actor: Admin/Teacher/Parent	**Use Case Type:** Detail, Essential	
Stakeholders and Interests: Admin/Teacher/Parent – wants to login to own dashboard to view specific information		
Brief Description: This use case describes users to login with specified email and password.		
Trigger: User clicks “Log In” on the Home screen of the web page **Type:** External		
Relationship **Association:** Admin/Teacher/Parent **Include:** **Extend:** - **Change Password for First-Time Login** (if password reset is required) - **Forgot Password** (if "Reset Password" option is triggered)		
Normal Flow of Events: 1. User navigates to the KinderEase Website Home Page. 2. User clicks on the "Log In" button in the website header. 3. The system displays the Login Form, including: 4. Email (required)		

5. Password (required) 6. CAPTCHA Verification (to prevent bots) 7. User enters their email and password and completes the CAPTCHA. 8. The system validates the input: - Checks if the email exists in the database. - Verifies if the password matches the stored hash. - Confirms the CAPTCHA is correct. 9. If credentials are valid, the system: - Identifies the user's role (Admin/Teacher/Parent). - Redirects the user to their respective dashboard. 10. The system displays the role-specific dashboard with appropriate permissions.
Sub Flows: • **First-Time Login:** - If the user logs in for the first time, the system may enforce a **password change** (extended use case). • **Session Management:** - The system maintains a **secure session** until logout or timeout.
Alternate/Exceptional Flows: 5a. Invalid Input: - Email empty: "Please Fill Up this Field" - Non-existing Email account: "User Not Found" - Wrong Password: "Invalid Password" - Wrong Captcha: "Invalid CAPTCHA code"

Table 4.1 Login Account Use Case Description

Activity Diagram

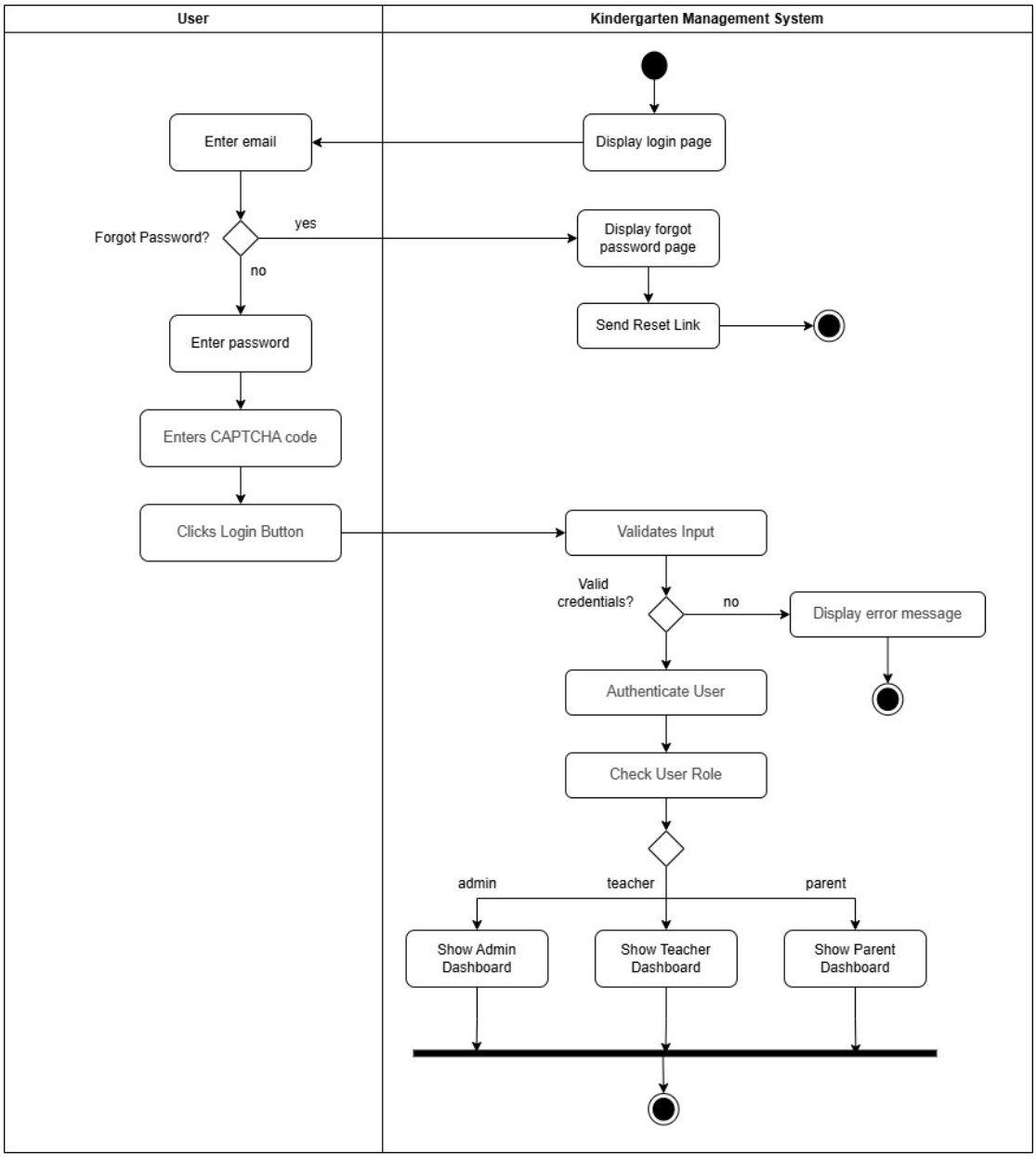


Figure 4.1 Login Account Activity Diagram

4.2.2 View Dashboard

Use Case Description

<table>
<tr><td colspan="2">Use Case Name: View Dashboard</td><td>ID: 2</td><td>Importance Level: High</td></tr>
<tr><td colspan="2">Primary Actor:
Admin/Teacher/Parent</td><td colspan="2">Use Case Type: Detail, Essential</td></tr>
<tr><td colspan="4">Stakeholders and Interests:
• Admin: Shows kindergarten basic analytics and announcement
• Teacher: Requires class schedules, student lists and announcement
• Parent: Wants child-specific progress, attendance, and payment status.
• System: Must load data efficiently and enforce role-based permissions.</td></tr>
<tr><td colspan="4">Brief Description: After successful login, the system displays a personalized dashboard based on the user's role (Admin/Teacher/Parent). The dashboard dynamically loads role-specific widgets and data while maintaining a consistent layout framework.</td></tr>
<tr><td colspan="4">Trigger: User completes successful login (Use Case: Login)
Type: External</td></tr>
<tr><td colspan="4"><u>Preconditions:</u>
1. User is authenticated.
2. User's role (Admin/Teacher/Parent) is identified.
<u>Postconditions:</u>
1. Correct dashboard is displayed with role-appropriate data.
2. User session is active.</td></tr>
<tr><td colspan="4">Normal Flow of Events:
1. System identifies user's role from session data.
2. System fetches role-specific data:
• Admin: School analytics, teacher/student counts, system alerts.
• Teacher: Assigned classes, today's schedule, pending assignments.
• Parent: Child's attendance, recent grades, upcoming events.
3. System renders dashboard components:
• Header with user's name/role.
• Sidebar menu (role-filtered).
• Widgets</td></tr>
</table>

4. User interacts with dashboard (e.g., clicks on a grade report).
Alternate/Exceptional Flows: **3a. No Data Available**: Displays empty state (e.g., "No classes assigned" for a new Teacher). **3b. Session Expired:** Redirects to login screen: *"Your session expired. Please log in again."*

Table 4.2 View Dashboard Use Case Description

Activity Diagram

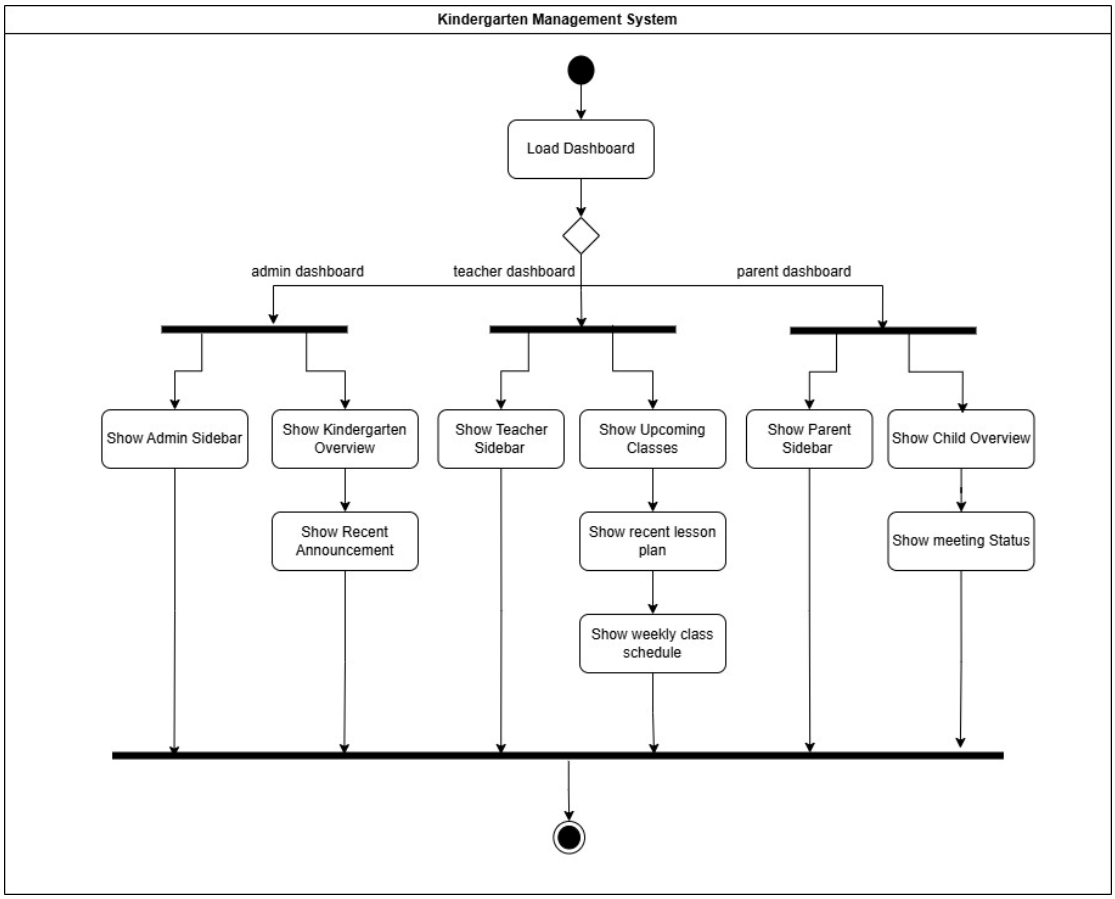


Figure 4.2 View Dashboard Activity Diagram

4.2.3 Manage Academic Year

Use Case Description

<table>
<tr><td>Use Case Name: Manage Academic Year</td><td>ID: 3</td><td>Importance Level: High</td></tr>
<tr><td>Primary Actor: Admin</td><td colspan="2">Use Case Type: Detail, Essential</td></tr>
<tr><td colspan="3">Stakeholders and Interests:
• Admin: Needs to configure academic years for school operations.
• System: Must ensure no overlapping/conflicting academic years.</td></tr>
<tr><td colspan="3">Brief Description:
• Allows Admins to add, modify, or delete academic years in the system.</td></tr>
<tr><td colspan="3">Trigger: Admin clicks on “Academic Year” from the sidebar</td></tr>
<tr><td colspan="3">Normal Flow of Events:
1. Admin navigates to Academic Years from the Admin Sidebar
2. The system shows the current and past academic year with start date and end date
3. Admin clicks “Add New Academic Year”
4. System show a form with:
• Year Name
• Start Date
• End Date
• Set to Current Year
5. Admin submits the form
6. System validates on the date to avoid overlapping with existing years and ensures end date is later that start date
7. System saves the new academic year
8. System show confirmation message</td></tr>
<tr><td colspan="3">Alternate/Exceptional Flows:
Edit Academic Year:
• Trigger: Admin clicks "Edit" on an existing year.
• Changes: Steps 3–7 apply, but with pre-filled form data.
• Validation: Edits don’t conflict with other years (except itself).
Delete Academic Year:</td></tr>
</table>

• Trigger: Admin clicks "Delete". • System checks: No dependent records (e.g., classes, schedules). • Confirmation dialog: - "Are you sure you want to delete this academic year?" - If confirmed: System removes the year and updates references.

Table 4.3 Manage Academic Year Use Case Description

Activity Diagram

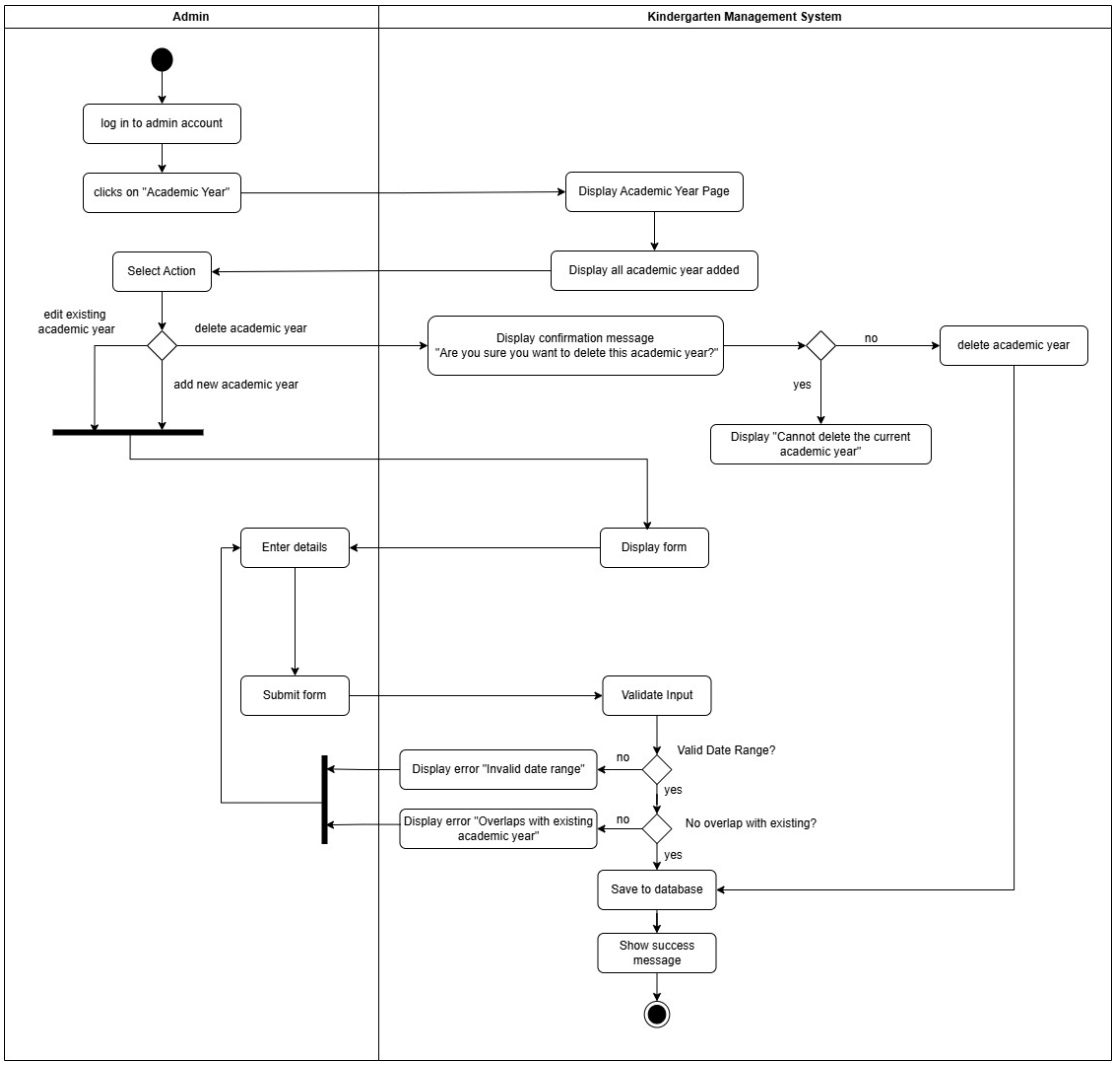


Figure 4.3 Manage Academic Year Activity Diagram

4.2.4 Manage Teacher Account

Use Case Description

Use Case Name: Add Teacher Account	**ID: 4**	**Importance Level:** High
Primary Actor: Admin	**Use Case Type:** Detail, Essential	
Stakeholders and Interests: • **Admin:** Needs to manage teacher accounts and data. • **Teacher:** Receives account credentials and expects accurate profile data. • **System:** Must ensure data integrity and security.		
Brief Description: Allows Admins to: • Add new teachers (sending a random password via email). • View the teacher list and individual profiles. • Edit teacher details. • Delete teacher accounts.		
Trigger: Admin clicks on Add Teacher from the Sidebar		
Preconditions: 1. Admin is logged in. 2. Admin has proper authorization to access teacher management functions **Postconditions:** 1. New teacher account is created in the system 2. Teacher receives login credentials via email 3. Teacher is marked for first-time login password change		
Normal Flow of Events: 1. Admin navigates to the Add Teacher page 2. System displays the teacher admission form 3. Admin enters the following teacher information: • Personal Details: • Full Name • Date of Birth • Gender		

- Nationality
- Home Address
- Contact Number
- Email Address
- Professional Information:
- Highest Qualification
- Field of Study
- Years of Experience
- Subjects Specialization
- Previous Workplace

4. Admin submits the form
5. System validates all input fields
6. System checks if email is unique
7. System generates a secure random password
8. System creates new teacher account
9. System sends email to teacher with login credentials
10. System redirects to teacher list page with success message

Alternative Flows:

1. Invalid Input:

- System displays error message for invalid fields
- Returns to form with previously entered data
- Highlights invalid fields

2. Email Already Exists:

- System displays "Email already exists" message
- Returns to form with previously entered data
- Admin must use different email

3. Email Sending Fails:

- Account is still created
- System displays warning that email couldn't be sent
- Admin may need to manually communicate credentials

Exception Flows:

1. Database Connection Error: • System displays database error message • Logs error details • Returns to form **2. Server Error:** • System displays general error message • Logs error details • Returns to form

Table 4.4 Manage Teacher Use Case Description

Activity Diagram

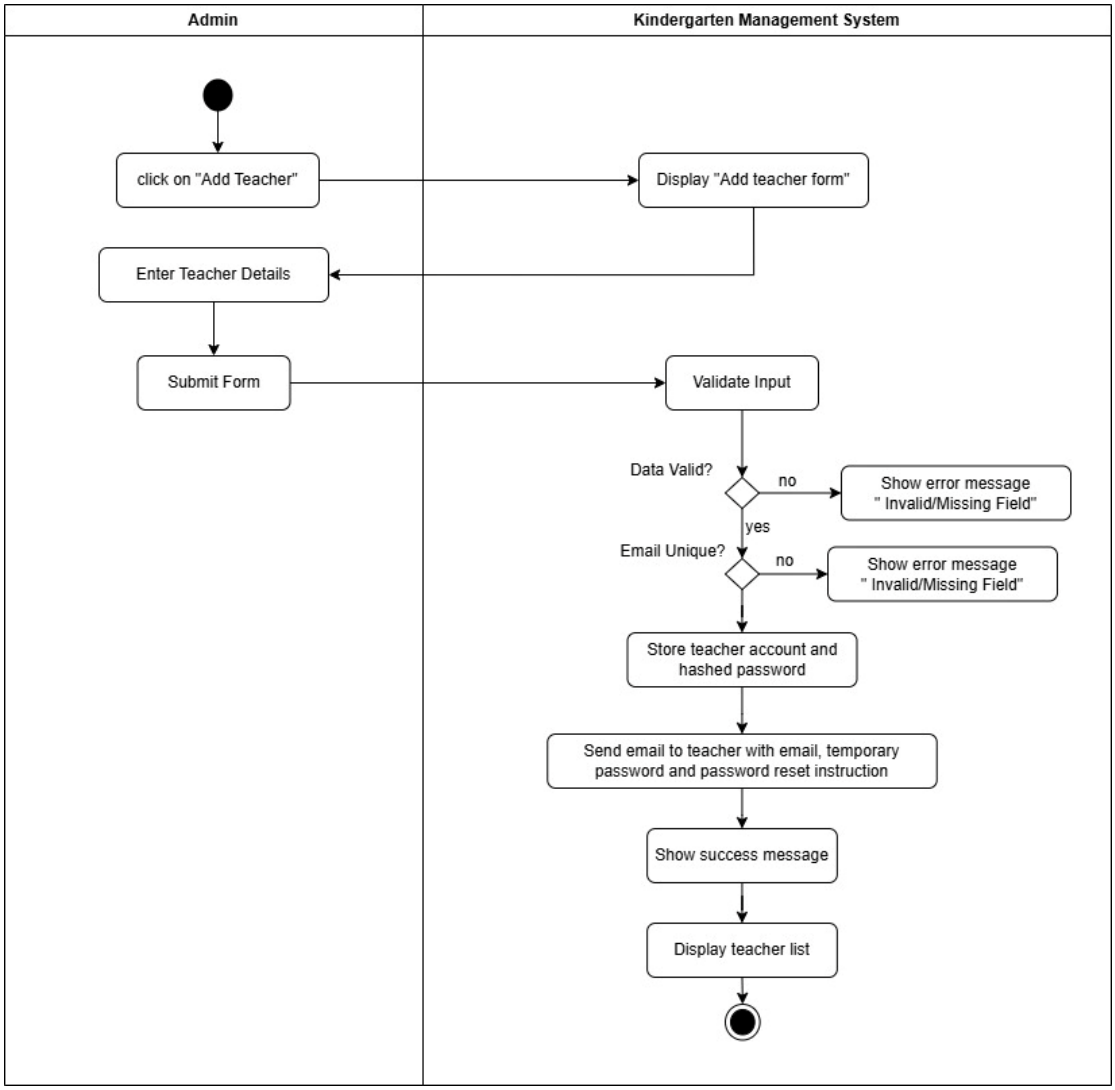


Figure 4.4 Manage Teacher Activity Diagram

4.2.5 Manage Announcement

Use Case Description

Use Case Name: Manage Announcement	**ID: 5**	**Importance Level:** High
Primary Actor: Admin	**Use Case Type:** Detail, Essential	
Stakeholders and Interests: • **Admin:** Needs to broadcast important information efficiently. • **Teachers/Parents:** Require timely, relevant announcements. • **System:** Must deliver announcements to correct user groups.		
Brief Description: This use case allows admins to create announcements with audience targeting to teachers, parents or both. Admin is also allowed to edit or delete existing announcements and ensure the announcements appear correctly on Teacher and Parent dashboards.		
Trigger: Admin clicks on Announcement from the Sidebar		
Preconditions: 1. Admin is logged in. **Postconditions:** **1.** New announcement is visible to targeted users. **2.** Edits/deletions are reflected in real-time.		
Normal Flow of Events: **1.** Admin clicks "Announcement" from the sidebar **2.** System displays all announcement created previously **3.** Admin clicks on "Create Announcement" **4.** System displays a form with • Title • Content • Target Audience (Teachers Only, Parents Only, All Users) • Priority (High/Medium/Low) 5. Admin submits the form		

6. System validates required fields

7. System store announcement in database

Alternate Flows

View Announcement List (Admin):

- Displays all announcement created

Edit Announcement:

1. Admin clicks "Edit" on an announcement.
2. System pre-fills the original form.
3. Admin modifies fields (e.g., changes audience from "Parents" to "Both").
4. On submit:
    - Updates database record
    - Triggers real-time updates for affected users.

Delete Announcement:

1. Admin clicks "Delete".
2. System shows confirmation:
    - *"Delete '[Announcement Title]'? Targeted users will no longer see this."*
3. **If confirmed:**
    - Delete from database
    - Removes from user dashboards.

View Announcement (Teacher/Parent Side):

1. User logs in.
2. System checks user role and fetches:
    - Active announcements where audience matches their role.
3. Displays announcements in a priority-sorted feed:
    - High-priority items pinned to top.
    - Expired announcements automatically hidden.

