

**VIETNAM GENERAL CONFEDERATION OF LABOR  
TON DUC THANG UNIVERSITY  
FALCUTY OF INFORMATION TECHNOLOGY**



## **Final Project**

**Subject: Service-Oriented Architecture**

**Topic: Primary School Classroom Management Module**

*Students perform:*

Tran Khai Hoang (520H0635)  
Nguyen Nhat Thong (52000808)  
Nguyen Gia Khiem (520H0464)

*Advising teacher:*

Mr. Duong Huu Phuc

April 27, 2023

# Contents

<b>1</b>	<b>System Overview</b>	<b>3</b>
1.1	Introduction . . . . .	3
1.2	System Specification . . . . .	3
1.3	Topic Scope . . . . .	3
1.3.1	Objects and Functions Boundaries . . . . .	3
1.3.2	Technologies Boundaries . . . . .	4
1.4	Practical Implications . . . . .	4
1.5	Thesis Layout . . . . .	4
<b>2</b>	<b>System Analysis and Design</b>	<b>5</b>
2.1	Functional Requirement . . . . .	5
2.1.1	Check Attendance . . . . .	5
2.1.2	Modify Teacher's Information . . . . .	5
2.1.3	View Student Information . . . . .	5
2.1.4	View List Student . . . . .	5
2.1.5	Send an Announcement via Email . . . . .	5
2.1.6	View Score Sheet . . . . .	5
2.1.7	View Schedule . . . . .	5
2.1.8	View Student Statistics . . . . .	5
2.1.9	Modify Score Sheet . . . . .	5
2.1.10	Verify Score . . . . .	5
2.1.11	Create Class . . . . .	6
2.1.12	Modify Schedule . . . . .	6
2.1.13	View Teacher's Information . . . . .	6
2.1.14	Modify Student Information . . . . .	6
2.2	Non-functional Requirement . . . . .	6
2.3	Use Case Diagram . . . . .	7
2.4	Use Case Specification . . . . .	8
2.4.1	Use Case Login . . . . .	8
2.4.2	Use Case Logout . . . . .	9
2.4.3	Use Case Check Attendance . . . . .	9
2.4.4	Use Case Modify Teacher's Information . . . . .	10
2.4.5	Use Case View Student's Information . . . . .	11
2.4.6	Use Case View List Students . . . . .	11
2.4.7	Use Case Send Announcement via Email . . . . .	12
2.4.8	Use Case View Score Sheet . . . . .	13
2.4.9	Use Case View Schedule . . . . .	14
2.4.10	Use Case View Statistics . . . . .	15
2.4.11	Use Case Modify Score Sheet . . . . .	16
2.4.12	Use Case Verify Score . . . . .	17
2.4.13	Use Case Create Class . . . . .	18
2.4.14	Use Case Modify Schedule . . . . .	19
2.4.15	Use Case View Teacher Information . . . . .	19
2.4.16	Use Case Modify Student Information . . . . .	20

2.5	Entity-Relationship Diagram . . . . .	21
2.6	Physical Database Model . . . . .	22
<b>3</b>	<b>System Implementation</b>	<b>23</b>
3.1	Technology Used and The Reasons for Use . . . . .	23
3.1.1	Front-end - HTML,CSS,Javascript . . . . .	23
3.1.2	Back-end - ExpressJS . . . . .	23
3.1.3	Database - MongoDB . . . . .	23
3.2	Project APIs . . . . .	24
3.2.1	Announcement APIs . . . . .	24
3.2.2	Attendance APIs . . . . .	26
3.2.3	Class APIs . . . . .	30
3.2.4	Parent APIs . . . . .	33
3.2.5	Subject APIs . . . . .	37
3.2.6	School Staff APIs . . . . .	39
3.2.7	Student APIs . . . . .	42
3.2.8	User APIs . . . . .	44
3.3	APIs documentation using Postman . . . . .	45
<b>4</b>	<b>System Demo</b>	<b>46</b>
4.1	System Demo script . . . . .	46
4.1.1	Principal Flow . . . . .	46
4.1.2	Teacher Flow . . . . .	46
4.1.3	Parent & Student Flow . . . . .	46
4.2	System Demo . . . . .	46
	<b>References</b>	<b>57</b>
	<b>Appendix A</b>	<b>58</b>
4.3	Interview Questions . . . . .	58
4.4	Questionnaire . . . . .	58

# 1 System Overview

## 1.1 Introduction

This report will present an overview of the general architecture for some functions of the **Primary School Classroom Management Module**, including interview questions, requirements collection questionnaire, functional and non-functional requirements described by the use case diagram and specification. The thesis will also mention the technology selected to implement the project and the reasons for the choice. In addition, the thesis also includes database design using Entity Relationship Diagram and Physical Database Model.

## 1.2 System Specification

An primary school has a need to develop a **Primary School Classroom Management Module** on a web app to improve the efficiency of student and classroom data management. In addition, the web app also supports parents to know the schedule, student's learning progress, class and school announcements. At the request of the school, the system will have 4 main types of users with separate functions including: parent, student, teacher and principal.

In this module, **students** can view schedule, score sheet of semesters in different school years, class and school announcement.

Meanwhile, **parents** can not only perform all the functions that students can perform, but also can change the student's personal information, view the student's attendance history, and view the teacher's information.

For **teachers**, Teachers can check attendance for students, view managed student lists and their information. Teachers can also view schedule and edit student score sheet. In addition, teachers can send class announcements to parents via email, view managed class statistics and modify teacher's information.

For **principal**, The principal can not only perform all the functions of the teacher, but also create and manage the class, the timetable of each class and verify the student's grades scored by the teacher.

## 1.3 Topic Scope

This project is a **Primary School Classroom Management Module**, only ensuring some basic functions just enough for teachers and parents to interact and understand the learning process of students. It also helps the principal to observe the students learning progress as well as manage the class and schedule of each class.

### 1.3.1 Objects and Functions Boundaries

- Objects Boundary: Parents, Students, Teachers, Principal

- Functions Boundary:

- Parents: view student's attendance, view teacher's information, modify student's information, view announcement, view score sheet, view schedule
- Students: view announcement, view score sheet, view schedule

- Teachers: check attendance, modify teacher's information, view student's information, view list students, make announcement, view score sheet, view schedule, view student statistics, modify score sheet.
- Principal: view attendances, view teacher's information, view student's information, view list students, make announcement, view score sheet, view & modify schedule, view student statistics, modify & verify score sheet, create class.

### 1.3.2 Technologies Boundaries

- Main technology used:

- Front-end: HTML, CSS, JavaScript, Bootstrap
- Back-end: ExpressJS
- Database: MongoDB

## 1.4 Practical Implications

- Supporting individuals to use the system in managing and monitoring the learning of elementary students
- Store the score sheet, the announcement for students.
- Support to manage class information and class timetables.

## 1.5 Thesis Layout

- Thesis consists of 4 parts:

- Part 1 - System Overview: Introduction to the goal, scope and practical significance of the topic.
- Part 2 - System Analysis and Design: Presents functional requirements, non-functional requirements, Use Case diagram, Use Case specification, Entity-relationship diagram, Physical Database Model
- Part 3 - System Implementation: Present the project APIs, the APIs Documentation, the expected technology to be used to deploy the module and the reasons for its choice.
- Part 4 - System Demo: Presents system demo script and system demo
- Appendix A - Question List Collect Request: Presents interview questions and questionnaire

## **2 System Analysis and Design**

### **2.1 Functional Requirement**

#### **2.1.1 Check Attendance**

- The system allows teachers to take attendance of students.
- The system allows principals, parents and students to view attendance history

#### **2.1.2 Modify Teacher's Information**

- The system allows teachers to change personal information.
- The system allows the principal to view the teacher's information

#### **2.1.3 View Student Information**

- The system allows teachers and principals to view student information

#### **2.1.4 View List Student**

- The system allows teachers to see the list of students in the class the teacher is teaching
- The system allows the principal to view a list of all students in the school or view a list by grade of school years

#### **2.1.5 Send an Announcement via Email**

- The system allows teachers and principals to send notifications to parents via email and on the system.
- The system allows parents and students to view announcements sent by teachers or principals.

#### **2.1.6 View Score Sheet**

- The system allows principals, teachers, parents and students to view students transcripts

#### **2.1.7 View Schedule**

- The system allows principals, teachers, parents and students to view the classes timetable

#### **2.1.8 View Student Statistics**

- The system allows principals and teachers to view class or student statistics - The system allows principals to view grade statistics

#### **2.1.9 Modify Score Sheet**

- The system allows teachers and principals to add and edit transcripts

#### **2.1.10 Verify Score**

- The system allows the principal to verify the student's grades are graded by the teacher.

#### **2.1.11 Create Class**

- The system allows the principal to create classes and add teachers and students to the class

#### **2.1.12 Modify Schedule**

- The system allows principals to add or adjust schedules by class, by semester, and by school year

#### **2.1.13 View Teacher's Information**

- The system allows parents to view teacher information

#### **2.1.14 Modify Student Information**

- The system allows parents to adjust students' personal information

### **2.2 Non-functional Requirement**

- Operational: The system can work on any browser and on most devices with Internet
- Performance:
  - System has a fast response time
  - The system is open 24/7
- Security:
  - No user can log in to another account without a password
  - To use the system, users must have an account and password provided by the system manager
- Cultural and Political:
  - The module fully complies with the legal provisions on network information security of Vietnam
  - The module complies with the general education standards of the Vietnamese Ministry of Education

## 2.3 Use Case Diagram



Figure 1: Use Case Diagram of Primary School Classroom Management Module



## 2.4 Use Case Specification

### 2.4.1 Use Case Login

<b>Use case ID</b>	UC001
<b>Use case Name</b>	Login
<b>Primary Actor</b>	Teacher, Student, Parent, System Manager, Principal
<b>Stakeholder and Interests</b>	None
<b>Summary</b>	The function helps users log into the system before using other functions
<b>Pre-Condition</b>	User must have an account in the database
<b>Trigger</b>	User click the Login button on the system interface
<b>Post-condition</b>	User successfully logged into the system
<b>Relationships</b>	Include: - All use cases in the diagram → Login
<b>Main scenario</b>	<ol style="list-style-type: none"><li>1. User want to access the system to use the functions for their work through the link.</li><li>2. The browser sends a request to the system. The system displays the login interface for User.</li><li>3. User fill in all the information that the system requires and click the Login button.</li><li>4. The system checks that the information is sufficiently entered.</li><li>5. The system checks that the information entered by staffs is correct.</li><li>6. The system switches the interface to the system interface corresponding to the logged in user</li></ol>
<b>Alternatives</b>	<ol style="list-style-type: none"><li>4.a. If the system checks the information entered is missing. The system will send a notification of additional missing information</li><li>5.a. If the system checks the information entered is incorrect. The system will send a notification to the user to check the entered information.</li></ol>

Table 1: Login use case specification

### 2.4.2 Use Case Logout

<b>Use case ID</b>	UC002
<b>Use case Name</b>	Logout
<b>Primary Actor</b>	Teacher, Student, Parent, System Manager, Principal
<b>Stakeholder and Interests</b>	None
<b>Summary</b>	The function helps Users log out of the system
<b>Pre-Condition</b>	User successfully logged in system
<b>Trigger</b>	Users click the Logout button on the system interface
<b>Post-condition</b>	Users successfully logged out of the system
<b>Relationships</b>	Include: Logout → Login
<b>Main scenario</b>	<ol style="list-style-type: none"> <li>1. After the session ends, the employee wants to log out of the system. Users presses the Logout button.</li> <li>2. The system records and logs out the User's account. The system finishes session of user in browser and switches the interface to the login interface</li> </ol>
<b>Alternatives</b>	None

Table 2: Logout use case specification

### 2.4.3 Use Case Check Attendance

<b>Use case ID</b>	UC003
<b>Use case Name</b>	Check Attendance
<b>Primary Actor</b>	Teacher
<b>Offstage Actor</b>	Student, Parent, Principal
<b>Summary</b>	Function to help teachers take attendance and allow students, parents and principals to view attendance history
<b>Pre-Condition</b>	<ul style="list-style-type: none"> <li>- Teachers, Student, Parent, Principal must log in to the system</li> <li>- Principals must create successful classrooms</li> </ul>
<b>Trigger</b>	The teacher clicks the attendance button
<b>Post-condition</b>	Teachers successfully take attendance for students
<b>Relationships</b>	Include: - Check Attendance → Login
<b>Main scenario</b>	<ol style="list-style-type: none"> <li>1. After the teacher successfully logged into the system and pressed the attendance button.</li> <li>2. The system displays a list of students in the class for the teacher to take attendance.</li> <li>3. The teacher checks the student and presses the corresponding attendance button of the student in the list.</li> <li>4. After taking attendance, the teacher presses the submit button</li> <li>5. The system records attendance information from the list and updates it to the system</li> </ol>
<b>Alternatives</b>	None

Table 3: Check Attendance use case specification

#### 2.4.4 Use Case Modify Teacher's Information

<b>Use case ID</b>	UC004
<b>Use case Name</b>	Modify Teacher Information
<b>Primary Actor</b>	Teacher
<b>Offstage Actor</b>	Principal
<b>Summary</b>	Function to help teachers edit personal information and help principals see that information
<b>Pre-Condition</b>	Teachers, Principal must log in to the system
<b>Trigger</b>	The teacher clicks the modify button
<b>Post-condition</b>	Teachers successfully modify teacher's information
<b>Relationships</b>	Include: - Modify Teacher Information → Login
<b>Main scenario</b>	<ol style="list-style-type: none"> <li>1. After the teacher successfully logged into the system and pressed the modify button.</li> <li>2. The system displays the teacher's information boxes for the teacher to enter new information</li> <li>3. The teacher fills in the information boxes that need to be changed</li> <li>4. The system checks that the information fields filled in by the teacher are correct</li> <li>5. The system updates that information to the system</li> </ol>
<b>Alternatives</b>	4.a. The system checks the information fields filled in by the teacher that do not meet the requirements of the system and sends a notice requesting the teacher to correct.

