

VIETNAM NATIONAL UNIVERSITY OF HOCHIMINH CITY  
THE INTERNATIONAL UNIVERSITY  
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING



**AN ADMINISTRATIVE SUPPORT SYSTEM FOR FACULTY STAFF**

By  
Nguyen Minh An – ITITIU19070

A thesis submitted to the School of Computer Science and Engineering  
in partial fulfillment of the requirements for the degree of  
Bachelor of Information Technology/Computer Science/Computer Engineering

Ho Chi Minh City, Vietnam  
2023

**Style Definition:** TOC 3: Tab stops: 0.92", Left + 6.29",  
Right, Leader: ...

## **AN ADMINISTRATIVE SUPPORT SYSTEM FOR FACULTY STAFF**

APPROVED BY:

\_\_\_\_\_,  
Huynh Kha Tu, Dr

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

THESIS COMMITTEE

## ACKNOWLEDGMENTS

First, I would like to express my sincere gratitude to Dr. Huynh Kha Tu, my guide, whose guidance, and support throughout my graduation thesis have been invaluable.

I am thankful to all the lecturers of the School of Computer Science at the International University for providing me with a strong foundation and valuable experiences during my studies. My heartfelt appreciation goes to my colleagues in the Computer Science Department - course K19 at the International University, who offered their encouragement and assistance during our studies and experiments. Moreover, I am grateful to my thesis reviewer and all the members of the examination and reading committee for dedicating their precious time to review my thesis report.

While I have worked diligently to complete my graduation thesis report, I acknowledge that there may still be areas that require improvement. I would greatly appreciate your recommendations and evaluations to ensure the accuracy of the report.

Lastly, I would like to extend my best wishes for good health to Dr. Huynh Kha Tu, all the lecturers and staffs at the School of Computer Science of the International University

## TABLE OF CONTENTS

<b>ACKNOWLEDGMENTS.....</b>	<b>3</b>
<b>TABLE OF CONTENTS.....</b>	<b>4</b>
<b>LIST OF FIGURES.....</b>	<b>6</b>
<b>LIST OF ABBREVIATIONS.....</b>	<b>8</b>
<b>ABSTRACT.....</b>	<b>9</b>
<b>CHAPTER 1. INTRODUCTION.....</b>	<b>10</b>
1.1. Background.....	10
1.2. Problem Statement.....	11
1.3. Scope and Objectives.....	12
1.4. Assumption and Solution.....	1344
1.5. Structure of thesis.....	1344
<b>CHAPTER 2. LITERATURE REVIEW/ TECHNOLOGIES.....</b>	<b>1544</b>
2.1. Literature review.....	1544
2.2. Technologies.....	1544
2.2.1. Overview dynamic web.....	1544
2.2.2. Front-end / Client-side.....	1645
2.2.2.1 NPM.....	1645
2.2.2.2. ReactJS with Vite.....	1746
2.2.2.3 Chart.js.....	1847
2.2.3. Back-end / Server-side.....	1948
2.2.3.1. Java Spring Boot.....	1948
2.2.3.1.1. Maven package management.....	2221
2.2.3.2. Liquibase.....	2322
2.2.4. Database.....	2423
<b>CHAPTER 3. METHODOLOGY.....</b>	<b>2624</b>
3.1. Overview.....	2624
3.2. System Design.....	2725
3.2.1. System Use Case.....	2725
3.2.2. System Sequence Diagram.....	2826
3.2.3. System Workflow.....	3129
3.2.4. Database design.....	3634
3.2.4.1. Database schema Diagram.....	3634

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Formatted: Tab stops: Not at 0.92" + 6.29"

Field Code Changed

Field Code Changed

Field Code Changed

Formatted: Tab stops: Not at 0.92" + 6.29"

Field Code Changed

Field Code Changed

Formatted: Tab stops: Not at 0.92" + 6.29"

Field Code Changed

Field Code Changed

Field Code Changed

Formatted: Tab stops: Not at 0.92" + 6.29"

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

3.2.5. User Interface design .....	3836	Formatted: Tab stops: Not at 0.92" + 6.29"
<b>CHAPTER 4. IMPLEMENT AND RESULTS.....</b>	<b>4139</b>	Field Code Changed
4.1. Setup Environment.....	4139	Field Code Changed
4.1.1. Operating System .....	4139	Formatted: Tab stops: Not at 0.92" + 6.29"
4.1.2. Required Tools .....	4139	Field Code Changed
4.1.2.1. Code editor and MySQL Workbench.....	4139	Field Code Changed
4.1.2.2. NodeJS and NPM .....	4240	Field Code Changed
4.1.2.3. ReactJS and Vite .....	4341	Field Code Changed
4.1.2.4. Maven.....	4442	Field Code Changed
4.1.2.5. Java JDK .....	4543	Field Code Changed
4.1.2.6. MySQL.....	4744	Field Code Changed
4.2. Implementations.....	4745	Field Code Changed
4.2.1. Front-end / Client-side.....	4745	Formatted: Tab stops: Not at 0.92" + 6.29"
4.2.2. Back-end / Server-side.....	5048	Field Code Changed
4.2.2.1 Source code .....	5048	Field Code Changed
4.2.2.2. Excel template format .....	5250	Field Code Changed
4.3 Result .....	5351	Field Code Changed
<b>CHAPTER 5. DISCUSSION AND EVALUATION .....</b>	<b>6362</b>	Field Code Changed
5.1. Discussion .....	6362	Field Code Changed
5.2. Evaluation .....	6362	Field Code Changed
<b>CHAPTER 6. CONCLUSION AND FUTURE WORK .....</b>	<b>6564</b>	Field Code Changed
6.1. Conclusion .....	6564	Field Code Changed
6.2. Future work .....	6564	Field Code Changed
<b>REFERENCES .....</b>	<b>6665</b>	Field Code Changed

## LIST OF FIGURES

Figure 1. REST API Structure .....	<a href="#">1645</a>
Figure 2. Dependency NPM installation in package. Json .....	<a href="#">1746</a>
Figure 3. ReactJS with Vite .....	<a href="#">1847</a>
Figure 4. Common charts that Chart.js support .....	<a href="#">1948</a>
Figure 5. JVM Architecture .....	<a href="#">2049</a>
Figure 6. Spring Framework Runtime .....	<a href="#">2120</a>
Figure 7. Spring boot overview .....	<a href="#">2120</a>
Figure 8. MVN workflow .....	<a href="#">2321</a>
Figure 9. pom.xml file structure .....	<a href="#">2322</a>
Figure 10. Liquibase configuration file structure .....	<a href="#">2423</a>
Figure 11. The application architecture .....	<a href="#">2624</a>
Figure 12. Use Case Diagram .....	<a href="#">2725</a>
Figure 13. Sequence Diagram for Adding new data .....	<a href="#">2826</a>
Figure 14. Sequence Diagram for Updating existed data .....	<a href="#">2927</a>
Figure 15. Sequence Diagram for Deleting existed data .....	<a href="#">3028</a>
Figure 16. Sequence Diagram for Importing excel file .....	<a href="#">3129</a>
Figure 17. Workflow of Adding new data .....	<a href="#">3230</a>
Figure 18. Workflow of Updating existed data .....	<a href="#">3331</a>
Figure 19. Workflow of Deleting existed data .....	<a href="#">3432</a>
Figure 20. Workflow of Importing excel file .....	<a href="#">3533</a>
Figure 21. Database schema relationship .....	<a href="#">3634</a>
Figure 22. Mockup Design for Task Management – Exam’s time Management – Teacher’s time Management .....	<a href="#">3937</a>
Figure 23. Table header of Task Management .....	<a href="#">3937</a>
Figure 24. Table header of Exam's time Management .....	<a href="#">3937</a>
Figure 25. Table header of Teacher's time Management .....	<a href="#">3937</a>
Figure 26. Mockup UI for Subject Filter .....	<a href="#">4038</a>
Figure 27. Mockup UI for Student Management .....	<a href="#">4038</a>
Figure 28. Verify installation node and npm .....	<a href="#">4341</a>
Figure 29. CLI to open VS Code .....	<a href="#">4341</a>
Figure 30. node_modules package result .....	<a href="#">4442</a>
Figure 31. Verify mvn result .....	<a href="#">4542</a>
Figure 32. Checkout maven path .....	<a href="#">4543</a>
Figure 33. Verify JDK installation .....	<a href="#">4644</a>
Figure 34. Checkout JDK path .....	<a href="#">4644</a>
Figure 35. Verification MySQL server .....	<a href="#">4745</a>
Figure 36. Front-end structure folder .....	<a href="#">4846</a>
Figure 37. Back-end structure folder .....	<a href="#">5048</a>
Figure 38. Excel template with Subject Filer .....	<a href="#">5250</a>
Figure 39. Excel template with Student tracking .....	<a href="#">5351</a>
Figure 40. Task Management UI .....	<a href="#">5351</a>
Figure 41. Adding Task form .....	<a href="#">5452</a>
Figure 42. Updating Task form .....	<a href="#">5553</a>
Figure 43. Exam time Management UI .....	<a href="#">5654</a>
Figure 44. Add Exam time form .....	<a href="#">5755</a>
Figure 45. Update Exam time form .....	<a href="#">5755</a>
Figure 46. Teacher's paid leave Management UI .....	<a href="#">5856</a>
Figure 47. Add Teacher's time form .....	<a href="#">5857</a>