Exceptional Flows:

- **Empty Title/Content:** *"Title and content are required."*
- **Past Expiry Date:** *"Expiry date must be in the future."*

Table 4.5 Manage Announcement Use Case Description

Activity Diagram

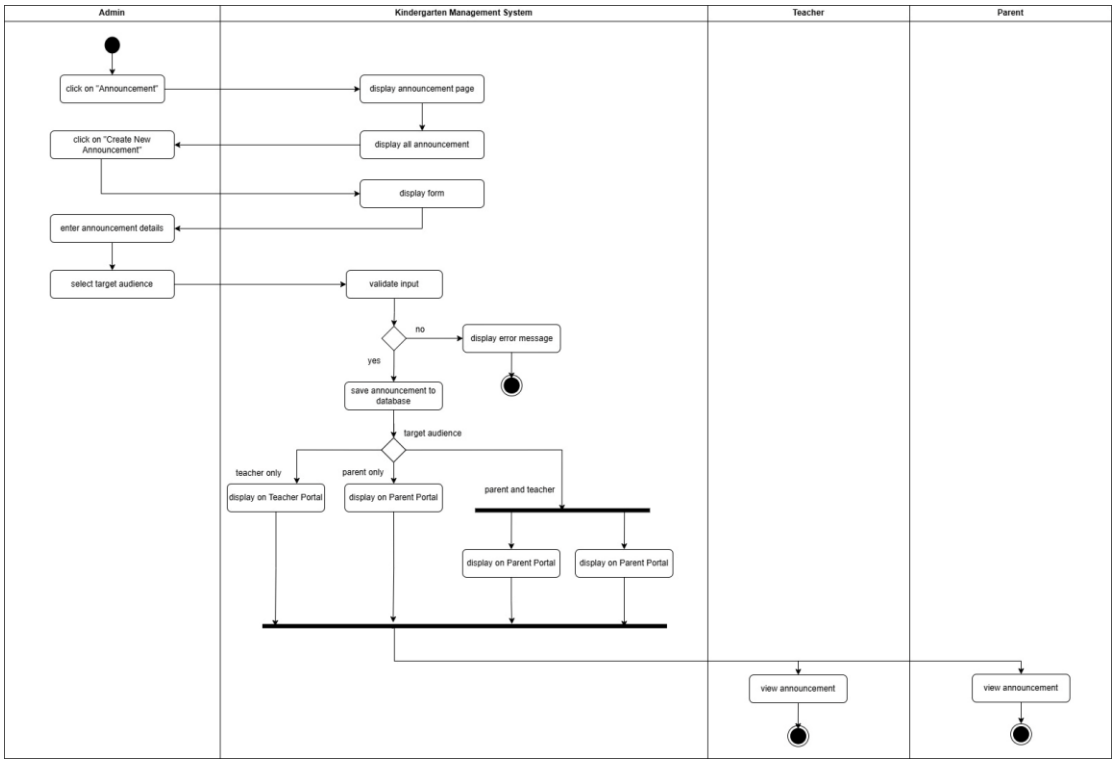


Figure 4.5 Manage Announcement Activity Diagram

4.2.6 Manage Student

Use Case Description

<table>
<tr><td colspan="2">Use Case Name: Manage Students</td><td>ID: 6</td><td>Importance Level: High</td></tr>
<tr><td colspan="2">Primary Actor: Admin
Secondary Actor: Parent, System</td><td colspan="2">Use Case Type: Detail, Essential</td></tr>
<tr><td colspan="4">Stakeholders and Interests:
Admin: Needs to enroll students efficiently and link them to parents.
Parent: Requires access to their child’s academic records.
System: Must ensure data integrity and automate account provisioning.</td></tr>
<tr><td colspan="4">Brief Description: This use case allows administrators to perform comprehensive student management, including adding new students, viewing student profiles, and updating student information.</td></tr>
<tr><td colspan="4">Trigger: Admin clicks on “Add Student” form the sidebar</td></tr>
<tr><td colspan="4"><u>Preconditions:</u>
Administrator is logged into the system
Valid academic year is set
Class levels are configured
Postconditions:
Student records are accurately maintained
Parent accounts are properly linked
Class assignments are updated
System logs are updated</td></tr>
<tr><td colspan="4">Normal Flow of Events:

Add New Student

1. Admin selects "Add New Student" option
2. System displays registration form
3. Admin enters:
Student personal details
Parent information
Emergency contacts</td></tr>
</table>

- Academic enrolment details

4. System validates inputs
5. System creates parent account
6. System registers student
7. System sends credentials to parents

View Student Profile

1. Admin selects student from list
2. System displays:
    - Student's personal information
    - Current class assignment
    - Medical details
    - Parents' contact information
    - Emergency contacts
3. Admin can navigate to edit profile

Edit Student Information

1. Admin initiates edit from profile view
2. System displays editable form
3. Admin modifies required information
4. System validates changes
5. System updates records
6. System confirms successful update

Alternative Flows:

1. Data Validation Errors

- System highlights invalid fields
- Displays error messages
- Preserves valid data
- Returns to form

2. Class Capacity Issues

- System checks class capacity
- Notifies if class is full
- Suggests alternative classes
- Prevents over-enrollment

3. Email System Failure

- System completes registration
- Logs email failure
- Notifies admin
- Provides manual communication option

Exception Flows:

1. Database Connection Error

- System displays error message
- Logs error details
- Returns to previous screen

2. Invalid Student ID

- System redirects to student list
- Displays error message

Table 4.6 Manage Student Use Case Description

Activity Diagram

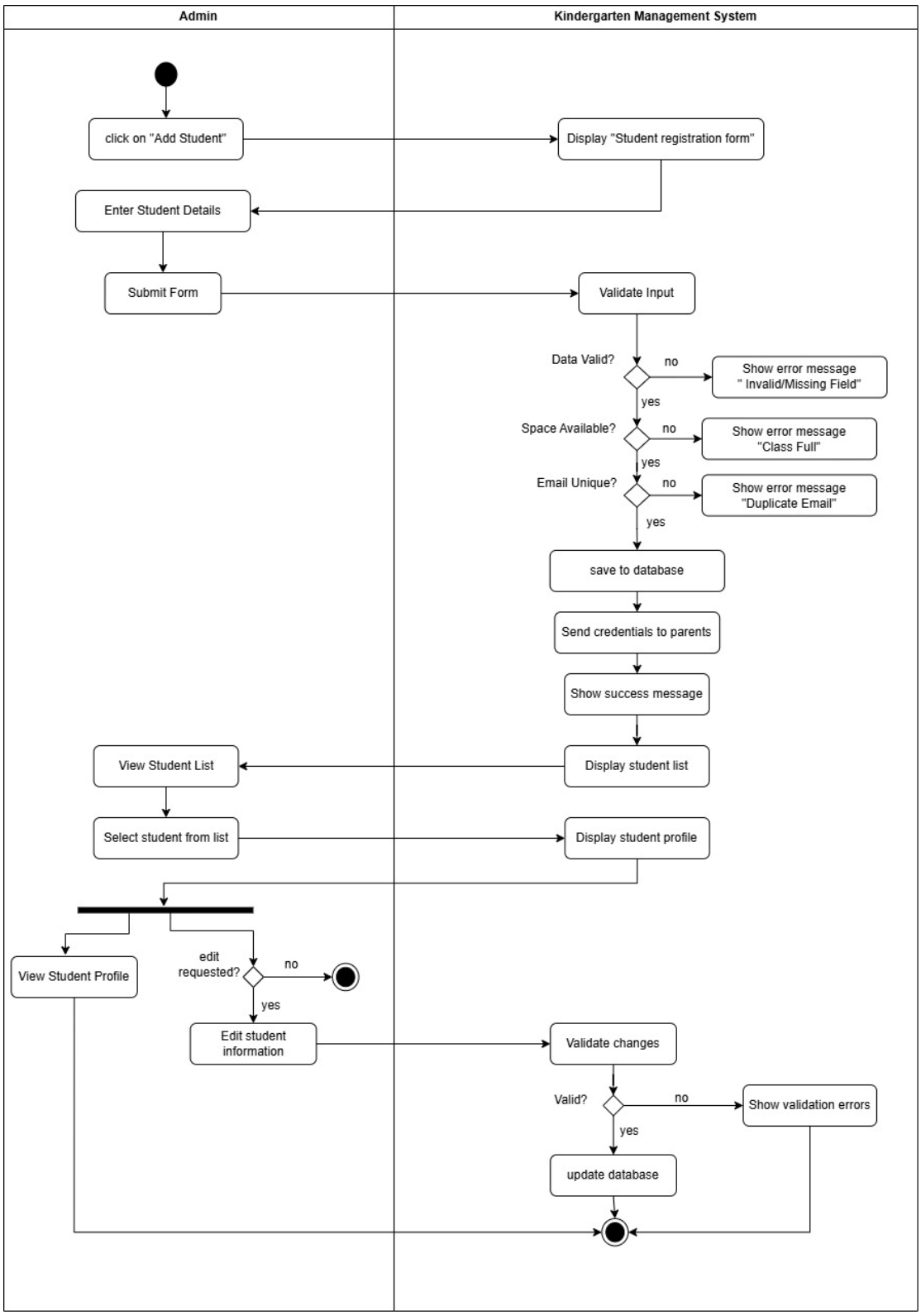


Figure 4.6 Manage Student Activity Diagram

4.2.7 Manage Class

Use Case Description

Use Case Name: Manage Class	**ID: 7**	**Importance Level:** High
Primary Actor: Admin **Secondary Actor:** Teacher, System	**Use Case Type:** Detail, Essential	
Stakeholders and Interests: • **Admin:** Needs to create and schedule subjects efficiently. • **Teacher**: Requires clear assignment to subjects and visibility in their schedule**.** • **System:** Must ensure no scheduling conflicts (time/location/teacher).		
Brief Description: This use case allows admins to add new subjects with details (subject title, subject description, assigned teacher) and filter the subjects added by class level. The admin is also allowed to schedule the subjects (day, time, location, class level) with calendar visibility and ensure the teachers to receive assigned subjects in their schedules)		
Trigger: Admin clicks on "Add Subject" form the sidebar		
Preconditions: • Administrator is logged into the system • Valid academic year is set • Teachers are registered in the system • Class levels are configured **Postconditions:** • Subject is created and/or scheduled • Teacher assignments are updated • Class schedule is maintained • System logs are updated		
Normal Flow of Events: **Add Subject** 1. Admin clicks "Add Subject". 2. System displays a form with: o Class Level (dropdown, e.g., "Grade 1").		

- Subject Title (required, e.g., "Mathematics").
- Description (optional, e.g., "Algebra basics").
- Assigned Teacher (dropdown of active teachers).

3. Admin submits the form.
4. System validates inputs (no duplicate subject titles per class level).
5. Subject is saved to database under subjects collection.
6. Confirmation: *"Subject added successfully"*

Schedule Subject

7. Admin clicks "Class Schedule" for the new subject.
8. System displays list of unscheduled subject for the particular class level.
9. If all the subjects for the particular class level is scheduled, it will shows "All subjects have been scheduled!"
10. For unscheduled subject, admin can click on the "Schedule Subject" button
11. System displays a scheduling form with:
    - Day of Week (dropdown: Monday-Friday).
    - Start/End Time (time picker).
    - Location (dropdown: e.g., "Room 101").
    - Capacity
12. Admin submits the schedule.
13. System checks for:
    - Teacher availability (no overlapping assignments).
    - Location availability (no double-booking).
14. If valid, system:
    - Saves schedule to database
    - Updates the calendar view in real-time.

Alternate Flows:

A1: Subject Already Exists

- System checks for duplicate subjects
- Notifies admin of existing subject
- Returns to subject form

- Preserves entered data

A2: Schedule Conflict

- System detects time/teacher conflict
- Displays conflict details
- Prevents schedule creation
- Returns to scheduling form

A3: Invalid Time Range

- System validates time slots
- Shows error for invalid times
- Maintains form data
- Allows correction

Exception Flows:

E1. Duplicate Subject:

- “Mathematics already exists for Class Level KD-Y3."

E2. Scheduling Conflict:

- "Teacher/room unavailable at this time. Choose another slot."

E3. Missing Required Field:

- "Subject title and teacher are required."

Table 4.7 Manage Class

Activity Diagram

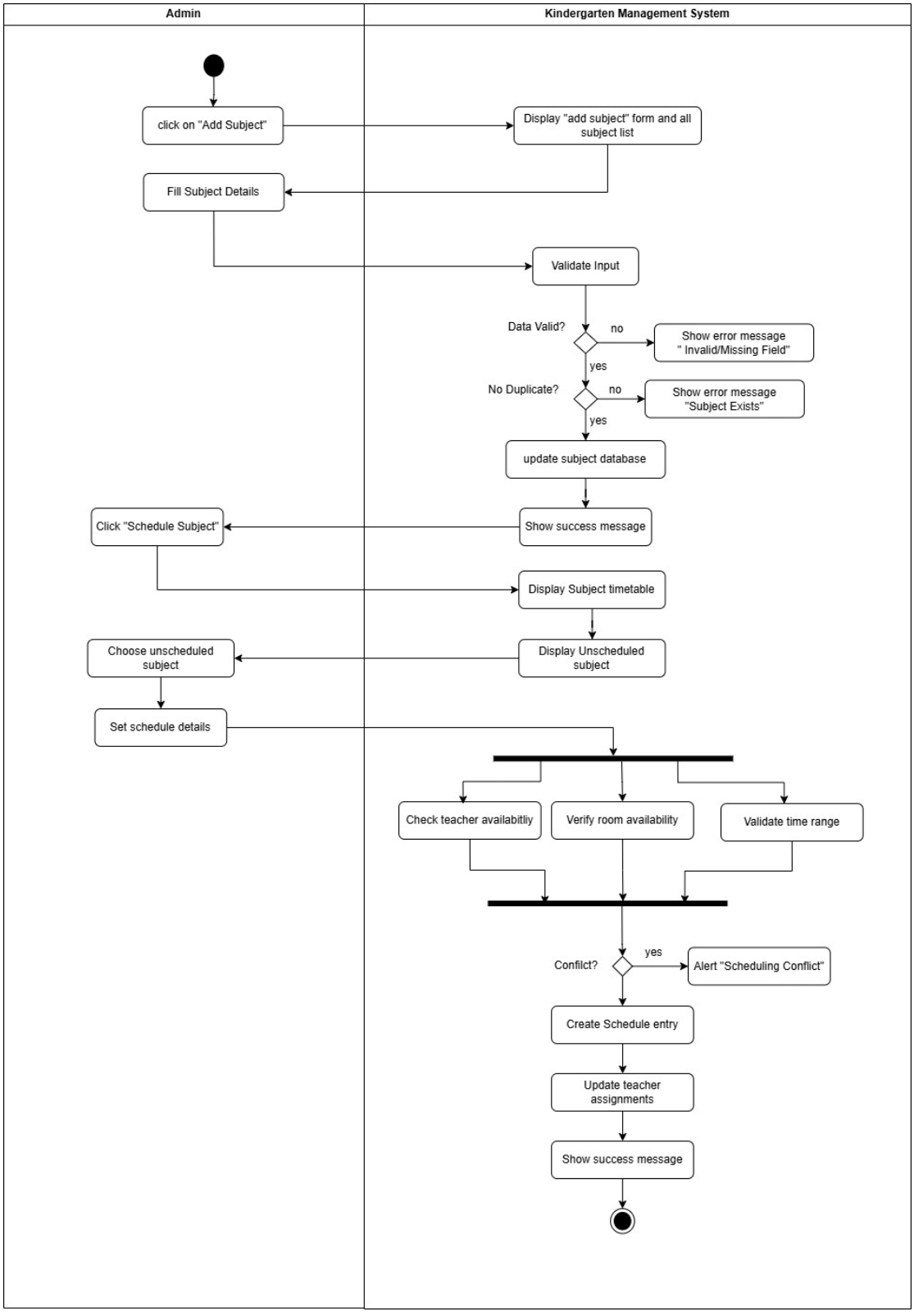


Figure 4.7 Manage Class Activity Diagram

4.2.8 Teacher Calendar

Use Case Description

Use Case Name: Teacher Calendar	**ID: 8**	**Importance Level:** High
Primary Actor: Teacher **Secondary Actor:** System	**Use Case Type:** Detail, Essential	
Stakeholders and Interests: • **Teacher:** Needs to view assigned classes with time/location details. • **System:** Must display accurate, real-time schedule data.		
Brief Description: This use case enables teachers to view their class schedule, manage tasks, and organize their academic responsibilities through an interactive calendar interface.		
Trigger: Teacher logs in and navigates to "Calendar" in sidebar		
Preconditions: • - Teacher is logged into the system • - Teacher has valid session • - Teacher has assigned classes **Postconditions:** • Calendar displays updated information • Tasks are properly managed • Class schedule is visible • Changes are saved to database		
Normal Flow of Events: **View Calendar** **1.** Teacher access calendar page **2.** System displays: • Monthly/weekly/daily calendar view • Class schedule with subject and level details **Manage Tasks** **1.** Teacher can: • Add new tasks • Enter task title		

- Provide description
- Set due date
- Assign priority (Low/Medium/High)

View existing tasks

1. Mark tasks as completed
2. Track task status

View Class Schedule

1. **System displays:**

- Subject name
- Class level
- Time slots
- Recurring schedule patterns
- Color-coded class sessions

Alternative Flows:

1. Task Creation Error

- System validates input
- Displays error message
- Preserves entered data
- Returns to form

2. Invalid Date Selection

- System prevents past dates
- Shows validation message
- Maintains form state

3. Session Timeout

- System detects invalid session
- Redirects to login page
- Preserves unsaved data

Exception Flows:

1. Database Connection Error

- System logs error
- Shows error message
- Returns to previous state

2. Calendar Load Failure - System attempts reload - Shows fallback view - Maintains basic functionality

Table 4.8 Manage Teacher Calendar Use Case Description

Activity Diagram

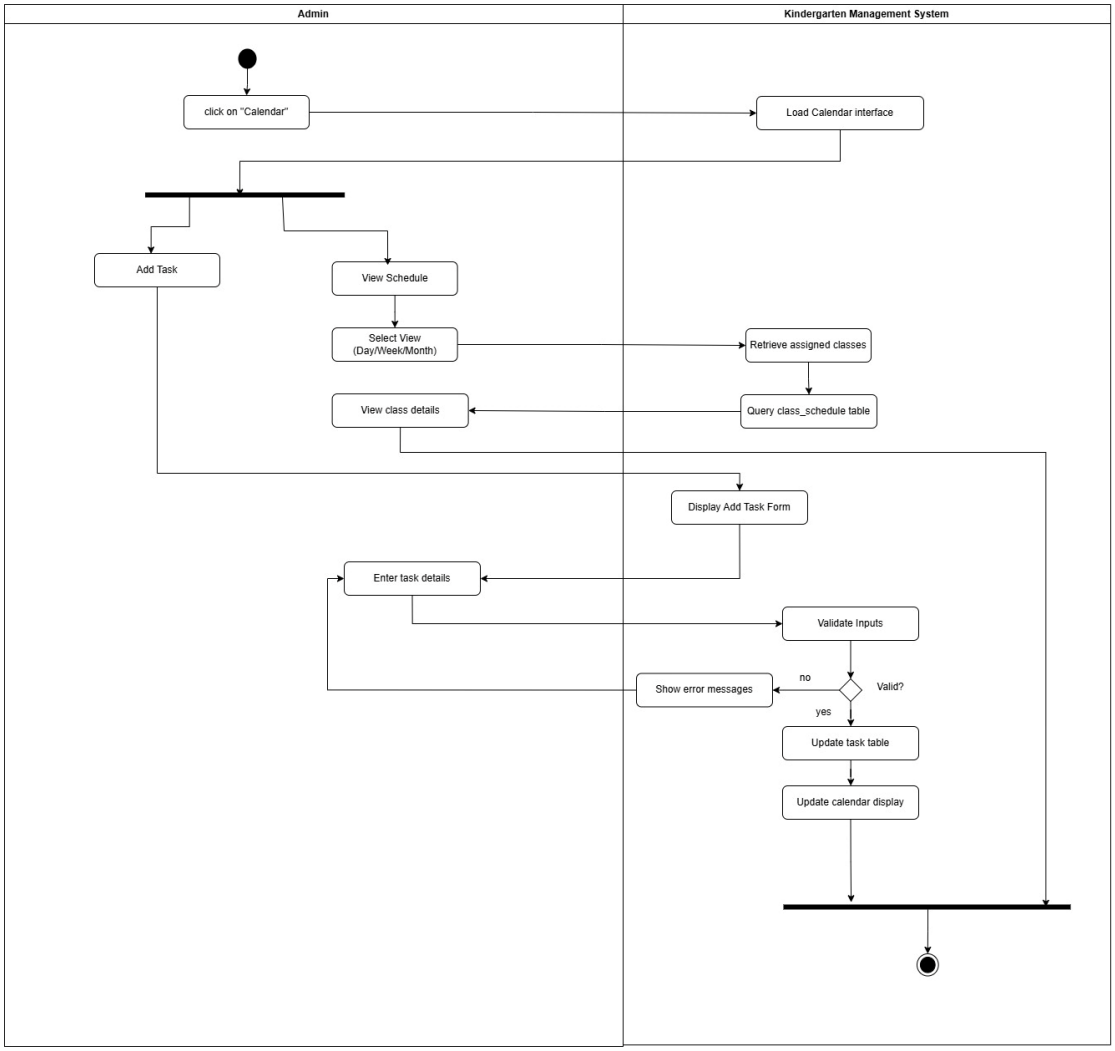


Figure 4.8 Manage Teacher Calendar Activity Diagram

4.2.9 View Class Assignments

Use Case Name: View Class Assignments	**ID: 9**	**Importance Level:** High
Primary Actor: Teacher **Secondary Actor:** System	**Use Case Type:** Detail, Essential	
Stakeholders and Interests: • **Teacher:** Needs to view their assigned classes with schedules, subjects, and student lists. • **System:** Must provide accurate, real-time class assignment data.		
Brief Description: This use case allows teachers to view a detailed list of classes assigned to them, including: • Subject information • Schedule (day/time/location) • Class level (e.g., KD – Y3)		
Trigger: Teacher clicks "Class Assigned" from the sidebar		
Preconditions: • Teacher is authenticated • Teacher has valid session credentials • Teacher has assigned classes in the database **Postconditions:** • Teacher views complete schedule • Classes are properly displayed • Session remains active • Schedule is current		
Normal Flow of Events: • Teacher selects "Class Assigned". • System displays the list of class assigned to the teacher with subject name, day, time, and class level		

Table 4.9 View Assigned Class Use Case Description

Activity Diagram

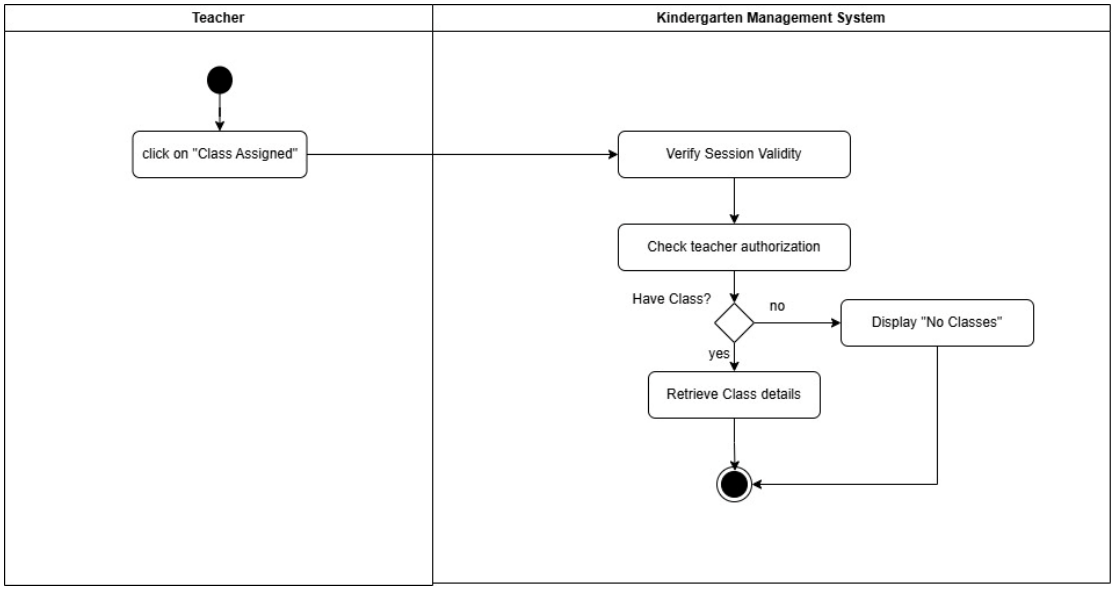


Figure 4.9 View Assigned Class Activity Diagram

4.2.10 Record Student Attendance

Use Case Description

<table>
<tr><td colspan="2">Use Case Name: Record Student Attendance</td><td>ID: 10</td><td>Importance Level: High</td></tr>
<tr><td colspan="2">Primary Actor: Teacher
Secondary Actor: System, Admin</td><td colspan="2">Use Case Type: Detail, Essential</td></tr>
<tr><td colspan="4">Stakeholders and Interests:
Teacher: Needs efficient attendance recording with minimal clicks.
Admin: Requires accurate attendance data for reporting.
Parents: Want real-time access to their child’s attendance.</td></tr>
<tr><td colspan="4">Brief Description: This use case enables teachers to record and manage daily attendance for their assigned classes.</td></tr>
<tr><td colspan="4">Trigger: Teacher clicks “Attendance” from the sidebar</td></tr>
<tr><td colspan="4">Preconditions:
Teacher is logged into the system
Teacher has assigned classes
Valid session exists
Students are enrolled in classes
Postconditions:
Attendance is recorded
Database is updated
Success message displayed
Records are available for review</td></tr>
<tr><td colspan="4">Normal Flow of Events:
View Classes
Teacher accesses the attendance page
System displays:
List of assigned classes
Date selector
Class selection dropdown
Teacher selects a specific class from the dropdown</td></tr>
</table>

4. Teacher insert the Attendance date
5. The system retrieves:
    - Complete student list for selected class
    - Current attendance status
6. Teacher record student attendance
7. For each student, teacher can mark attendance status:
    - Present
    - Absent
    - Late
    - Excused
8. Techer can also add optional remarks
9. Teacher submit attendance record

Alternative Flows:

1. No Classes Assigned

- System detects no classes
- Displays warning message
- Shows empty class list
- Maintains page structure

2. No Students in Class

- System shows empty student list
- Displays "No students found" message
- Disables submission button

3. Invalid Session

- System detects unauthorized access
- Redirects to login page
- Terminates current session

Exception Flows:

1. Database Error

- System logs error
- Shows error message
- Preserves form data

2. Submission Error

- System maintains form state - Shows error message - Allows resubmission

Table 4.10 Record Student Attendance

Activity Diagram

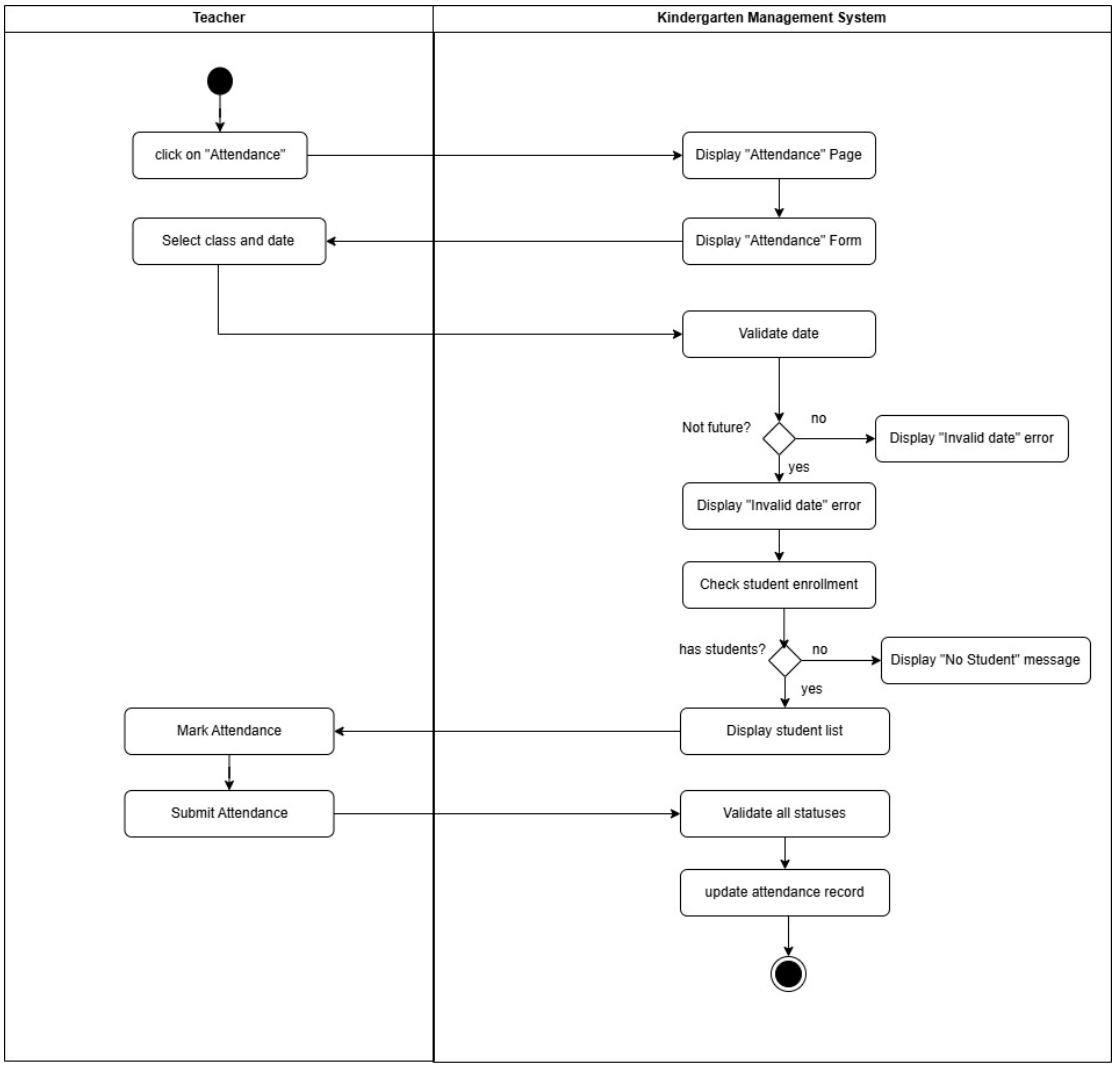


Figure 4.10 Record Student Attendance Activity Diagram

4.2.11 Manage Meeting Slot

Use Case Name: Manage Meeting Slot	**ID: 11**	**Importance Level:** High
Primary Actor: Teacher **Secondary Actor:** System, Admin	**Use Case Type:** Detail, Essential	
Stakeholders and Interests: • **Teacher:** Needs to control availability for parent meetings. • **Parent:** Requires visibility into available slots. • **System:** Must enforce booking rules and sync data in real-time.		
Brief Description: This use case enables teachers to manage their meeting slots, view class schedules, and handle parent meeting availability.		
Trigger: Teacher clicks "Meeting Slot" from the sidebar		
Preconditions: • Teacher is logged into the system • Teacher has valid session • Database connection is established **Postconditions:** • New slots appear in the parent portal within 1 minute. • Booked slots are locked for edits.		
Normal Flow of Events: **Adding Meeting Slot** **1.** The teacher clicks on the "Meeting Slot" from the sidebar **2.** The system displays the teacher's upcoming class for the next 7 days and the previously added slot with status(available/unavailable) **3.** Teacher clicks on "Add Meeting Slot" **4.** The system displays a form for the teacher to enter the meeting date, start time and end time **5.** The teacher inputs the meeting date, start time and end time **6.** The teacher submits the new meeting slot **7.** The system validates the meeting time will not clash the teacher's class schedule or overlapping with existing slots **8.** The system update the teacher meeting slots in database **9.** The updated meeting slots is shown in teacher's and parent's side		

Delete Meeting Slot

1. Teacher click on the delete button of the meeting slot
2. The system validates whether the meeting slot has been booked
3. If the meeting slot has been booked, an error message "cannot delete a booked slot" will be shown
4. If the meeting slot is not been booked, a confirmation message will prompt user to confirm on the deletion, a successful message "Meeting Slot deleted" will be shown

Alternative Flows:

A1. Slot Creation Error

- The system detects time conflict, invalid time range, and overlapping slots when teacher creates a new meeting slot.
- The system shows an error message for maintaining to form data and allows correction by the teacher.