Table 4: Modify Teacher Information use case specification

#### 2.4.5 Use Case View Student's Information

<b>Use case ID</b>	UC005
<b>Use case Name</b>	View Student Information
<b>Primary Actor</b>	Teacher
<b>Stakeholder and Interests</b>	Principal
<b>Summary</b>	Function to help teachers and principals view students information
<b>Pre-Condition</b>	- Teachers, Principals must log in to the system - Principals must create successful classrooms
<b>Trigger</b>	The teacher clicks a student from the list of student
<b>Post-condition</b>	Teachers and Principal successfully view the list of student
<b>Relationships</b>	Include: - View Student Information → Login
<b>Main scenario</b>	1. After the teacher successfully logged into the system and pressed the list student button. 2. The system displays the list of students in the class managed by the teacher 3. The teacher clicks on a student from the list 4. The system displays the student's information
<b>Alternatives</b>	None

Table 5: View Student Information use case specification

#### 2.4.6 Use Case View List Students

<b>Use case ID</b>	UC006
<b>Use case Name</b>	View List Student
<b>Primary Actor</b>	Teacher
<b>Stakeholder and Interests</b>	Principal
<b>Summary</b>	Function to help teachers and principals view list of students
<b>Pre-Condition</b>	- Teachers, Principals must log in to the system - Principals must create successful classrooms
<b>Trigger</b>	The teacher clicks the list student button
<b>Post-condition</b>	Teachers and Principal successfully view the list of student
<b>Relationships</b>	Include: - View List Student → Login
<b>Main scenario</b>	1. After the teacher successfully logged into the system and pressed the list student button. 2. The system displays a list of all students for the principal and a list of students by class for the teacher
<b>Alternatives</b>	None

Table 6: View List Student use case specification

#### 2.4.7 Use Case Send Announcement via Email

<b>Use case ID</b>	UC007
<b>Use case Name</b>	Send Announcement via Email
<b>Primary Actor</b>	Teacher, Principal
<b>Offstage Actor</b>	Parent, Student
<b>Summary</b>	Function to help teachers and principals send announcement via email and on system. It also help parent and student view the announcement
<b>Pre-Condition</b>	- Teachers, Principals, Parent, Student must log in to the system - Principals must create successful classrooms
<b>Trigger</b>	The teacher and principal clicks the compose button
<b>Post-condition</b>	Teachers and Principal successfully send announcement to parents. Parent and Student successfully view the announcement
<b>Relationships</b>	Include: - Send Announcement via Email → Login
<b>Main scenario</b>	1. After the teacher successfully logged into the system and pressed the compose button. 2. The system displays information boxes to help teachers and principals compose announcements 3. Teachers and principals enter information into the information boxes provided by the system 4. The system checks that the information entered by the principal and the teacher is correct 5. The system records to the system and sends notifications to parents and students via Email and on system
<b>Alternatives</b>	4.a. If the system checks the information fields filled in by the teacher that do not meet the requirements of the system and sends a notice requesting the teacher to correct.

Table 7: Send Announcement via Email use case specification

#### 2.4.8 Use Case View Score Sheet

<b>Use case ID</b>	UC008
<b>Use case Name</b>	View Score Sheet
<b>Primary Actor</b>	Teacher, Principal, Parent, Student
<b>Stakeholder and Interests</b>	None
<b>Summary</b>	Function to help Teacher, Principal, Parent and Student view student's score sheet
<b>Pre-Condition</b>	- Teachers, Principals, Parent, Student must log in to the system - Principals must create successful classrooms
<b>Trigger</b>	- The teacher and principal clicks a student from the list of student - Student and Parent click the score sheet button
<b>Post-condition</b>	Teachers, Principals, Parent, Student successfully view the student's Score sheet
<b>Relationships</b>	Include: - View Score Sheet → Login
<b>Main scenario</b>	1.1. After the Teachers, Principals successfully logged into the system. The teacher and principal pressed the list student button. 1.2. The system displays the list of students in the class managed by the teacher 1.3. The teacher and principal clicks on a student from the list 1.4. The system displays the student's score sheet 2.1. After the Parent and Student successfully logged into the system. The Parent and Student pressed the score sheet button. 2.2. The system displays the student's score sheet
<b>Alternatives</b>	None

Table 8: View Score Sheet use case specification

#### 2.4.9 Use Case View Schedule

<b>Use case ID</b>	UC009
<b>Use case Name</b>	View Schedule
<b>Primary Actor</b>	Teacher, Parent, Student
<b>Stakeholder and Interests</b>	Principal
<b>Summary</b>	Function to help Teacher, Principal, Parent and Student view schedule
<b>Pre-Condition</b>	<ul style="list-style-type: none"> <li>- Teachers, Principals, Parent, Student must log in to the system</li> <li>- Principals must create successfully classrooms</li> <li>- Principals must create successfully schedule</li> </ul>
<b>Trigger</b>	<ul style="list-style-type: none"> <li>- The principal clicks a class from the list of classes</li> <li>- Teacher, Student and Parent click the schedule button</li> </ul>
<b>Post-condition</b>	Teachers, Principals, Parent, Student successfully view the schedule
<b>Relationships</b>	Include: - View Schedule → Login
<b>Main scenario</b>	<p>1.1. After the principal successfully logged into the system.The principal pressed the list classes button.</p> <p>1.2. The system displays the list of classes</p> <p>1.3. The principal clicks on a class from the list</p> <p>1.4. The system displays the class's schedule</p> <p>2.1. After the Parent, Student and Teacher successfully logged into the system.The Parent, Student and Teacher pressed the schedule button.</p> <p>2.2. The system displays the class's schedule</p>
<b>Alternatives</b>	None

Table 9: View Schedule use case specification

#### 2.4.10 Use Case View Statistics

<b>Use case ID</b>	UC010
<b>Use case Name</b>	View Statistic
<b>Primary Actor</b>	Teacher
<b>Stakeholder and Interests</b>	Principal
<b>Summary</b>	Function to help Teacher, Principal view the statistics
<b>Pre-Condition</b>	- Teachers, Principals must log in to the system - Principals must create successful classrooms
<b>Trigger</b>	- The principal and teacher clicks the statistic button
<b>Post-condition</b>	Teachers, Principals successfully view the statistics
<b>Relationships</b>	Include: - View Student Statistic → Login
<b>Main scenario</b>	1. After the Teachers, Principals successfully logged into the system. They pressed the statistic button. 2. The system displays the class's statistic for teacher and grade's statistic for principal
<b>Alternatives</b>	None

Table 10: View Statistic use case specification



#### 2.4.11 Use Case Modify Score Sheet

<b>Use case ID</b>	UC011
<b>Use case Name</b>	Modify Score Sheet
<b>Primary Actor</b>	Teacher
<b>Stakeholder and Interests</b>	Principal
<b>Summary</b>	Function to help Teacher, Principal Modify Score Sheet
<b>Pre-Condition</b>	- Teachers, Principals must log in to the system - Principals must create successful classrooms
<b>Trigger</b>	- The principal and teacher clicks the modify score sheet button
<b>Post-condition</b>	Teachers, Principals successfully modify the score sheet
<b>Relationships</b>	Include: - Modify Score Sheet → Login - Modify Score Sheet → Verify Score Extend: - Enter Score → Modify Score Sheet - Edit Score → Modify Score Sheet
<b>Main scenario</b>	1. After the Teachers, Principals successfully logged into the system. The teacher pressed the modify score sheet button. 2. The system displays the list of students in the class managed by the teacher 3. The teacher clicks on a student from the list 4. The system displays the student's score sheet 5. Teacher enter the new score or edit the old score and click the submit button 6. The system updates that score to the system and sends it to the principal for verification
<b>Alternatives</b>	None

Table 11: Modify Score Sheet use case specification

#### 2.4.12 Use Case Verify Score

<b>Use case ID</b>	UC012
<b>Use case Name</b>	Verify Score
<b>Primary Actor</b>	Principal
<b>Stakeholder and Interests</b>	None
<b>Summary</b>	Function to help Principal Verify Score
<b>Pre-Condition</b>	- Principals must log in to the system
<b>Trigger</b>	- The principal clicks the verify score button
<b>Post-condition</b>	Teachers, Principals successfully verify the score sheet
<b>Relationships</b>	Include:- Verify Score → Login
<b>Main scenario</b>	<ol style="list-style-type: none"><li>1. After the Principals successfully logged into the system.The teacher pressed the verify button.</li><li>2. The system displays the list of students</li><li>3. The principal clicks on a student from the list</li><li>4. The system displays the student's score sheet</li><li>5. The principal checks and accepts the grade given by the teacher. Principal clicks submit button</li><li>6. system to record and update the status of the scoreboard</li></ol>
<b>Alternatives</b>	<ol style="list-style-type: none"><li>5.a. Principal checks and does not accept grades graded by teachers. Principal presses deny button</li><li>5.b. system to record and update the status of the scoreboard</li></ol>

Table 12: Verify Score use case specification

### 2.4.13 Use Case Create Class

<b>Use case ID</b>	UC013
<b>Use case Name</b>	Create Class
<b>Primary Actor</b>	Principal
<b>Stakeholder and Interests</b>	None
<b>Summary</b>	Function to help Principal Create Class
<b>Pre-Condition</b>	- Principals must log in to the system
<b>Trigger</b>	- The principal clicks the create class button
<b>Post-condition</b>	Principals successfully create a new class
<b>Relationships</b>	Include: - Create Class → Login - Create Class → Appointment of teachers - Create Class → Add Student
<b>Main scenario</b>	<ol style="list-style-type: none"> <li>1. After the Principals successfully logged into the system. The teacher pressed the create class button.</li> <li>2. The system displays the information boxes of the class and the list of teachers and students in the school</li> <li>3. The principal enters the class information and adds teachers and students to the classroom. Principal presses the create button</li> <li>4. The system checks that the principal enters all the information that the system requires. the system instantiates a new class on the system</li> <li>5. The system sends a successful initialization message</li> </ol>
<b>Alternatives</b>	4.a. If the system checks that the principal enters missing or incorrect information that the system requires. The system sends a notification to the principal to adjust.

Table 13: Create Class use case specification

#### 2.4.14 Use Case Modify Schedule

<b>Use case ID</b>	UC014
<b>Use case Name</b>	Modify Schedule
<b>Primary Actor</b>	Principal
<b>Stakeholder and Interests</b>	None
<b>Summary</b>	Function to help Principal Modify Schedule
<b>Pre-Condition</b>	- Principals must log in to the system
<b>Trigger</b>	- The principal clicks the modify schedule button
<b>Post-condition</b>	Principals successfully modify schedule
<b>Relationships</b>	Include:- Modify Schedule → Login
<b>Main scenario</b>	<ol style="list-style-type: none"> <li>1. After the Principals successfully logged into the system.The teacher pressed the modify schedule button.</li> <li>2. The system displays the list of classes</li> <li>3. The principal clicks on a class from the list</li> <li>4. The system displays the class's schedule</li> <li>5. The principal click the modify button</li> <li>6. The system displays the information boxes of the schedule of the class</li> <li>7. The principal enter the new schedule</li> <li>8. The system checks that the principal enters all the information that the system requires and update the information into the system</li> </ol>
<b>Alternatives</b>	8.a. If the system checks that the principal enters missing or incorrect information that the system requires. The system sends a notification to the principal to adjust.

Table 14: Modify Schedule use case specification

#### 2.4.15 Use Case View Teacher Information

<b>Use case ID</b>	UC015
<b>Use case Name</b>	View Teacher Information
<b>Primary Actor</b>	Parent
<b>Stakeholder and Interests</b>	Principal
<b>Summary</b>	Function to help parent and principals view teacher information
<b>Pre-Condition</b>	<ul style="list-style-type: none"> <li>- Parent, Principals must log in to the system</li> <li>- Principals must create successful classrooms</li> </ul>
<b>Trigger</b>	The teacher clicks the teacher information button
<b>Post-condition</b>	Parent and Principal successfully the teacher information
<b>Relationships</b>	Include: - View Teacher Information → Login
<b>Main scenario</b>	<ol style="list-style-type: none"> <li>1. After the teacher successfully logged into the system and pressed the teacher information button.</li> <li>2. The system displays the teacher's information</li> </ol>
<b>Alternatives</b>	None

Table 15: View Teacher Information use case specification

#### 2.4.16 Use Case Modify Student Information

<b>Use case ID</b>	UC016
<b>Use case Name</b>	Modify Student Information
<b>Primary Actor</b>	Parent
<b>Stakeholder and Interests</b>	Student
<b>Summary</b>	Function to help parent edit student's information and help student see that information
<b>Pre-Condition</b>	Parent, Student must log in to the system
<b>Trigger</b>	The parent clicks the modify button
<b>Post-condition</b>	Parent successfully modify student's information
<b>Relationships</b>	Include: - Modify Student Information → Login
<b>Main scenario</b>	<ol style="list-style-type: none"><li>1. After the parent successfully logged into the system and pressed the modify button.</li><li>2. The system displays the student's information boxes for the parent to enter new information</li><li>3. The parent fills in the information boxes that need to be changed</li><li>4. The system checks that the information fields filled in by the parent are correct</li><li>5. The system updates that information to the system</li></ol>
<b>Alternatives</b>	<ol style="list-style-type: none"><li>4.a. The system checks the information fields filled in by the parent that do not meet the requirements of the system and sends a notice requesting the teacher to correct.</li></ol>

Table 16: Modify Student Information use case specification

## 2.5 Entity-Relationship Diagram

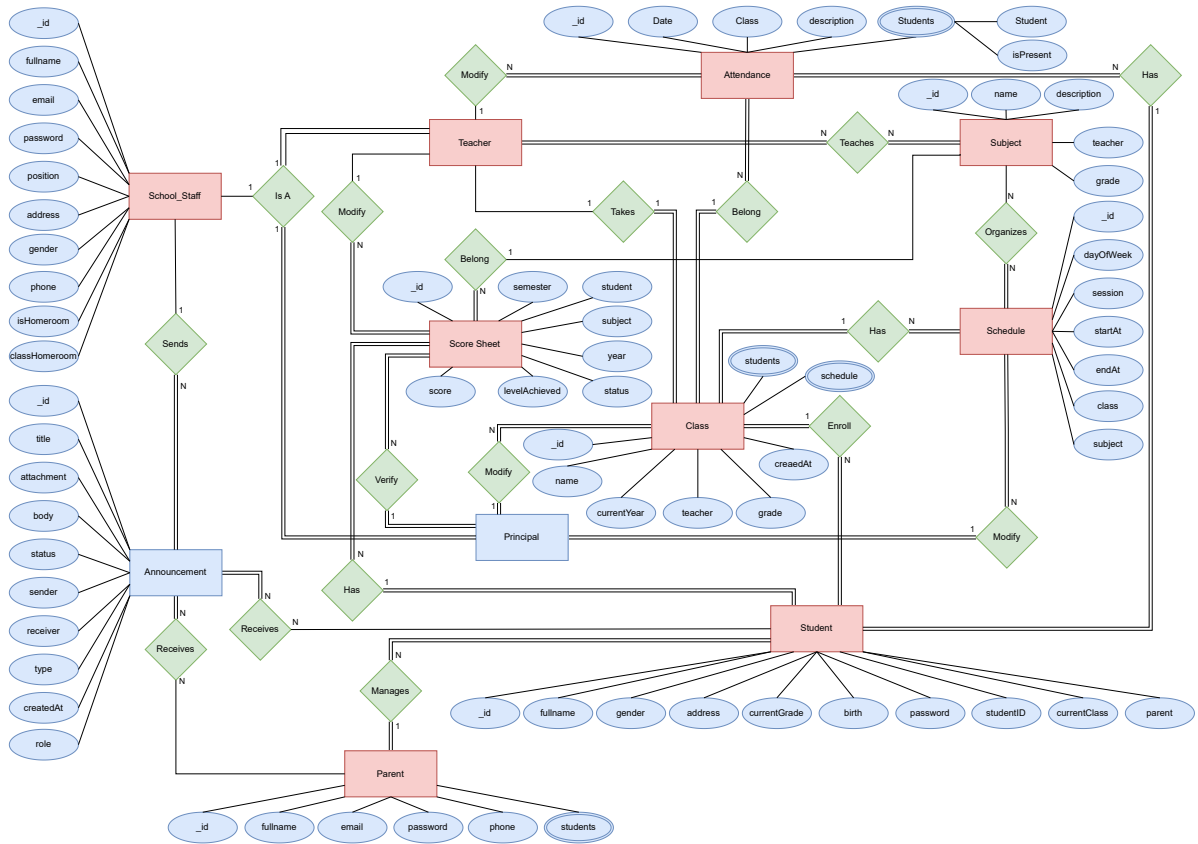


Figure 2: Entity-Relationship Diagram of Primary School Classroom Management Module