Figure 48. Update Teacher's time form .....	<del>59</del> <a href="#">57</a>
Figure 49. Subject Filter UI.....	<del>60</del> <a href="#">58</a>
Figure 50. Statistic Chart for subject .....	<del>60</del> <a href="#">58</a>
Figure 51. Re-rendering chart when select other subject .....	<del>60</del> <a href="#">59</a>
Figure 52. Add new Subject form and import excel file .....	<del>61</del> <a href="#">60</a>
Figure 53. Student Management UI .....	<del>62</del> <a href="#">60</a>
Figure 54. Table filter will change when select other semester .....	<del>62</del> <a href="#">61</a>

## LIST OF ABBREVIATIONS

Abbreviation	Description
AOP	Aspect Oriented Programming
API	Application Programming Interface
CLI	Command Line Interface
CRUD	Create/Read/Update/Delete
DOM	Document Object Model
DTO	Data Transfer Object
ERD	Entity Relationship Diagram
I/O	Input / Output
JPA	Java Persistence API
JVM	Java Virtual Machine
NPM	Node Package Management
POM	Project Object Model
SSL/TLS	Secure Sockets Layer/ Transport Layer Security
UI	User Interface



## **ABSTRACT**

Currently, many managerial positions are supported by applications, including both desktop and web-based applications. At the International University, various faculties and departments have implemented applications to enhance work productivity. For instance, the student affairs department utilizes Edusoft website for managing student information and subjects, the student affairs room employs iuoss-website to announce student activities, and Blackboard serves as a platform for subject teaching management, including subject and lesson announcements as well as testing. Nevertheless, the administrative tasks handled by Computer Science department secretaries are predominantly carried out manually. Recognizing this, the objective of this thesis is to develop a web application that assists in streamlining administrative work for both the faculty and the faculty secretariat.

## CHAPTER 1. INTRODUCTION

### 1.1. Background

An administrative support system for faculty staff plays a crucial role in educational institutions by providing efficient management and support for various administrative tasks. Faculty staff members, including administrators, secretaries, and support staff, are responsible for managing tasks such as exam scheduling, teacher day-off management, student statistics per semester, and subject result statistics.

Traditionally, these administrative tasks were handled manually, leading to potential errors, inefficiencies, and time-consuming processes. Faculty staff had to rely on paper-based systems and manual calculations to track and manage important information related to exams, teacher availability, student statistics, and subject results.

Recognizing the need for a more streamlined and automated approach, educational institutions have turned to administrative support systems tailored specifically to the needs of faculty staff. These systems leverage technology to provide a centralized platform for managing and organizing essential administrative tasks, improving efficiency, accuracy, and overall productivity.

An effective administrative support system for faculty staff offers a range of features to simplify and streamline these tasks. For exam management, the system provides tools to schedule exams, assign rooms and proctors, and generate exam timetables. It helps ensure proper coordination and avoids scheduling conflicts, allowing faculty staff to focus on delivering exams smoothly.

Managing teacher day-offs becomes easier with the support system, as it provides a centralized calendar where faculty staff can request and track leave requests. The system enables administrators to efficiently manage and approve teacher absences, ensuring appropriate coverage and minimal disruption to the academic schedule.

When it comes to student statistics per semester, the administrative support system allows faculty staff to record and track information related to student enrollment, class attendance, and performance. It simplifies the process of generating comprehensive reports and statistics, enabling faculty staff to analyze and monitor student progress effectively.

Additionally, the system helps in the statistical analysis of subject results. It provides tools to input and manage grades, calculate averages, and generate reports on student performance. This data allows faculty staff to identify trends, assess teaching effectiveness, and make informed decisions to improve the learning experience.

By implementing an administrative support system tailored to these specific needs, educational institutions can significantly enhance the efficiency, accuracy, and management of tasks related to exam scheduling, teacher day-offs, student statistics per semester, and subject result statistics. Faculty staff can save time, reduce administrative burdens, and focus more on their core responsibilities of teaching and providing quality education to students.

## **1.2. Problem Statement**

In educational institutions, faculty staff members face numerous challenges in managing administrative tasks such as exam scheduling, teacher day-offs, student statistics per semester, and subject result statistics. These tasks are traditionally handled manually, leading to inefficiencies, errors, and time-consuming processes. The lack of an efficient and centralized administrative support system tailored to the specific needs of faculty staff creates several problems that hinder productivity and effectiveness.

Firstly, the manual management of exam scheduling poses difficulties in avoiding scheduling conflicts, allocating rooms and proctors, and generating accurate exam timetables. This can result in confusion among faculty staff and students, leading to disruptions in the examination process.

Secondly, the absence of a streamlined system for managing teacher day-offs results in challenges such as inefficient leave request processes, difficulties in tracking teacher availability, and potential scheduling conflicts. This can lead to increased administrative workload, inadequate coverage for classes, and disruptions in the academic calendar.

Thirdly, the manual handling of student statistics per semester, including enrollment data, class attendance, and performance records, creates challenges in generating comprehensive reports and tracking student progress effectively. Faculty staff may struggle with time-consuming data entry, calculation errors, and difficulty in analyzing trends and patterns.

Lastly, the manual compilation and analysis of subject result statistics pose challenges in accurately inputting grades, calculating averages, and generating meaningful reports. This can result in delayed access to critical information, difficulties in assessing teaching effectiveness, and inadequate data for decision-making.

Overall, the absence of an administrative support system tailored to the needs of faculty staff in managing tasks such as exam scheduling, teacher day-offs, student statistics per semester, and subject result statistics creates inefficiencies, errors, and challenges that impact the productivity and effectiveness of the faculty staff. A comprehensive and streamlined solution is required to address these problems and provide faculty staff with the necessary tools and functionalities to handle these administrative tasks efficiently and accurately.

### 1.3. Scope and Objectives

The main goal of the system for faculty staff will encompass the management of tasks related to exam scheduling, teacher day-offs, student statistics per semester, and subject result statistics. The system will provide a centralized platform for faculty staff to input, track, and manage data relevant to these tasks. It will include functionalities for generating reports, conducting statistical analysis, and facilitating communication among faculty members and administrators.

This thesis encompasses the identification of system requirements and the demonstration of user interactions within the proposed system. Additionally, the thesis showcases examples of user tasks and activities within the system, highlighting how they would interact with the various features and components. The objective is to provide a comprehensive understanding of the system requirements and user workflows to guide the development and implementation process effectively:

- **Streamline Exam Scheduling:** The system aims to automate and streamline the process of exam scheduling, ensuring efficient allocation of rooms, proctors, and exam timetables. It will help avoid scheduling conflicts and facilitate smooth administration of exams.
- **Facilitate Teacher Day-off Management:** The system will provide a user-friendly interface for faculty staff to request and track teacher day-offs. It will enable efficient leave request processes, ensure adequate coverage for classes, and minimize disruptions in the academic calendar.
- **Manage Student Statistics:** The system will allow faculty staff to record and track student-related data, including enrollment information, class attendance, and performance records. It will generate comprehensive reports and statistics to facilitate effective monitoring of student progress.

- **Analyze Subject Result Statistics:** The system will enable faculty staff to input and manage grades, calculate averages, and generate reports on subject result statistics. It will facilitate meaningful analysis of student performance, assist in assessing teaching effectiveness, and support data-driven decision-making.
- **Enhance Communication and Collaboration:** The system will include communication features to facilitate seamless collaboration among faculty members and administrators. It will enable easy sharing of information, notifications, and updates related to the managed tasks.
- **Ensure Data Security and Privacy:** The system will prioritize the security and privacy of sensitive data, including student records, teacher information, and exam results. It will implement appropriate security measures to protect data from unauthorized access and ensure compliance with data protection regulations.
- **User-Friendly Interface:** The system will provide a user-friendly interface that is intuitive and easy to navigate. It will be designed with the needs of faculty staff in mind, ensuring a positive user experience and minimizing the learning curve for adopting the system.

#### 1.4. Assumption and Solution

The assumption underlying the proposed solution is that there is a need for an administrative support system for faculty staff in order to improve efficiency, accuracy, and productivity in managing various tasks such as exam scheduling, teacher day-offs, student statistics per semester, and subject result statistics. It is assumed that manual methods of managing these tasks are time-consuming, prone to errors, and hinder overall effectiveness.

The solution to address the aforementioned challenges is to develop an integrated administrative support system specifically designed for faculty staff. This system will leverage technological advancements to streamline administrative processes, providing a centralized platform for efficient management of tasks.

#### 1.5. Structure of thesis

The structure of this thesis will be organized as follow:

- Chapter 2 introduces literature review and overview of technologies which applied in this thesis.
- All the requirements, system design, system architecture will be shown in Chapter 3.

- Chapter 4 shows how to set up environments, implementations for both Front-end and Back-end and the result.
- The discussions, comparisons and evaluations will be shown in Chapter 5.
- Finally, Chapter 6 presents the conclusion and future work of this project.

## **CHAPTER 2. LITERATURE REVIEW/ TECHNOLOGIES**

### **2.1. Literature review**

Currently, International University has several office support applications like OAA Online Request, Edusoft, IUOSS, and Blackboard. However, within the computer science department, there is a lack of a dedicated application specifically designed to manage the faculty secretary's tasks. This could potentially serve as the inaugural application aimed at providing assistance to the staff within the Faculty of Computer Science department.