A2. Delete Booked Slot

- The system checks the booking status to prevent deletion by showing a warning message when the teacher deleting a meeting slot

A3. Update Slot Status

- The teacher toggles the availability of meeting slot
- The system updates the status and show confirmation of the modification.

Exception Flows:

E1. Database Error

- The system logs errors and shows error message to maintains to current view of the user interface

E2. Session Timeout

- The system detects invalid session and redirect the user to login.

Table 4.11 Manage Meeting Slot Use Case Description

Activity diagram

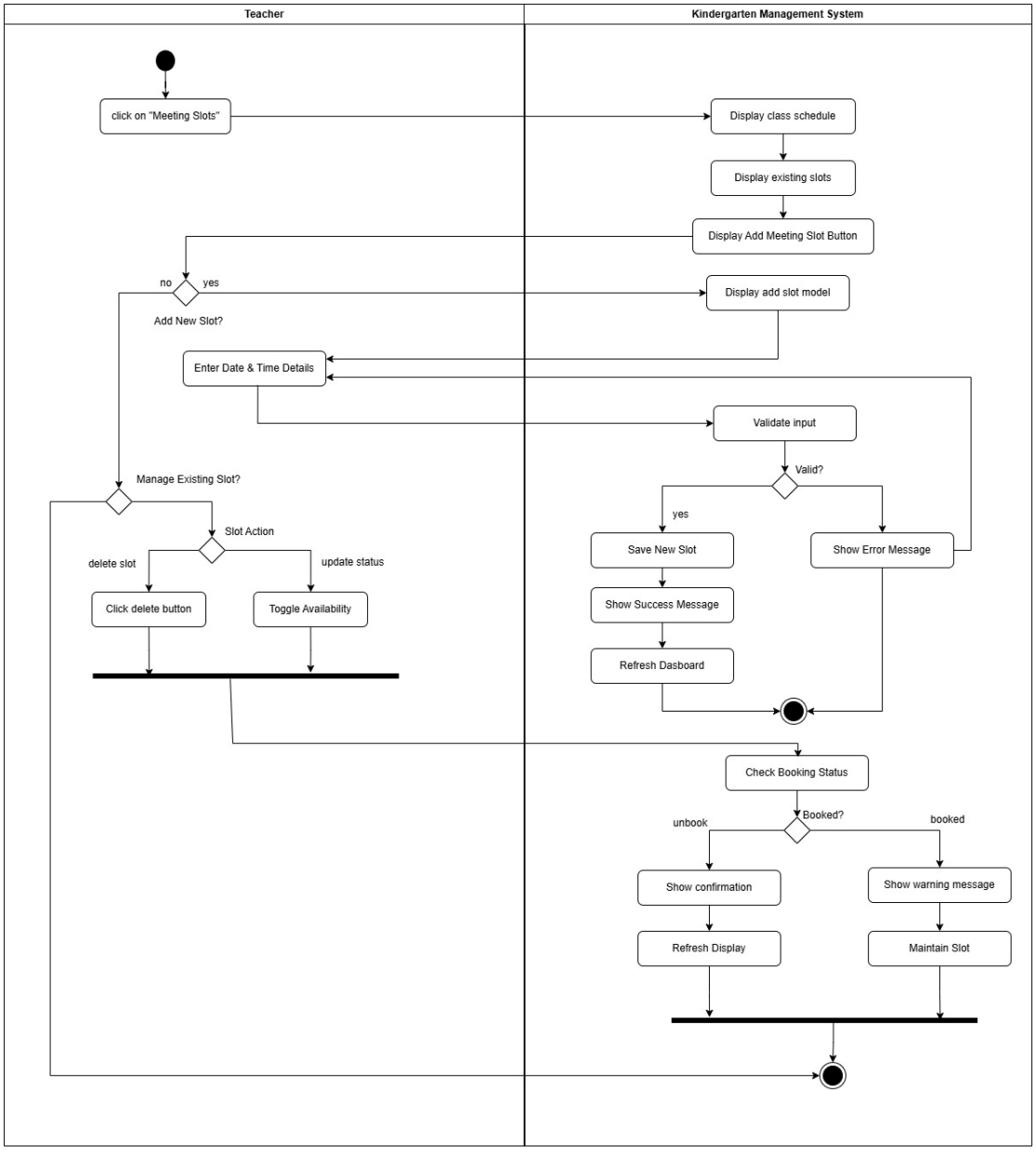


Figure 4.11 Manage Meeting Slot Activity Diagram

4.2.12 Manage Meeting Request

Use Case Name: Manage Meeting Request	**ID: 12**	**Importance Level:** High
Primary Actor: Teacher **Secondary Actor:** System, Parent	**Use Case Type:** Detail, Essential	
Stakeholders and Interests: • **Teacher:** Needs to efficiently manage meeting requests. • **Parent:** Wants timely responses to meeting requests. • **System:** Must notify parents and update database		
Brief Description: This use case enables teachers to manage parent meeting requests, including approving, rescheduling, and submitting meeting reports.		
Trigger: • New meeting request submitted by parent. • Teacher navigates to **"Meeting Request"** tab.		
Preconditions: • Teacher is logged into the system • Teacher has valid session • Database connection is established • Email system is configured **Postconditions:** • Request status updated • Email notifications sent • Meeting report stored • Database synchronized		
Normal Flow: **1.** The system displays the meeting requests from parents with parent's name, requested date and time, and the purpose of the meeting **2.** The teacher takes action to approve by clicking the “approve” button, and the slot is reserved instantly. An email notification will be sent to the parent **3.** If the teacher clicks on “reschedule meeting”, an email notification will be sent to the parents to reschedule a new meeting with the reason		

4. The system will update on the Meeting Request database on the approval of the request 5. For approved meetings, the teacher can add child's behavior assessment, note social development, provide recommendation and submit in a meeting report. 6. The system then marks the meeting as completed
Alternative Flows: **A1. No Meeting Requests** • The system shows "No requests" message when there is no meeting requested by the parents **A2. View Existing Report** • The system displays all report section of the meeting including the report submission data **Exception Flows:** **E1. Database Error** - System logs error - Shows error message - Preserves form data - Maintains current view **2. Invalid Request Data** - Validates input - Shows validation errors - Allows correction - Preserves valid data

Table 4.12 Manage Meeting Request Use Case Description

Activity Diagram

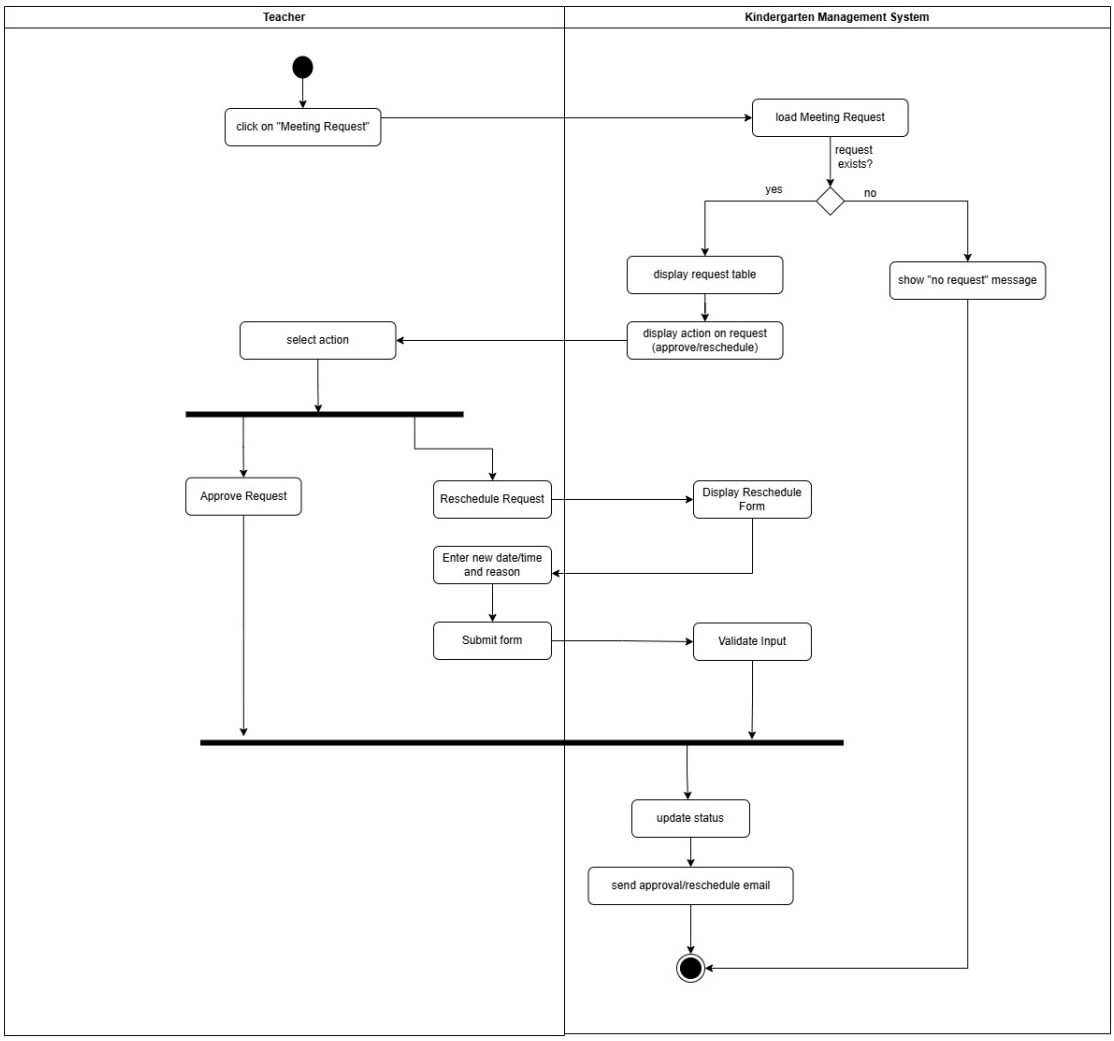


Figure 4.12 Manage Meeting Request Activity Diagram

4.2.13 Manage Lesson Plan

Use Case Description

Use Case Name: Manage Lesson Plan	**ID: 13**	**Importance Level:** High
Primary Actor: Teacher	**Use Case Type:** Detail, Essential	
Stakeholders and Interests: **Teacher:** • Ability to easily create and organize lesson plans by subject. • Option to reuse lesson plans via templates • Control over visibility (public/private). • Ability to edit or delete their own plans efficiently.		
Brief Description: This use case allows teachers to create, edit, save, delete, and manage lesson plans. Lesson plans can be tagged to specific subjects, saved as templates, and shared publicly or kept private**.**		
Trigger: • Teacher clicks on “add lesson plan” from the sidebar		
Preconditions: • Teacher is authenticated • Teacher is assigned to the particular subject • Database connection is established **Postconditions:** • A new lesson plan is created, modified, or deleted. • The lesson plan may be saved as a reusable template. • Sharing settings are updated.		
Normal Flow: **Create Lesson Plan** 1. Teacher navigates to the "Lesson Plan" section. 2. Teacher selects "Create New Lesson Plan". 3. System displays form to enter subject, title, content, and other necessary details. 4. Teacher fills in the lesson plan and selects:		

- Save as template (optional)
- Set visibility: Public / Private

5. Teacher clicks "Save".
6. System confirms that the lesson plan is successfully created.

Manage Existing Plans

1. Teacher can view all created plans and do edition on them
2. Teacher can delete lesson plans
3. **Teacher can view plan details**

Work With Templates

1. Teacher can view public templates
2. Teacher can use existing templates created
3. Teacher can create new templates for lesson plans

Alternate Flows

A1. Editing an Existing Lesson Plan

1. Teacher selects an existing lesson plan they created.
2. System displays current content.
3. Teacher makes changes and saves.
4. System confirms updates.

A2. Deleting a Lesson Plan

1. Teacher selects a lesson plan they created.
2. Teacher clicks "Delete".
3. System prompts for confirmation.
4. Teacher confirms deletion.
5. System removes the lesson plan.

Exception Flows:

E1. Database Error

System:

- Logs error
- Shows error message
- Maintains current view
- Preserves form data

E2. Unauthorized Access

System: • Blocks access • Redirects to login • Logs attempt • Terminates session

Table 4.13 Manage Lesson Plan Use Case Description

Activity Diagram

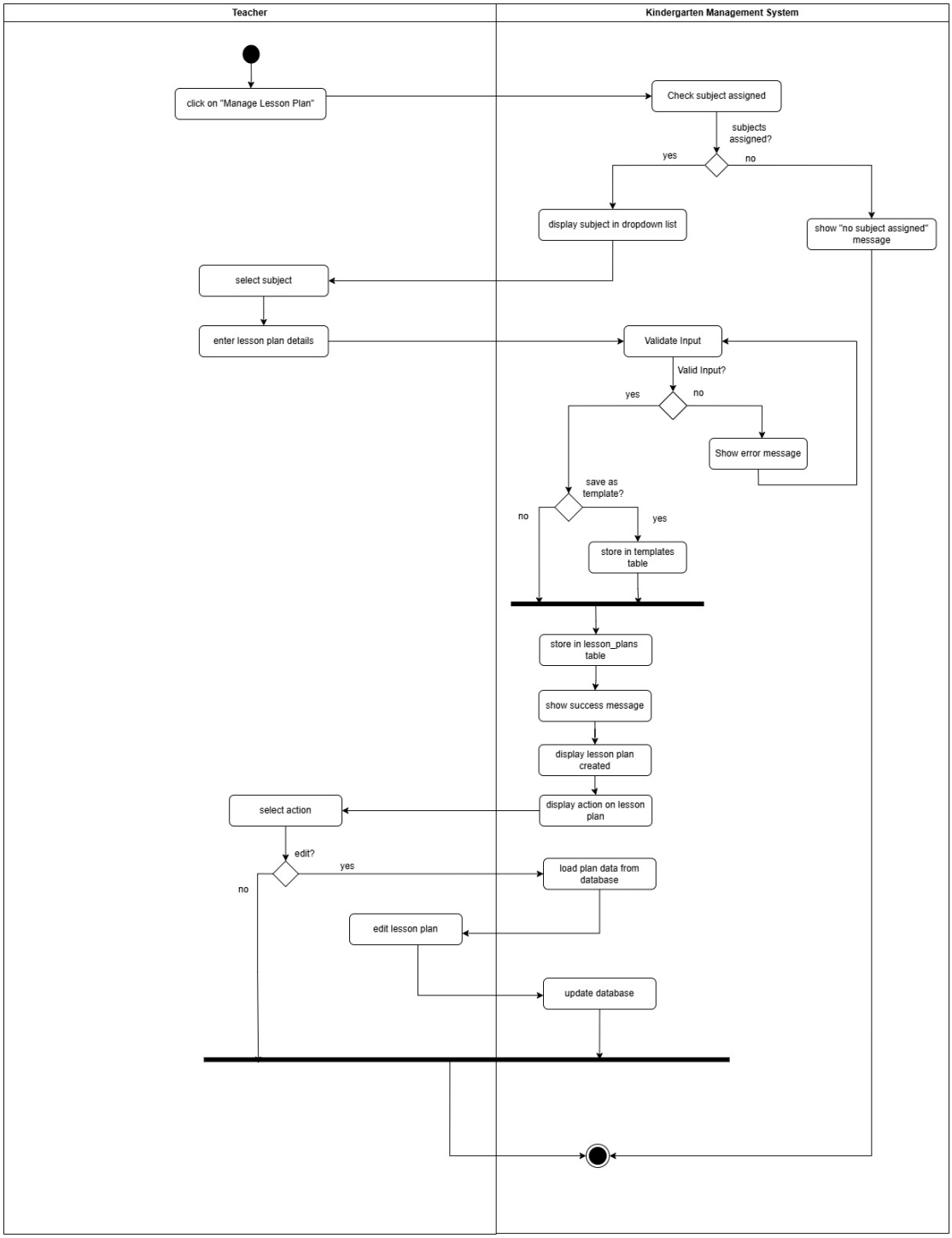


Figure 4.13 Manage Lesson Plan Activity Diagram

4.2.14 Book Meeting

Use Case Description

Use Case Name: Book Meeting	**ID: 14**	**Importance Level:** High
Primary Actor: Parent	**Use Case Type:** Detail, Essential	

Secondary Actor: Teacher	
Stakeholders and Interests: **Parent:** - Easy access to view teacher's available time slot - Ability to book a meeting with clear communication of purpose - Receive timely updates (approval or reschedule email notification) **Teacher:** - Receiving clear meeting requests with stated purpose - Having control over availability and approval of meetings	
Brief Description: This use case allows a parent to view available time slots of a teacher, select a preferred slot, and book a meeting for a specific purpose. An email notification is sent to the teacher, and the parent awaits approval.	
Trigger: • Parent clicks on "book meeting" from the sidebar	
Preconditions: • Parent is logged into the system • Parent has valid session • Database connection is established • Email system is configured **Postconditions:** • A meeting request is created. • Email notification is sent to the teacher. • Parent is notified upon approval or rejection.	
Normal Flow: 1. Parent logs in and navigates to the "Book Meeting" section. 2. System displays a list of teachers and their available time slots. 3. Parent selects a teacher and a suitable time slot. 4. Parent enters the purpose of the meeting. 5. Parent clicks "Book Meeting".	

6. System creates a pending meeting request. **7.** System sends an email notification to the selected teacher. **8.** Parent is shown a confirmation that the request was sent. **9.** Parent waits for a response, which will be received via email once the teacher approves or rejects the request.
Alternate Flows **A1. No Available Time Slots** - If no time slots are available for a selected teacher, the system displays a message: "No available meeting slots at this time."

Table 4.14 Manage Available Meeting Slot Use Case Description

Activity Diagram

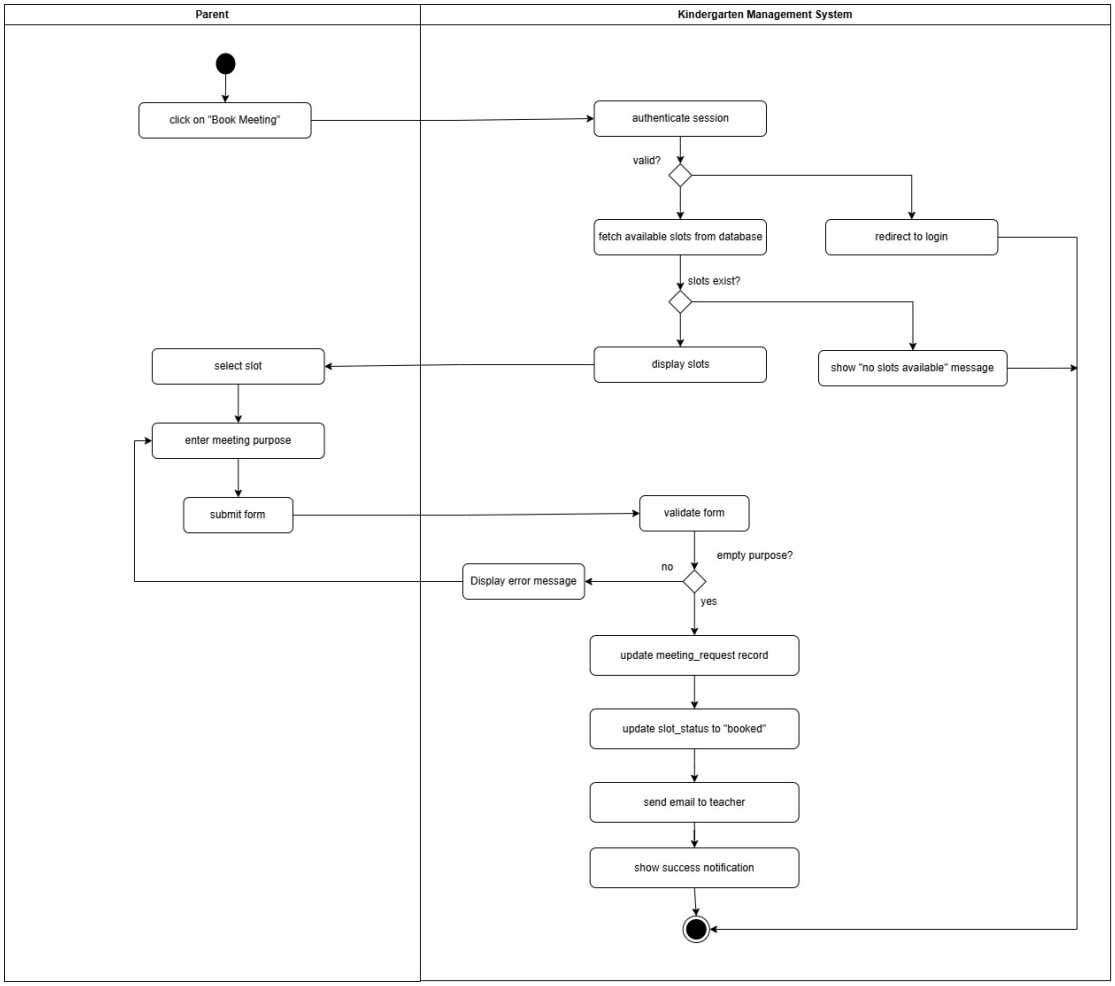


Figure 4.14 Manage Available Meeting Slot Activity Diagram

4.2.15 Payment Management

Use Case Description

Use Case Name: Book Meeting	**ID: 14**	**Importance Level:** High
Primary Actor: Admin	**Use Case Type:** Detail, Essential	
Stakeholders and Interests: **Admin:** - Efficiently manage all billing and payment operations - Generate accurate invoices and track payments - Monitor financial performance and outstanding balances - Ensure timely fee collection and financial reporting		

Parent:

- Receive clear, accurate invoices for their children's fees
- Make payments conveniently and securely
- Track payment history and outstanding balances
- Receive timely payment confirmations

Brief Description: The Payment Management use case enables administrators to comprehensively manage the financial operations of the kindergarten system. This includes creating and managing invoices (both individual and bulk), tracking payments, monitoring financial performance through dashboards, and maintaining billing records. The system provides complete visibility into revenue collection, outstanding balances, and payment trends while ensuring accurate financial reporting and efficient billing processes.

Trigger:

Primary Triggers:

- Administrator needs to generate invoices for student fees
- New payment received that requires verification and recording
- Monthly/periodic financial reporting requirements
- Parent payment inquiry or dispute
- Overdue payment follow-up needed

Temporal Triggers:

- Monthly billing cycle initiation
- Payment due date reminders
- End-of-month financial reconciliation
- Academic year fee structure updates

Preconditions:

- Admin is logged into the system
- Parent has valid session
- Database connection is established
- Student enrollment records must exist in the system
- Parent contact and billing information must be available
- Fee structures and templates must be configured

- Academic year and class information must be current

Postconditions:

- All financial transactions are accurately recorded in the database
- Invoice records are created with unique identification numbers
- Payment status is updated and reflected across all relevant modules
- Financial reports and dashboards display current information
- Database integrity is maintained
- All stakeholders receive appropriate notifications
- Outstanding balance calculations are updated
- Financial statistics and trends are refreshed

Normal Flow:

1. Administrator accesses the billing dashboard
2. System displays financial statistics (monthly collections, outstanding balances, overdue invoices)
3. Administrator reviews current financial status and identifies required actions
4. Administrator navigates to invoice management section
5. Selects between individual invoice creation or bulk invoice generation
6. For bulk invoices: selects target students, fee template, and due date
7. System validates selections and generates invoices with unique numbers
8. Invoices are saved to database and notifications sent to parents

Alternate Flows

A1: Bulk Invoice Generation with Custom Fees

Trigger: Administrator needs to create invoices with non-standard fee structures

1. Administrator selects "Custom Fee Template" option
2. System prompts for fee item details (description, amount, category)
3. Administrator defines custom fee structure
4. System validates fee template and proceeds with bulk generation
5. Custom invoices are created and distributed

A2: Payment Verification Failure

Trigger: Payment verification process identifies discrepancies

1. System flags payment as "Pending Verification"

2. Administrator reviews payment details and supporting documentation
3. If payment is legitimate: Administrator manually approves payment
4. If payment is fraudulent/incorrect: Administrator rejects payment and notifies relevant parties
5. System updates payment status accordingly

A3: Overdue Payment Management

Trigger: System identifies overdue invoices during regular monitoring

1. System generates overdue payment report
2. Administrator reviews overdue accounts and amounts
3. Administrator initiates follow-up actions:
   a. Sends automated reminder notifications to parents
   b. Schedules personal contact for high-value overdue amounts
   c. Applies late fees if configured in system policies
4. System tracks follow-up actions and updates account status

A4: Invoice Modification/Cancellation

Trigger: Invoice requires correction or cancellation after generation

1. Administrator locates the specific invoice in manage invoices section
2. If invoice is unpaid: Administrator can modify or cancel invoice
3. If invoice is partially/fully paid: Administrator creates credit note or adjustment
4. System maintains audit trail of all modifications
5. Affected parties receive notifications of changes

A5: Payment Method Integration Failure

Trigger: External payment gateway or method becomes unavailable

1. System detects payment processing failure
2. Administrator receives system alert about payment method issues
3. Administrator switches to manual payment recording mode
4. Payments are recorded manually with appropriate documentation
5. System synchronizes with payment gateway once service is restored

A6: Financial Report Generation

Trigger: Management requests specific financial reports or analysis

1. Administrator accesses reporting module from billing dashboard
2. Selects report parameters (date range, payment methods, student groups)
3. System generates comprehensive financial reports including:
    a. Revenue analysis by time period
    b. Outstanding balance summaries
    c. Payment method distribution
    d. Overdue account listings

Chapter 5 System Implementation

5.1 Login Page

In the login page, the user is required to enter their email account and password, and input the captcha code displayed on the screen. When a user clicks on the login button, the system will validate the user authentication.