## 2.6 Physical Database Model

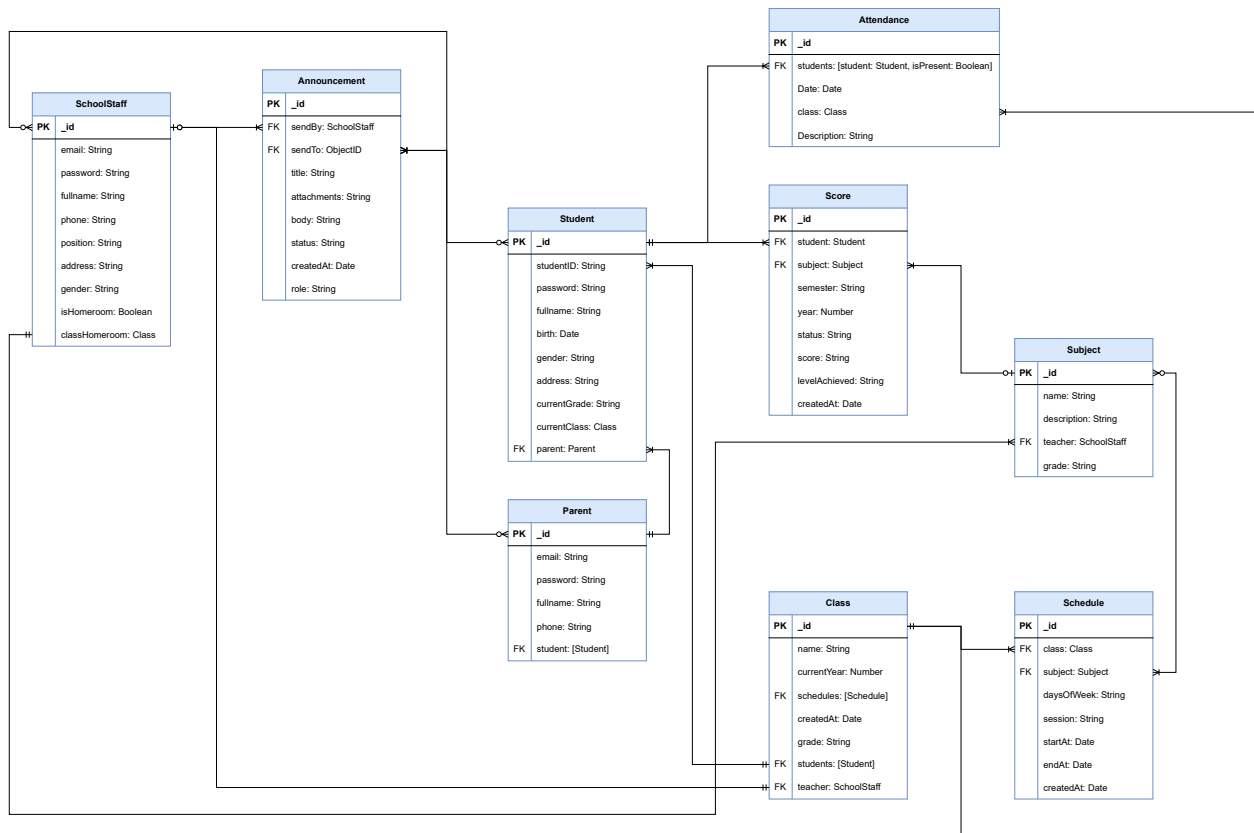


Figure 3: Physical Database Model of Primary School Classroom Management Module

## 3 System Implementation

### 3.1 Technology Used and The Reasons for Use

#### 3.1.1 Front-end - HTML,CSS,Javascript

- Basic easy to use with support for many libraries and frameworks

#### 3.1.2 Back-end - ExpressJS

- **Rapid server development:** Expressjs provides many features in the form of functions for easy use anywhere in the program. This eliminated the need to write code thereby saving time.
- **Routing:** Express js provides a routing mechanism that maintains the state of the website with the help of URLs.
- **Templating:** The templating tools provided by Express.js allow developers to build dynamic content on websites by building HTML templates on the server side.
- **Debugging:** For successful development of web applications, debugging is not necessary. Now with Expressjs debugging has become easier thanks to the ability to pinpoint the parts of the web application that have errors.
- **Middleware:** This is middleware that has access to databases, client requests, and other middleware. This middleware is primarily responsible for the systematic organization of Expressjs functionality.

#### 3.1.3 Database - MongoDB

- Document oriented
- High performance
- High Scalability – Sharding
- Flexible – add/remove fields with little or no impact on the application
- Data is not consistent
- No joins
- Distributable
- Representing data in JSON or BSON



## 3.2 Project APIs

### 3.2.1 Announcement APIs

```
1 // [GET] All Sent Announcements -> /api/ancm/list/sent
2 - Input:
3   Params: {null}
4   Body: {null}
5 - Output: [
6   {
7     "_id": "644620cd6797253c70c801aa",
8     "sendBy": "6444fb7fe14137d2c1586b02",
9     "sendTo": [
10      "6445427c934b44c3693c49da",
11      "6445602896858bdf4d8313db"
12    ],
13     "type": "announcement",
14     "role": "all",
15     "title": "Test 1",
16     "body": "hello day la thong bao",
17     "status": "created",
18     "createdAt": "2023-04-24T06:25:17.291Z",
19     "updatedAt": "2023-04-24T06:25:17.291Z",
20     "__v": 0
21   }
22 ]
```

Listing 1: All Sent Announcements

```
1 // [GET] All Announcement Received -> /api/ancm/list/receive
2 - Input:
3   Params: {null}
4   Body: {null}
5 - Output: [
6   {
7     "_id": "644620cd6797253c70c801aa",
8     "sendBy": "6444fb7fe14137d2c1586b02",
9     "sendTo": [
10      "6445427c934b44c3693c49da",
11      "6445602896858bdf4d8313db"
12    ],
13     "type": "announcement",
14     "role": "all",
15     "title": "Test 1",
16     "body": "hello day la thong bao",
17     "status": "created",
18     "createdAt": "2023-04-24T06:25:17.291Z",
19     "updatedAt": "2023-04-24T06:25:17.291Z",
20     "__v": 0
21   }
22 ]
```

Listing 2: All Announcement Received

```
1 // [GET] Announcement Detail by ID -> /api/ancm/:id
2 - Input:
3   Params: {:id = 644620cd6797253c70c801aa}
4   Body: {null}
5 - Output:{
6   "_id": "644620cd6797253c70c801aa",
7   "sendBy": "6444fb7fe14137d2c1586b02",
8   "sendTo": [
9     "6445427c934b44c3693c49da",
10    "6445602896858bdf4d8313db"
11  ],
12  "type": "announcement",
13  "role": "all",
14  "title": "Test 1",
15  "body": "hello day la thong bao",
16  "status": "created",
17  "createdAt": "2023-04-24T06:25:17.291Z",
18  "updatedAt": "2023-04-24T06:25:17.291Z",
19  "__v": 0
20 }
```

Listing 3: Announcement Detail by ID

```
1 // [POST] Create a New Announcement -> /api/ancm/create
2 - Input:
3   Params: {null}
4   Body: {
5     "sendTo"="6445427c934b44c3693c49da,6445602896858bdf4d8313db",
6     "title"="Test 1",
7     "type"="announcement",
8     "body"="hello day la thong bao",
9     "sendBy"="6444fb7fe14137d2c1586b02",
10    "role"="all"
11    "status"="created"
12  }
13 - Output:{"message": "New announcement has been created successfully"}
```

Listing 4: Create a New Announcement

### 3.2.2 Attendance APIs

```
1 // [GET] All Attendance -> /api/attend/list
2 - Input:
3   Params: {null}
4   Body: {null}
5 - Output:
6   {
7     "_id": "6446c08c7ca3afa5098c305d",
8     "student": {
9       "_id": "64468a7cd87974dbbae0fb1e",
10      "studentID": "STU238992",
11      "password": "$2a$10$ydFdkGGXfZS3YmrBNRr0gu5rABPdH5n7EWDa8JIJEtqLgDsNL63bq"
12    },
13     "fullname": "Do Thi Tram Anh",
14     "birth": "2016-07-18T17:00:00.000Z",
15     "gender": "female",
16     "address": "Long Xuyen",
17     "currentGrade": "1st",
18     "parent": "64468a7cd87974dbbae0fb1c",
19     "__v": 0,
20     "currentClass": "64468b09d87974dbbae0fb2b"
21   },
22   "currentDate": "2023-04-24T17:00:00.000Z",
23   "isPresent": true,
24   "description": "...",
25   "createdAt": "2023-04-24T17:46:52.583Z",
26   "updatedAt": "2023-04-24T17:46:52.583Z",
27   "__v": 0
28 }
```

Listing 5: All Attendance

```
1 // [GET] List by Student ID -> /api/attend/list/:studentID
2 - Input:
3   Params: {:studentID=6446c08c7ca3afa5098c305d}
4   Body: {null}
5 - Output:
6   {
7     "_id": "6446c08c7ca3afa5098c305d",
8     "student": {
9       "_id": "64468a7cd87974dbbae0fb1e",
10      "studentID": "STU238992",
11      "password": "$2a$10$ydFdkGGXfZS3YmrBNRr0gu5rABPdH5n7EWDa8JIJEtqLgDsNL63bq"
12    },
13    "fullname": "Do Thi Tram Anh",
14    "birth": "2016-07-18T17:00:00.000Z",
15    "gender": "female",
16    "address": "Long Xuyen",
17    "currentGrade": "1st",
18    "parent": "64468a7cd87974dbbae0fb1c",
19    "__v": 0,
20    "currentClass": "64468b09d87974dbbae0fb2b"
21  },
22  "currentDate": "2023-04-24T17:00:00.000Z",
23  "isPresent": true,
24  "description": "...",
25  "createdAt": "2023-04-24T17:46:52.583Z",
26  "updatedAt": "2023-04-24T17:46:52.583Z",
27  "__v": 0
28 ]
```

Listing 6: List by Student ID

```
1 // [GET] List by Class ID and date -> /api/attend/list/:classID/:date
2 - Input:
3   Params: {
4     :classID = 64468b09d87974dbbae0fb2b,
5     :date = 04-25-2023
6   }
7   Body: {null}
8 - Output:[
9   {
10    "_id": "6446c08c7ca3afa5098c305d",
11    "student": {
12      "_id": "64468a7cd87974dbbae0fb1e",
13      "studentID": "STU238992",
14      "password": "$2a$10$ydFdkGGXfZS3YmrBNRr0gu5rABPdH5n7EWDa8JIJEtqLgDsNL63bq"
15    },
16    "fullname": "Do Thi Tram Anh",
17    "birth": "2016-07-18T17:00:00.000Z",
18    "gender": "female",
19    "address": "Long Xuyen",
20    "currentGrade": "1st",
21    "parent": "64468a7cd87974dbbae0fb1c",
22    "__v": 0,
23    "currentClass": "64468b09d87974dbbae0fb2b"
24  },
25  "currentDate": "2023-04-24T17:00:00.000Z",
26  "isPresent": true,
27  "description": "...",
28  "createdAt": "2023-04-24T17:46:52.583Z",
29  "updatedAt": "2023-04-24T17:46:52.583Z",
30  "__v": 0
31 ]
```

Listing 7: List by Class ID and date

```
1 // [POST] Create a New Attendance Card -> /api/attend/create
2 - Input:
3   Params: {null}
4   Body: {
5     "student"="64468a7cd87974dbbae0fb1e"
6     "currentDate"="4-25-2023"
7     "isPresent"="true"
8     "description"="..."
9   }
10 - Output:[
11   {
12     "_id": "6446c08c7ca3afa5098c305d",
13     "student": {
14       "_id": "64468a7cd87974dbbae0fb1e",
15       "studentID": "STU238992",
16       "password": "$2a$10$ydFdkGGXfZS3YmrBNRr0gu5rABPdH5n7EWDa8JIJEtqLgDsNL63bq"
17     },
18     "fullname": "Do Thi Tram Anh",
19     "birth": "2016-07-18T17:00:00.000Z",
20     "gender": "female",
21     "address": "Long Xuyen",
22     "currentGrade": "1st",
23     "parent": "64468a7cd87974dbbae0fb1c",
24     "__v": 0,
25     "currentClass": "64468b09d87974dbbae0fb2b"
26   },
27   "currentDate": "2023-04-24T17:00:00.000Z",
28   "isPresent": true,
29   "description": "...",
30   "createdAt": "2023-04-24T17:46:52.583Z",
31   "updatedAt": "2023-04-24T17:46:52.583Z",
32   "__v": 0
33 ]
```

Listing 8: Create a New Attendance Card

### 3.2.3 Class APIs

```
1 // [GET] List All of Classes -> /api/class/list
2 - Input:
3   Params: {null}
4   Body: {null}
5 - Output: [
6   {
7     "_id": "64468b09d87974dbbae0fb2b",
8     "name": "1A",
9     "currentYear": 2023,
10    "teacher": {
11      "_id": "644689177661a30b6942f653",
12      "email": "thanh@gmail.com",
13      "password": "$2a$10$BUa4iXspSkMU/5GHaBtDA0rdLc720C//7m7gVTZPut2xpCh54u0QK"
14    },
15    "fullname": "Hoai Thanh",
16    "phone": "18001234",
17    "position": "teacher",
18    "address": "Binh Dinh",
19    "gender": "male",
20    "isHomeroom": true,
21    "__v": 0,
22    "classHomeroom": "64468b09d87974dbbae0fb2b"
23  },
24  "grade": "1st",
25  "schedules": [],
26  "students": [
27    "64468a7cd87974dbbae0fb1e",
28    "64468aa7d87974dbbae0fb23",
29    "64468adad87974dbbae0fb28"
30  ],
31  "createdAt": "2023-04-24T13:58:33.066Z",
32  "updatedAt": "2023-04-24T13:59:58.050Z",
33  "__v": 3
34 ]
```