### **2.2. Technologies**

#### **2.2.1. Overview dynamic web**

Dynamic websites are websites that are built using dynamic web technologies and can generate content in real-time. Unlike static websites that display the same information to all users, dynamic websites can customize content based on user interactions, database queries, and other factors.

Dynamic websites are built on the previous generation web architecture but place more emphasis on software that works at backend side (Server side).

The advantage of dynamic website:

- Web applications can easily be scalable.
- Developers easily update data.
- Works as a Functional application.
- Friendly User interface

The disadvantage of Dynamic website:

- The function that website serves.
- Technical Dept.

In this project, dynamic websites use REST API as bridge for frontend and backend.

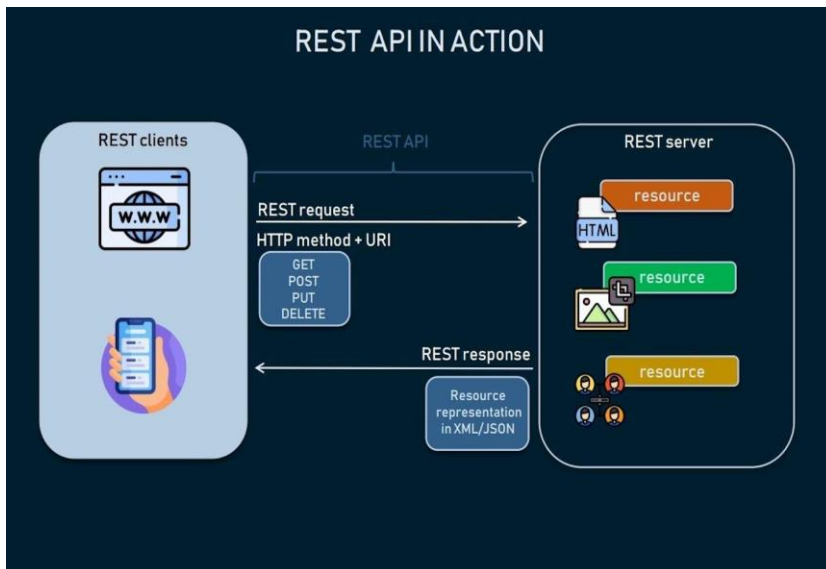


Figure 1. REST API Structure

#### 2.2.2. Front-end / Client-side

##### 2.2.2.1 NPM

NPM is a widely used package manager in the JavaScript ecosystem. It plays a vital role in managing dependencies, facilitating package installation, and enabling package distribution. With NPM, developers can easily install, update, and manage libraries and frameworks needed for their JavaScript projects.

Using NPM, we can define project dependencies in a `package.json` file, which includes information about the required packages and their versions. NPM provides commands to install dependencies, ensuring that the project has all the necessary packages to run successfully.



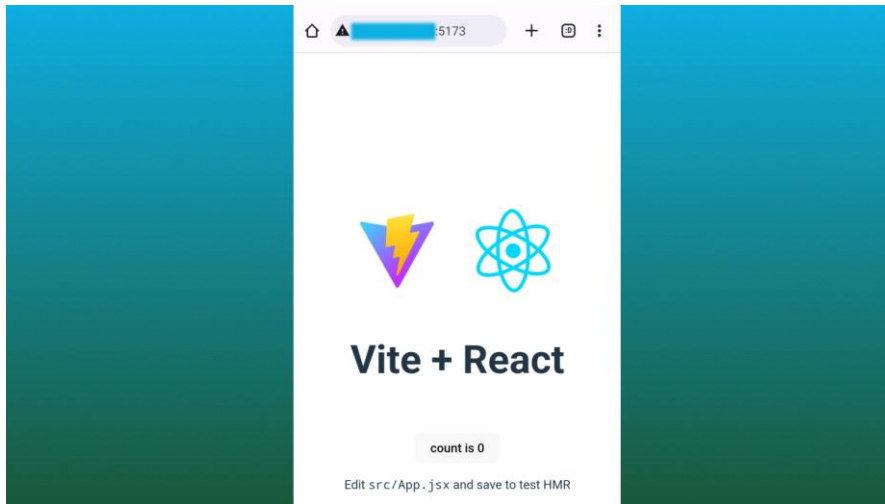
```

12  "dependencies": {
13    "@emotion/react": "^11.11.0",
14    "@emotion/styled": "^11.11.0",
15    "@mui/icons-material": "^5.11.16",
16    "@mui/material": "^5.12.3",
17    "@mui/x-data-grid": "^6.3.1",
18    "@reduxjs/toolkit": "^1.9.5",
19    "axios": "^1.4.0",
20    "chart.js": "^4.3.0",
21    "react": "^18.2.0",
22    "react-chartjs-2": "^5.2.0",
23    "react-dom": "^18.2.0",
24    "react-hook-form": "^7.43.9",
25    "react-icons": "^4.8.0",
26    "react-router-dom": "^6.11.1"
27  },
28  "devDependencies": {
29    "@types/react": "^18.0.28",
30    "@types/react-dom": "^18.0.11",
31    "@typescript-eslint/eslint-plugin": "^5.57.1",
32    "@typescript-eslint/parser": "^5.57.1",
33    "@vitejs/plugin-react": "^4.0.0",
34    "eslint": "^8.38.0",
35    "eslint-plugin-react-hooks": "^4.6.0",
36    "eslint-plugin-react-refresh": "^0.3.4",
37    "typescript": "^5.0.4",
38    "vite": "^4.3.2"
39  }

```

Figure 2. Dependency NPM installation in package. Json

#### 2.2.2.2. ReactJS with Vite



**Figure 3. ReactJS with Vite**

ReactJS with Vite is a powerful combination for building modern web applications. ReactJS is a popular JavaScript library for building user interfaces, while Vite is a fast and efficient build tool and development server. [\[6\]](#)

ReactJS provides a component-based architecture that allows developers to create reusable UI components. It offers a declarative approach to building user interfaces, making it easier to manage application state and handle UI updates efficiently. ReactJS also supports virtual DOM (Document Object Model) rendering, which improves performance by efficiently updating only the necessary parts of the UI.

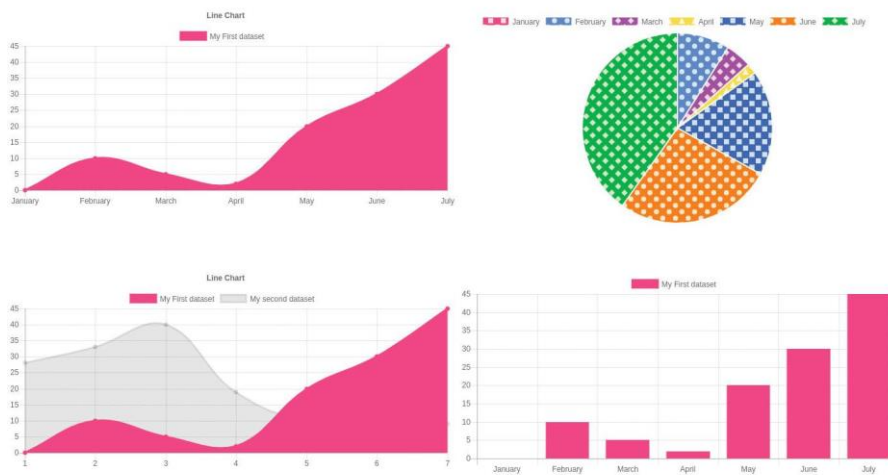
Vite, on the other hand, is a build tool specifically designed for modern JavaScript frameworks like ReactJS. It offers fast development server capabilities with instant hot module replacement (HMR), allowing developers to see changes in real-time without needing to refresh the browser. Vite also optimizes the build process by leveraging ES module imports, resulting in faster build times and improved development experience [\[5\]](#).

When combining ReactJS and Vite, we can take advantage of React's powerful UI capabilities and component-based architecture.

### **2.2.2.3 Chart.js**

Chart.js is a versatile and popular JavaScript library used for creating interactive and visually appealing charts and graphs on the web. It provides a simple and intuitive API for developers to generate various types of charts [\[7\]](#).

Chart.js provides extensive options for customizing the appearance and behavior of charts. You can easily modify colors, labels, tooltips, axes, and other visual elements to match your application's design and requirements. It also supports animations, allowing for smooth transitions and interactions when updating or interacting with the chart. Furthermore, Chart.js supports responsive for UI (The charts automatically resize and adapt to fit different screen sizes and devices) and various data integration (arrays, JSON objects, and APIs).



**Figure 4. Common charts that Chart.js support**

### 2.2.3. Back-end / Server-side

#### 2.2.3.1. Java Spring Boot

Before delving into the specifics of the Spring framework, it's helpful to understand the Java Virtual Machine (JVM) [\[4\]](#) and its role as the environment in which Spring Boot applications run. The Java Virtual Machine is a crucial component of the Java platform. It provides an execution environment for Java bytecode, which is the compiled form of Java source code. The JVM acts as an intermediary between the platform-independent Java code and the underlying hardware and operating system.