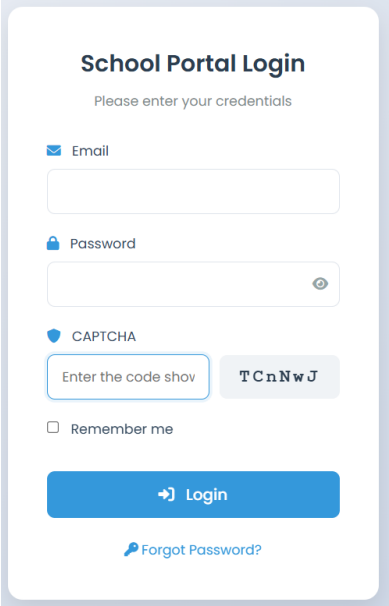


Figure 4.15 Login Page

If the user enters an email that is not exist or incorrect password, the system will prompt an error message to the user to reenter the email or password.

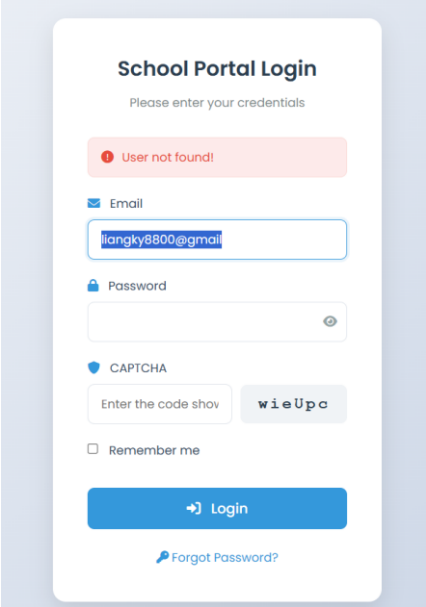


Figure 4.16 User Not Found Error Message

If the user enters a wrong captcha code, the system will prompt an error message.

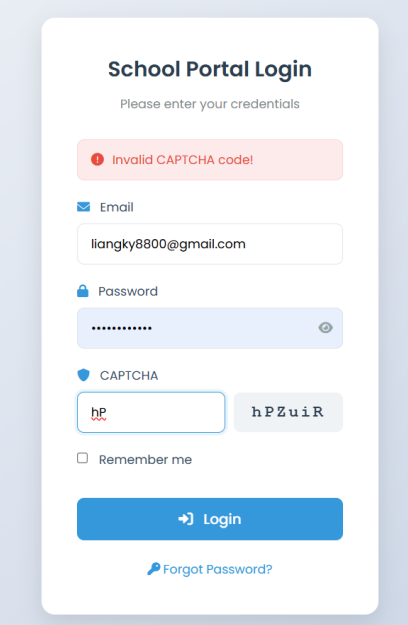


Figure 4.17 Invalid Captcha Code Error Message

If the user forgot their password, they can click on the “Forget Password” link, and they will be directed to the reset password page. Users are required to enter their email address.

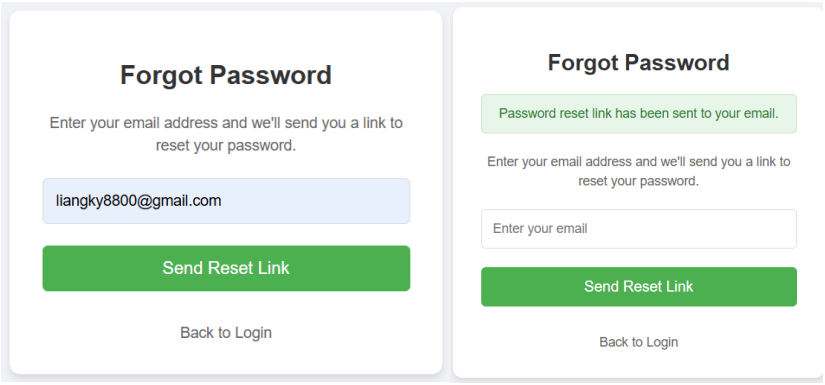


Figure 4.18 Forgot Password

A reset password link will be sent to the email address. User then clicks on the RESET link and is directed to a page to enter their new password.

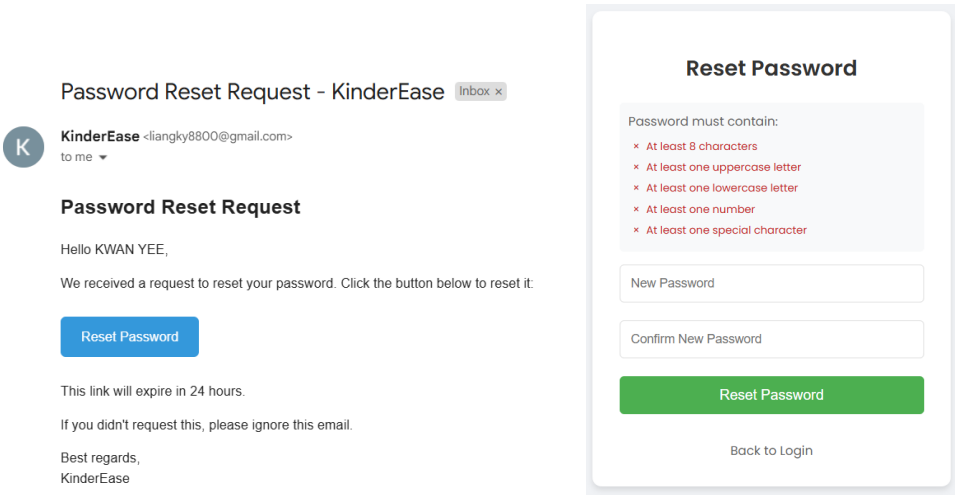


Figure 4.19 Reset Password

5.2 Administrative Role Task

Administration plays a crucial role in the Kindergarten Management System. In this project system, the admin is responsible for many tasks in the kindergarten. Upon arrival at the system's login page, the admin is able to log in with their email account and password. When the system validates the credentials and authentication, the admin is transitioned seamlessly to the admin dashboard. Once the admin logs in, the admin can scroll through the sidebar allocated on the left-hand side with several essential features that can be done. The first listed in the sidebar is the admin dashboard, which shows all the necessary information and an overview of the kindergarten. The admin dashboard includes the total number of students, teachers, and subjects of the

kindergarten. The student distribution throughout the class level and a summary of students' attendance are shown. It also includes the recent announcement.

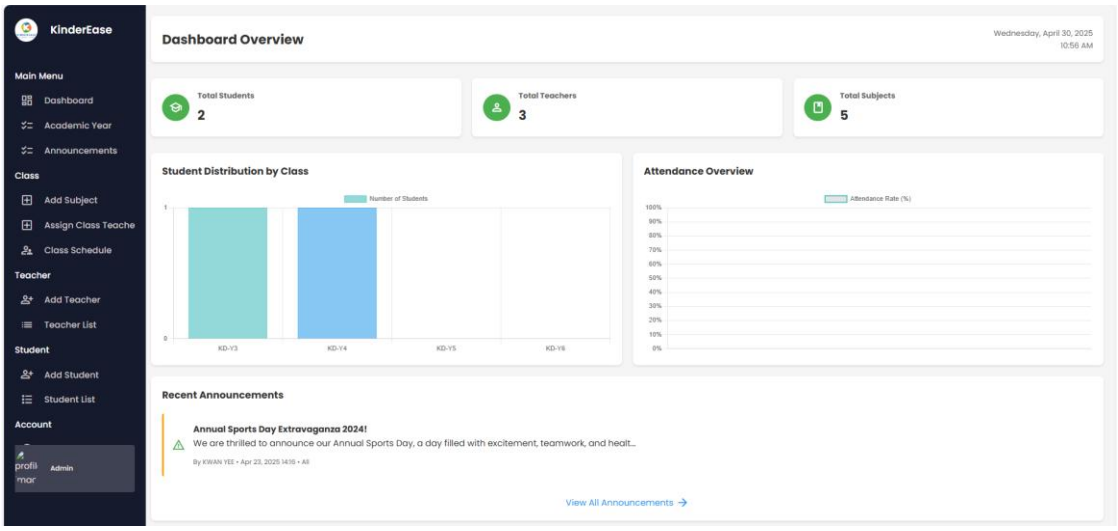


Figure 4.20 Admin Dashboard

Besides, the admin can create new academic year with specified start date and end date from the academic year selection at the sidebar. By creating new academic year, admin fills up the necessary details in a form.

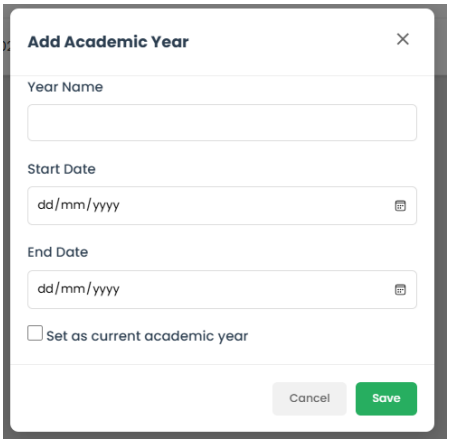


Figure 4.21 Add Academic Year

Then, the valid academic year will be saved and displayed

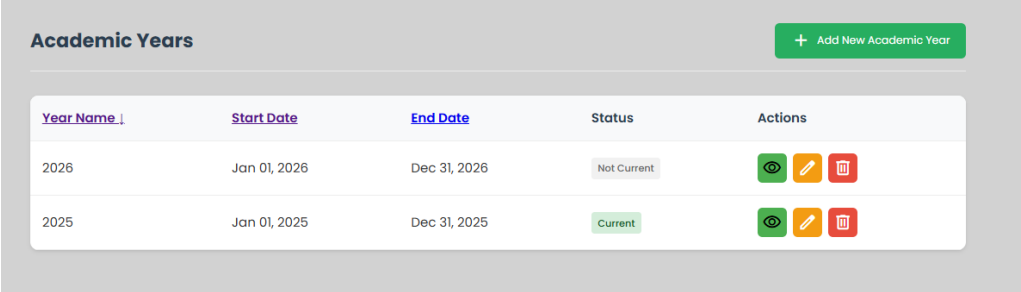


Figure 4.22 Academic Year List

When the admin added an invalid input for the academic year, like date range from Jan 01, 2025 to Dec 31, 2025 twice, the system will show a warning message.

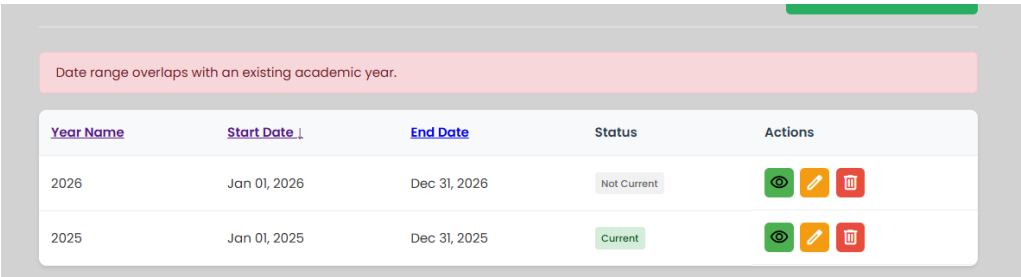


Figure 4.23 Invalid Academic Year Error Message

The admin is able to create a new announcement with an announcement title, details, and select for target audience, which includes teachers only, parents only, or both teachers and parents. The announcement will be saved to the database, admin is able to edit and delete on the announcement created, whereas teachers and parents can only read on the announcement sent by the admin.

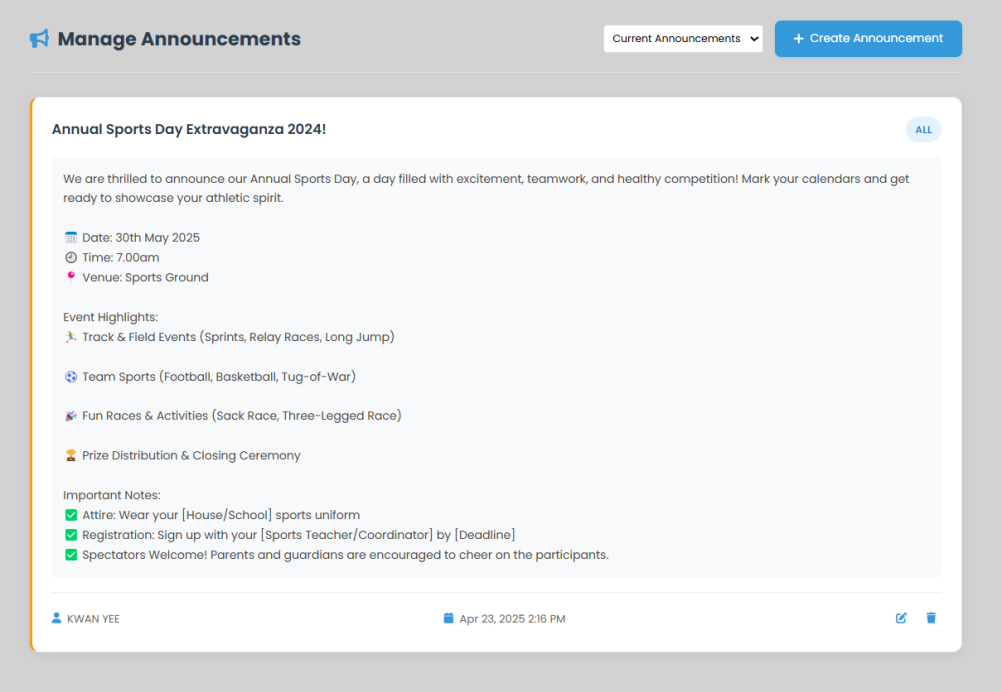


Figure 4.24 Admin Announcement Page

The system will prompt an error message for empty input in the announcement creation form.

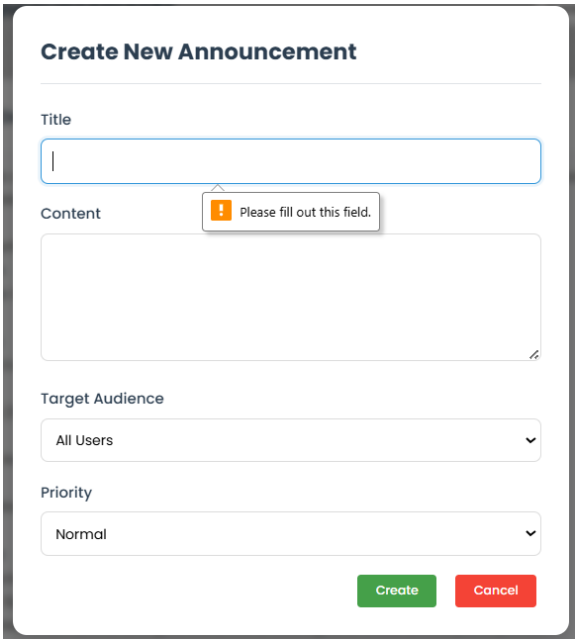


Figure 4.25 Create New Announcement

When the admin clicks on the edit button on the current announcement, an edit dialog will be shown.

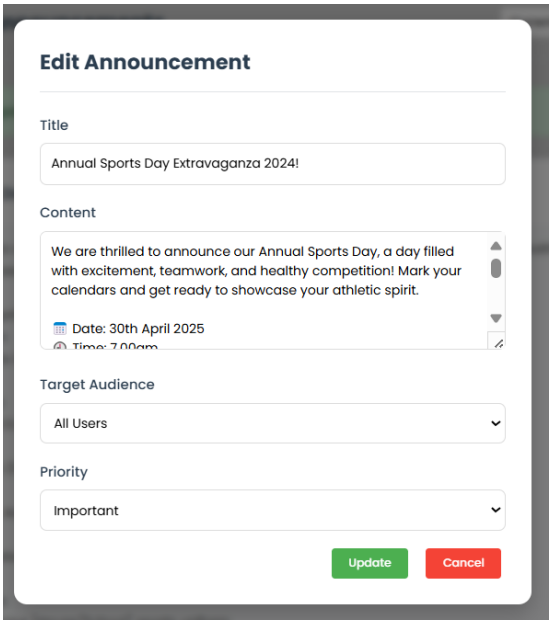


Figure 4.26 Edit Announcement

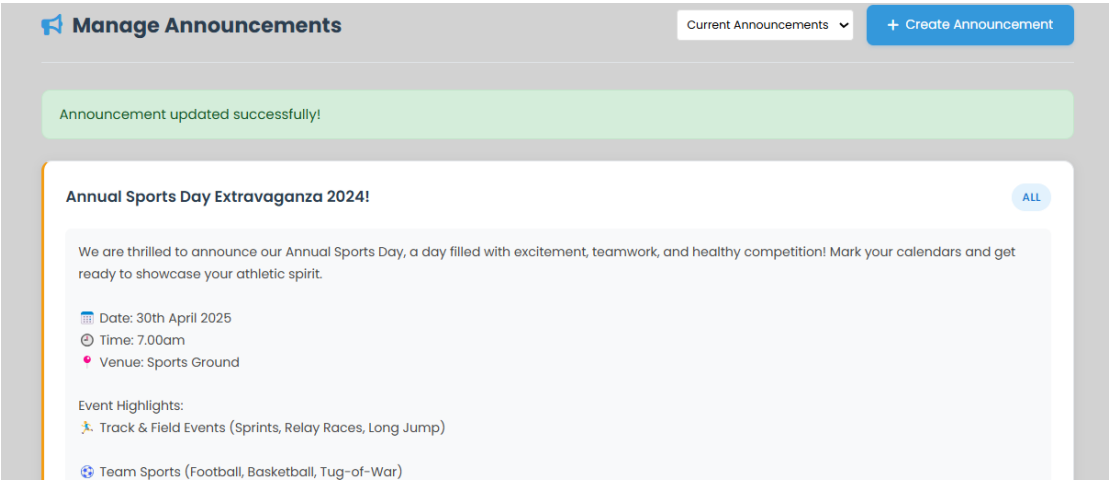


Figure 4.27 Edit Announcement Success Message

A success message will be shown upon successful updates.

In the class management system, the admin is first required to add subjects to each class level. The subject is added with the subject name, subject description, and with an assigned teacher to a particular class level. After adding the subject, a success message will be out and the subject will be saved in the database.

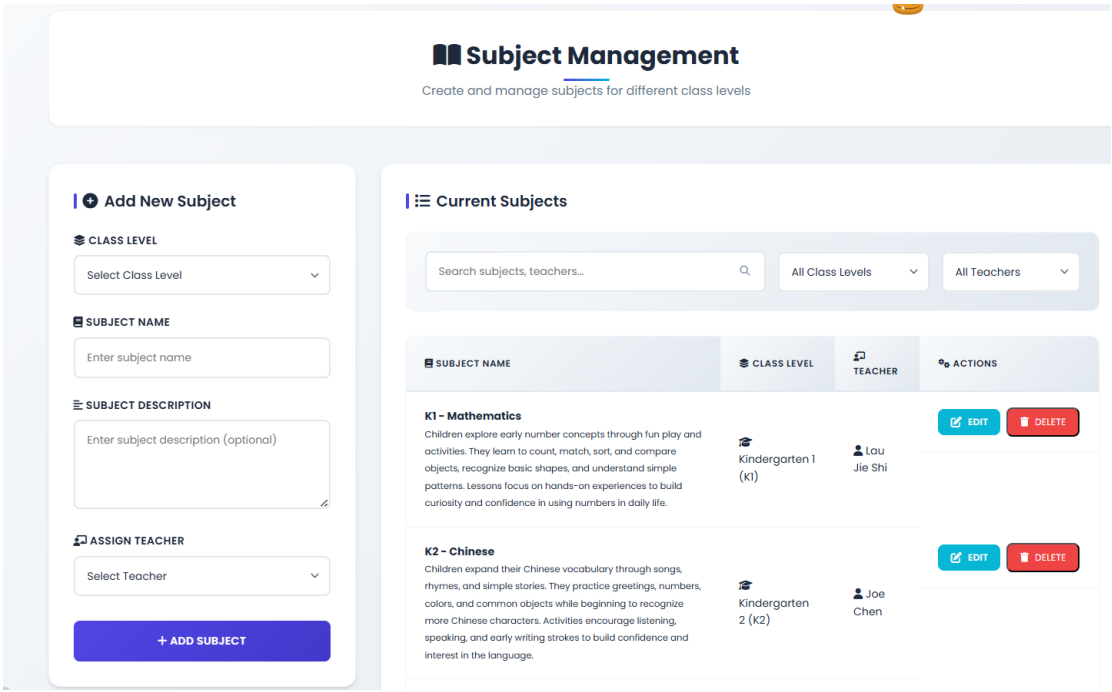


Figure 4.28 Add Subject Page

Admin can edit on the subject details by clicking on the edit button of the added subject. A dialog will be prompt out for admin to edit.

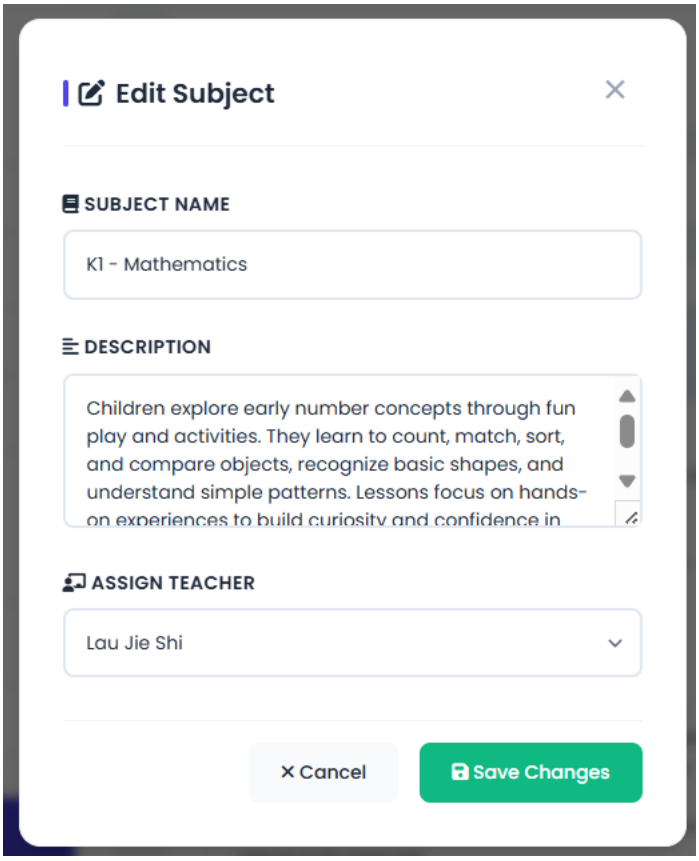


Figure 4.29 Edit Subject

After that, the unscheduled subject will be shown beside the timetable and the admin is required to schedule the day and time for each subject.

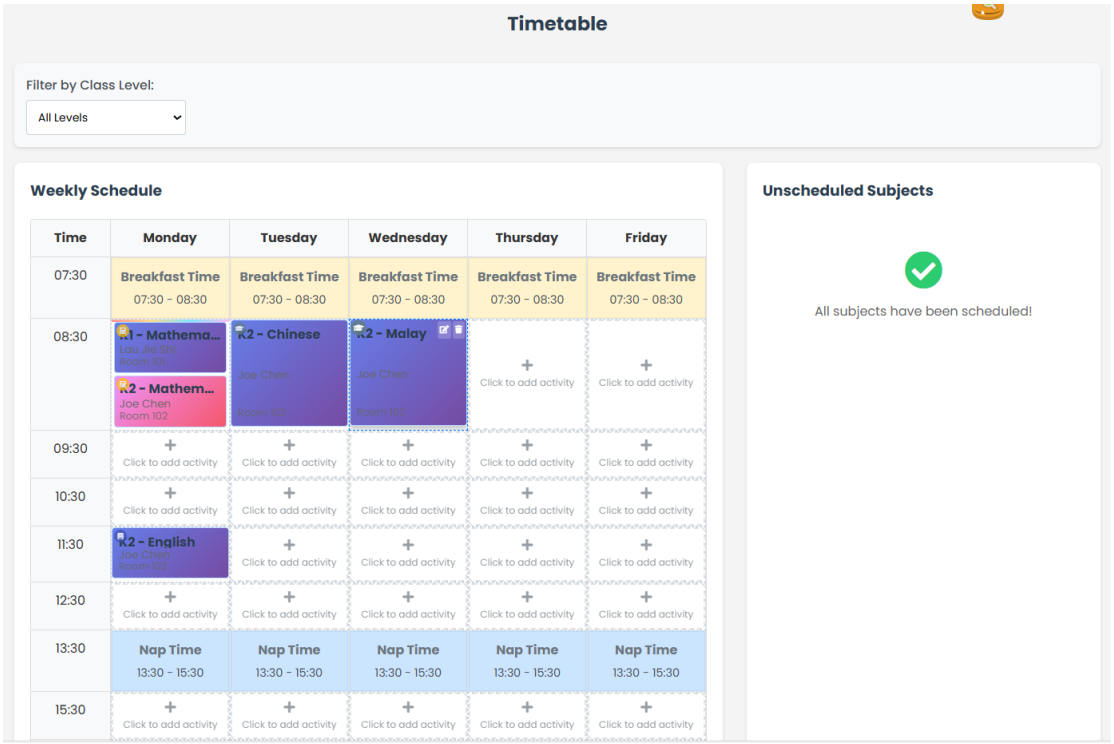


Figure 4.30 Schedule Subject Page

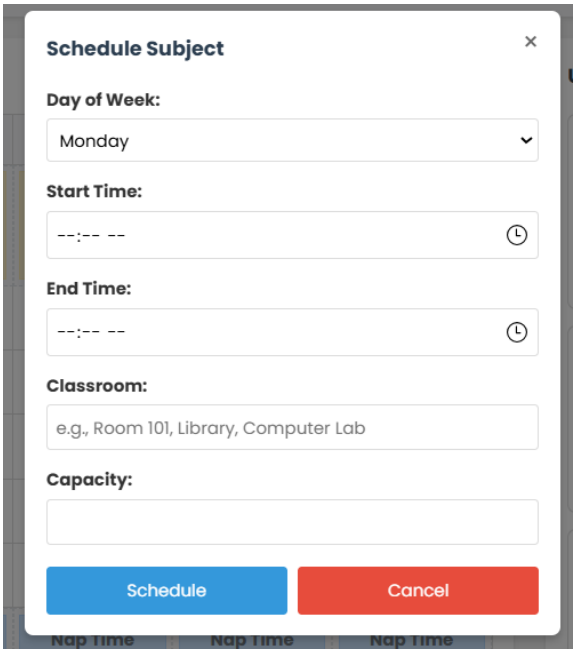


Figure 4.31 Schedule Subject Dialog

The scheduled subject will be shown in the timetable and is allowed to edit and delete by the admin. The admin can also add a custom activity by clicking on the empty timeslot.