Listing 9: List All of Classes

```
1 // [GET] List of Class by Grade -> /api/class/list/:grade
2 - Input:
3   Params: {:grade=1st}
4   Body: {null}
5 - Output:[
6   {
7     "_id": "64468b09d87974dbbae0fb2b",
8     "name": "1A",
9     "currentYear": 2023,
10    "teacher": {
11      "_id": "644689177661a30b6942f653",
12      "email": "thanh@gmail.com",
13      "password": "$2a$10$BUa4iXspSkMU/5GHaBtDA0rdLc720C//7m7gVTZPut2xpCh54u0QK"
14    },
15    "fullname": "Hoai Thanh",
16    "phone": "18001234",
17    "position": "teacher",
18    "address": "Binh Dinh",
19    "gender": "male",
20    "isHomeroom": true,
21    "__v": 0,
22    "classHomeroom": "64468b09d87974dbbae0fb2b"
23  },
24  "grade": "1st",
25  "schedules": [],
26  "students": [
27    "64468a7cd87974dbbae0fb1e",
28    "64468aa7d87974dbbae0fb23",
29    "64468adad87974dbbae0fb28"
30  ],
31  "createdAt": "2023-04-24T13:58:33.066Z",
32  "updatedAt": "2023-04-24T13:59:58.050Z",
33  "__v": 3
34 ]
```

Listing 10: List of Class by Grade



```
1 // [GET] Class Detail by ID -> /api/class/:id
2 - Input:
3   Params: {:id=64468b09d87974dbbae0fb2b}
4   Body: {null}
5 - Output:{
6   "_id": "64468b09d87974dbbae0fb2b",
7   "name": "1A",
8   "currentYear": 2023,
9   "teacher": "644689177661a30b6942f653",
10  "grade": "1st",
11  "schedules": [],
12  "students": [
13    "64468a7cd87974dbbae0fb1e",
14    "64468aa7d87974dbbae0fb23",
15    "64468adad87974dbbae0fb28"
16  ],
17  "createdAt": "2023-04-24T13:58:33.066Z",
18  "updatedAt": "2023-04-24T13:59:58.050Z",
19  "__v": 3
20 }
```

Listing 11: Class Detail by ID

```
1 // [POST] Create a New Class -> /api/class/create
2 - Input:
3   Params: {null}
4   Body: {
5     "name"="1A",
6     "currentYear"="2023",
7     "teacher"="644689177661a30b6942f653",
8     "grade"="1st"
9   }
10 - Output:{"message": "New class has been created successfully"}
```

Listing 12: Create a New Class

```
1 // [PUT] Add Student to Class -> /api/class/addStudent/:classID/:studentID
2 - Input:
3   Params: {
4     :classID = 64468b09d87974dbbae0fb2b
5     :studentID = 64468adad87974dbbae0fb28
6   }
7   Body: {null}
8 - Output: {"message": "Add Student Successfully"}
```

Listing 13: Add Student to Class

### 3.2.4 Parent APIs

```
1 // [GET] List All of Parents -> /api/parent/list
2 - Input:
3   Params: {null}
4   Body: {null}
5 - Output: [
6   {
7     "_id": "64468a7cd87974dbbae0fb1c",
8     "email": "dohungdung@gmail.com",
9     "password": "$2a$10$h8oK.krf9BZHIYqIgC0vK0lGuJX/TVMfQDgxm4dKJV5T2d6S00d1W",
10    "fullname": "Do Hung Dung",
11    "address": "Long Xuyen",
12    "phone": "0925312312",
13    "__v": 0,
14    "student": {
15      "_id": "64468a7cd87974dbbae0fb1e",
16      "studentID": "STU238992",
17      "password": "$2a$10$ydFdkGGXfZS3YmrBNRr0gu5rABPdH5n7EWDa8JIJEtqLgDsNL63bq"
18    },
19    "fullname": "Do Thi Tram Anh",
20    "birth": "2016-07-18T17:00:00.000Z",
21    "gender": "female",
22    "address": "Long Xuyen",
23    "currentGrade": "1st",
24    "parent": "64468a7cd87974dbbae0fb1c",
25    "__v": 0,
26    "currentClass": "64468b09d87974dbbae0fb2b"
27  }, {
28    "_id": "64468aa6d87974dbbae0fb21",
29    "email": "info.trankhaihoang@gmail.com",
30    "password": "$2a$10$eeNEoqq.GOPslBBMZN4NyezsSLcZjzFHnwrrQQg..u36hr00jawjW",
31    "fullname": "Tran Khai Hoang",
32    "address": "Long Xuyen",
33    "phone": "0378807854",
34    "__v": 0,
35    "student": {
36      "_id": "64468aa7d87974dbbae0fb23",
37      "studentID": "STU233576",
38      "password": "$2a$10$n8fJ.ln6dt.hkMpCPVyvg.7Uih5Ued2xuDg91cb2W6CVngy/UW9Vy"
39    },
40    "fullname": "Tran Minh Tri",
41    "birth": "2016-07-14T17:00:00.000Z",
42    "gender": "male",
43    "address": "Long Xuyen",
44    "currentGrade": "1st",
45    "parent": "64468aa6d87974dbbae0fb21",
46    "__v": 0,
47    "currentClass": "64468b09d87974dbbae0fb2b"
48  }
49 ]
```

Listing 14: List All of Parents

```
1 // [GET] List Parents by ClassID -> /api/parent/list/:classID
2 - Input:
3   Params: {:classID=6445252374c5fbcd83bd661b}
4   Body: {null}
5 - Output:[
6   {
7     "_id": "6445427c934b44c3693c49da",
8     "email": "kyn@gmail.com",
9     "password": "$2a$10$cocIhjfgNdZs745PSDoZ801VAnli0NKmU6nS4rkoSWcF6UswGjn/S",
10    "fullname": "Khuu Yen Ngoc",
11    "address": "Dinh My",
12    "phone": "0372360506",
13    "__v": 0,
14    "student": {
15      "_id": "6445427c934b44c3693c49dc",
16      "studentID": "STU232242",
17      "password": "$2a$10$vDnWJjdpVki01/x6t0ga70xt/Xmhd7FKlGealisAP/KpDeezklHAW"
18    },
19    "fullname": "Tran Trong Dao",
20    "birth": "2030-12-11T17:00:00.000Z",
21    "gender": "male",
22    "address": "Dinh My",
23    "currentGrade": "1st",
24    "parent": "6445427c934b44c3693c49da",
25    "__v": 0,
26    "currentClass": "6445252374c5fbcd83bd661b"
27  }
28 ]
```

Listing 15: List Parents by ClassID

```
1 // [GET] Detail by ID -> /api/parent/:id
2 - Input:
3   Params: {:id=64468a7cd87974dbbae0fb1c}
4   Body: {null}
5 - Output:{
6   "_id": "64468a7cd87974dbbae0fb1c",
7   "email": "dohungdung@gmail.com",
8   "password": "$2a$10$h8oK.krf9BZHIYqIgC0vK0lGuJX/TVMfQDgxm4dKJV5T2d6S00d1W",
9   "fullname": "Do Hung Dung",
10  "address": "Long Xuyen",
11  "phone": "0925312312",
12  "__v": 0,
13  "student": {
14    "_id": "64468a7cd87974dbbae0fb1e",
15    "studentID": "STU238992",
16    "password": "$2a$10$ydFdkGGXfZS3YmrBNRr0gu5rABPdH5n7EWDa8JIJEtqLgDsNL63bq",
17    "fullname": "Do Thi Tram Anh",
18    "birth": "2016-07-18T17:00:00.000Z",
19    "gender": "female",
20    "address": "Long Xuyen",
21    "currentGrade": "1st",
22    "parent": "64468a7cd87974dbbae0fb1c",
23    "__v": 0,
24    "currentClass": "64468b09d87974dbbae0fb2b"
25  }
26 }
```

Listing 16: Detail by ID

```
1 // [GET] Detail by Student ID -> /api/parent/detail/:studentID
2 - Input:
3   Params: {:studentID=64468adad87974dbbae0fb28}
4   Body: {null}
5 - Output:{
6   "_id": "64468adad87974dbbae0fb26",
7   "email": "nhatthong33333@gmail.com",
8   "password": "$2a$10$0qH5xhGCS2syxdTEyYkMj.8YB/EoTdj0bx7hh9B/gokUHbhd1D/G",
9   "fullname": "Nguyen Nhat Thong",
10  "address": "TP HCM",
11  "phone": "0333333333",
12  "__v": 0,
13  "student": {
14    "_id": "64468adad87974dbbae0fb28",
15    "studentID": "STU230150",
16    "password": "$2a$10$Mr9fA0obpuFG0bpOXbs12eMtE0eNUT/v/jKq4jFKvyUL.4F6Jnfem",
17    "fullname": "Nguyen Nhat He",
18    "birth": "2016-12-14T17:00:00.000Z",
19    "gender": "male",
20    "address": "TP HCM",
21    "currentGrade": "1st",
22    "parent": "64468adad87974dbbae0fb26",
23    "__v": 0,
24    "currentClass": "64468b09d87974dbbae0fb2b"
25  }
26 }
```

Listing 17: Detail by Student ID

```
1 // [POST] Create a new Parent and new a Student -> /api/parent/create
2 - Input:
3   Params: {null}
4   Body: {
5     "email"="nhatthong33333@gmail.com",
6     "fullname"="Nguyen Nhat Thong",
7     "phone"="0333333333",
8     "address"="TP HCM",
9     "stu_fullname"="Nguyen Nhat He",
10    "birth"="12-15-2016",
11    "gender"="male",
12    "currentGrade"="1st"
13  }
14 - Output:{"message": "New student has been created successfully"}
```

Listing 18: Create a new Parent and new a Student

### 3.2.5 Subject APIs

```
1 // [POST] Get list all subject -> /api/subject/list
2 - Input:
3   Params: {null}
4   Body: {null}
5 - Output: [
6   {
7     "_id": "6449f1c46a6a7e9b533daee",
8     "name": "Tieng Viet",
9     "description": "....",
10    "teacher": {
11      "_id": "6444fb42e14137d2c1586aff",
12      "email": "it.trankhaihoang@gmail.com",
13      "password": "$2a$10$Sv0xj5gfRMDWdV/xrj9ksev10CgIogbbWxFexa9YHHa1R/6Cv60wi"
14    },
15    "fullname": "Tran Khai Hoang",
16    "phone": 378807854,
17    "position": "teacher",
18    "address": "Thoai Son",
19    "gender": "male",
20    "__v": 0,
21    "isHomeroom": true,
22    "classHomeroom": "6447975b1f2ec28f4f7b199d"
23  },
24  "grade": "1st"
25 ],
26 {
27   "_id": "6449f2526a6a7e9b533daeb0",
28   "name": "Toan",
29   "description": "....",
30   "teacher": {
31     "_id": "6444fb42e14137d2c1586aff",
32     "email": "it.trankhaihoang@gmail.com",
33     "password": "$2a$10$Sv0xj5gfRMDWdV/xrj9ksev10CgIogbbWxFexa9YHHa1R/6Cv60wi"
34   },
35   "fullname": "Tran Khai Hoang",
36   "phone": 378807854,
37   "position": "teacher",
38   "address": "Thoai Son",
39   "gender": "male",
40   "__v": 0,
41   "isHomeroom": true,
42   "classHomeroom": "6447975b1f2ec28f4f7b199d"
43 },
44 ]
```

Listing 19: Get list all subject

```
1 // [POST] list subjects by grade -> /api/subject/list/:grade
2 - Input:
3   Params: {:grade=1st}
4   Body: {null}
5 - Output:[
6   {
7     "_id": "6449f1c46a6a7e9b533daee",
8     "name": "Tieng Viet",
9     "description": "....",
10    "teacher": {
11      "_id": "6444fb42e14137d2c1586aff",
12      "email": "it.trankhaihoang@gmail.com",
13      "password": "$2a$10$Sv0xj5gfRMDWdV/xrj9ksevl0CgIogbbWxFexa9YHHa1R/6Cv60wi "
14    },
15    "fullname": "Tran Khai Hoang",
16    "phone": 378807854,
17    "position": "teacher",
18    "address": "Thoai Son",
19    "gender": "male",
20    "__v": 0,
21    "isHomeroom": true,
22    "classHomeroom": "6447975b1f2ec28f4f7b199d"
23  },
24  "grade": "1st"
25 },
26 {
27   "_id": "6449f2526a6a7e9b533daeb0",
28   "name": "Toan",
29   "description": "....",
30   "teacher": {
31     "_id": "6444fb42e14137d2c1586aff",
32     "email": "it.trankhaihoang@gmail.com",
33     "password": "$2a$10$Sv0xj5gfRMDWdV/xrj9ksevl0CgIogbbWxFexa9YHHa1R/6Cv60wi "
34   },
35   "fullname": "Tran Khai Hoang",
36   "phone": 378807854,
37   "position": "teacher",
38   "address": "Thoai Son",
39   "gender": "male",
40   "__v": 0,
41   "isHomeroom": true,
42   "classHomeroom": "6447975b1f2ec28f4f7b199d"
43 },
44 ]
```

Listing 20: list subjects by grade

### 3.2.6 School Staff APIs

```

1 // [GET] List All of Teacher -> /api/SchoolStaff/teacher
2 - Input:
3   Params: {null}
4   Body: {null}
5 - Output:
6   {
7     "_id": "6444fb42e14137d2c1586aff",
8     "email": "it.trankhaihoang@gmail.com",
9     "password": "$2a$10$Sv0xj5gFRMDWdV/xrj9ksevl0CgIogbbWxFexa9YHHa1R/6Cv60wi",
10    "fullname": "Tran Khai Hoang",
11    "phone": 378807854,
12    "position": "teacher",
13    "address": "Thoai Son",
14    "gender": "male",
15    "__v": 0,
16    "isHomeroom": true,
17    "classHomeroom": {
18      "_id": "6447975b1f2ec28f4f7b199d",
19      "name": "2a",
20      "currentYear": 2023,
21      "teacher": "6444fb42e14137d2c1586aff",
22      "grade": "2nd",
23      "schedules": [],
24      "students": [
25        "6448048165ae1a30b8822566",
26        "64478cc6b125341ed0501c61"
27      ],
28      "createdAt": "2023-04-25T09:03:23.717Z",
29      "updatedAt": "2023-04-25T17:00:49.550Z",
30      "__v": 2
31    },
32  },
33  {
34    "_id": "6447f4f372e8c9cecb8af073",
35    "email": "dung@gmail.com",
36    "password": "$2a$10$Pb/de3n8k8D8bjz50IgMg0/dIZWJ9aP7W9.NBeCEuYomFFIEP9NGy",
37    "fullname": "Nhan Dung",
38    "phone": "19006767",
39    "position": "teacher",
40    "address": "Long An",
41    "gender": "male",
42    "isHomeroom": false,
43    "__v": 0
44  }
45 ]