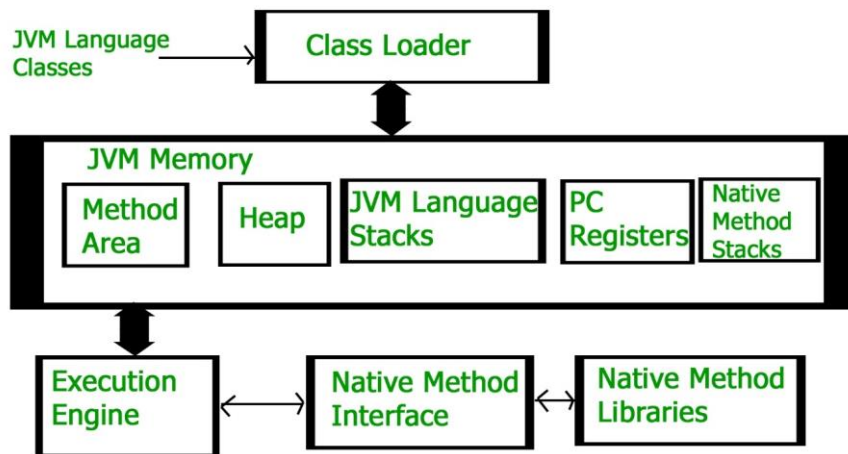
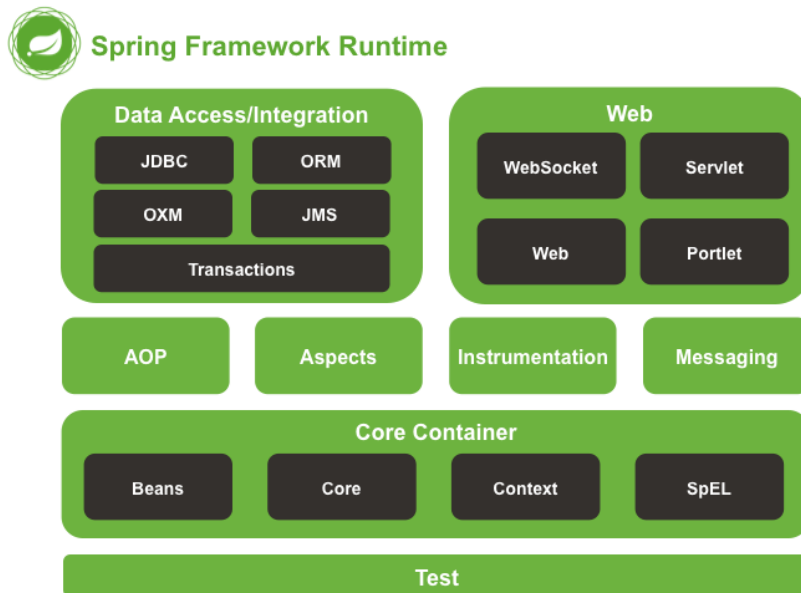


Figure 5. JVM Architecture

Spring is a widely used framework in the Java programming language that offers a wide range of functionalities. It is structured into multiple modules, with each module serving a specific purpose, such as Spring Core, Web, Data access, AOP, [Security \[2\]](#) and more. The framework is based on two essential principles: Dependency Injection and Aspect-Oriented Programming.

The disadvantage of Spring is complicated configuration:

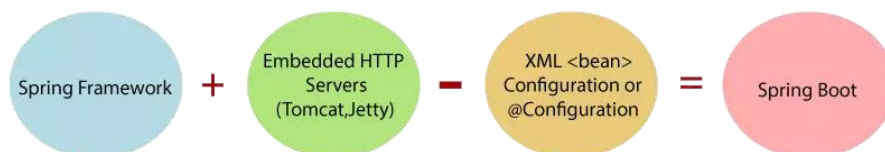
- Create a Maven or Gradle project.
- Add the necessary libraries.
- Generate XML to configure the project, configure the beans.
- Code and build to a war. File
- Configure Tomcat server to run the newly built WAR file.



**Figure 6. Spring Framework Runtime**

Spring Boot was developed as a solution to address the complexities associated with traditional Spring configuration methods [\[1\]](#):

- Auto config: automatically configures for you, just start the code, and run it. (Provide annotation).
- Build annotation-based beans instead of XML file.
- Server Tomcat is embedded right in the JAR build file, just run anywhere java can run.



**Figure 7. Spring boot overview**

### 2.2.3.1. Maven package management

MVN is an acronym for Apache Maven, which is a widely used build automation and dependency management tool in Java-based projects. Maven helps streamline the development process by providing a structured approach to managing project dependencies, compiling source code, running tests, and creating distributable artifacts [\[3\]](#).

There are some key aspects of Maven:

- **Project Structure:** Maven follows a standard project structure that promotes organization and consistency. It defines specific directories for source code, resources, test cases, and configuration files.
- **pom.xml:** Maven projects are configured using an XML file called pom.xml. The pom.xml file contains project-specific information, such as project dependencies, build plugins, repositories, and other project configurations.
- **Dependency Management:** Maven simplifies dependency management by allowing you to declare project dependencies in the pom.xml file. Maven retrieves these dependencies from remote repositories and automatically resolves transitive dependencies, ensuring that the required libraries are available for your project.
- **Plugins:** Maven provides a rich ecosystem of plugins that extend its functionality. Plugins can be used to perform various tasks, such as code analysis, generating documentation, running tests, and deploying artifacts to repositories. Plugins can be configured in the pom.xml file to automate these tasks during the build process.