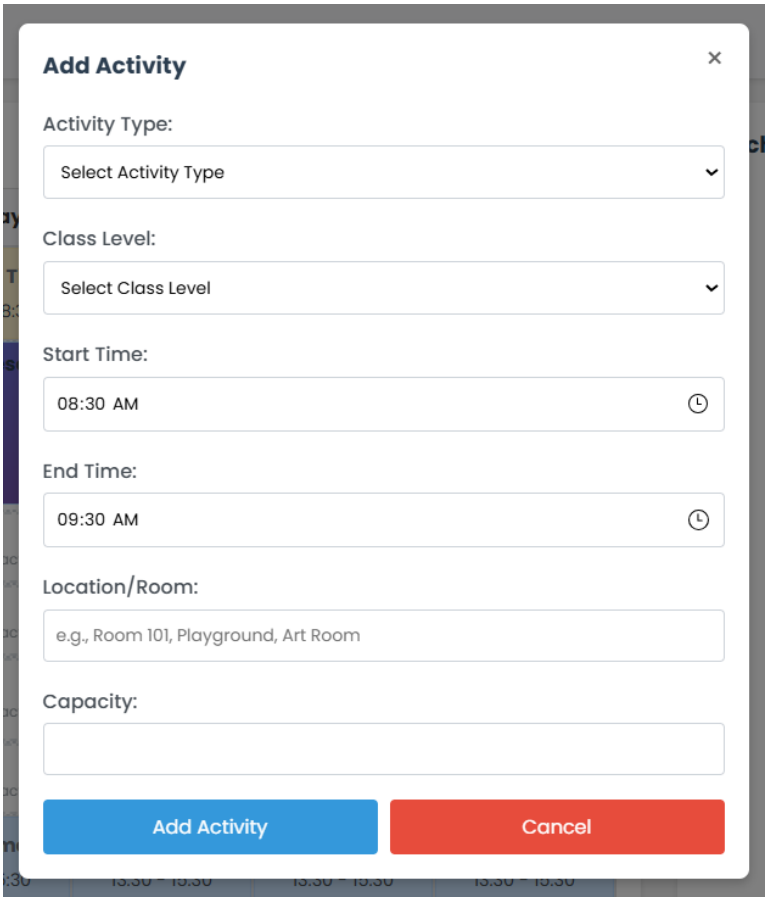


Figure 4.32 Add Activity

The admin can add a new teacher by filling up all the particulars in the add teacher form; the teacher added will receive an email for their account with a randomly generated password. the teacher is then required to log in with the email and the given password for the first time login. When the system detects a first-time login teacher, the teacher will be required to change their password before they can log in to the system. The added teacher will be shown in the teacher list, The admin is allowed to sort the list view and perform view, edit, and deletion on the teacher.

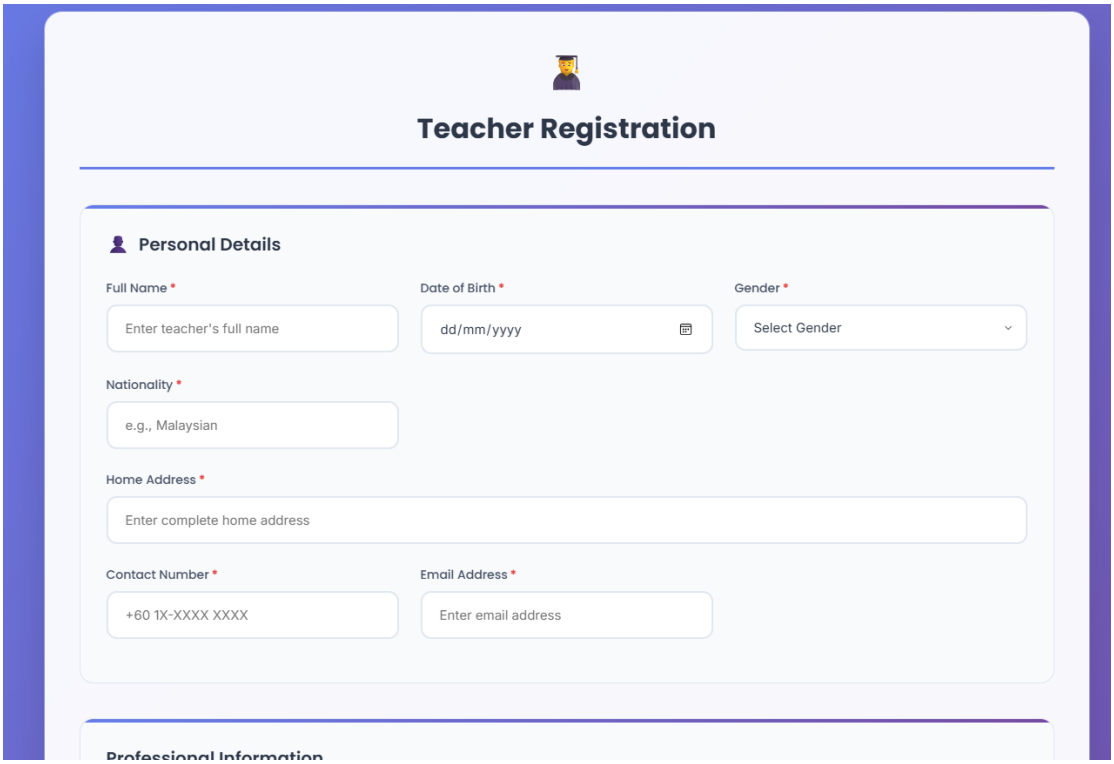


Figure 4.33 Teacher Admission Form

After successfully completing the teacher registration, the teacher list will be shown, the admin can also click on the edit, and view button of each teacher to perform action.

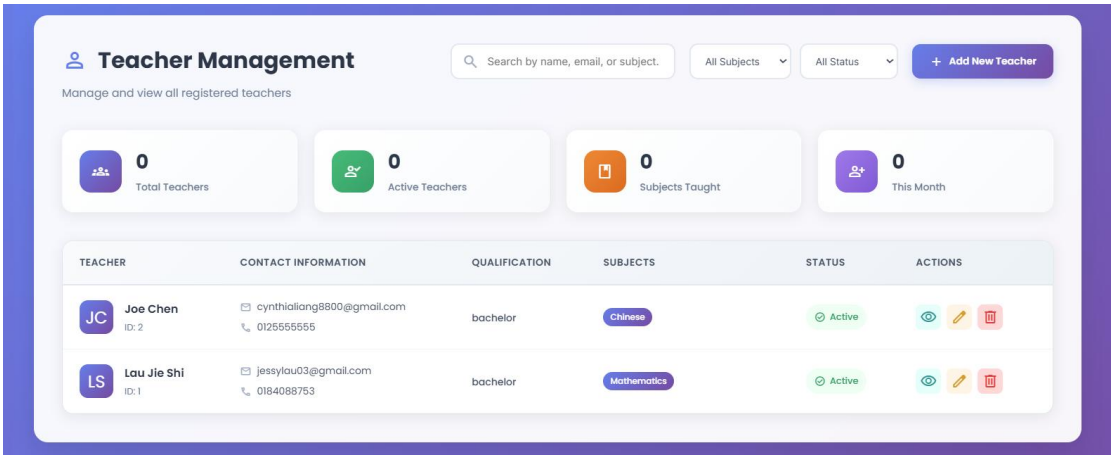


Figure 4.34 Teacher List

The admin can add a new student by filling up the student registration form; the student's parents' information is required. Same as the adding teacher process, the email provided by the parents will receive an email with a randomly generated password. The parent is then required to log in with the given password.

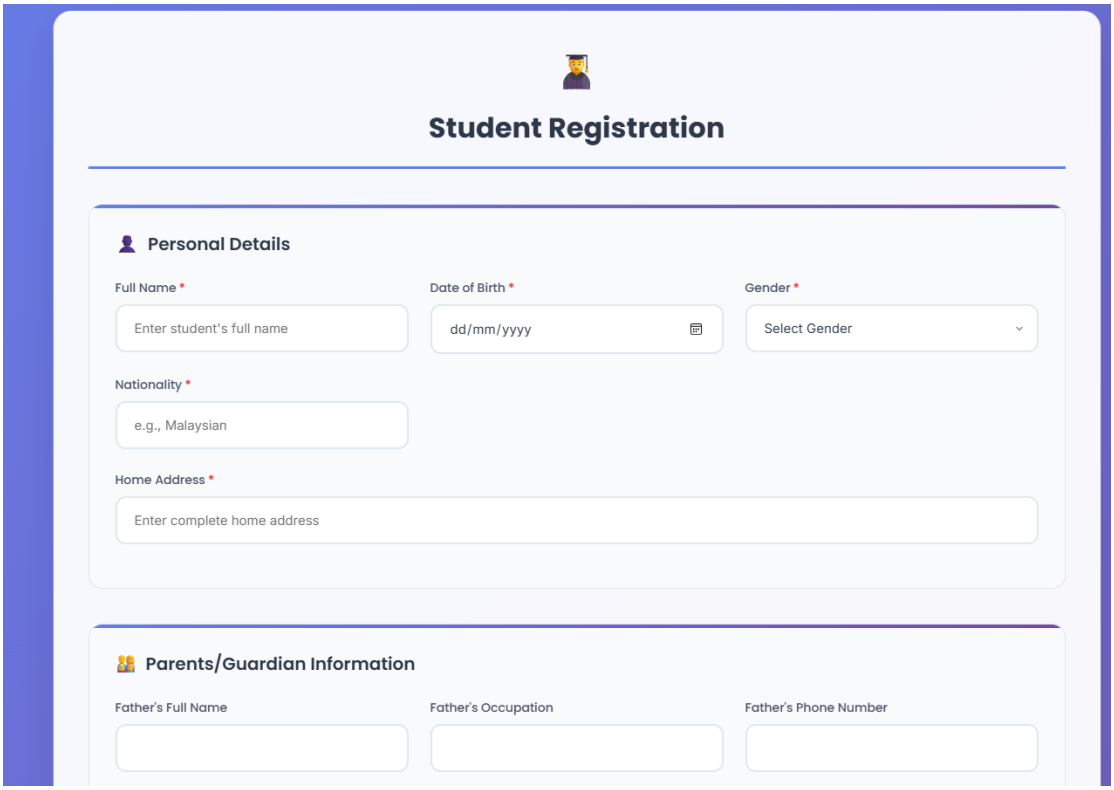


Figure 4.35 Student Registration Form

The student added will be shown in the student list. The admin is allowed to perform view, edit, and delete on the student.

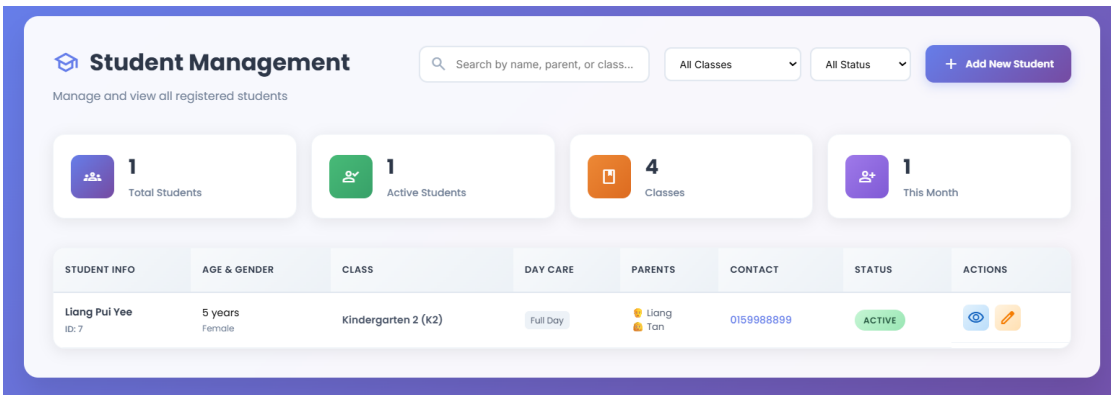


Figure 4.36 Student List

On the financial management, admin can view the overall payment analysis on the billing dashboard.

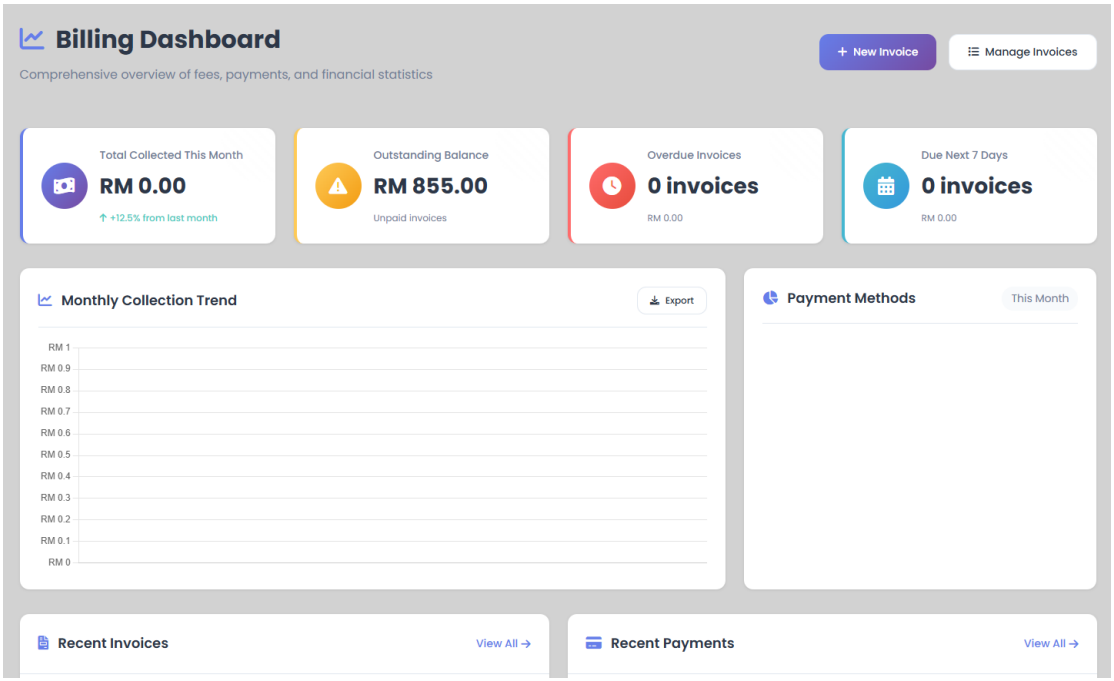
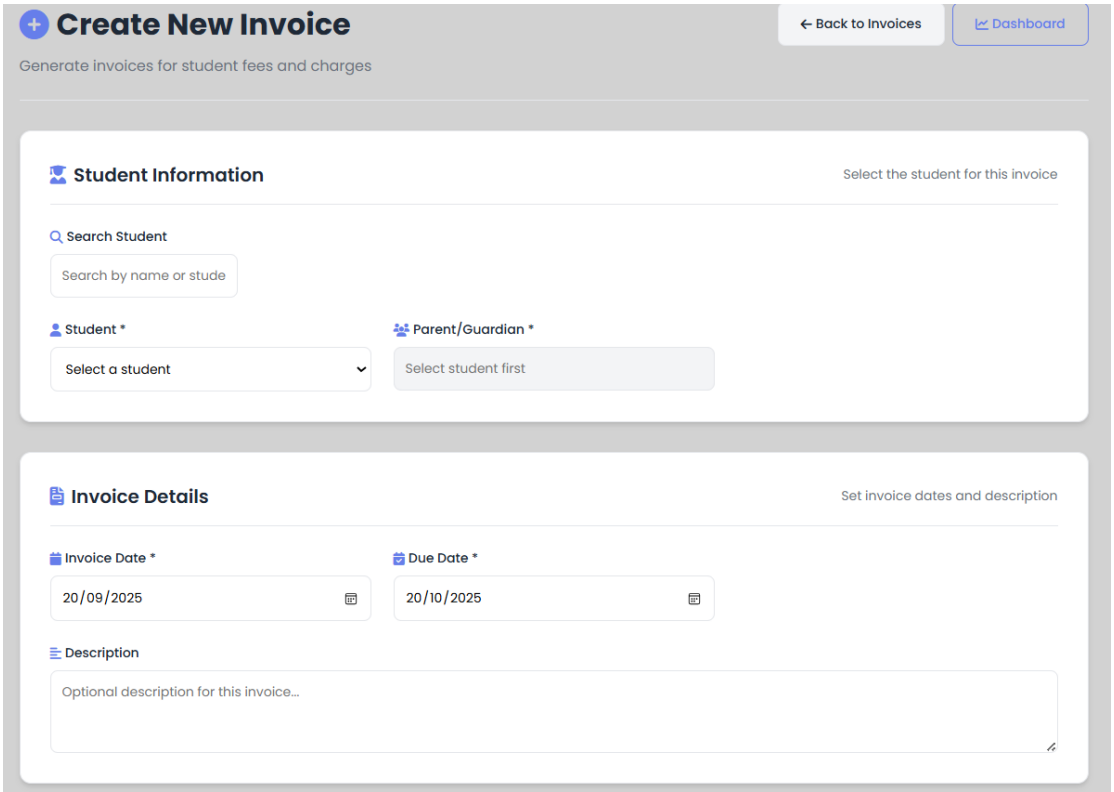


Figure 4.37 Billing Dashboard

By clicking on the create new invoice button, a create new invoice form will be shown. The admin is required to insert the details of the invoice, including the fee items.



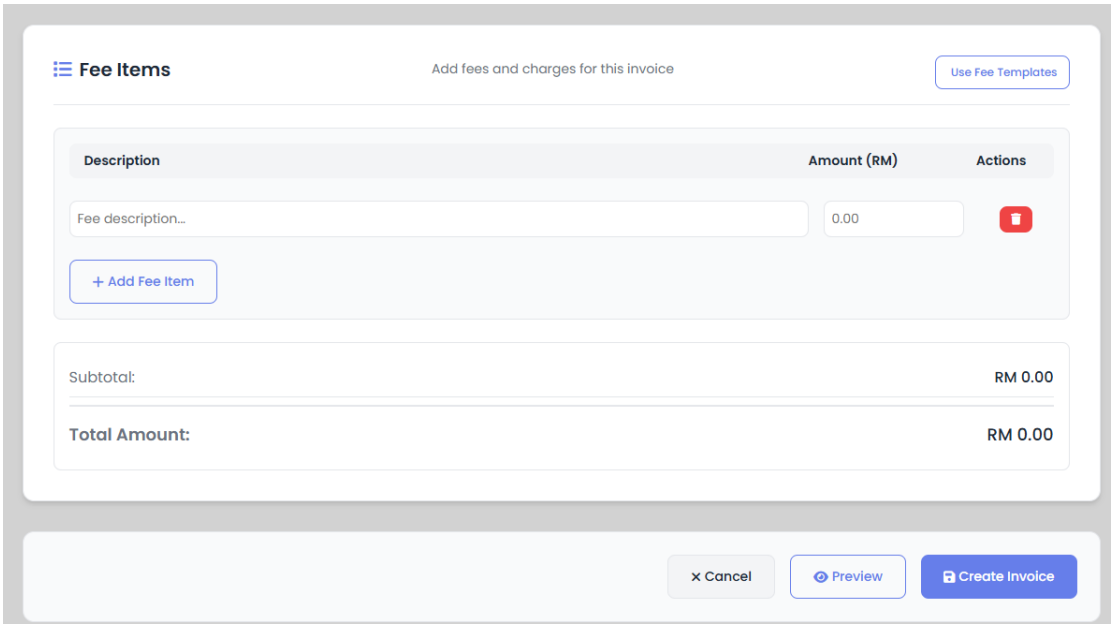


Figure 4.38 Create New Invoice Form

The created invoice form can be viewed when the admin successfully create a new invoice.

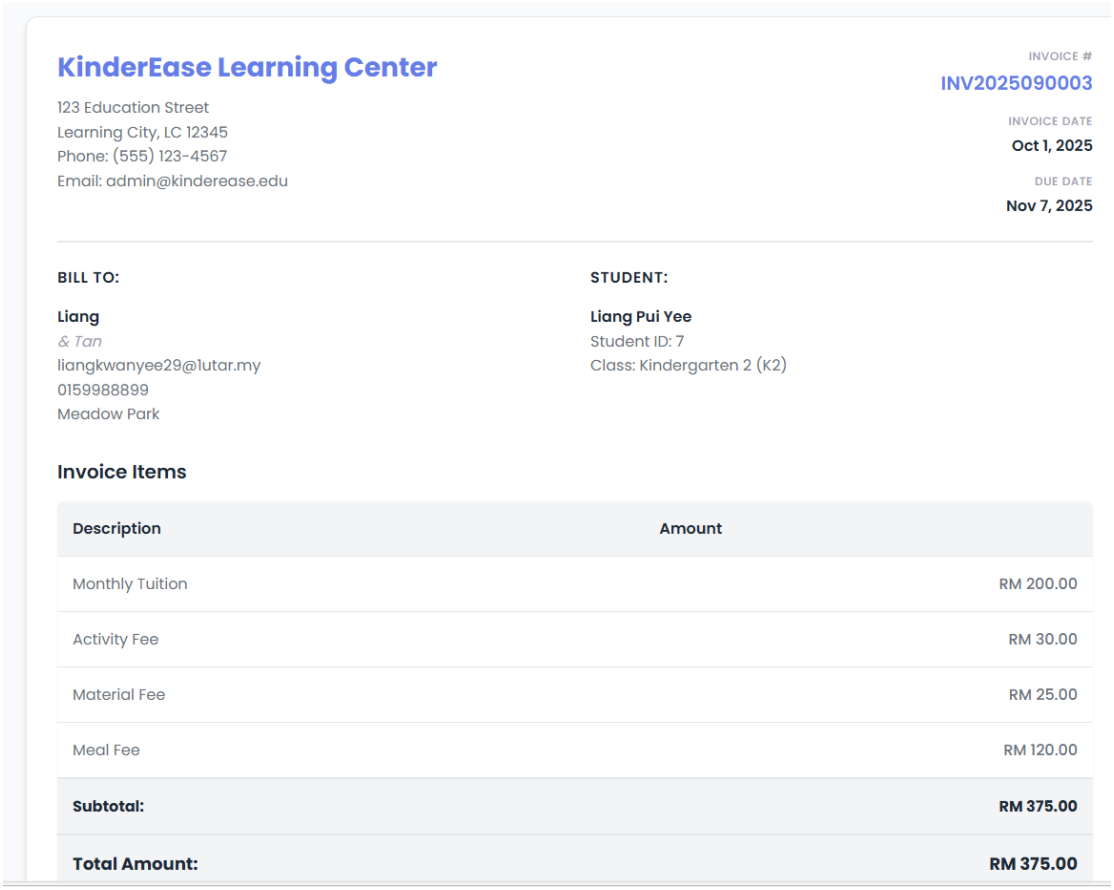


Figure 4.39 View Invoice Form

When there are multiple invoices that need to be created, the admin can choose to create a bulk invoice by selecting multiple students with the same invoice structure.

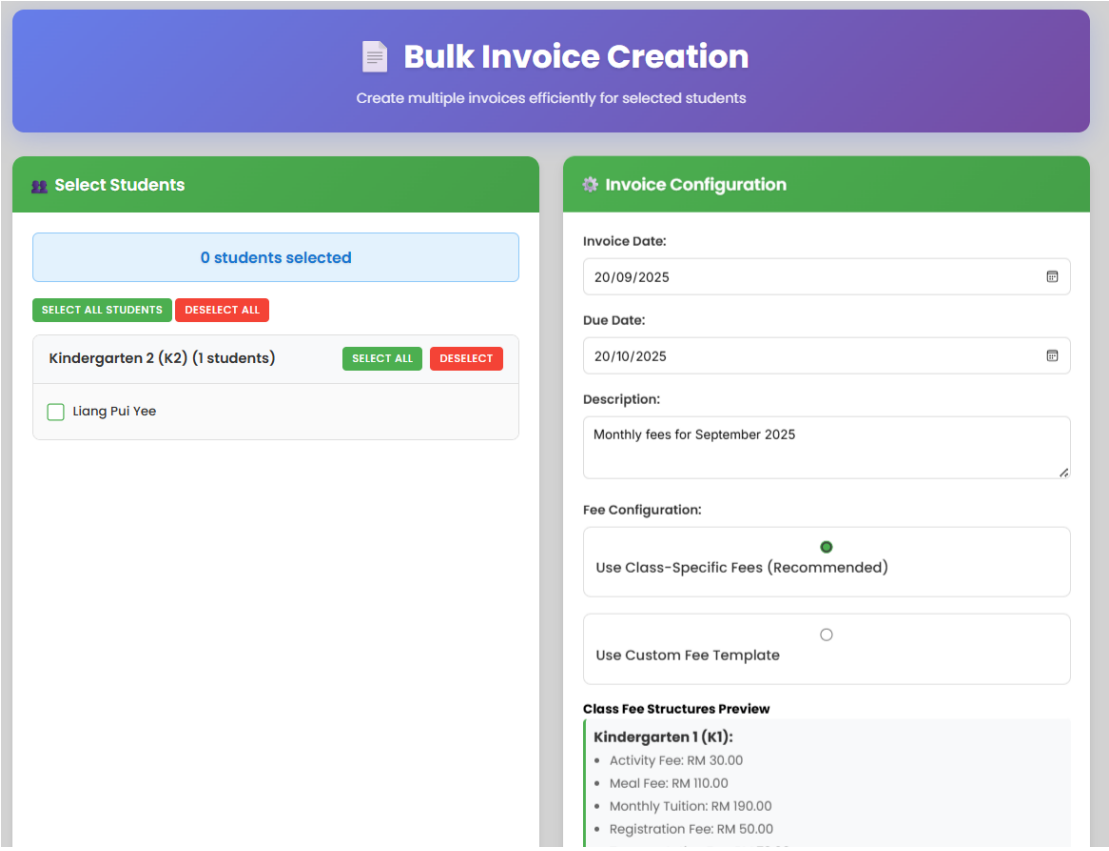
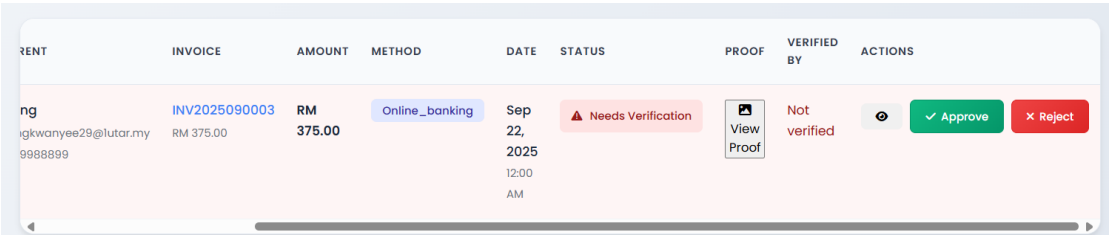


Figure 4.40 Create Bulk Invoice

After the parent makes payment, the admin is required to take action on the payment by approving or rejecting.