```

Listing 21: List All of Teacher



```
1 // [GET] Get Available Teachers -> /api/SchoolStaff/available
2 - Input:
3   Params: {null}
4   Body: {null}
5 - Output:[
6   {
7     "_id": "6447f4d872e8c9cecb8af071",
8     "email": "khiem@gmail.com",
9     "password": "$2a$10$Kwthjvff0NCBEuaQawaed0DfkSxxM6vwu4n00YdjtHJS0rmbeDs3C",
10    "fullname": "Gia Khiem",
11    "phone": "18004321",
12    "position": "teacher",
13    "address": "Dong Nai",
14    "gender": "male",
15    "isHomeroom": false,
16    "__v": 0
17  },
18  {
19    "_id": "6447f4f372e8c9cecb8af073",
20    "email": "dung@gmail.com",
21    "password": "$2a$10$Pb/de3n8k8D8bjz50IgMg0/dIZWJ9aP7W9.NBeCEuYomFFIEP9NGy",
22    "fullname": "Nhan Dung",
23    "phone": "19006767",
24    "position": "teacher",
25    "address": "Long An",
26    "gender": "male",
27    "isHomeroom": false,
28    "__v": 0
29  }
30 ]
```

Listing 22: Get Available Teachers

```
1 // [GET] Detail by ID -> /api/SchoolStaff/:id
2 - Input:
3   Params: {:id=6444fb42e14137d2c1586aff}
4   Body: {null}
5 - Output:{
6   "_id": "6444fb42e14137d2c1586aff",
7   "email": "it.trankhaihoang@gmail.com",
8   "password": "$2a$10$Sv0xj5gfRMDWdV/xrj9ksevl0CgIogbbWxFexa9YHHa1R/6Cv60wi",
9   "fullname": "Tran Khai Hoang",
10  "phone": "378807854",
11  "position": "teacher",
12  "address": "Thoai Son",
13  "gender": "male",
14  "__v": 0,
15  "isHomeroom": true,
16  "classHomeroom": "6447975b1f2ec28f4f7b199d"
17 }
```

Listing 23: Detail by ID

```
1 // [POST] Create a New School Staff -> /api/SchoolStaff/create
2 - Input:
3   Params: {null}
4   Body: {
5     "email"="phule@gmail.com" ,
6     "fullname"="Phu Le" ,
7     "phone"="19006794" ,
8     "position"="teacher" ,
9     "address"="Ha Noi" ,
10    "gender"="male"
11  }
12 - Output:{"message": "New school staff has been created successfully"}
```

Listing 24: Create a New School Staff

### 3.2.7 Student APIs

```
1 // [GET] List All of Students -> /api/student/list
2 - Input:
3   Params: {null}
4   Body: {null}
5 - Output: [{ "_id": "64468a7cd87974dbbae0fb1e",
6   "studentID": "STU238992",
7   "password": "$2a$10$ydFdkGGXfZS3YmrBNRrOgu5rABPdH5n7EWDa8JIJEtqLgDsNL63bq",
8   "fullname": "Do Thi Tram Anh",
9   "birth": "2016-07-18T17:00:00.000Z",
10  "gender": "female",
11  "address": "Long Xuyen",
12  "currentGrade": "1st",
13  "parent": {
14    "_id": "64468a7cd87974dbbae0fb1c",
15    "email": "dohungdung@gmail.com",
16    "password": "$2a$10$h8oK.krf9BZHIYqIgC0vK0lGuJX/TVMfQDgxm4dKJV5T2d6S00d1W"
17  },
18  "fullname": "Do Hung Dung",
19  "address": "Long Xuyen",
20  "phone": "0925312312",
21  "__v": 0,
22  "student": "64468a7cd87974dbbae0fb1e"
23 }, {"_id": "64468b09d87974dbbae0fb2b",
24 "currentClass": {
25   "_id": "64468b09d87974dbbae0fb2b",
26   "name": "1A",
27   "currentYear": 2023,
28   "teacher": {
29     "_id": "644689177661a30b6942f653",
30     "email": "thanh@gmail.com",
31     "password": "$2a$10$BUa4iXspSkMU/5GHaBtDA0rdLc720C//7
32     m7gVTZPut2xpCh54u0QK",
33     "fullname": "Hoai Thanh",
34     "phone": "18001234",
35     "position": "teacher",
36     "address": "Binh Dinh",
37     "gender": "male",
38     "isHomeroom": true,
39     "__v": 0,
40     "classHomeroom": "64468b09d87974dbbae0fb2b"
41   },
42   "grade": "1st",
43   "schedules": [],
44   "students": [
45     "64468a7cd87974dbbae0fb1e",
46     "64468aa7d87974dbbae0fb23",
47     "64468adad87974dbbae0fb28"
48   ], "createdAt": "2023-04-24T13:58:33.066Z",
49   "updatedAt": "2023-04-24T13:59:58.050Z",
50   "__v": 3
51 } } ]
```

Listing 25: List All of Students

```

1 // [GET] List Students by Class ID -> /api/student/:classID
2 - Input:
3   Params: {:classID=6445427c934b44c3693c49dc}
4   Body: {null}
5 - Output:
6   {
7     "_id": "6445427c934b44c3693c49dc",
8     "studentID": "STU232242",
9     "password": "$2a$10$vDnWJjdpVki01/x6t0ga70xt/Xmhd7FKlGealisAP/KpDeezklHAW",
10    "fullname": "Tran Trong Dao",
11    "birth": "2030-12-11T17:00:00.000Z",
12    "gender": "male",
13    "address": "Dinh My",
14    "currentGrade": "1st",
15    "parent": {
16      "_id": "6445427c934b44c3693c49da",
17      "email": "kyn@gmail.com",
18      "password": "$2a$10$cocIhjfgNdZs745PSDoZ801VAnli0NKmU6nS4rkoSWcF6UswGjn/S"
19    },
20    "fullname": "Khuu Yen Ngoc",
21    "address": "Dinh My",
22    "phone": "0372360506",
23    "__v": 0,
24    "student": "6445427c934b44c3693c49dc"
25  }, "__v": 0,
26  "currentClass": {
27    "_id": "6445252374c5fbcd83bd661b",
28    "name": "1A",
29    "currentYear": 2023,
30    "teacher": {
31      "_id": "6444fb42e14137d2c1586aff",
32      "email": "info.trankhaihoang@gmail.com",
33      "password": "$2a$10$Sv0xj5gfrMDWdV/xrj9ksevl0CgIogbbWxFexa9YHHa1R/6Cv60wi",
34      "fullname": "Tran Khai Hoang",
35      "phone": 378807854,
36      "position": "teacher",
37      "address": "Thoai Son",
38      "gender": "male",
39      "__v": 0,
40      "isHomeroom": false
41    }, "grade": "1st",
42    "schedules": [],
43    "students": [
44      "6445427c934b44c3693c49dc"
45    ],
46    "createdAt": "2023-04-23T12:31:31.206Z",
47    "updatedAt": "2023-04-23T14:50:15.505Z",
48    "__v": 3
49  }
50 ]

```

Listing 26: List Students by Class ID

### 3.2.8 User APIs

```
1 // [POST] Login -> /api/user/login
2 - Input:
3   Params: {null}
4   Body: {
5     "email"="STU238992",
6     "password"="STU238992"
7   }
8 - Output:{
9   "message": "Login successful.",
10  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NDQ2OGE3Y2Q4Nzk3NGRiYmF1MGZiMWUuLCJpYXQiOiJlMjE2ODIzNTkwNzZ9.L5MWCcvjbPh0wbAvvQbejb_QIi9kW3541lmVr_kAABA",
11  "user": {
12    "_id": "64468a7cd87974dbbae0fb1e",
13    "studentID": "STU238992",
14    "password": "$2a$10$ydFdkGGXfZS3YmrBNRr0gu5rABPdH5n7EWDa8JIJEtqLgDsNL63bq",
15    "fullname": "Do Thi Tram Anh",
16    "birth": "2016-07-18T17:00:00.000Z",
17    "gender": "female",
18    "address": "Long Xuyen",
19    "currentGrade": "1st",
20    "parent": "64468a7cd87974dbbae0fb1c",
21    "__v": 0,
22    "currentClass": "64468b09d87974dbbae0fb2b"
23  }
24 }
```

Listing 27: Login

```
1 // [POST] Logout -> /api/user/logout
2 - Input:
3   Params: {null}
4   Body: {null}
5 - Output:{"message": "Logout successful"}
```

Listing 28: Logout

```
1 // [POST] Change password -> /api/user/changePassword
2 - Input:
3   Params: {null}
4   Body: {
5     "Password"="hoang9999",
6     "NewPassword"="hoang"
7   }
8 - Output:{"message": "Password changed successfully"}
```

Listing 29: Change password

### 3.3 APIs documentation using Postman

Please follows this link for the complete Postman documentation: [Postman Documentation](#)

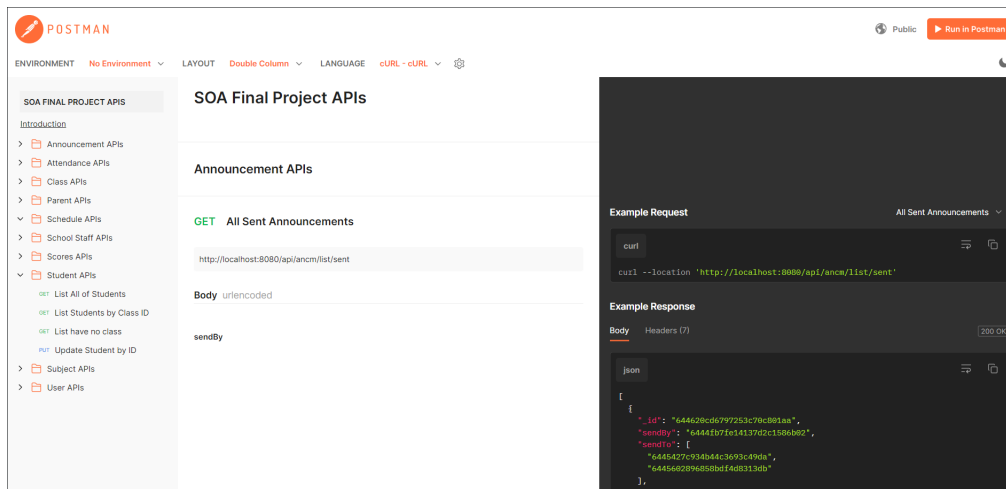


Figure 4: APIs documentation preview via Postman

## 4 System Demo

### 4.1 System Demo script

#### 4.1.1 Principal Flow

Login → List of Classes → Detail of Class → Create Class → Add Student to Class → List of Teacher → Schedule → Create Announcement → History of Announcement

#### 4.1.2 Teacher Flow

Login → Dash Board → List Students → Check Attendance → Scoreboard → Schedule → Inbox → Create Announcement → History of Announcement

#### 4.1.3 Parent & Student Flow

Login → Dash Board → List Announcements → Schedule → Scoreboard

### 4.2 System Demo



Figure 5: Login Interface

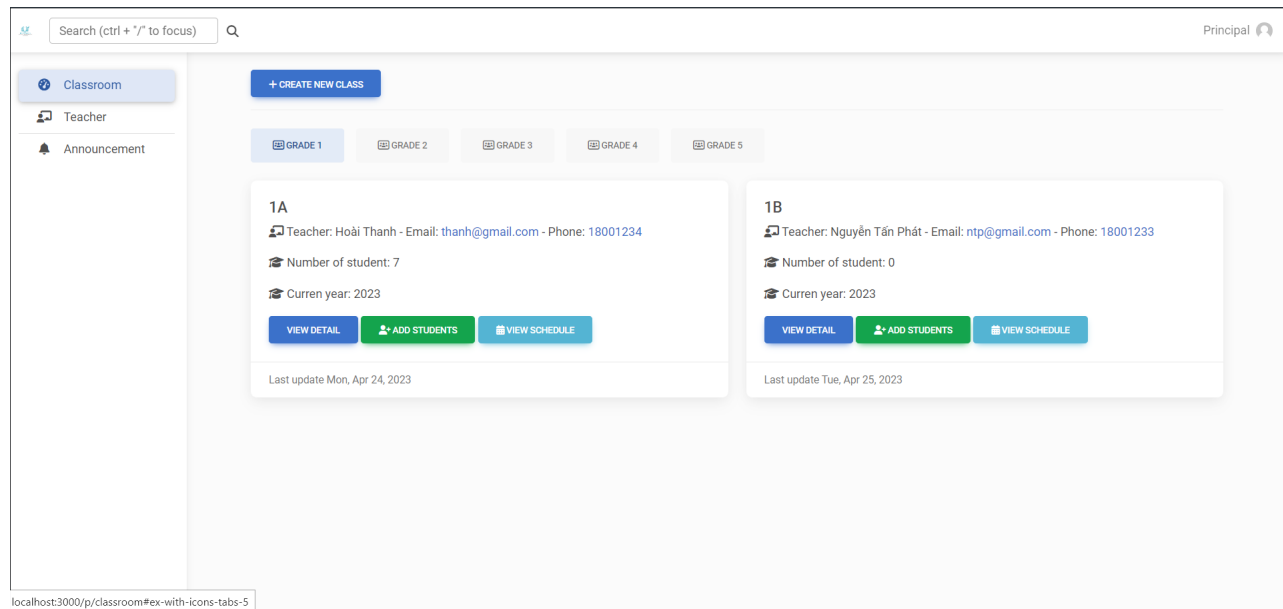


Figure 6: Principal List of Classes Interface

Detail

X

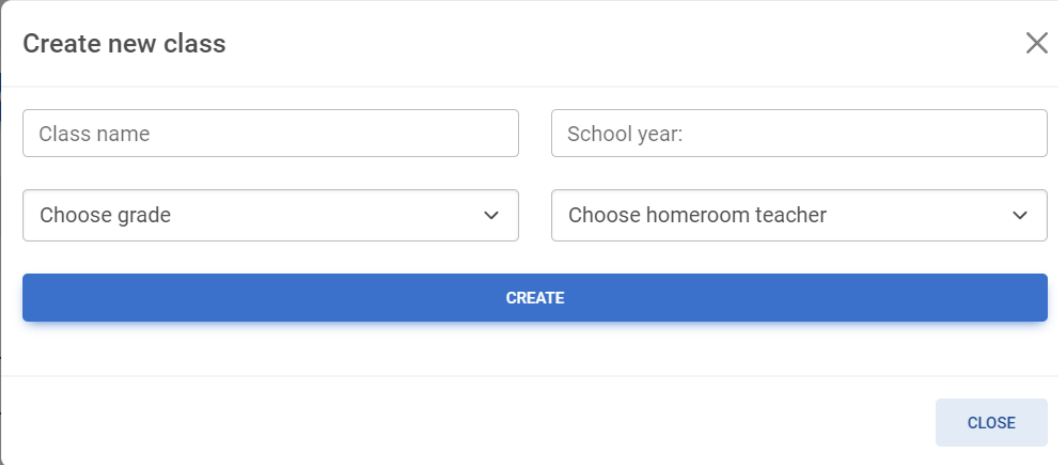
Show 10 entries

Search:

Student ID	Full name	Parent	Birth	Gender	Address
STU230150	Nguyễn Nhật Hệ	Nguyễn Nhật Thống Email: <a href="mailto:nhatthong33333@gmail.com">nhatthong33333@gmail.com</a> Phone: 0333333333	14th Dec 2016	male	TP HCM
STU233576	Trần Minh Trí	Trần Khải Hoàng Email: <a href="mailto:info.trankhaihoang@gmail.com">info.trankhaihoang@gmail.com</a> Phone: 0378807854	14th Jul 2016	male	Long Xuyên
STU233930	Nguyễn Văn C	Nguyễn Văn D Email: <a href="mailto:vanDgmail.com">vanDgmail.com</a> Phone: 0333322222	14th Dec 2016	male	TP HCM
STU237991	Nguyễn Văn Hòa	Nguyễn Minh Thiện Email: <a href="mailto:it.trankhaihoang@gmail.com">it.trankhaihoang@gmail.com</a> Phone: 03333212321	14th Dec 2016	male	TP HCM
STU238496	Nguyễn Văn C	Nguyễn Văn B Email: <a href="mailto:vanBgmail.com">vanBgmail.com</a> Phone: 0333322222	14th Dec 2016	male	TP HCM
STU238860	Nguyễn Bò Viên	Nguyễn Đình Quý Email: <a href="mailto:nguyndinhquy@gmail.com">nguyndinhquy@gmail.com</a> Phone: 0333322222	14th Dec 2016	male	TP HCM

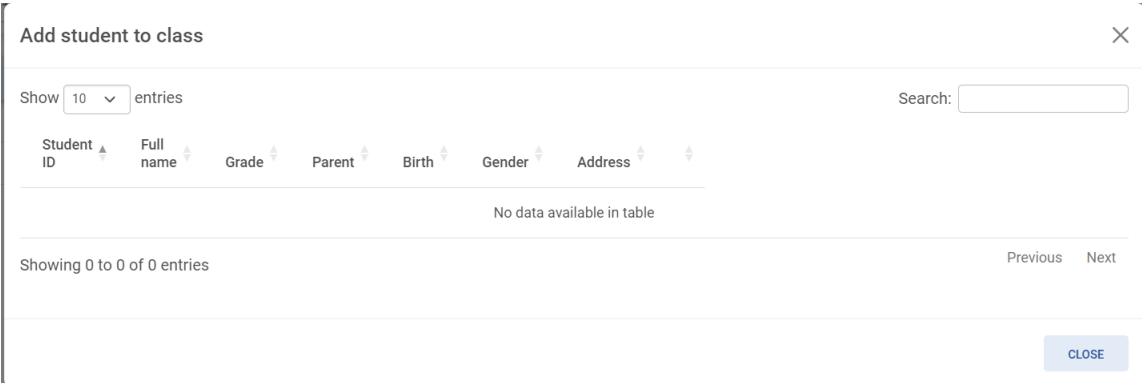
Figure 7: Principal Detail of Class Interface





The 'Create new class' interface is a modal window with a title bar containing the text 'Create new class' and a close button (X). The main content area contains four input fields arranged in a 2x2 grid: 'Class name', 'School year:', 'Choose grade' (with a dropdown arrow), and 'Choose homeroom teacher' (with a dropdown arrow). Below these fields is a large blue button labeled 'CREATE'. At the bottom right of the modal is a light blue button labeled 'CLOSE'.

Figure 8: Create Class Interface



The 'Add student to class' interface is a modal window with a title bar containing the text 'Add student to class' and a close button (X). The main content area features a table with columns: 'Student ID', 'Full name', 'Grade', 'Parent', 'Birth', 'Gender', and 'Address'. Each column header has a small upward arrow icon. Above the table, there is a 'Show 10 entries' dropdown menu and a 'Search:' input field. Below the table, the text 'No data available in table' is displayed. At the bottom left, it says 'Showing 0 to 0 of 0 entries'. At the bottom right, there are 'Previous' and 'Next' navigation links. A light blue 'CLOSE' button is located at the bottom right of the modal.

Figure 9: Add Student to Class Interface

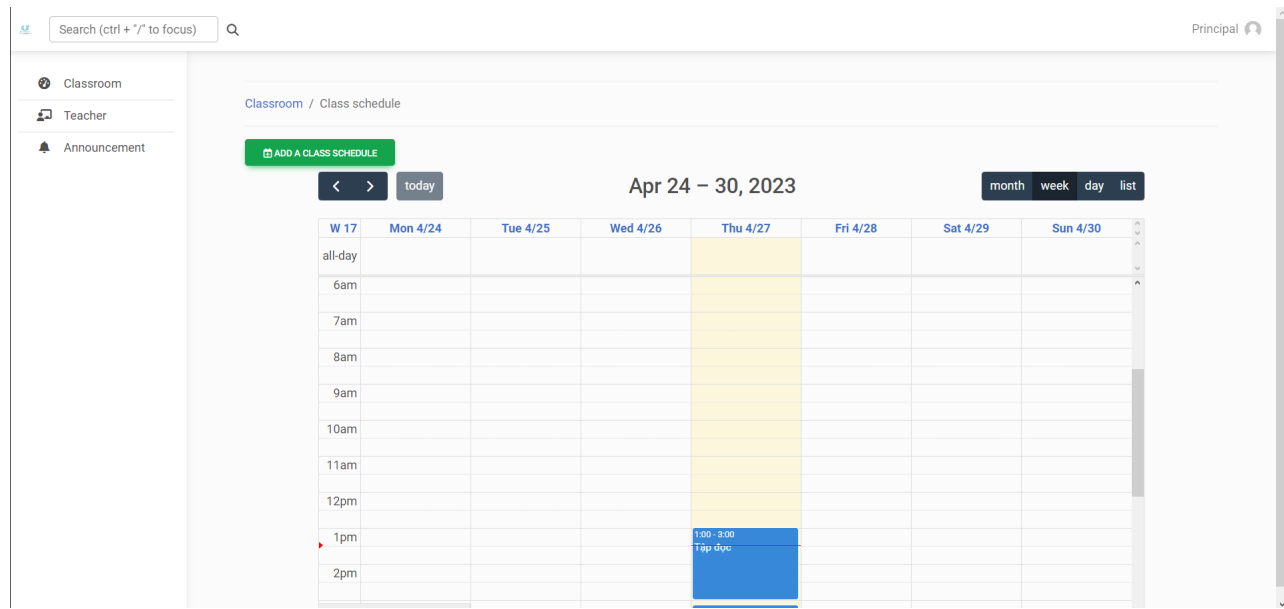


Figure 10: Principal Schedule of Class Interface

Add schedule

Choose subject

Choose date of week

dd/mm/yyyy

dd/mm/yyyy

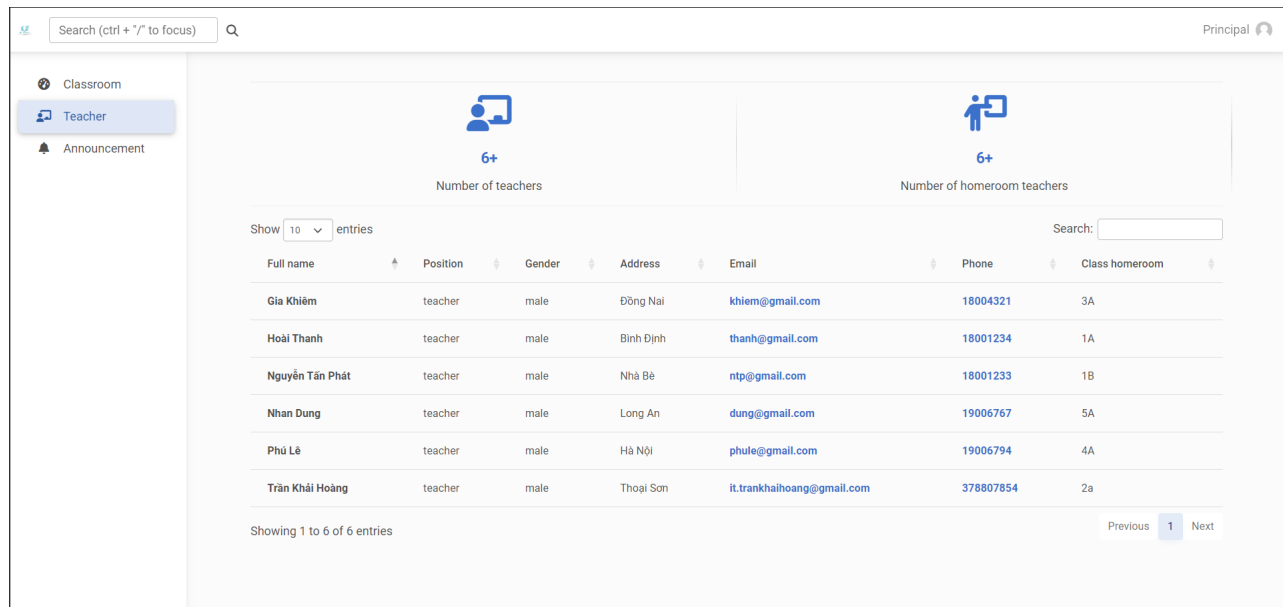
Starting time

Ending time

CREATE

CLOSE

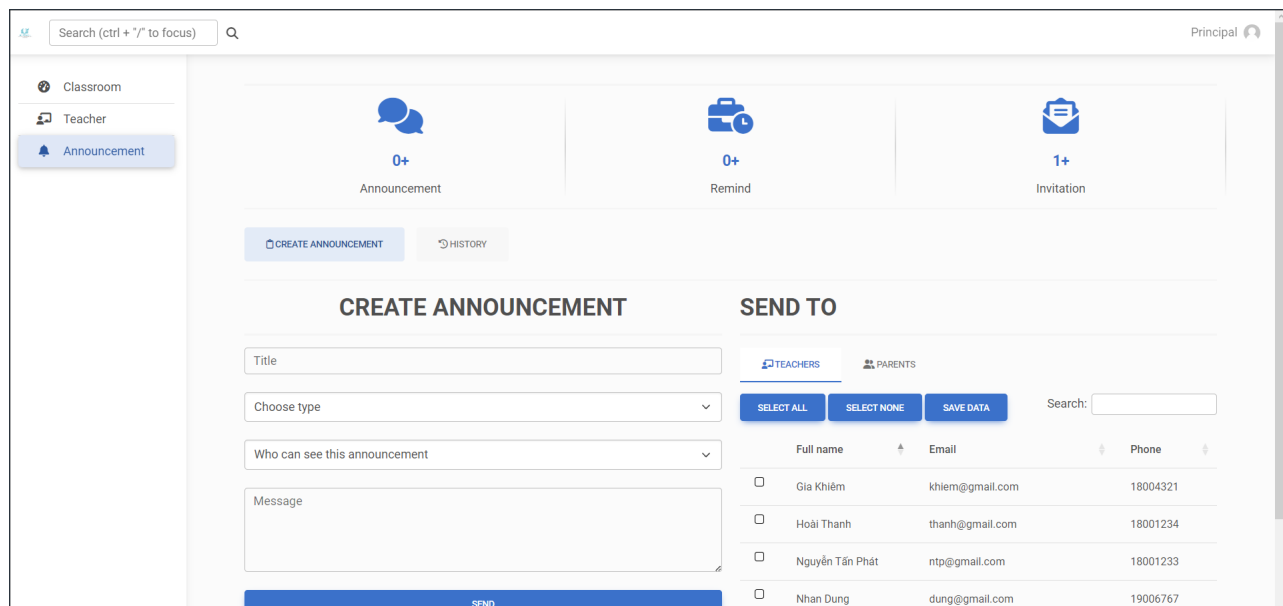
Figure 11: Principal Create Schedule of Class Interface



The interface shows a sidebar with 'Classroom', 'Teacher', and 'Announcement' options. The main area displays two summary cards: 'Number of teachers' (6+) and 'Number of homeroom teachers' (6+). Below these is a table of teachers with columns for Full name, Position, Gender, Address, Email, Phone, and Class homeroom. The table lists 6 teachers. A search bar and pagination controls are also present.

Full name	Position	Gender	Address	Email	Phone	Class homeroom
Gia Khiêm	teacher	male	Đồng Nai	khiem@gmail.com	18004321	3A
Hoài Thanh	teacher	male	Bình Định	thanh@gmail.com	18001234	1A
Nguyễn Tấn Phát	teacher	male	Nhà Bè	ntp@gmail.com	18001233	1B
Nhan Dung	teacher	male	Long An	dung@gmail.com	19006767	5A
Phú Lê	teacher	male	Hà Nội	phule@gmail.com	19006794	4A
Trần Khải Hoàng	teacher	male	Thoại Sơn	it.trankhaihoang@gmail.com	378807854	2a

Figure 12: Principal List of Teacher Interface



The interface shows a sidebar with 'Classroom', 'Teacher', and 'Announcement' options. The main area displays three summary cards: 'Announcement' (0+), 'Remind' (0+), and 'Invitation' (1+). Below these is a 'CREATE ANNOUNCEMENT' form with fields for Title, Choose type, Who can see this announcement, and Message. There is also a 'SEND TO' section with a list of teachers and a search bar.

**CREATE ANNOUNCEMENT**

Title:

Choose type:

Who can see this announcement:

Message:

**SEND TO**

TEACHERS: ☐ SELECT ALL ☐ SELECT NONE

Search:

Full name	Email	Phone
<input type="checkbox"/> Gia Khiêm	khiem@gmail.com	18004321
<input type="checkbox"/> Hoài Thanh	thanh@gmail.com	18001234
<input type="checkbox"/> Nguyễn Tấn Phát	ntp@gmail.com	18001233
<input type="checkbox"/> Nhan Dung	dung@gmail.com	19006767