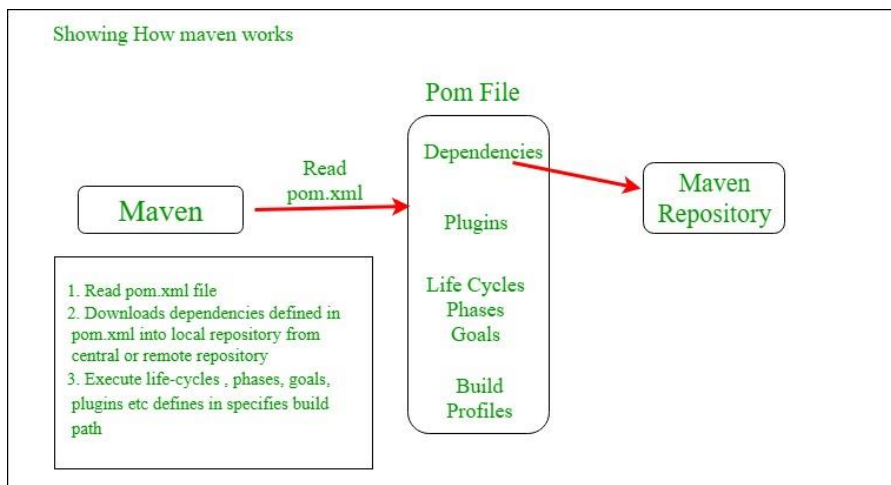


Figure 8. MVN workflow

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.5.6</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.api</groupId>
  <artifactId>libapi</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>libapi</name>
  <description>libapi</description>
  <properties>
    <java.version>11</java.version>
  </properties>
  <dependencies>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
  </dependencies>
</project>
  
```

Figure 9. pom.xml file structure

### 2.2.3.2. Liquibase

As this is a new project, it is expected that the database structure may undergo frequent changes. To handle these changes efficiently, I opted to utilize Liquibase for managing database migrations.

Liquibase is an open-source database migration tool that allows developers to manage and automate database schema changes over time. It provides a way to version control and track database changes, making it easier to collaborate on database development and deploy changes across different environments.

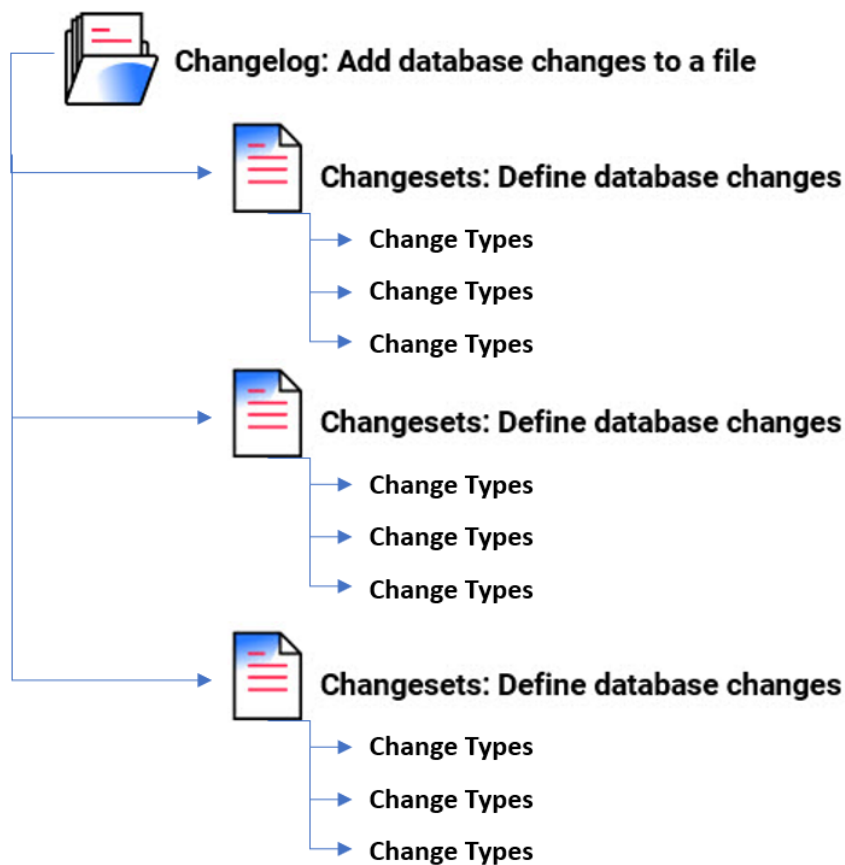


Figure 10. Liquibase configuration file structure

#### 2.2.4. Database

MySQL is an open-source relational database management system (RDBMS) that is widely used for storing and managing structured data. It is one of the most popular databases in the world due to its ease of use, performance, reliability, and extensive community support.

MySQL is designed to handle large amounts of data and supports high concurrency. It provides various techniques like indexing, query optimization, and caching mechanisms to improve performance and ensure efficient data retrieval.



MySQL offers various security features, including user authentication, access control, and data encryption. It supports secure connections using SSL/TLS protocols, ensuring data privacy and integrity.

## CHAPTER 3. METHODOLOGY

### 3.1. Overview

As mentioned earlier, the system has been developed to cater to the requirements of faculty and staff. It incorporates CRUD functionality to facilitate five key management tasks related to users: task management, exam time management, teacher's day off management, subject results management, and student management.

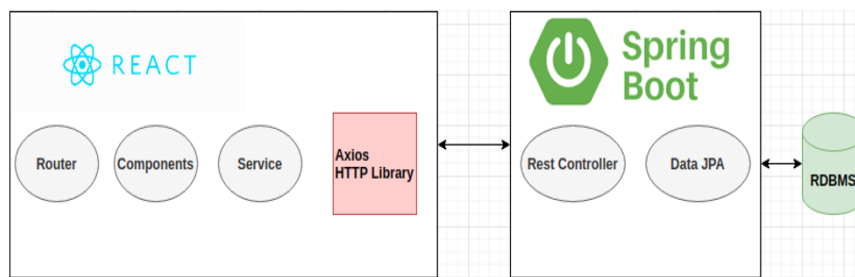
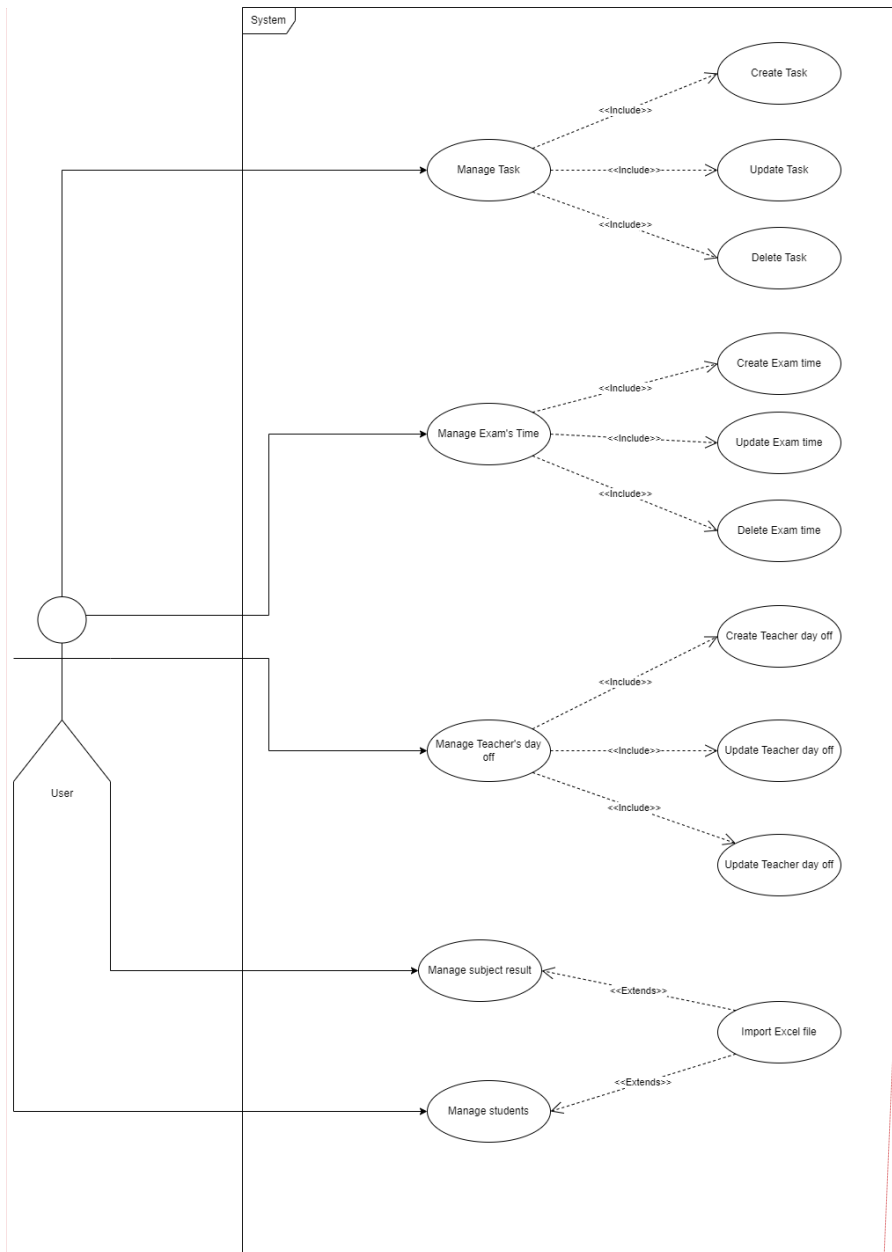


Figure 11. The application architecture

## 3.2. System Design

### 3.2.1. System Use Case



**Commented [A1]:** Em nên canh cho hình nằm trọng vào trang nhé.

Figure 12. Use Case Diagram

### 3.2.2. System Sequence Diagram

The figures below will show the Sequence Diagram for functionalities of this thesis paper.

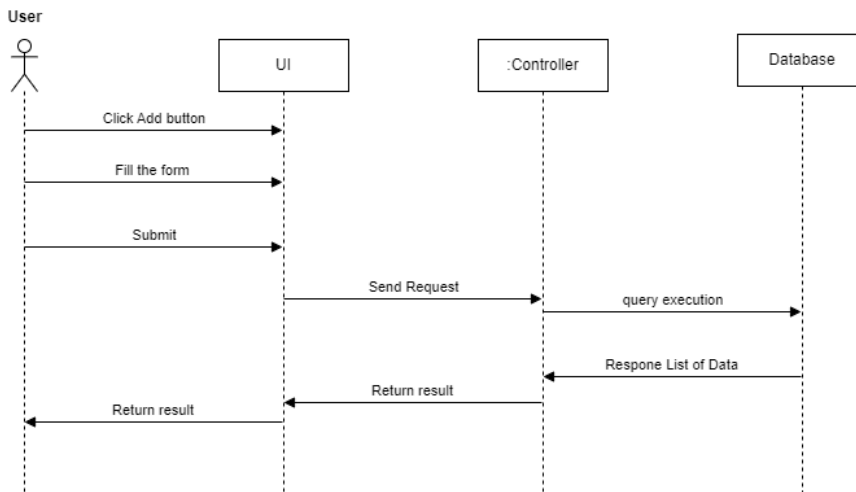


Figure 13. Sequence Diagram for Adding new data

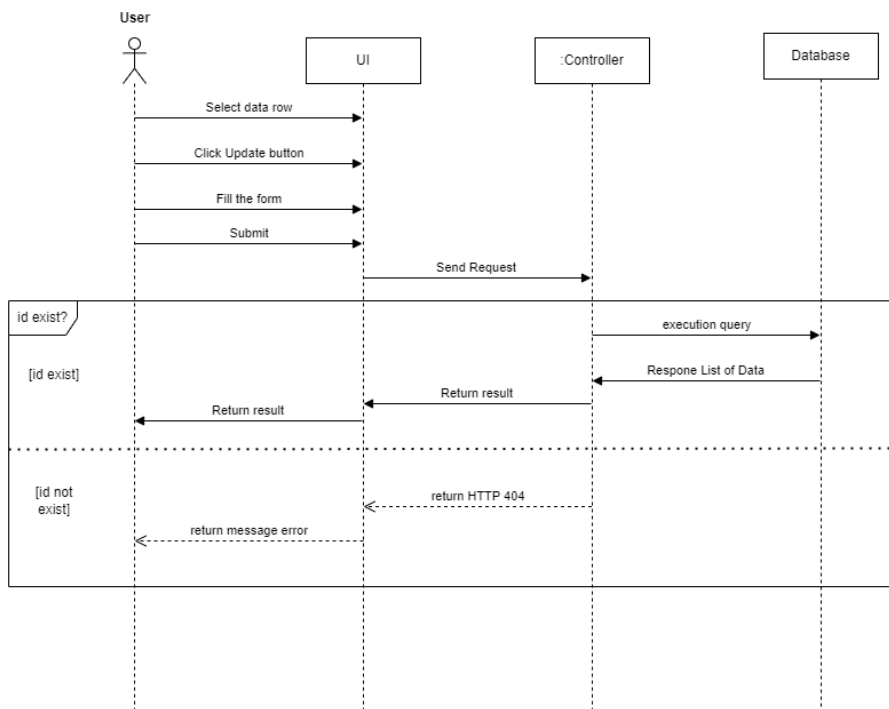


Figure 14. Sequence Diagram for Updating existed data

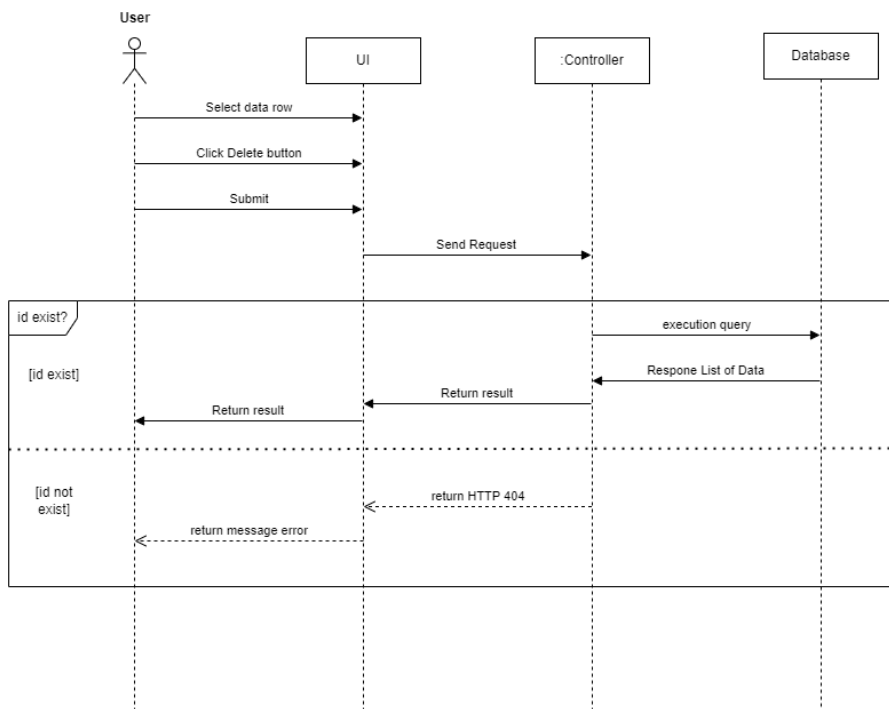


Figure 15. Sequence Diagram for Deleting existed data

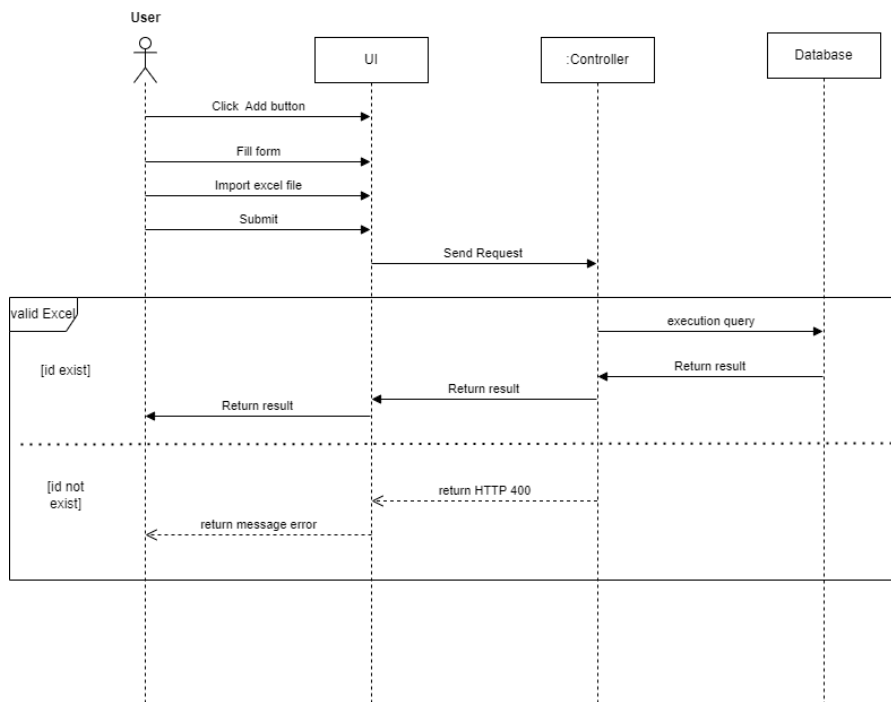
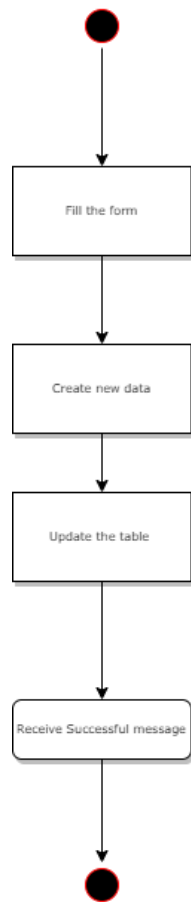


Figure 16. Sequence Diagram for Importing excel file

### 3.2.3. System Workflow

The figures below will show the main function workflow of this thesis project [\[12\]](#).

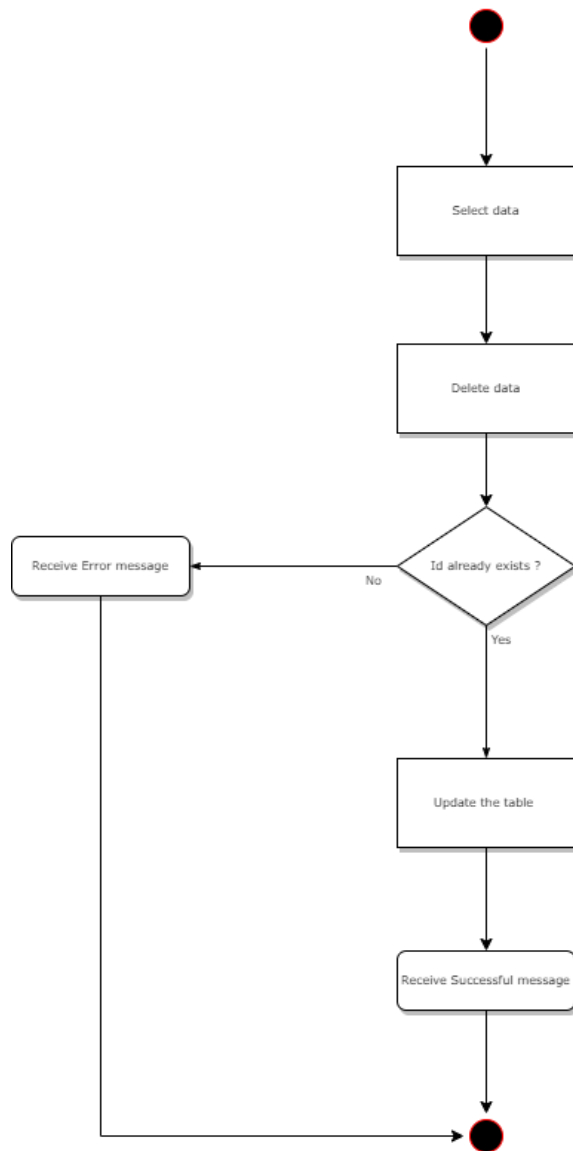


**Figure 17. Workflow of Adding new data**

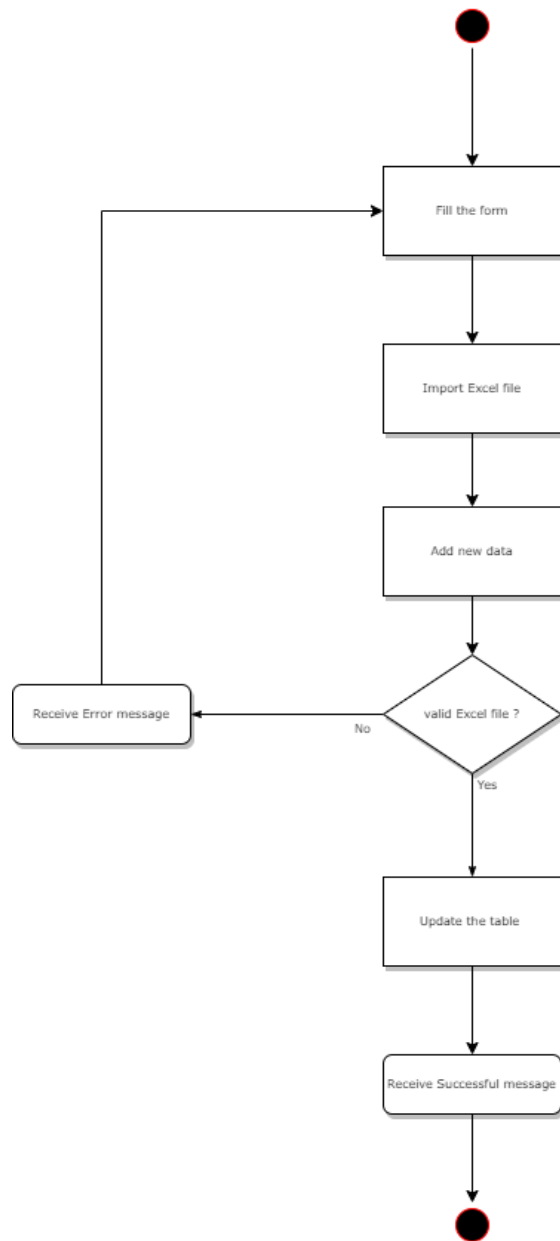




**Figure 18. Workflow of Updating existed data**



**Figure 19. Workflow of Deleting existed data**



**Figure 20. Workflow of Importing excel file**

### 3.2.4. Database design

#### 3.2.4.1. Database schema Diagram

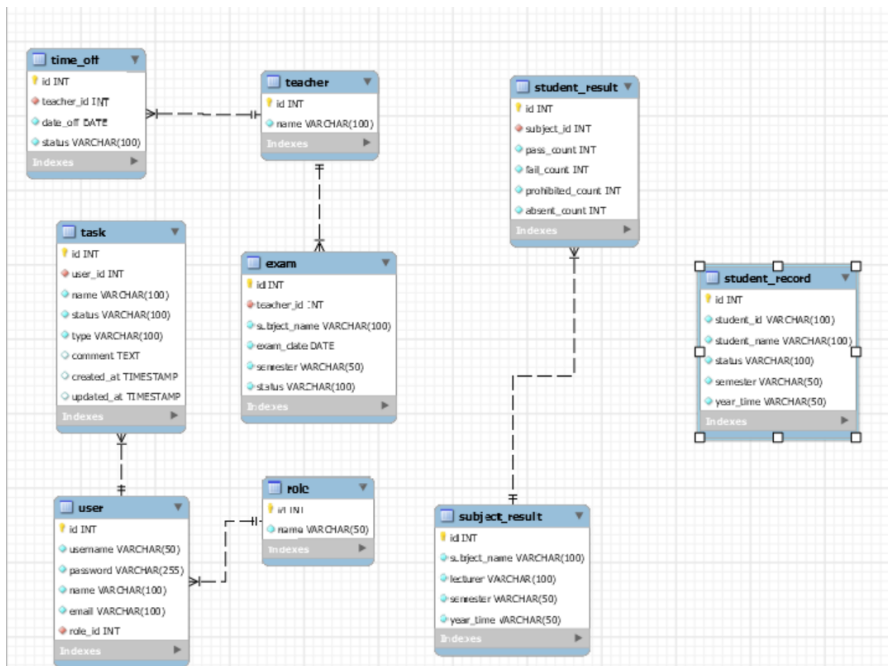


Figure 21. Database schema relationship

This are description for this schema diagram:

Table: role

Fields:

- id: An auto-incremented integer representing the unique identifier for a role.
- name: A string field storing the name of the role.

Table: user

Fields:

- id: An auto-incremented integer representing the unique identifier for a user.
- username: A string field storing the username of the user.
- password: A string field storing the password of the user. name: A string field storing the name of the user.

- email: A string field storing the email address of the user. role\_id: An integer field referencing the role id from the 'role' table.