5.3 Teacher Role Task

For the teacher in the Kindergarten System, after they logged in to their account, they will be directed to the teacher dashboard. In the dashboard, there will be a sidebar and some necessary overview information shown such as upcoming lesson, today's schedule, recent lesson plans and weekly schedule.

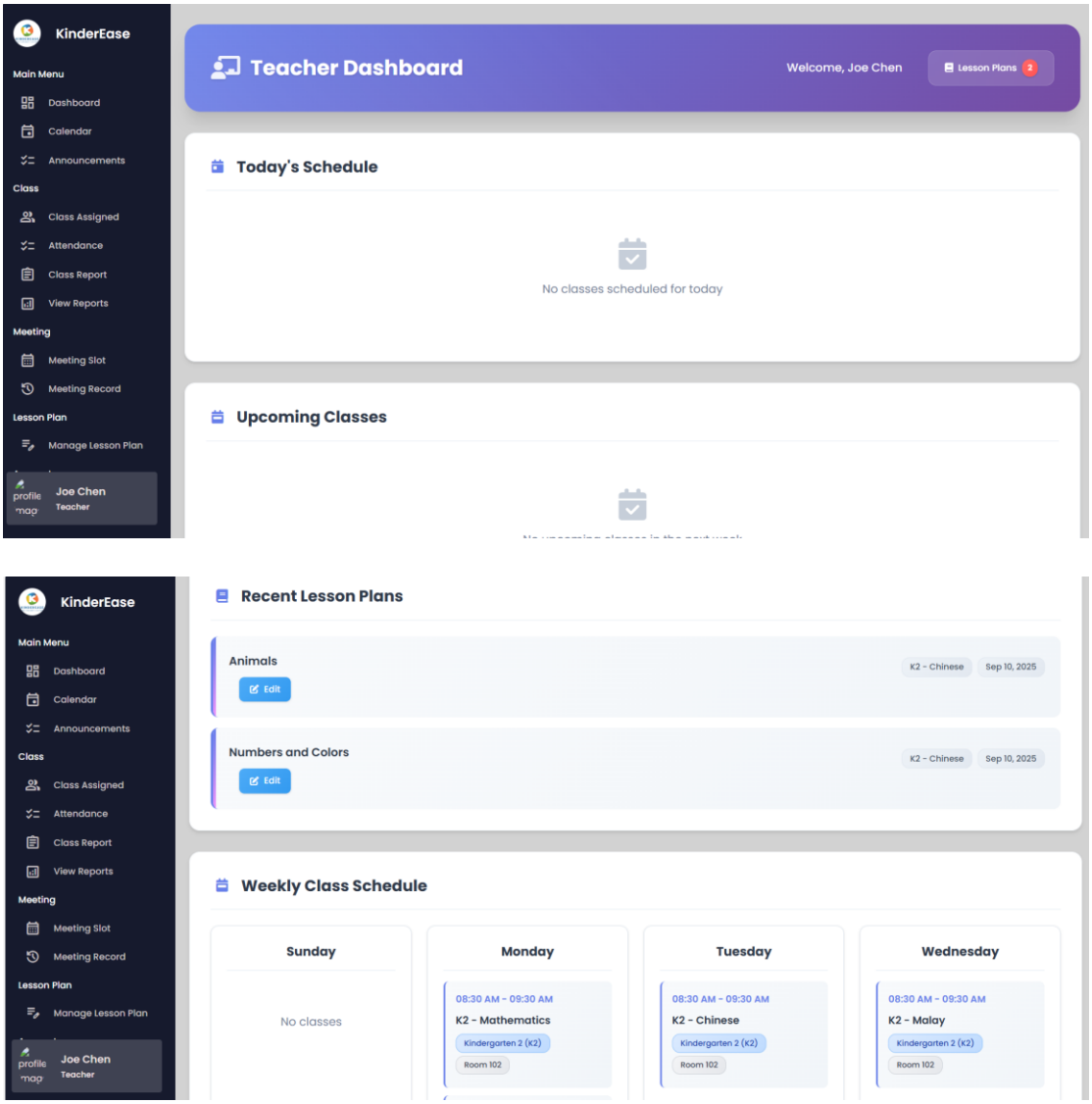


Figure 4.41 Teacher Dashboard

Teachers are able to view their calendar in daily, weekly or monthly view. Their assigned class will also be shown in the calendar.

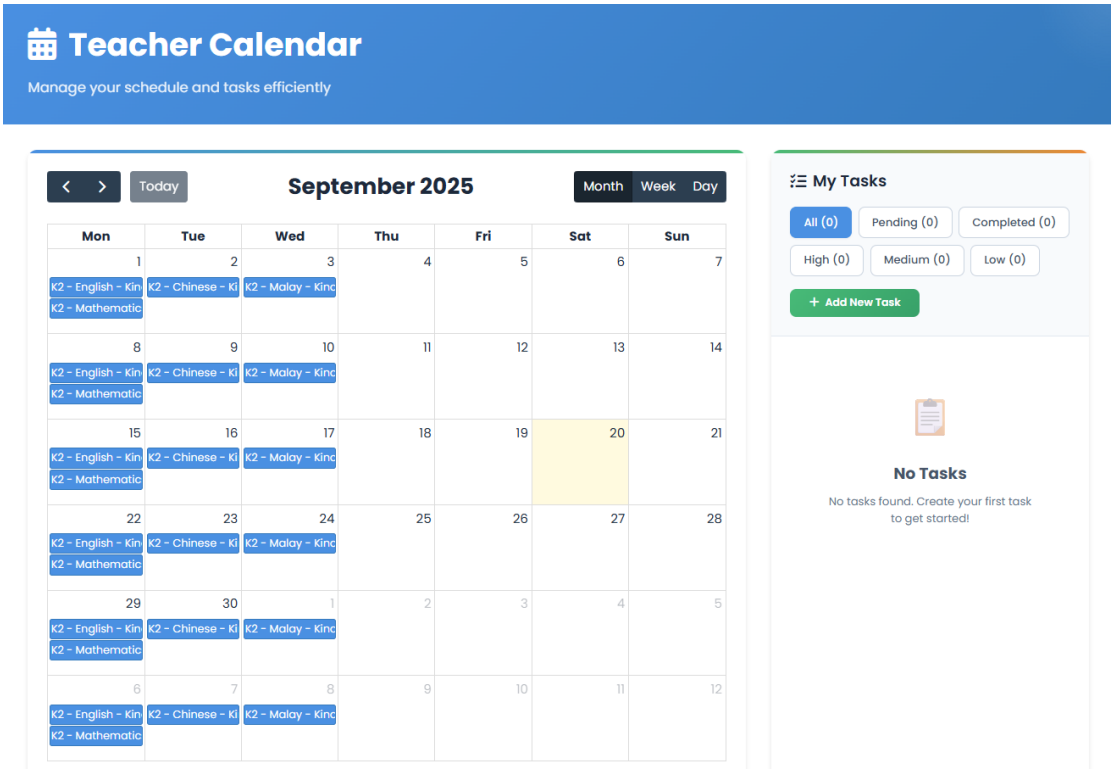


Figure 4.42 Teacher Calendar

Teacher can add new task by defining the due date and priority, the added task will be shown in the task list beside the calendar.

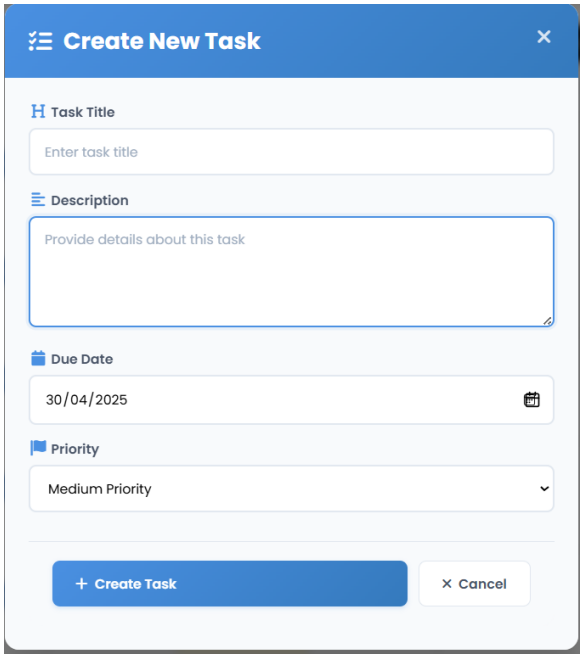


Figure 4.43 Create New Task Form

Teacher can also view the announcement created by administrators and filter the past and current announcements.

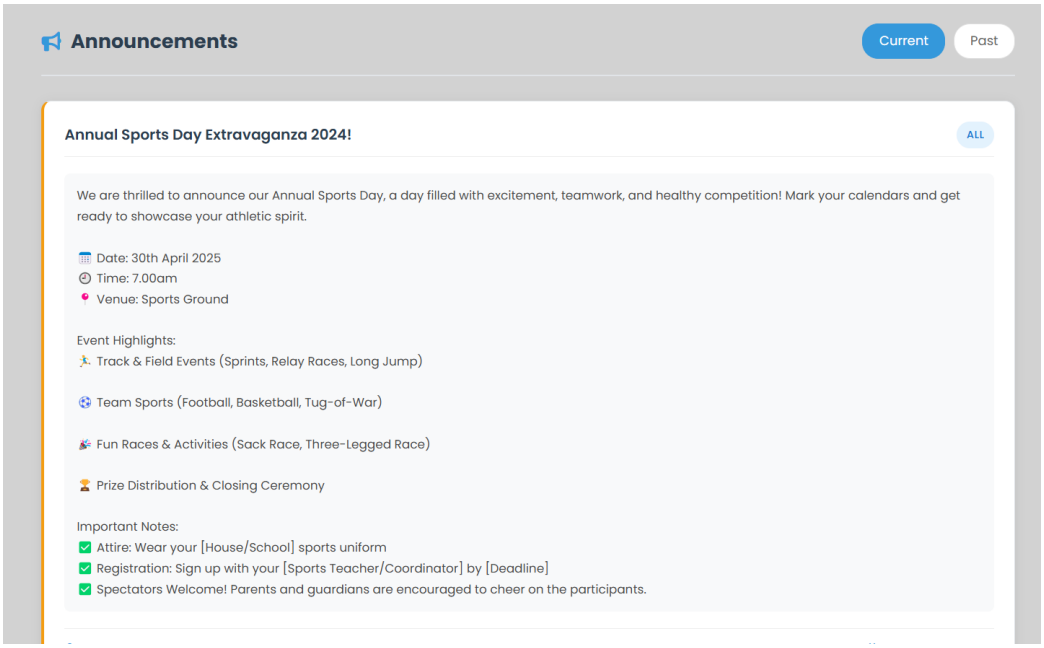


Figure 4.44 Teacher Announcement View

Besides, teacher can view on their assigned class with detailed day, time, and subjects.

Figure 4.45 View Assigned Class

Teacher can manage the student attendance by selecting the class and date in the attendance page, the system will then display the student list of the class level. Teacher can mark student's attendance as attend, absent, leave or excused. Teachers can also add in remarks for each student during the class session and submit a class report for every class.

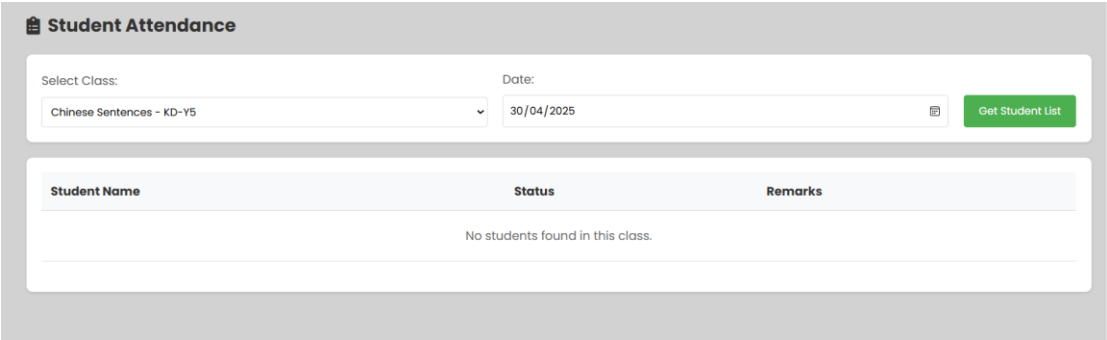


Figure 4.46 Student Attendance

The admin is also required to submit a class report after completing a class

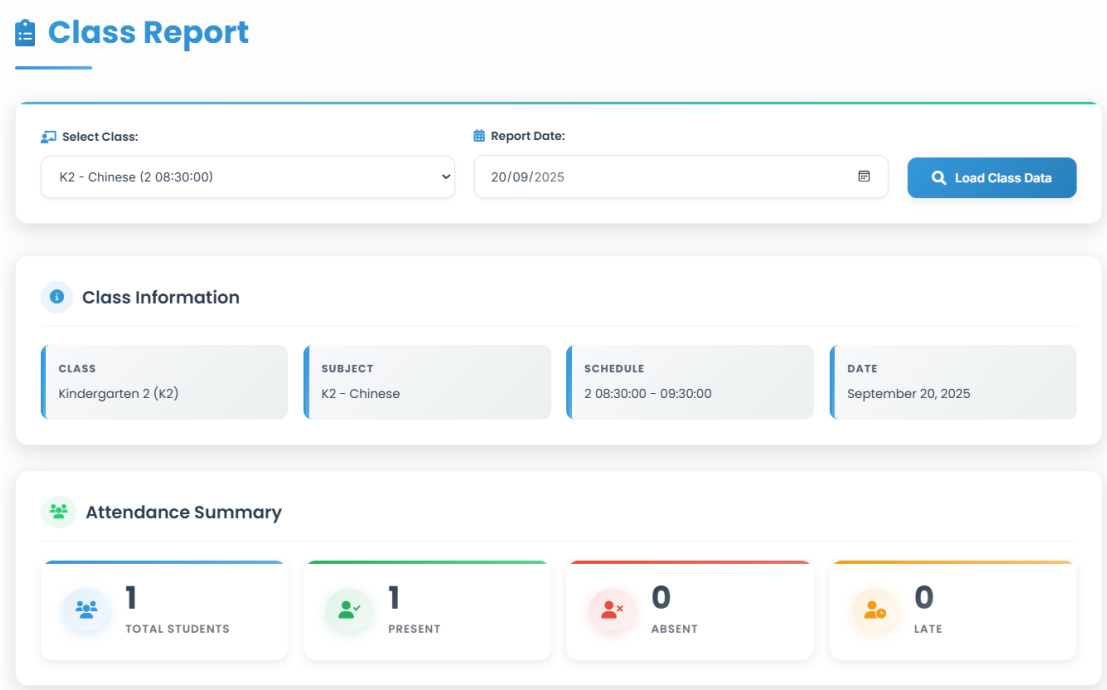


Figure 4.47 Add Clas Report

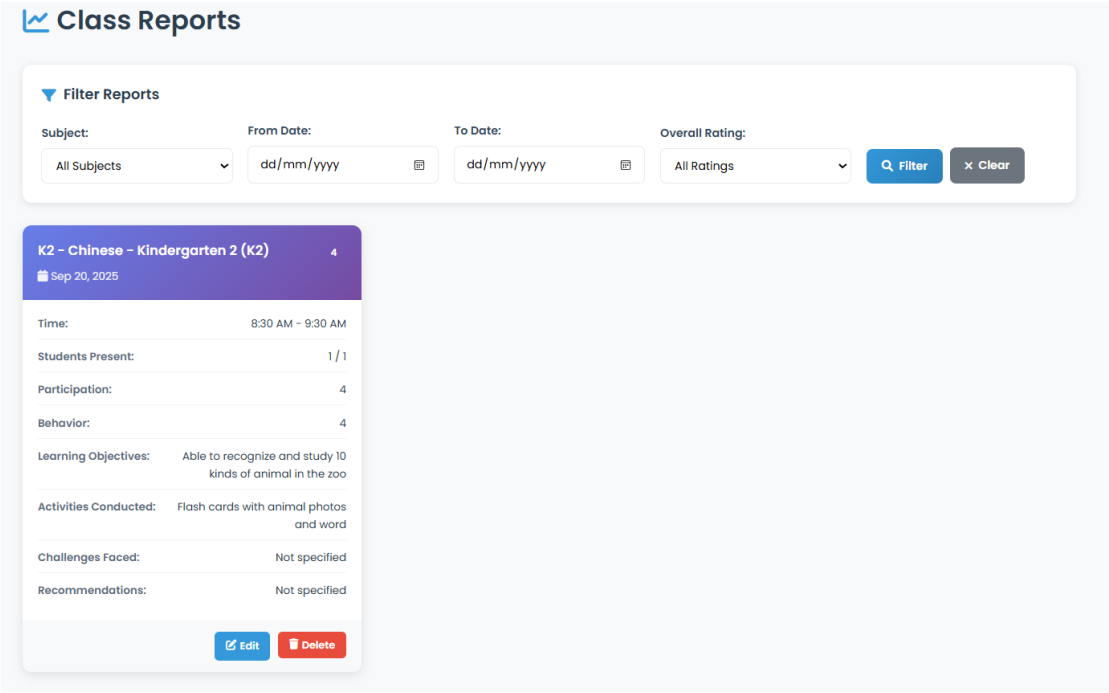


Figure 4.48 View Class Report

To ease the communication between teachers and parents, teacher can manage their own meeting slots by specifying the date, start time and end time. Then a list of created slot will be shown.

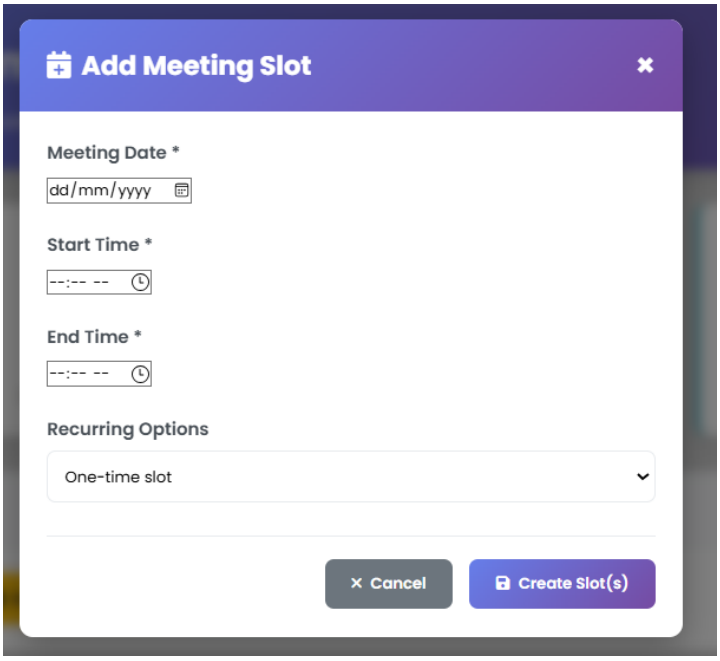


Figure 4.49 Manage Meeting Slot

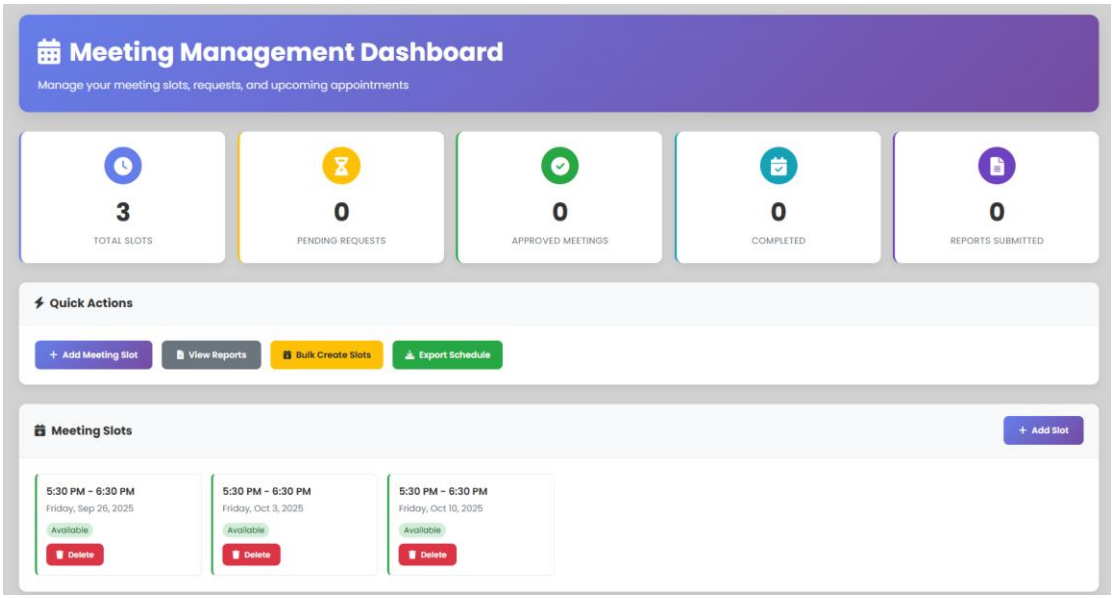


Figure 4.50 View Meeting Slot

When a parent requests on an available meeting slot, it will be shown in the meeting request tab, whereby the teacher needs to take action to approve the meeting request or ask for rescheduling.

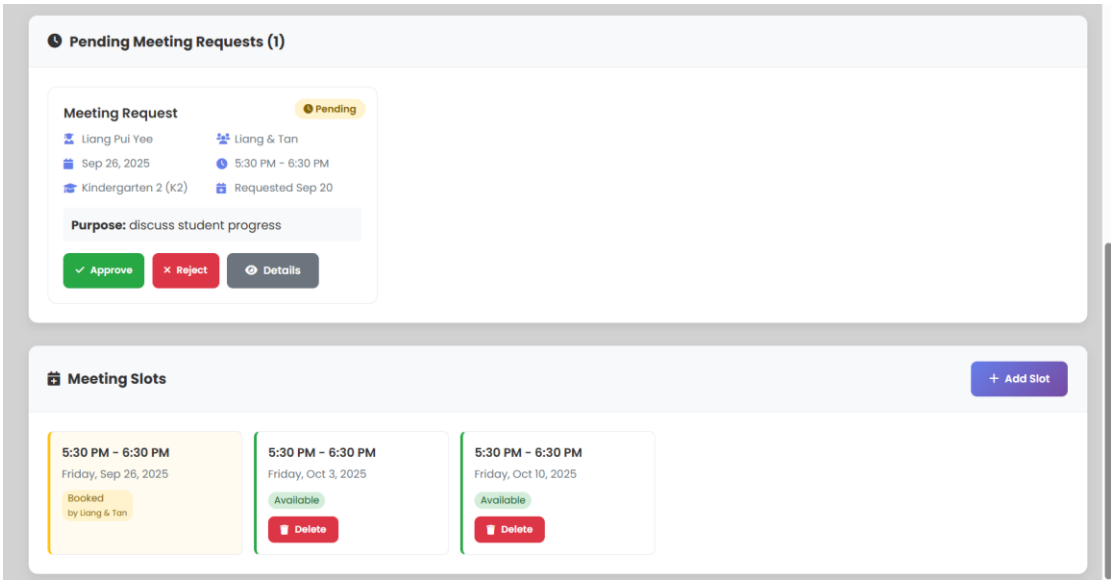
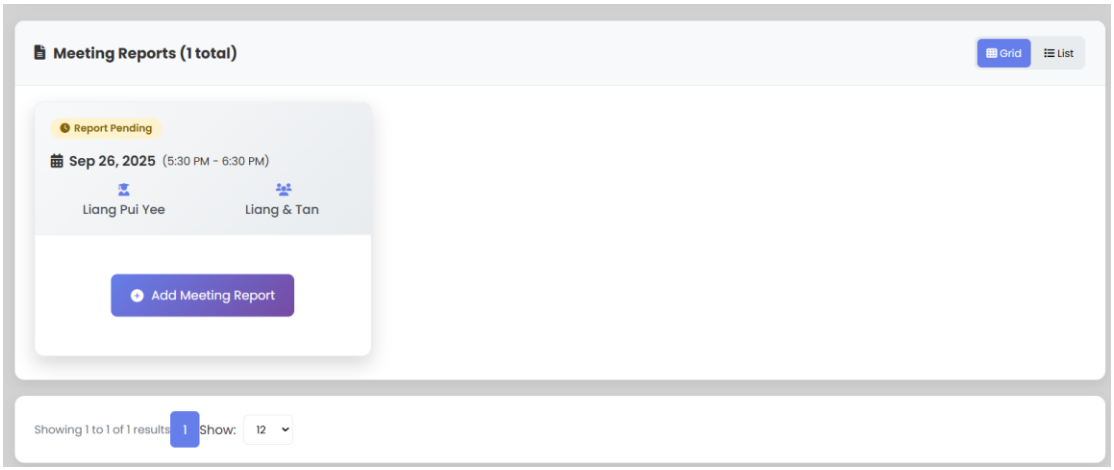


Figure 4.51 Manage Meeting Request

During the meeting, the teacher will be required to submit a meeting report with particular student updates.