Figure 13: Principal Create Announcement Interface

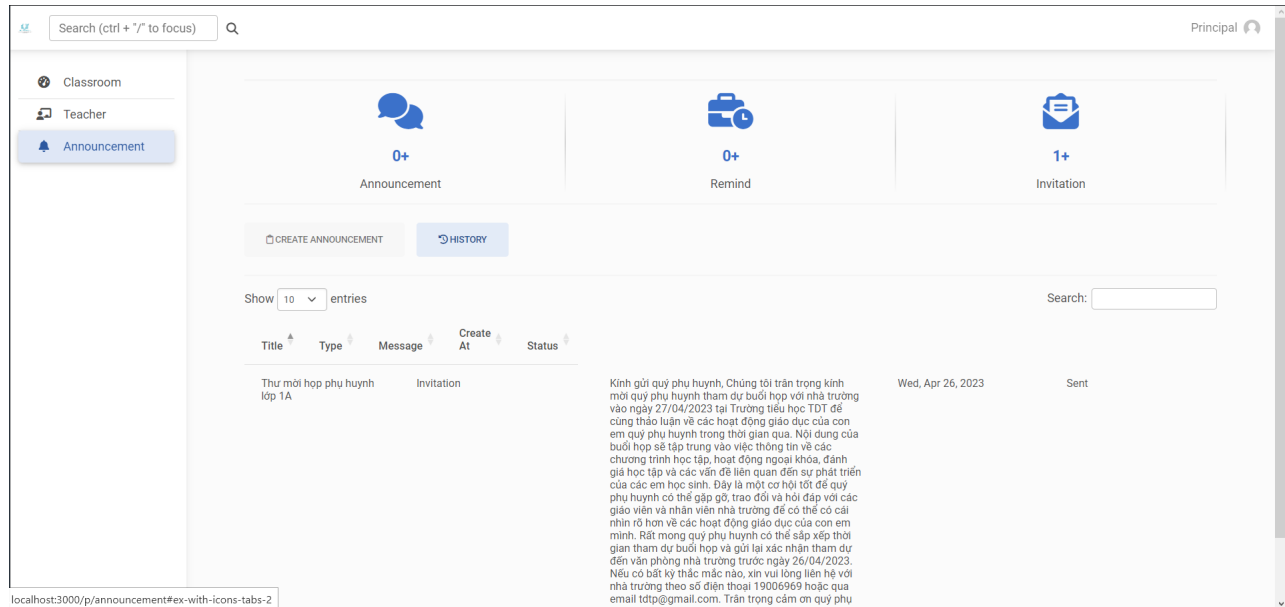


Figure 14: Principal Announcement History Interface

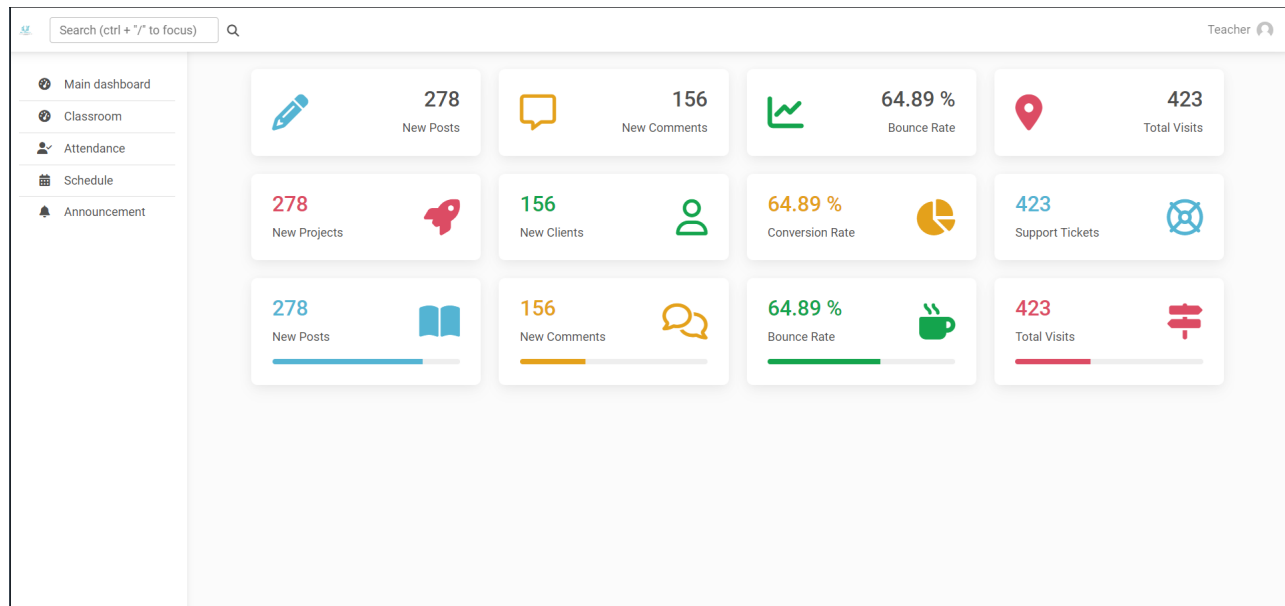
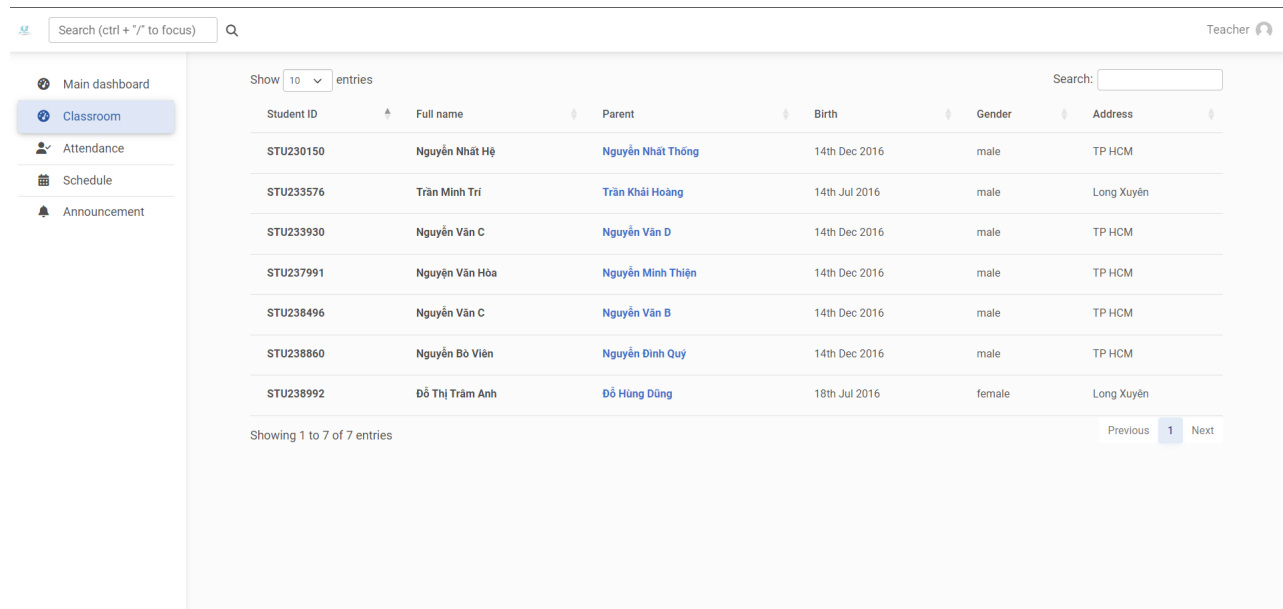


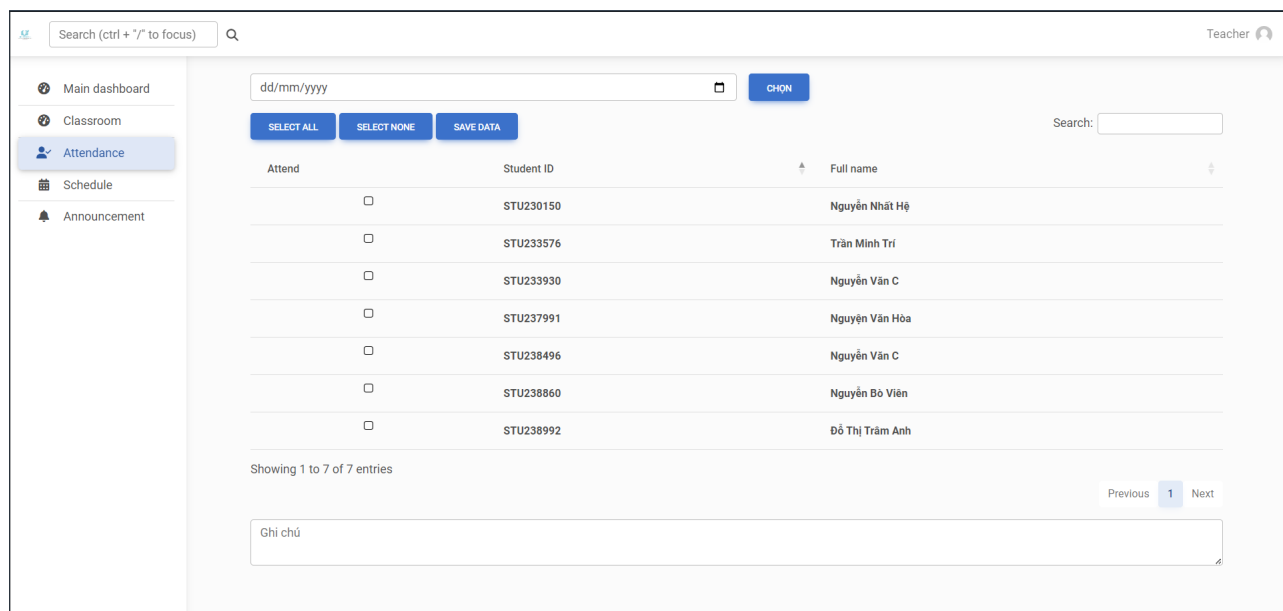
Figure 15: Teacher Dashboard Interface



The interface shows a sidebar with navigation links: Main dashboard, Classroom (selected), Attendance, Schedule, and Announcement. The main area displays a table of students with columns: Student ID, Full name, Parent, Birth, Gender, and Address. A search bar is at the top right, and a 'Show 10 entries' dropdown is at the top left of the table. The table contains 7 entries, and pagination shows 'Showing 1 to 7 of 7 entries' with 'Previous' and 'Next' buttons.

Student ID	Full name	Parent	Birth	Gender	Address
STU230150	Nguyễn Nhật Hề	Nguyễn Nhật Thống	14th Dec 2016	male	TP HCM
STU233576	Trần Minh Trí	Trần Khải Hoàng	14th Jul 2016	male	Long Xuyên
STU233930	Nguyễn Văn C	Nguyễn Văn D	14th Dec 2016	male	TP HCM
STU237991	Nguyễn Văn Hòa	Nguyễn Minh Thiện	14th Dec 2016	male	TP HCM
STU238496	Nguyễn Văn C	Nguyễn Văn B	14th Dec 2016	male	TP HCM
STU238860	Nguyễn Bò Viên	Nguyễn Đình Quý	14th Dec 2016	male	TP HCM
STU238992	Đỗ Thị Trâm Anh	Đỗ Hùng Dũng	18th Jul 2016	female	Long Xuyên

Figure 16: Teacher List of Students Interface



The interface shows a sidebar with navigation links: Main dashboard, Classroom, Attendance (selected), Schedule, and Announcement. The main area displays a table for attendance with columns: Attend, Student ID, and Full name. Above the table are input fields for date (dd/mm/yyyy), a 'CHỌN' button, and buttons for 'SELECT ALL', 'SELECT NONE', and 'SAVE DATA'. A search bar is at the top right. The table contains 7 entries, and pagination shows 'Showing 1 to 7 of 7 entries' with 'Previous' and 'Next' buttons. There is also a 'Ghi chú' (Note) field at the bottom.

Attend	Student ID	Full name
<input type="checkbox"/>	STU230150	Nguyễn Nhật Hề
<input type="checkbox"/>	STU233576	Trần Minh Trí
<input type="checkbox"/>	STU233930	Nguyễn Văn C
<input type="checkbox"/>	STU237991	Nguyễn Văn Hòa
<input type="checkbox"/>	STU238496	Nguyễn Văn C
<input type="checkbox"/>	STU238860	Nguyễn Bò Viên
<input type="checkbox"/>	STU238992	Đỗ Thị Trâm Anh

Figure 17: Teacher Attendance Interface

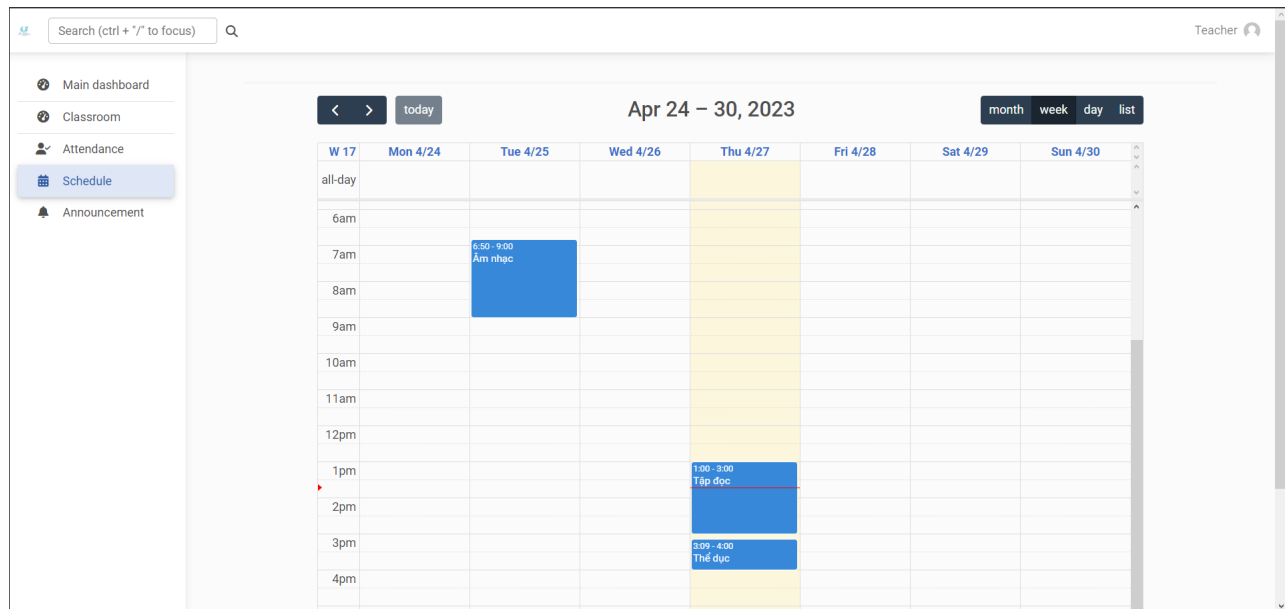


Figure 18: Teacher Schedule Interface

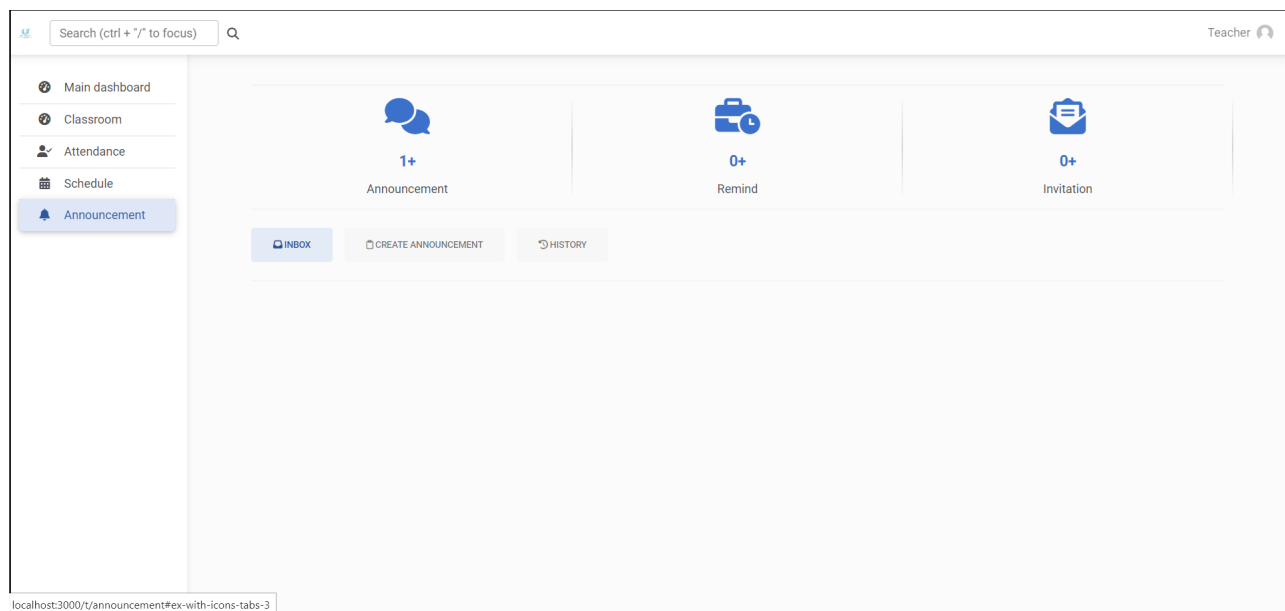


Figure 19: Teacher Inbox Interface

Search (ctrl + "/" to focus) Q Teacher

Main dashboard  
Classroom  
Attendance  
Schedule  
Announcement

1+ Announcement  
0+ Remind  
0+ Invitation

INBOX CREATE ANNOUNCEMENT HISTORY

**CREATE ANNOUNCEMENT**

Title  
Choose type  
Who can see this announcement  
Message

**SEND TO**

STUDENTS PARENTS  
SELECT ALL SELECT NONE SAVE DATA Search:

Full name	Gender	Birth	
<input type="checkbox"/> Nguyễn Bô Viên	Male	Thu, Dec 15, 2016	
<input type="checkbox"/> Nguyễn Nhật Hề	Male	Thu, Dec 15, 2016	
<input type="checkbox"/> Nguyễn Văn C	Male	Thu, Dec 15, 2016	
<input type="checkbox"/> Nguyễn Văn C	Male	Thu, Dec 15, 2016	

localhost:3000/t/announcement#ex-with-icons-tabs-1 SEND

Figure 20: Teacher Create Announcement Interface

Search (ctrl + "/" to focus) Q Teacher

Main dashboard  
Classroom  
Attendance  
Schedule  
Announcement

1+ Announcement  
0+ Remind  
0+ Invitation

INBOX CREATE ANNOUNCEMENT HISTORY

Show 10 entries Search:

Title	Type	Message	Create At	Status
Nội dung học tuần sau	Announcement	Nội dung lý thuyết:..... Căn chuẩn bi:.....	Thu, Apr 27, 2023	Sent

Showing 1 to 1 of 1 entries Previous 1 Next

localhost:3000/t/announcement#ex-with-icons-tabs-2

Figure 21: Teacher Announcement History Interface

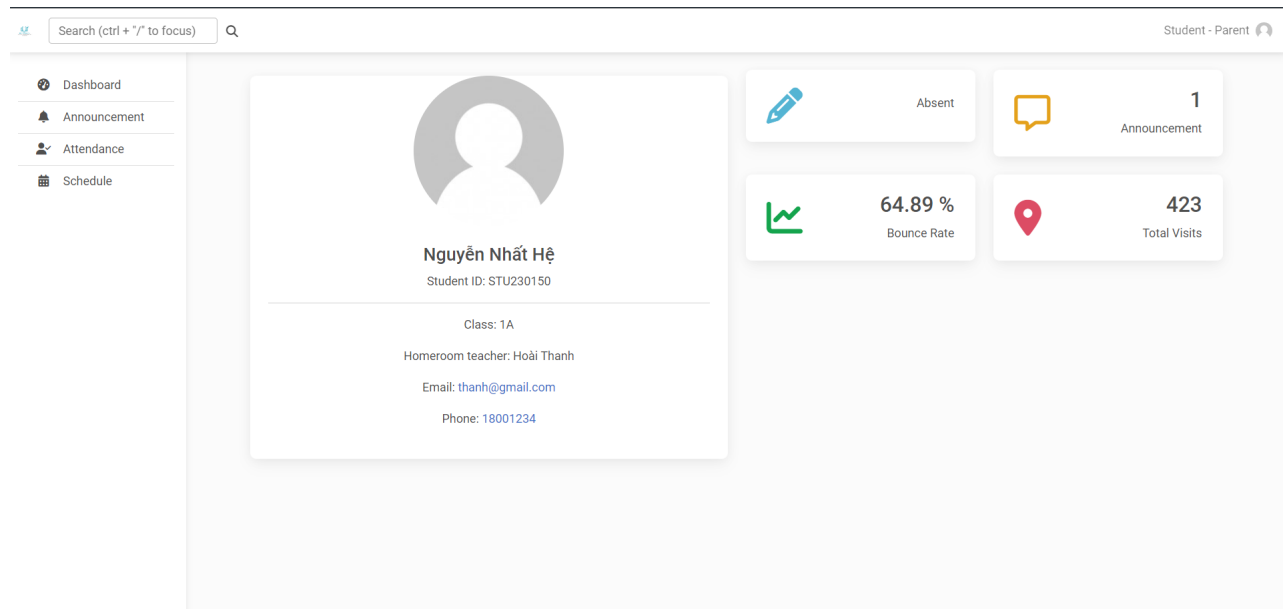


Figure 22: Parent & Student Dashboard Interface

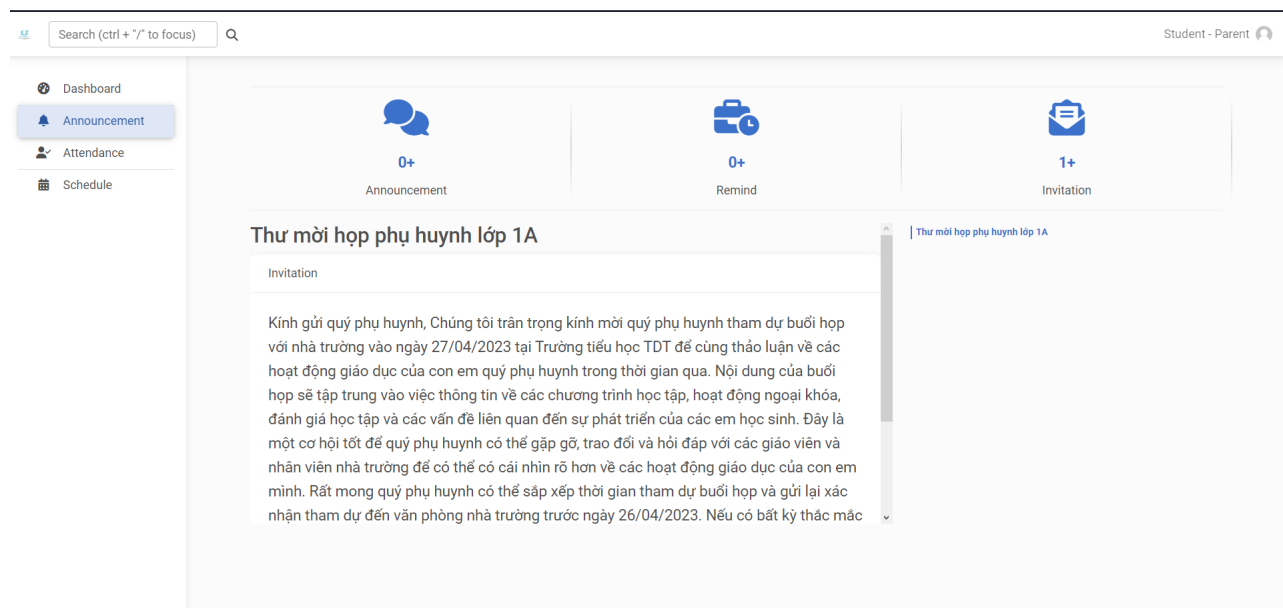


Figure 23: Parent & Student Inbox Interface



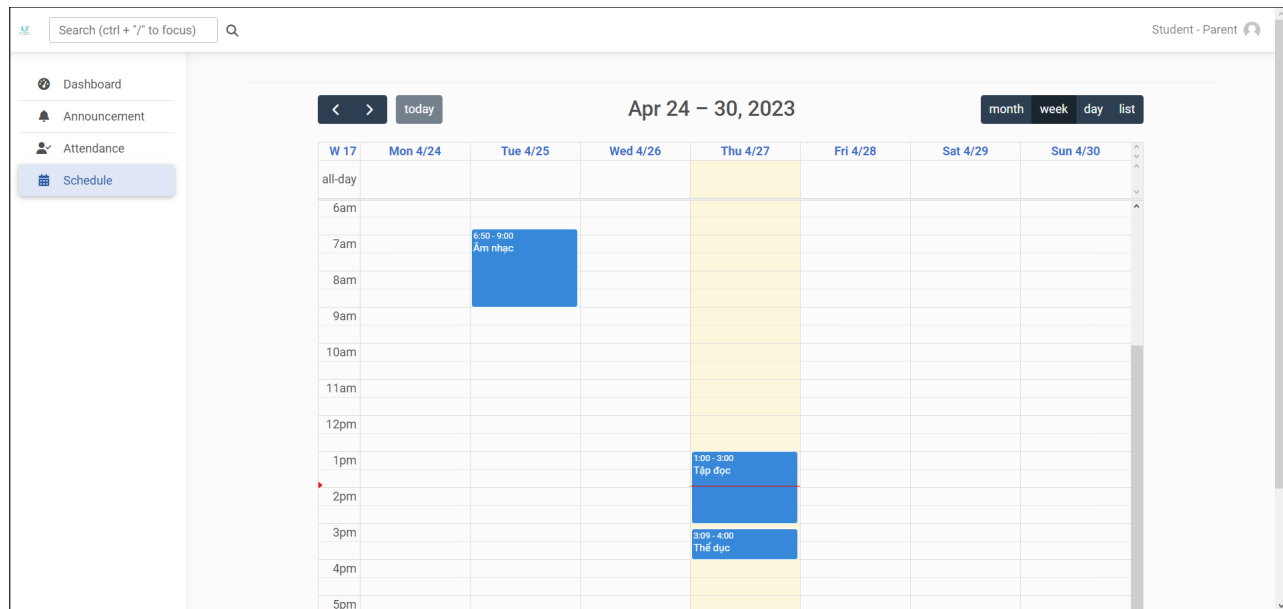


Figure 24: Parent & Student Schedule Interface

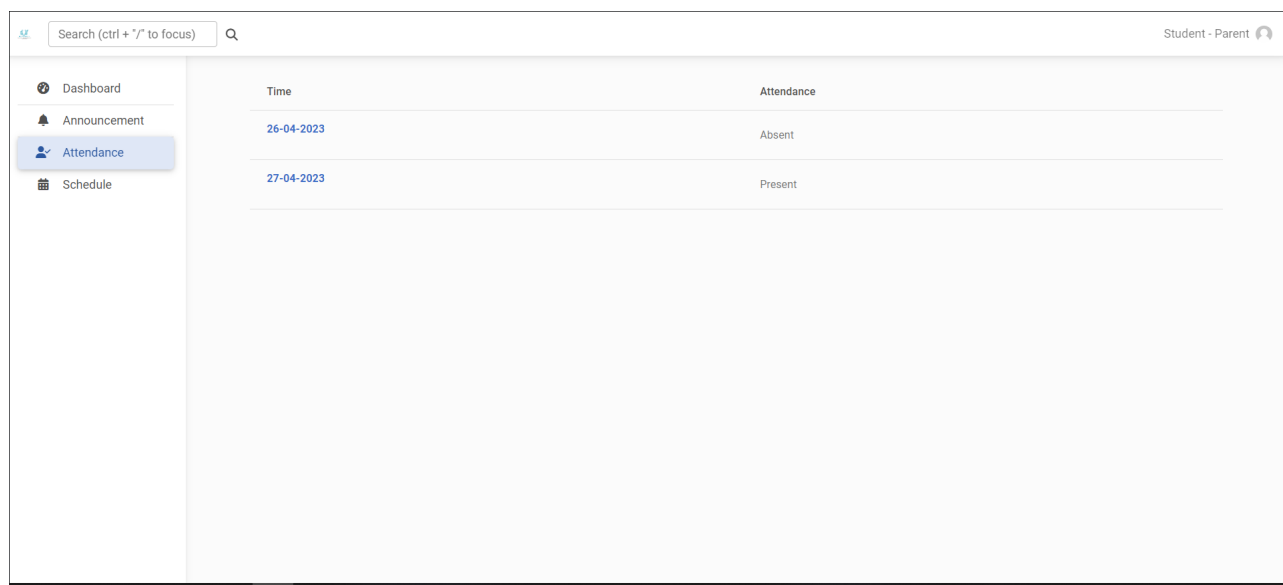


Figure 25: Parent & Student Attendance Interface

# References

- [1] Phuc H.Duong, M.Sc. **Requirement Determination & Use-Case Analysis.** Ton Duc Thang University. 2022
- [2] Phuc H.Duong, M.Sc. **Process Modeling & Data Modeling.** Ton Duc Thang University. 2022

# Appendix A-Question List Collect Request

## 4.3 Interview Questions

- On which device do you usually use the system? **Phone**
- What browser do you usually use to use the system? **Chrome**
- Maximum number of students in a class? **45**
- What functions can parents perform in this module? **view attendance, score sheet, schedule, announcement, teacher's information and modify student's information**
- What functions can student perform in this module? **view score sheet, schedule, announcement**
- which method do the parents want to receive the announcement? **Email and on system**
- What standards do primary school students' score sheet follow? **Circular 27 on assessment of primary school students in Vietnam**
- How many class period do students have in a day? **7 class period**
- What days of the week do students study? **From Monday to Friday**
- Typically, what are the types of users that need to visit the website? **parents, students, teachers, school board**
- What criteria do you usually use to search for a student? **name, class, grade**

## 4.4 Questionnaire

1. On which device do you usually use the system?
  - (a) **Smart Phone**
  - (b) Tablet
  - (c) Laptop
  - (d) PC
2. What browser do you usually use to use the system?
  - (a) **Chrome**
  - (b) Opera
  - (c) Edge
  - (d) FireFox
  - (e) Another choice
3. How do parents want to contact teachers?
  - (a) **Phone Number**
  - (b) Email
  - (c) SMS
4. Maximum number of students in a class? **45**

5. Parents want to receive notifications via?
  - (a) **Email**
  - (b) **Browser Notification**
  - (c) Another choice
6. On what factors do you usually look for students?
  - (a) **Full Name**
  - (b) Class
  - (c) Grade
  - (d) Another choice
7. What is the maximum time you feel satisfied with the system's task response rate?
  - (a) **5s**
  - (b) 2s
  - (c) 7s
  - (d) Not more than 10s
8. Regarding the password, what type of password is both convenient for customers to remember and ensures high security?
  - (a) **Minimum length of 8 characters with both numbers, special characters, and uppercase letters**
  - (b) Minimum length of 8 characters including at least one special character
  - (c) Minimum length of 8 characters with both numbers and special characters
  - (d) Minimum length of 8 characters with both numbers and uppercase letters
9. who will verify the grades graded by the teacher? **Principal**
10. What days of the week do students study? **From Monday to Friday**