Table: task

Fields:

- id: An auto-incremented integer representing the unique identifier for a task.
- user\_id: An integer field referencing the user id from the 'user' table.
- name: A string field storing the name of the task.
- status: An enumeration field indicating the status of the task (Done, In Progress, TODO, or Pending).
- type: An enumeration field indicating the type of task (In-going or Out-going).
- comment: A text field for additional comments on the task.
- created\_at: A timestamp field representing the creation timestamp of the task.
- updated\_at: A timestamp field representing the last update timestamp of the task.

Table: teacher

Fields:

- id: An auto-incremented integer representing the unique identifier for a teacher.
- name: A string field storing the name of the teacher.

Table: exam

Fields:

- id: An auto-incremented integer representing the unique identifier for an exam.
- teacher\_id: An integer field referencing the teacher id from the 'teacher' table.
- subject\_name: A string field storing the name of the subject for the exam.
- exam\_date: A date field representing the date of the exam.
- semester: A string field indicating the semester of the exam.
- status: An enumeration field indicating the status of the exam (Done, In Progress, TODO, or Pending).

Table: time\_off

Fields:

- id: An auto-incremented integer representing the unique identifier for a time off request.

- `teacher_id`: An integer field referencing the teacher id from the 'teacher' table.
- `date_off`: A date field representing the requested date off.
- `status`: An enumeration field indicating the status of the time off request (Done, In Progress, TODO, or Pending).

Table: `subject_result`

Fields:

- `id`: An auto-incremented integer representing the unique identifier for a subject result.
- `subject_name`: A string field storing the name of the subject.
- `lecturer`: A string field storing the name of the lecturer for the subject.
- `semester`: A string field indicating the semester of the subject.
- `year_time`: A string field indicating the year of the subject.

Table: `student_result`

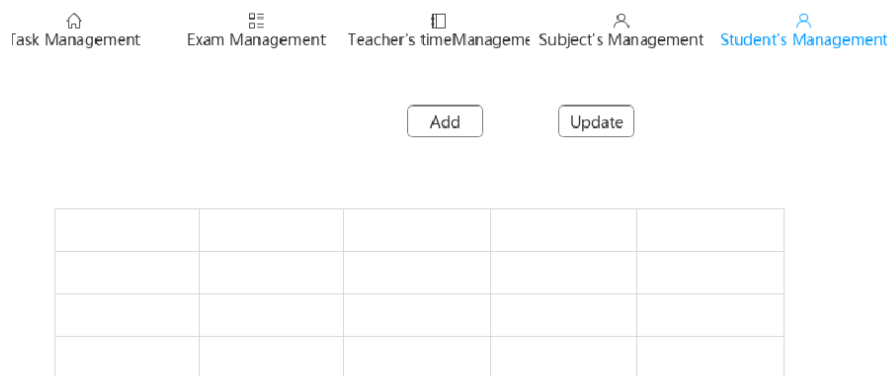
Fields:

- `id`: An auto-incremented integer representing the unique identifier for a student result.
- `subject_id`: An integer field referencing the subject id from the 'subject\_result' table.
- `pass_count`: An integer field representing the count of students who passed the subject.
- `fail_count`: An integer field representing the count of students who failed the subject.
- `prohibited_count`: An integer field representing the count of students with prohibited status.
- `absent_count`: An integer field

### 3.2.5. User Interface design

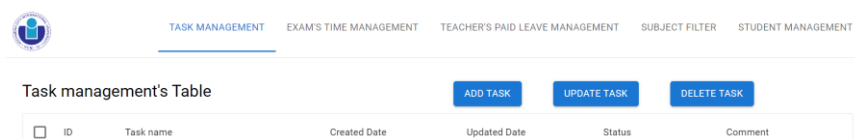
User interface design refers to the process of creating visually appealing and user-friendly interfaces for software applications or websites. It involves designing the layout, navigation, and visual elements that users interact with, aiming to provide an intuitive and engaging user experience. For this project, I applied Single Page Application with 5 tabs so that I can easily handle state when implementing and helpful with users.

With 3 tabs: Task Management – Exam's time Management – Teacher's time Management, all of them have the same business goals so the mockup design is the same.



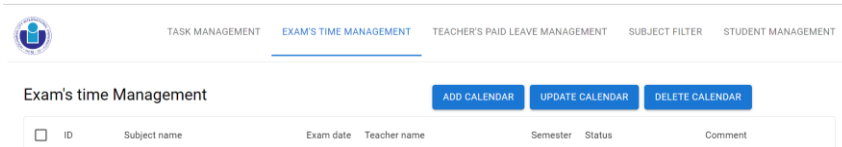
**Figure 22. Mockup Design for Task Management – Exam's time Management – Teacher's time Management**

- With Task Management the table will have these value:



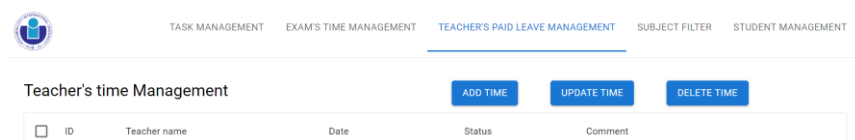
**Figure 23. Table header of Task Management**

- With Exam's time Management the table will have these value:



**Figure 24. Table header of Exam's time Management**

- With Teacher's time Management the table will have these value:



**Figure 25. Table header of Teacher's time Management**

With the Subject filter, it will have import excel file and rendering table with chart. The details of logic I will explain in Chapter 4.



Figure 26. Mockup UI for Subject Filter

With Student Management, it will import excel files and render 2 table. The details of logic I will explain in Chapter 4.