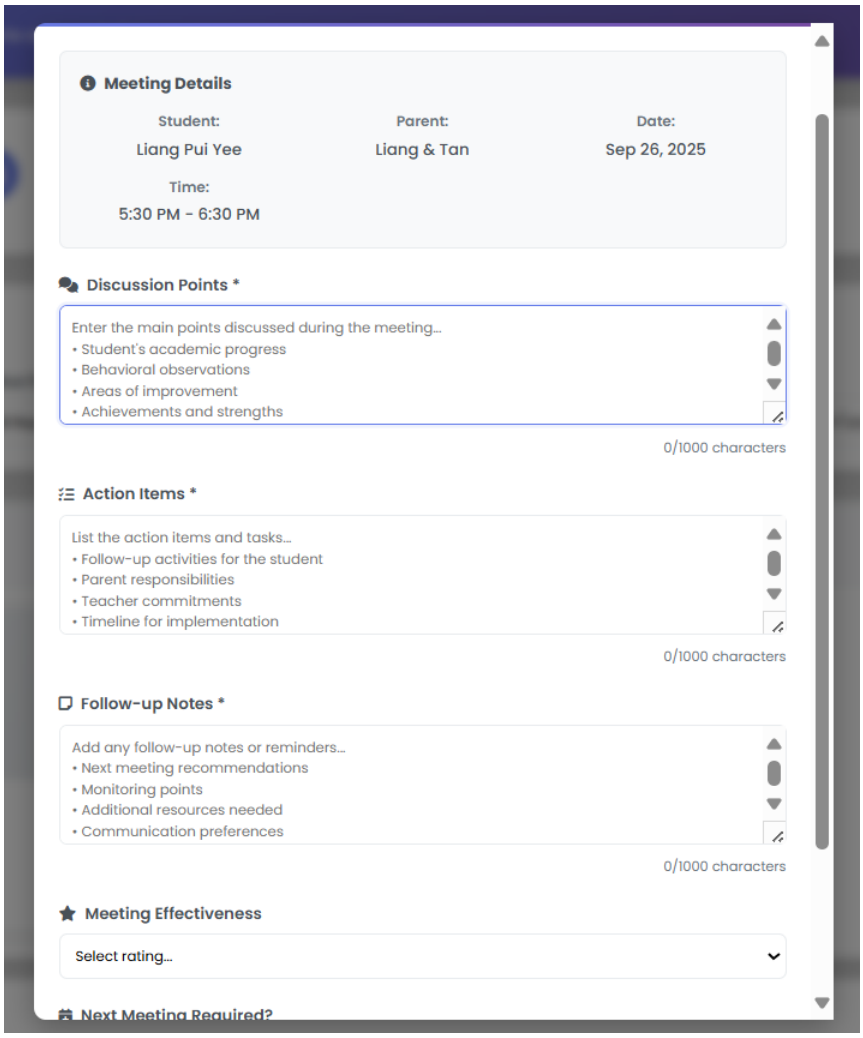


Figure 4.52 Submit Meeting Report

The submitted report will then be saved in the database and updated in the meeting records.

Figure 4.53 Meeting Record

The teacher can create their lesson plan by clicking on the Manage Lesson Plan from the sidebar. A form will be displayed for the teacher to select on the subject assigned to them and fill up the details of the lesson plan. The lesson plan can be set as public that will be shown for every teacher site or private for personal use only. Besides, the teacher can also save the lesson plan as a template for future use.

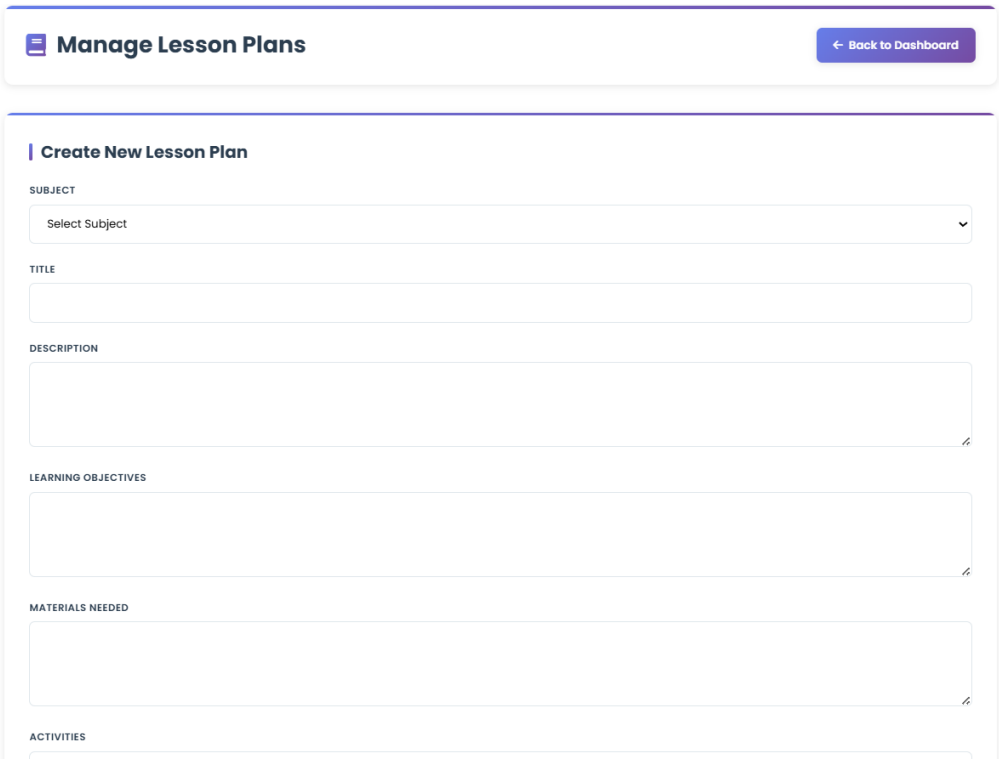


Figure 4.54 Manage Lesson Plan

The created lesson plan will be shown in “My Lesson Plans”

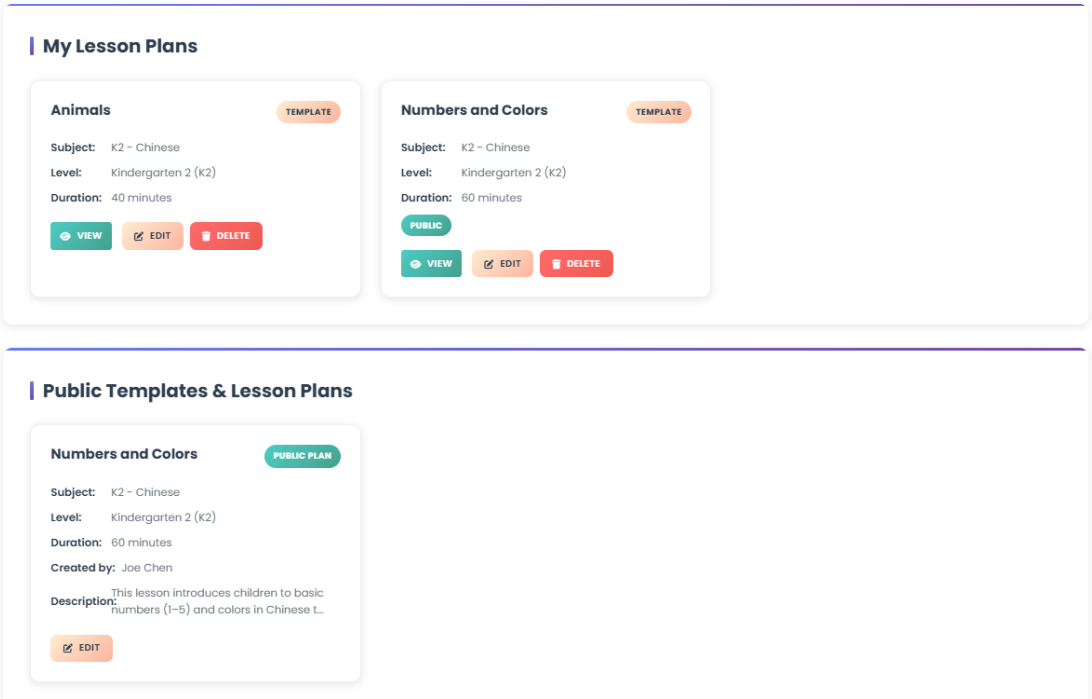


Figure 4.55 Lesson Plan Template

5.4 Parent Role Task

Upon parent's login, the parent will be directed to the parent dashboard. Their children overview, attendance summary, upcoming classes, announcements, and meeting status will be displayed in card view in the dashboard.

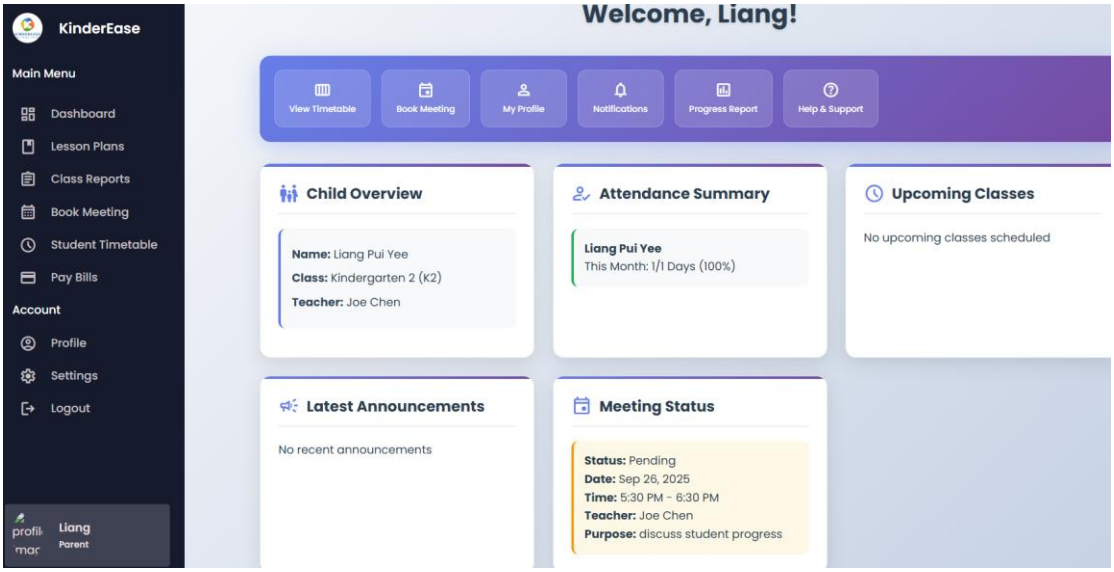
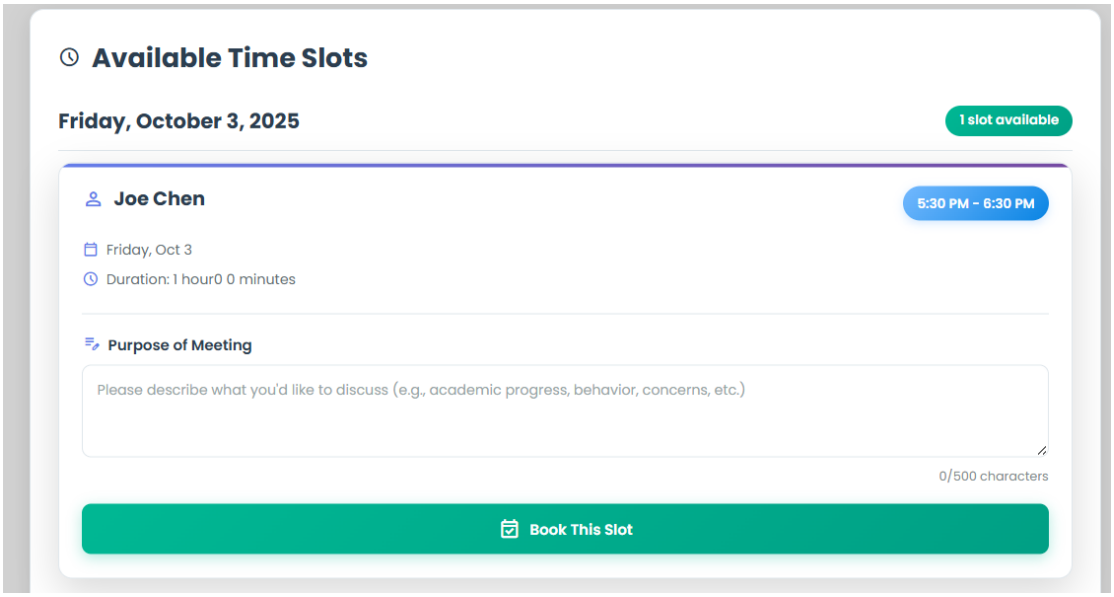


Figure 4.56 Parent Dashboard

The parent is able to book meeting with teachers by clicking on the available slots created by the teacher. The parent needs to enter the purpose of meeting at their desire slot.



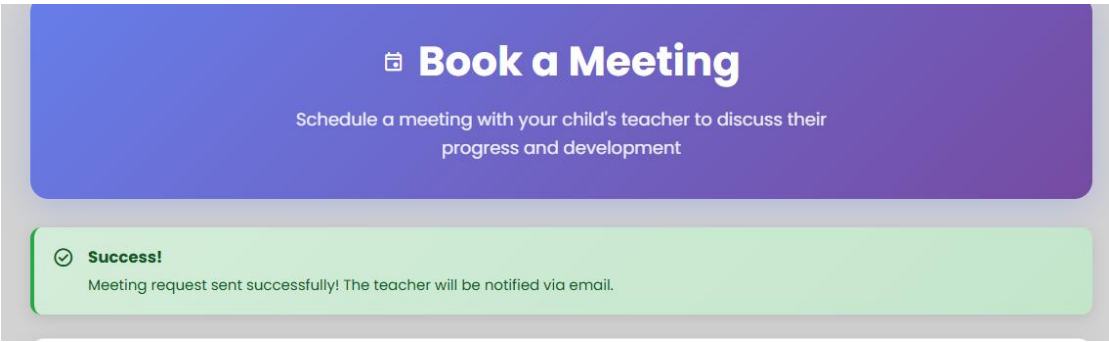


Figure 4.57 Book Meeting

Then, an email from the parent will be sent to the teacher and awaiting for the teacher's approval for the meeting request.

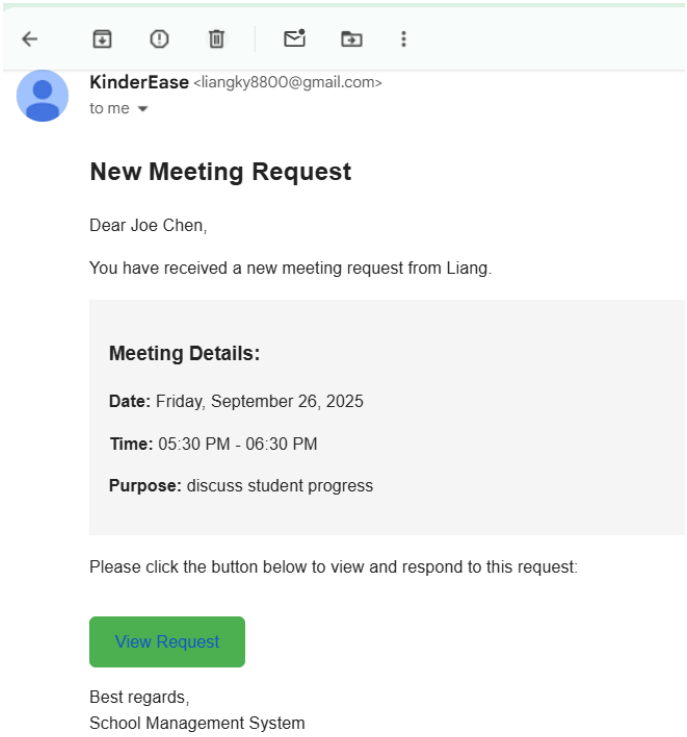


Figure 4.58 Meeting Request Email

Beside the core function in parent dashboard, parents are also allowed to view the lesson plan created within their children's class level.

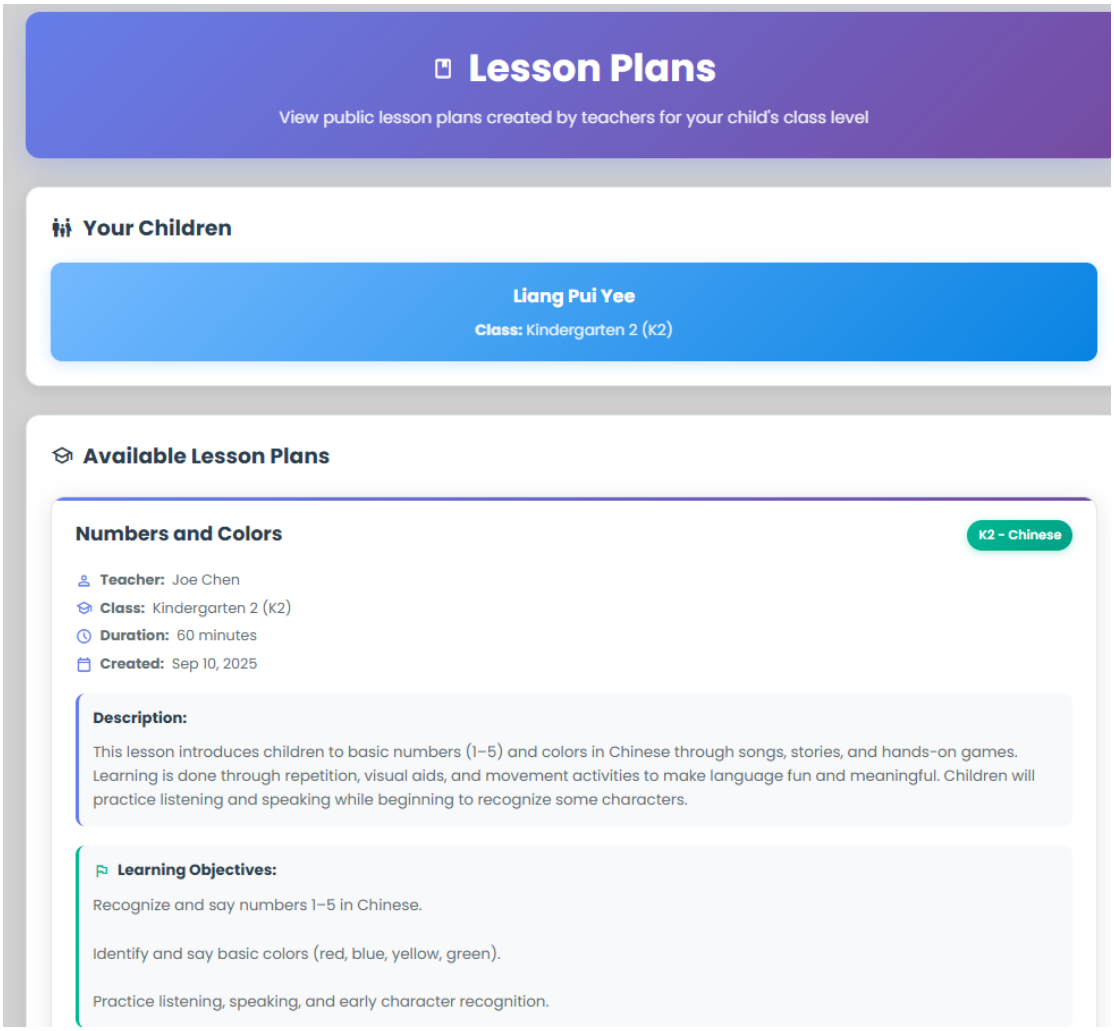


Figure 4.59 View Lesson Plan

Parents can also view the class report submitted by the teacher in order to know the how the class is being conducted.

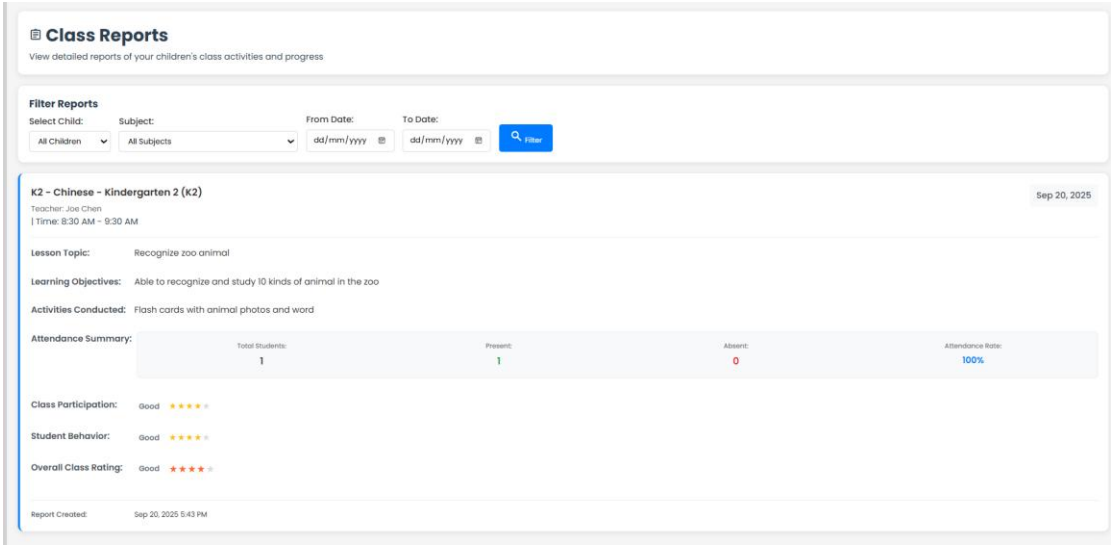


Figure 4.60 View Class Report

Parents are also able to view their children's weekly timetable.

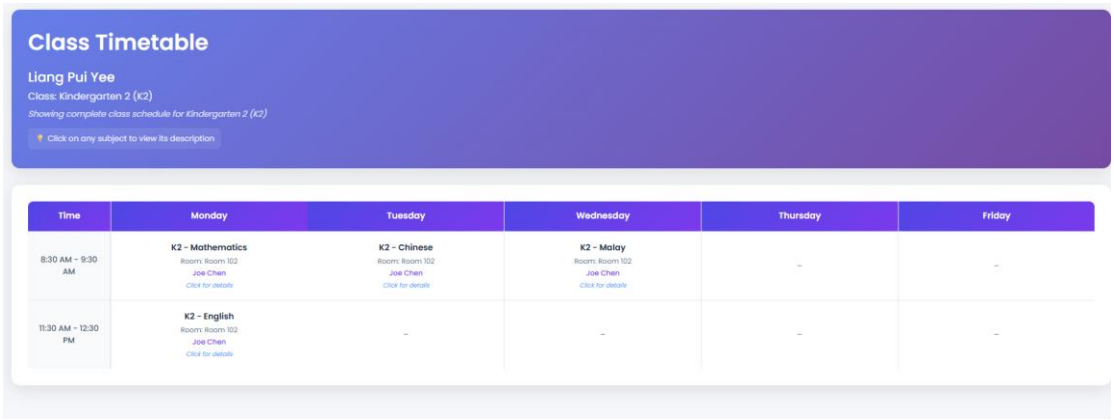


Figure 4.61 View Student's Timetable

Lastly, parents can also make payment by viewing the upcoming bills from their dashboard page.

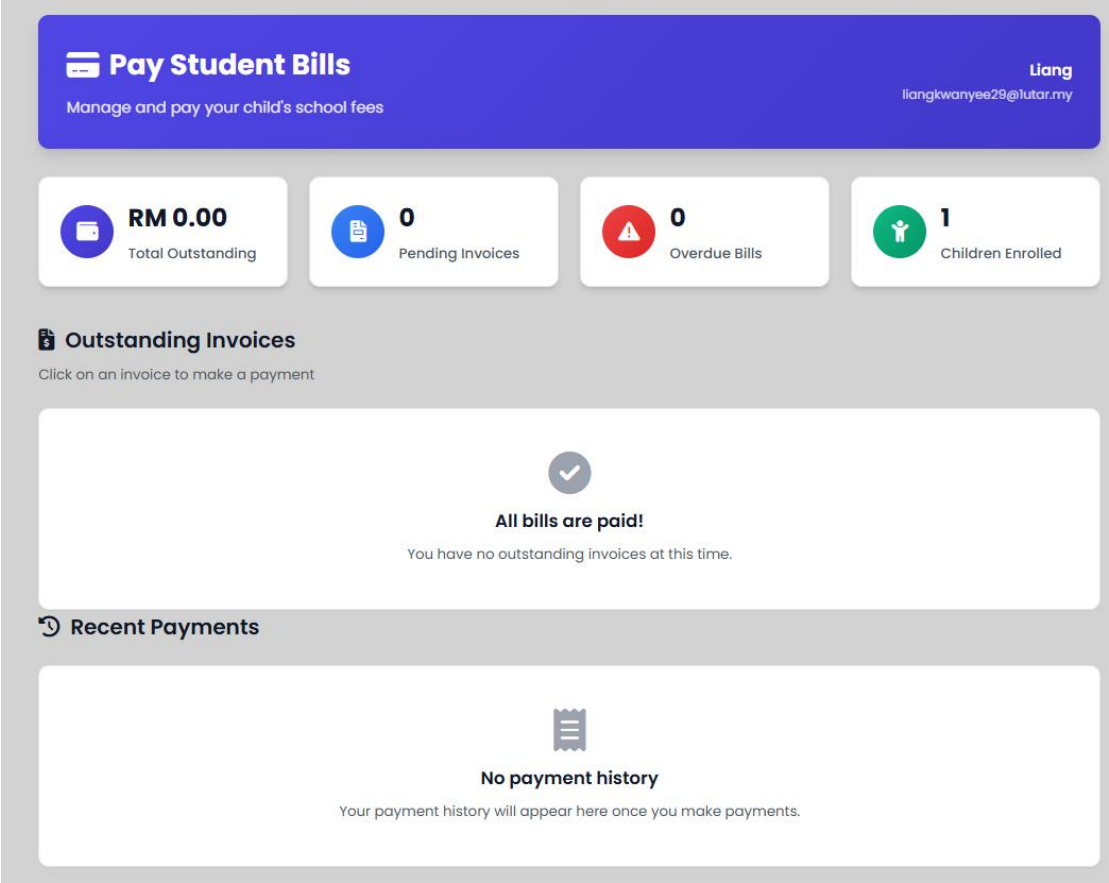


Figure 4.62 Pay Bills

CHAPTER 6 System Evaluation and Discussion

6.1 System Testing and Result

6.1.1 Login Module

No.	Test Case	Input	Expected Output	Actual Result
1.	Login with correct email and password	Email: admin@kinderease.com Password: kinderease123 CAPTCHA: correct_captcha_code	- Redirect to admin_Dashboard.php - Session bariables set: user_id, full_name, user_type = 'admin' - Display success message	Same as expected output
2.	Login with invalid email format	Email: admin.com	- Error message: "Invalid email format" - Remain on login Page - No session created	Same as expected output
3.	Wrong Password	Email: admin@kinderease.com Password: wrongpassword CAPTCHA: correct_captcha_code	- Error message: "Invalid Password" - Remain on login Page - Login attempt counter incremented	Same as expected output
4.	Attempt to login more the 5 times	6 consecutive failed login attempts	- Error message: "Too many failed attempts. Please try again in 15 minutes." - Account locked for 15 minutes - No further login attempts allowed	Same as expected output
5.	Invalid CAPTCHA	Email: admin@kinderease.com Password: kinderease123 CAPTCHA: wrong_captcha_code	- Error message: "Invalid CAPTCHA code" - Remain on login page	Same as expected output