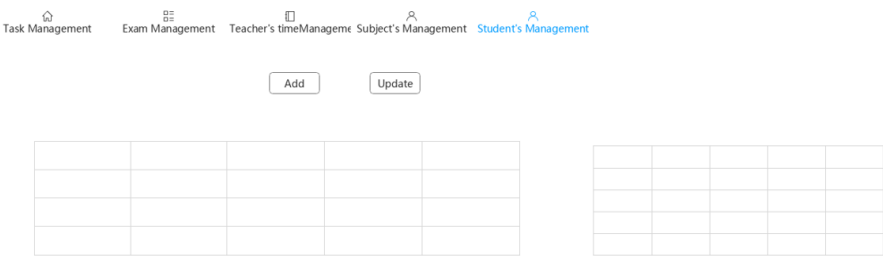


Figure 27. Mockup UI for Student Management



## CHAPTER 4. IMPLEMENT AND RESULTS

### 4.1. Setup Environment

#### 4.1.1. Operating System

The web back-end and front-end are implemented and run in a Windows 11 Home Single Language 64-bit Operating System. This system is applicable to all the Operating System.

#### 4.1.2. Required Tools

##### 4.1.2.1. Code editor and MySQL Workbench

Front-end side: Visual Studio Code

- Visit the official VS Code website: Go to the official Visual Studio Code website at <https://code.visualstudio.com/>
- Choose the appropriate installer: On the VS Code website, you will see download links for different operating systems (Windows, macOS, and Linux). Click on the download link that corresponds to your operating system.
- Download the installer: The file size may vary depending on your operating system.
- Run the installer: Locate the downloaded installer file on your computer and run it.
- Install recommended extensions: Visual Studio Code provides a wide range of extensions that enhance its functionality. Click on the Extensions icon in the sidebar, search for extensions, and click the "Install" button to add them to your VS Code installation.

Back-end side: IntelliJ IDEA Community Edition

- Visit the JetBrains website: Go to the JetBrains website at <https://www.jetbrains.com/idea/download/>
- Choose the IntelliJ IDEA edition: On the IntelliJ IDEA download page, you will find different editions available for download. Make sure to select the "Community" edition, as it is the free version.
- Choose the operating system: Select your operating system from the list provided. IntelliJ IDEA supports Windows, macOS, and Linux. Choose the appropriate version for your system.

- Download the IntelliJ IDEA installer: Click on the download link for the IntelliJ IDEA Community Edition installer for your operating system. The file size may vary depending on your system and version.
- Run the installer: Locate the downloaded installer file on your computer and run it.
- Configure IntelliJ IDEA: Upon opening IntelliJ IDEA for the first time, you will be prompted to customize the settings according to your preferences.

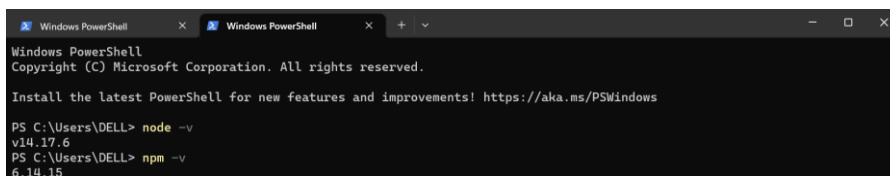
#### Database: MySQL Workbench

- Visit the official MySQL website: Go to the official MySQL website at <https://dev.mysql.com/downloads/workbench/>
- Choose the operating system: On the MySQL Workbench download page, select the appropriate version for your operating system (Windows, macOS, or Linux).
- Download MySQL Workbench: The file size may vary depending on your operating system.
- Run the MySQL Workbench installer: Locate the downloaded installer file on your computer and run it.
- Configure MySQL Workbench: After the installation is complete, you can open MySQL Workbench.
- Connect to MySQL server: Once MySQL Workbench is launched, you can establish a connection to your MySQL server. Click on the "New Connection" button and provide the necessary details such as hostname, port, username, and password.

#### 4.1.2.2. NodeJS and NPM

- Visit the official Node.js website: Go to the official Node.js website at <https://nodejs.org/>
- Choose the Node.js version: On the Node.js website, you will see two versions available for download: LTS (Long-Term Support) and Current.
- Choose the operating system: On the download page, you will find different installers based on your operating system.
- Download the Node.js installer: Click on the download link for the Node.js installer that corresponds to your operating system.

- Run the Node.js installer: Locate the downloaded installer file on your computer and run it.
- Verify the installation: After the installation is complete, you can verify that Node.js is installed correctly. Open a terminal or command prompt and run the following commands to check the installed version of Node.js and npm:
  - To check the Node.js version, run: `node -v`
  - To check the npm version, run: `npm -v`



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

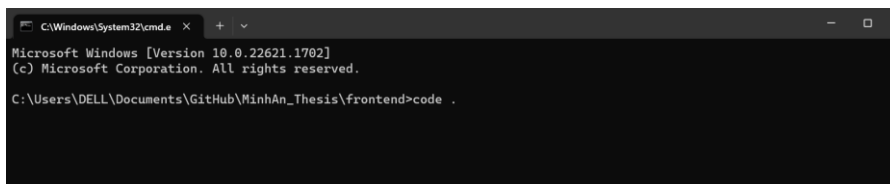
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\DELL> node -v
v14.17.6
PS C:\Users\DELL> npm -v
6.14.15
```