Table 6.1 Login Module Testing

6.1.2 Student Management

No.	Test Case	Input	Expected Output	Actual Result

6.	Valid Student Registration	Full Name: "John Doe" DOB: "2019-05-15" Gender: "Male" Nationality: "Malaysian" Home Address: "Kampar, Perak" Father Name: "Robert Doe" Father Phone: "0123456789" Father Email: "robert@email.com" Mother Name: "Jane Doe" Mother Phone: "0198765432" Class Assigned: "Nursery 1" Academic Year: "2025"	- Success message: "Student registered successfully" - Student ID generated automatically - Parent account created with login credentials - Email sent to parents with login details	Same as expected output
7.	Missing Required Fields	Full Name: "" (empty) DOB: "2019-05-15" Father Name: "Robert Doe"	- Error message: "Full name is required" - Form validation highlights empty fields - No student record created	Same as expected output
8.	Invalid Phone Number	Father Phone: "123456" (invalid format) Mother Phone: "abcdefghij" (non-numeric)	- Error message: "Invalid father's phone number format" - Error message: "Invalid mother's phone number format" - Form submission blocked	Same as expected output
9.	Invalid Email Format	Email: "invalid-email-format"	- Error message: "Invalid mother's phone number format" - Form submission blocked	Same as expected output
10.	Future Date of Birth	DOB: "2025-12-31" (future date) CAPTCHA: wrong_captcha_code	- Error message: "Invalid date of birth" - Form validation fails	Same as expected output

11.	Class at Full Capacity	Class Assigned: "Nursery 1" (already at 20/20 capacity)	- Error message: "Selected class is at full capacity" - Student registration blocked	Same as expected output
12.	Edit Student Information	Student ID: 123 Updated Phone: "0111234567" Updated Address: "456 New Street, KL"	- Success message: "Student information updated successfully" - Database record updated Change log created	Same as expected output

Table 6 .2 Student Management Testing

6.1.3 Teacher Management

No.	Test Case	Input	Expected Output	Actual Result
13.	Valid Teacher Registration	Full Name: "Sarah Johnson" Email: "sarah.johnson@email.com" Contact: "0123456789" Qualification: "Bachelor of Education" Field of Study: "Early Childhood Education" Experience: "5" Subjects: "Mathematics, English"	- Success message: "Teacher registered successfully" - Teacher ID generated - Login credentials created - Welcome email sent	Same as expected output
14.	Duplicate Email Registration	Email: "existing.teacher@email.com" (already exists)	- Error message: "Email already exists in the system" - Registration blocked - No duplicate record created	Same as expected output
15.	Invalid Experience Value	Experience: "-5" (negative value)	- Error message: "Experience must be a positive number" - Form validation fails	Same as expected output
16.	Missing Required Qualification	Qualification: "" (empty) Field of Study: "Mathematics"	- Error message: "Qualification is required" - Form submission blocked	Same as expected output

Table 6.3 Teacher Management Testing

6.1.4 Class Management and Scheduling

No.	Test Case	Input	Expected Output	Actual Result
17.	Valid Subject Scheduling	Class Level: "Nursery 1" Subject: "Mathematics" Day: "Monday" Start Time: "09:00" End Time: "10:00" Teacher: "Sarah Johnson" Classroom: "Room 101" Capacity: "20"	- Success message: "Subject scheduled successfully" - Timetable updated - Teacher assigned to class	Same as expected output
18.	Time Conflict Scheduling	Teacher: "Sarah Johnson" Day: "Monday" Start Time: "09:30" (conflicts with existing 09:00-10:00) End Time: "10:30"	- Error message: "Teacher has a scheduling conflict at this time" - Scheduling blocked - Suggest alternative times	Same as expected output
19.	Invalid Time Range	Start Time: "10:00" End Time: "09:00" (end before start)	- Error message: "End time must be after start time" - Form validation fails	Same as expected output

Table 6 .4 Class Management and Scheduling Testing

6.1.5 Financial Management

No.	Test Case	Input	Expected Output	Actual Result
20.	Valid Invoice Creation	Student: "John Doe" Parent: "Robert Doe" Due Date: "2024-02-28" Fee Items: [{description: "Monthly Tuition", amount: "500.00"}, {description: "Activity Fee", amount: "50.00"}] Description: "February 2024 Fees"	- Error message: "Invoice must have at least one fee item with amount greater than 0" - Invoice creation blocked	Same as expected output
21.	Invoice with Zero Amount	Fee Items: [{description: "Tuition", amount: "0.00"}]	- Error message: "Invoice must have at least one fee item with amount greater than 0" - Invoice creation blocked	Same as expected output
22.	Invalid Due Date	Due Date: "2023-01-01" (past date)	- Warning message: "Due date is in the past"	Same as expected output

			- Allow creation with confirmation	
23.	Bulk Invoice Generation	Class: "Nursery 1" Fee Type: "Monthly Tuition" Amount: "500.00" Students Selected: [All 20 students]	- Success message: "20 invoices created successfully" - Individual invoices for each student - Batch processing confirmation	Same as expected output

Table 6.5 Financial Management Testing

6.1.6 Academic Year Management

No.	Test Case	Input	Expected Output	Actual Result
24.	Create New Academic Year	Year Name: "2024-2025" Start Date: "2024-01-01" End Date: "2024-12-31" Status: "Active"	- Success message: "Academic year created successfully" - Year becomes current active year - Previous year status updated to "Completed"	Same as expected output
25.	Overlapping Academic Years	Start Date: "2024-06-01" End Date: "2025-05-31" (Overlaps with existing 2024-2025 year)	- Error message: "Academic year dates overlap with existing year" - Creation blocked	Same as expected output
26.	Invalid Date Range	Start Date: "2024-12-31" End Date: "2024-01-01"	- Error message: "End date must be after start date" - Form validation fails	Same as expected output

Table 6.6 Academic Year Management Testing

6.1.7 Announcement Management

No.	Test Case	Input	Expected Output	Actual Result
27.	Create New Academic Year	Year Name: "2024-2025" Start Date: "2024-01-01" End Date: "2024-12-31" Status: "Active"	- Success message: "Academic year created successfully" - Year becomes current active year - Previous year status updated to "Completed"	Same as expected output
28.	Empty Announcement Content	Title: "Test Announcement" Content: "" (empty)	- Error message: "Announcement content is required" - Publication blocked	Same as expected output

Table 6.7 Announcement Management Testing

6.1.8 Calendar and Task Management

No.	Test Case	Input	Expected Output	Actual Result
29.	Create New Task	Title: "Prepare Math Worksheets" Description: "Create worksheets for addition practice" Due Date: "2024-02-15" Priority: "High"	- Success message: "Task created successfully!" - Task appears in calendar view - Task counter updated - Modal closes automatically	Same as expected output
30.	Invalid Task Creation - Empty Title	Title: "" (empty) Description: "Test description" Due Date: "2024-02-15"	- Error message: "This field is required" - Form validation highlights empty title field - Task not created - Modal remains open	Same as expected output
31.	Invalid Task Creation - Past Due Date	Title: "Test Task" Due Date: "2023-01-01" (past date)	- Error message: "Due date cannot be in the past" - Form validation fails - Task not created	Same as expected output
32.	Edit Existing Task	Task ID: 123 Updated Title: "Modified Task Title" Updated Priority: "Medium"	- Success message: "Task updated successfully!"	Same as expected output

			- Changes reflected in the calendar - Task list updated	
33.	Complete Task	- Click the complete button on task ID 123 - Confirm completion dialog	- Confirmation dialog: "Mark this task as completed?" - Task status changed to completed - Task moved to completed filter - Completed counter incremented	Same as expected output
34.	Delete Task	- Click delete button on task ID 123 - Confirm deletion dialog	- Confirmation dialog: "Are you sure you want to delete this task? This action cannot be undone." - Task removed from the calendar - Task counters updated	Same as expected output

Table 6.8 Calendar and Task Management Testing

6.1.9 Lesson Plan Management

No.	Test Case	Input	Expected Output	Actual Result
35.	Create Valid Lesson Plan	Title: "Introduction to Addition" Subject: "Mathematics" Description: "Basic addition concepts for nursery students" Objectives: "Students will learn to add numbers 1-10" Materials: "Counting blocks, worksheets, whiteboard" Activities: "Hands-on counting, group exercises" Assessment: "Oral questioning and worksheet completion" Duration: "45" Is Template: false	- Success message: "Lesson plan created successfully!" - Plan ID generated automatically - Plan appears in lesson plans list -	Same as expected output

		Is Public: true		
36.	Missing Required Fields	Title: "" (empty) Subject: "Mathematics" Objectives: "" (empty)	- Error messages: "Title is required", "Objectives are required" - Form validation highlights empty fields - Lesson plan not created	Same as expected output
37.	File Uploaded with Lesson Plan	Valid lesson plan data + Files: ["worksheet.pdf" (2MB), "activity_image.jpg" (1MB), "demo_video.mp4" (10MB)]	- Lesson plan created successfully - All files uploaded and linked to plan - File attachments accessible from plan view - File types validated correctly	Same as expected output
38.	Invalid File Upload	Valid lesson plan data + Files: ["malicious.exe" (executable file), "huge_file.mp4" (60MB - exceeds limit)]	- Error message: "Invalid file type" for .exe file - Error message: "File size exceeds maximum limit of 50MB" - Lesson plan created but files rejected - Only valid files uploaded	Same as expected output

Table 6 .9 Lesson Plan Management Testing

6.1.10 Meeting Management

No.	Test Case	Input	Expected Output	Actual Result
39.	Create Meeting Slot	Date: "2024-02-20" Start Time: "14:00" End Time: "14:30" Max Bookings: "1" Notes: "Parent consultation available"	- Success message: "Meeting slot created successfully!" - Slot appears in calendar - Available for parent booking	Same as expected output

40.	Overlapping Meeting Slot	Date: "2024-02-20" Start Time: "14:15" End Time: "14:45" (Overlaps with existing 14:00-14:30 slot)	- Error message: "Time slot conflicts with existing meeting" - Slot creation blocked - Suggest alternative times - Show existing schedule	Same as expected output
41.	Complete Meeting With Report	Meeting ID: 123 Student Discussion: "Discussed math progress and areas for improvement" Parent Feedback: "Parents concerned about homework difficulty" Action Items: "Provide additional practice materials" Follow-up Required: true	- Success message: "Meeting report submitted successfully!" - Meeting status changed to completed - Report accessible to admin and parents - Follow-up reminder created if required	Same as expected output
42.	Cancel Meeting Slot	Slot ID: 123 Reason: "Teacher unavailable due to emergency" Notify Parents: true	- Confirmation dialog: "Cancel this meeting slot?" - Slot removed from calendar - Notification sent to booked parents - Alternative slots suggested	Same as expected output

Table 6.10 Meeting Management Testing

6.1.11 Meeting Booking System

No.	Test Case	Input	Expected Output	Actual Result
43.	Valid Meeting Booking	- Teacher ID: 1 - Available slot ID: 5 - Purpose: "Discuss child's progress"	- Success message: "Booking successful, confirmation message with booking reference, email sent to teacher"	Same as expected output
44.	Duplicate Slot Booking	Attempt to book already booked slot	- Error message "This slot is no longer available"	Same as expected output
45.	Empty Purpose Field	Valid teacher and slot, empty purpose field Action Items: "Provide	- Validation error "Purpose is required"	Same as expected output

		additional practice materials" Follow-up Required: true		

Table 6.11 Meeting Management Testing

6.1.10 Meeting Booking System

No.	Test Case	Input	Expected Output	Actual Result
46.	Valid Meeting Booking	- Teacher ID: 1 - Available slot ID: 5 - Purpose: "Discuss child's progress"	- Success message: "Booking successful, confirmation message with booking reference, email sent to teacher"	Same as expected output
47.	Duplicate Slot Booking	Attempt to book already booked slot	- Error message "This slot is no longer available"	Same as expected output
48.	Empty Purpose Field	Valid teacher and slot, empty purpose fieldAction Items: "Provide additional practice materials" Follow-up Required: true	- Validation error "Purpose is required"	Same as expected output

Table 6.12 Meeting Management Testing

6.2 Project Challenges

During the development of the Kindergarten Management System, there are several issues and challenges were encountered. In order to guarantee the system's effectiveness, usability, and scalability, these problems have to be resolved. There are 3 critical issues during the development of the system.

There were numerous obstacles in the way of KinderEase's development, particularly in managing **student enrollment and attendance**. The enrollment process required coordination across multiple tables, such as students, class enrollment, class schedules, subjects, and class levels. While attendance tracking required real-time accuracy and validation for multiple status kinds, both human and automatic enrollment included the danger of inconsistent data. Complexity was increased by maintaining data integrity, which included handling changes in enrollment status, eliminating duplicate enrollments, and guaranteeing proper class placement.

Another significant challenge was the **integration of multi-role dashboards** for Admin, Teacher, and Parent users. Each role demanded different layouts and functionalities, including system management for admins, class handling for teachers, and child-specific views for parents. Implementing secure role-based access control and maintaining a consistent user experience across dashboards proved difficult, especially with the need for responsive design and smooth JavaScript interactions.

The **database design and query performance** also introduced difficulties. With more than 20 interconnected tables, ensuring data consistency, handling cascading updates, and optimizing complex queries was essential. Real-time statistics, bulk operations like auto-enrollment, and academic year transitions required careful transaction management. Strong validation, error handling, and reliable backup processes were necessary to prevent data corruption and ensure system reliability.

Lastly, defining the **payment structure** was particularly challenging. The system needed to handle multiple fee types, partial payments, payment plans, and flexible invoice generation. Supporting various payment methods and integrating with external gateways required both technical and security considerations. Additionally, financial reporting, overdue tracking, and compliance with audit and tax regulations added to the complexity of implementation.

Overall, these challenges highlighted the need for careful planning, robust database design, and iterative development to ensure the system delivered reliable functionality while supporting the diverse needs of kindergarten management.

6.3 Objectives Evaluation

The project has successfully met the research objectives outlined in the early stages of development.

The first objective, which was to develop a communication platform between teachers and parents, was achieved through the implementation of real-time messaging, notifications, and appointment scheduling features. Teachers are able to share school announcements, timetables, and event reminders directly with parents via the system. Furthermore, the meeting arrangement function allows teachers to set available slots

and parents to request discussions regarding their child's progress, thus strengthening teacher-parent collaboration and engagement.

The second objective, focused on providing teachers with a user-friendly environment for lesson planning, has also been realized. The system supports the creation, editing, and saving of lesson plans within a streamlined interface. Teachers are able to manage their lesson content efficiently and attach supplementary materials, which simplifies the teaching preparation process. This functionality reduces paperwork and ensures that lesson content is well-structured, accessible, and reusable for future classes.

The third objective, to build a comprehensive kindergarten management system that integrates various operational processes, was effectively accomplished. The system provides a web-based interface accessible to administrators, teachers, and parents with role-based access control for security and usability. Administrative tasks such as enrollment, attendance tracking, billing, and reporting are managed within a centralized platform. Additionally, analytics and reporting tools were integrated, enabling administrators and parents to monitor student performance, attendance rates, and financial transactions. This demonstrates that the system is capable of supporting the daily operations of a kindergarten while also enhancing transparency and efficiency.

Overall, the project has achieved its intended objectives, delivering a robust and user-friendly kindergarten management system that addresses communication, lesson planning, and administrative needs in a single integrated solution.

CHAPTER 7 Conclusion and Recommendation

7.1 Conclusion

In conclusion, developing the Kindergarten Management System represents a significant step toward addressing the inefficiencies in traditional kindergarten operations by leveraging modern technology. The existing systems often lead to time-consuming administrative tasks, errors in data management, and poor communication between teachers and parents. These issues ultimately hindered the productivity and parental involvement in the children's education. The development of the Kindergarten Management System is motivated by the need for a more efficient and user-friendly solution and aims to create a comprehensive web-based system that automates the key administrative tasks, enhances teacher-parent communication, and simplifies lesson planning.

This proposed system stands out as a comprehensive and integrated platform that effectively addresses the diverse and complex operational needs of a kindergarten. Through its well-designed multi-role architecture, the system ensures that administrators, teachers, and parents can each access dedicated interfaces tailored to their unique responsibilities. Role-based access control safeguards sensitive information while maintaining a smooth and personalized user experience. This architecture not only streamlines day-to-day operations but also enhances transparency and communication among stakeholders, thereby fostering a more collaborative learning environment.

In sum, the Kindergarten Management System provides an efficient, user-friendly solution that addresses the key challenges of traditional kindergarten operations. By automating administrative tasks, strengthening communication between teachers and parents, and supporting lesson planning, the system enhances both productivity and collaboration. With its role-based design and integrated features, it serves as a comprehensive platform that improves transparency, efficiency, and the overall management of kindergarten activities.

7.2 Recommendation

One key recommendation for future development is the creation of a **mobile application** to complement the web system. While the current system is fully functional on desktop and responsive on mobile browsers, a dedicated mobile app for parents and teachers would greatly improve accessibility and convenience. Features such as instant push notifications for attendance updates, fee reminders, and appointment confirmations would keep users engaged and informed in real time. Additionally, offline capabilities could be introduced to allow teachers to record attendance or notes without constant internet access, syncing automatically when a connection is restored.

Another important enhancement involves the implementation of **automated reminders and notifications**. Currently, parents must log in to the portal to check updates such as invoices, appointments, or class announcements. By integrating SMS, email, or app-based notifications, the system could proactively alert parents about overdue payments, upcoming parent-teacher meetings, or special school events. This would not only improve communication between the kindergarten and parents but also reduce the administrative burden on staff who currently rely on manual reminders.

In terms of security and compliance, strengthening **data protection measures** should be prioritized. Although the system currently uses role-based access control, further enhancements such as end-to-end encryption, multi-factor authentication, and regular data backups would significantly improve the security posture. At the same time, compliance with regional data protection laws, such as the Personal Data Protection Act (PDPA) in Malaysia, would ensure that sensitive information about children, parents, and financial transactions is handled responsibly. These improvements would build trust among stakeholders and safeguard the institution against potential data breaches.

Looking ahead, the integration of **artificial intelligence (AI) features** could position the system as a forward-looking educational management platform. AI can be applied to automate complex tasks such as generating class timetables, predicting fee collection trends, and analyzing student attendance patterns to identify early signs of absenteeism. Additionally, an AI-powered chatbot could be integrated into the parent portal or mobile app to answer frequently asked questions, such as fee structures or

upcoming events, reducing the workload for administrative staff. These innovations would enhance efficiency, personalize user experience, and provide actionable insights for decision-making.

REFERENCES

[1] "Preschool Education a Case Study of Students' Professional Identity Enhancement," *Adult and Higher Education*, vol. 4, no. 10, 2022, doi: https://doi.org/10.23977/aduhe.2022.041016.

[2] J. Denisyak, "PRE-SCHOOL EDUCATION ESTABLISHMENT AS A SYSTEM INSTITUTION FOR THE FORMATION OF HEALTH CARE COMPETENCES OF A CHILD," *Mountain School of Ukrainian Carpaty*, no. 26, pp. 12–15, Apr. 2022, doi: https://doi.org/10.15330/msuc.2022.26.12-15.

[3] Clipa and P. Gavriluță, "Early Childhood Education – Parents' Perceptions," *The European Proceedings of Social and Behavioural Sciences*, May 2017, doi: https://doi.org/10.15405/epsbs.2017.05.02.210.

[4] G. Liu, Z. Jin, X. Zheng, Z. Wang, and W. Liu, "Associations between teacher–parent relationships and preschool children's social behavior problems—the chain mediating roles of work–family conflict and parenting self-efficacy," *Frontiers in Psychology*, vol. 15, Jul. 2024, doi: https://doi.org/10.3389/fpsyg.2024.1349652.

[5] T. M. Paes, R. Duncan, D. J. Purpura, and S. A. Schmitt, "The relations between teacher-child relationships in preschool and children's outcomes in kindergarten," *Journal of Applied Developmental Psychology*, vol. 86, p. 101534, May 2023, doi: https://doi.org/10.1016/j.appdev.2023.101534.

[6] Leshtarova, "SOCIAL INTERACTIONS WITH FAMILIES.," *Ebsco.com*, 2024. https://openurl.ebsco.com/EPDB%3Agcd%3A10%3A16579178/detailv2?sid=ebsco%3Aplink%3Ascholar&id=ebsco%3Agcd%3A127682741&crl=c&link_origin=scholar.google.com.my (accessed Nov. 20, 2024).

[7] X. Ke and J. Wu, "Design and implementation of kindergarten information management system under the background of Internet," *MATEC Web of Conferences*, vol. 365, p. 01046, 2022, doi: https://doi.org/10.1051/matecconf/202236501046.

[8] A. Papakonstantinou, "Teachers' Perceptions Regarding School Parents' Online Groups," *Education Sciences*, vol. 13, no. 1, p. 60, Jan. 2023, doi: https://doi.org/10.3390/educsci13010060.

[9] M. N. Velasco *et al.*, "Enhancing Parent-Teacher Collaboration in Early Childhood Education through a Web-Based App," *Enhancing Parent-Teacher Collaboration in*

Early Childhood Education through a Web-Based App, pp. 131–136, Jul. 2024, doi: https://doi.org/10.1109/icicos62600.2024.10636878.

[10] Muratović, A. Muratović, and E. Dacić, "Modern approach to management as a factor of effective functioning of kindergarten," *Ekonomski izazovi*, vol. 12, no. 23, pp. 57–67, 2023, doi: https://doi.org/10.5937/ekoizazov2323057m.

[11] Dionísia Laranjeiro, Maria João Antunes, and P. Santos, "Using Design-Based Research for a Technological Intervention to Promote Parental Involvement in Kindergarten," *Using Design-Based Research for a Technological Intervention to Promote Parental Involvement in Kindergarten*, vol. 4, no. 3, Mar. 2023, doi: https://doi.org/10.1007/s42979-023-01666-8.

[12] "Anak2U | About The Kindergarten & Nursery Management System 👶 📚 🤗," *Anak2U*, Feb. 20, 2024. https://anak2u.com.my/about/ (accessed Nov. 20, 2024).

[13] L. Inc, "Make an impact on every aspect of school.," *Make an impact on every aspect of school.* https://www.littlelives.com/

[14] "Childcare billing software to make accounting easy | Illumine," *Illumine childcare software*. https://illumine.app/childcare-billing-software/

[15] Angaiz, D., Jabeen, Z., & Karim, N. (2020). Perceptions of experienced and novice teachers about the importance of lesson planning for effective learning in private elementary schools of gilgit city. *Pakistan Journal of Social Research*, 3(3), 268-275. https://doi.org/10.52567/pjsr.v3i3.268

[16] Li, Y. (2024). Mastering the Art of Lesson Planning: A Practitioner Guide for Teachers (Backed by Scientific Evidence). https://doi.org/10.46679/9788196780579

[17] Y. Wang and Y. Hu, "Intelligent Kindergarten Management System Based on Web Server and Android Platform," *Lecture Notes in Electrical Engineering*, pp. 923–928, Oct. 2022, doi: https://doi.org/10.1007/978-981-19-4132-0_122.

[18] Teuku Djauhari and Baso Irfandi, "SISTEM INFORMASI PENGOLAHAN DATA SISWA BERBASIS WEB PADA RA AN-NASYWA (STUDI KASUS PENGOLAHAN DATA SISWA)," *JURNAL AKADEMIKA*, vol. 14, no. 2, pp. 21–26, Apr. 2022, doi: https://doi.org/10.53564/akademika.v14i2.855.

[19] "Comprehensive School Management Solutions in Singapore | Taidii," *Taidii.com*, 2024. https://www.taidii.com/en/studentCare (accessed Nov. 20, 2024).

[20] Adib Rifqi Setiawan and Umi Nur Avivah, “Importance of Early Childhood Education,” *Thesis Commons (OSF Preprints)*, Jun. 2022, doi: https://doi.org/10.31237/osf.io/9324s.

[21] E. Major, “Parent-Teacher Communication from the Perspective of the Educator,” *Central European journal of educational research*, vol. 5, no. 2, Dec. 2023, doi: https://doi.org/10.37441/cejer/2023/5/2/13281.

[22] Sufyan bin Uzayr, *PHP*. 2022. doi: https://doi.org/10.1201/9781003308669.

[23] M. I. N. Saroni and B. Mulyanti, “Hypertext preprocessor framework in the development of web applications,” *IOP Conference Series: Materials Science and Engineering*, vol. 830, p. 022096, May 2020, doi: https://doi.org/10.1088/1757-899x/830/2/022096.

[24] H. Shah, “Harnessing customized built-in elements -- Empowering Component-Based Software Engineering and Design Systems with HTML5 Web Components,” *arXiv.org*, Nov. 25, 2023. https://arxiv.org/abs/2311.16601 (accessed Nov. 21, 2024).

[25] K. J. Grant, “CSS in Depth, Second Edition,” *Google Books*, 2024. https://books.google.com/books?hl=en&lr=&id=j7wWEQAAQBAJ&oi=fnd&pg=PA1&dq=CSS+in+website+development&ots=jQI_pQTiht&sig=Bh1ECBQisYMtjE8dxeRv1vMGoE8 (accessed Dec. 01, 2024).

[26] Ranjan, A. Sinha, and R. Battewad, “JavaScript for Modern Web Development,” *Google Books*, 2020. https://books.google.com/books?hl=en&lr=&id=b2bdDwAAQBAJ&oi=fnd&pg=PT24&dq=CSS+provides+the+convenience+for+developers+to+create+and+apply+consistent+style+across+all+pages+of+the+system+to+ensure+a+uniform+experience+for+users&ots=6geC1DzrH1&sig=GlHKzGrMC39uIvW39RgMXBengNg (accessed Dec 01, 2024).

[27] S. Botros and J. Tinley, “High Performance MySQL,” *Google Books*, 2021. https://books.google.com.my/books?hl=en&lr=&id=G3ZPEAAAQBAJ&oi=fnd&pg=PR4&dq=MySQL+provides+a+structured+and+scalable+solution+to+handle+this+data+securely+and+reliably&ots=Ei3NvqOF7Q&sig=zy-nes-XeDxDs-Hn_58KdyDYHZo&redir_esc=y#v=onepage&q&f=false (accessed Nov. 21, 2024).

[28] K. Lamsal, “Designing and Developing a dynamic website using PHP,” 2020. Available: https://www.theseus.fi/bitstream/handle/10024/346960/Kamal_lamsal.pdf

[29] M. A. Hossain, “Software Development Life Cycle (SDLC) Methodologies for Information Systems Project Management,” *International Journal For Multidisciplinary Research*, vol. 5, no. 5, Sep. 2023, doi: https://doi.org/10.36948/ijfmr.2023.v05i05.6223.

[30] B. Rogers, “Agile XP, Stress, Trust and Empowerment in Distributed Workforce Environments - ProQuest,” *Proquest.com*, 2023. https://search.proquest.com/openview/a432c72a0222678c2b0b5903334c3618/1?pq-origsite=gscholar&cbl=18750&diss=y

[31] C. Greg, “Agile Project Management,” *Google Books*, 2021. https://books.google.com/books?hl=en&lr=&id=SZoXEAAAQBAJ&oi=fnd&pg=PT4&dq=XP%27s+ability+to+make+fast+adjustments+makes+it+perfect+for+projects+whose+requirements+change+frequently&ots=N82tcrhzFv&sig=c6PE4MQTElY7gpo6YMYa_5TbpKM (accessed Dec. 02, 2024).

APPENDIX

A.1 Poster