Figure 28. Verify installation node and npm

#### 4.1.2.3. ReactJS and Vite

- Apply cli ``cd`` to the front-end project folder and type ``code .`` to open VS Code.



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL\Documents\GitHub\MinhAn_Thesis\frontend>code .
```

Figure 29. CLI to open VS Code

- Run ``npm install`` to download all the dependencies and packages automatically. After downloading you will see `node_modules` folder in your source code. It means you install successfully ReactJS and Vite.

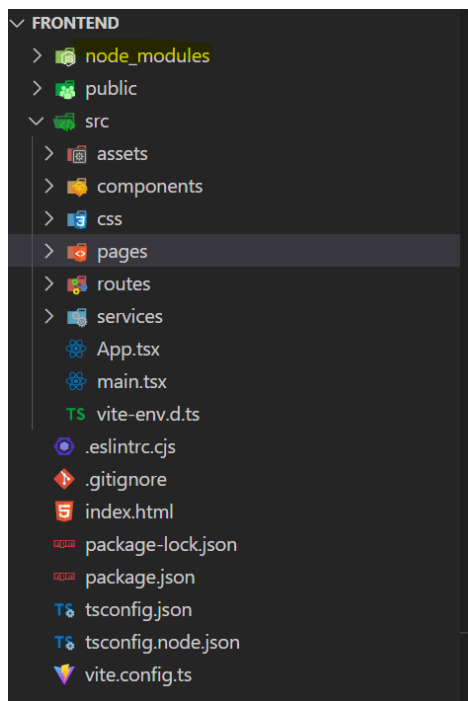


Figure 30. node\_modules package result

#### 4.1.2.4. Maven

- Visit the Apache Maven website: Go to the Apache Maven website at <https://maven.apache.org/>
- Download the Maven distribution.
- Choose the Maven distribution: Select the distribution package that suits your operating system.
- Extract the Maven distribution: Once the download is complete, locate the downloaded distribution file on your computer and extract its contents.
- Set up the environment variables: To use Maven from the command line, you need to set up the necessary environment variables.
- Verify the installation: Open a new terminal or command prompt and run the following command: ``mvn -v``. You should see the installed Maven version information if the installation was successful.

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\DELL> mvn -v
Apache Maven 3.9.0 (9b58d2bad23a66be161c4664ef21ce219c2c8584)
Maven home: C:\apache-maven-3.9.0
Java version: 11.0.8, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-11.0.8
Default locale: en_US, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"
PS C:\Users\DELL>

```

Figure 31. Verify mvn result

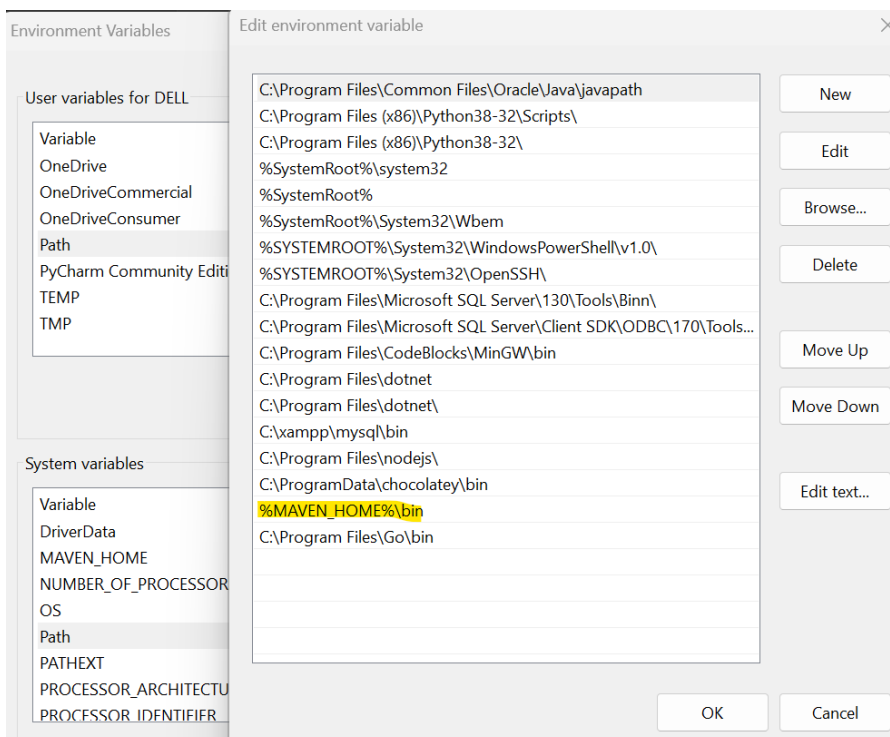


Figure 32. Checkout maven path

#### 4.1.2.5. Java JDK

- Visit the Oracle website: Go to the Oracle website's Java SE Downloads page at <https://www.oracle.com/java/technologies/downloads/>
- Choose the JDK package: Look for the package suitable for your operating system.
- Download the JDK
- Run the JDK installer: Locate the downloaded installer file on your computer and run it.

- Set the PATH environment variable: After the installation is complete, you need to set up the PATH environment variable so that your operating system can find the Java executables. The process to set the PATH variable differs depending on your operating system.
- Verify the installation: Open a new terminal or command prompt and run the following command: `java -version`. You should see the installed JDK version information if the installation was successful.

```

Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL>java -version
java version "11.0.8" 2020-07-14 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.8+10-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.8+10-LTS, mixed mode)

C:\Users\DELL>

```

Figure 33. Verify JDK installation

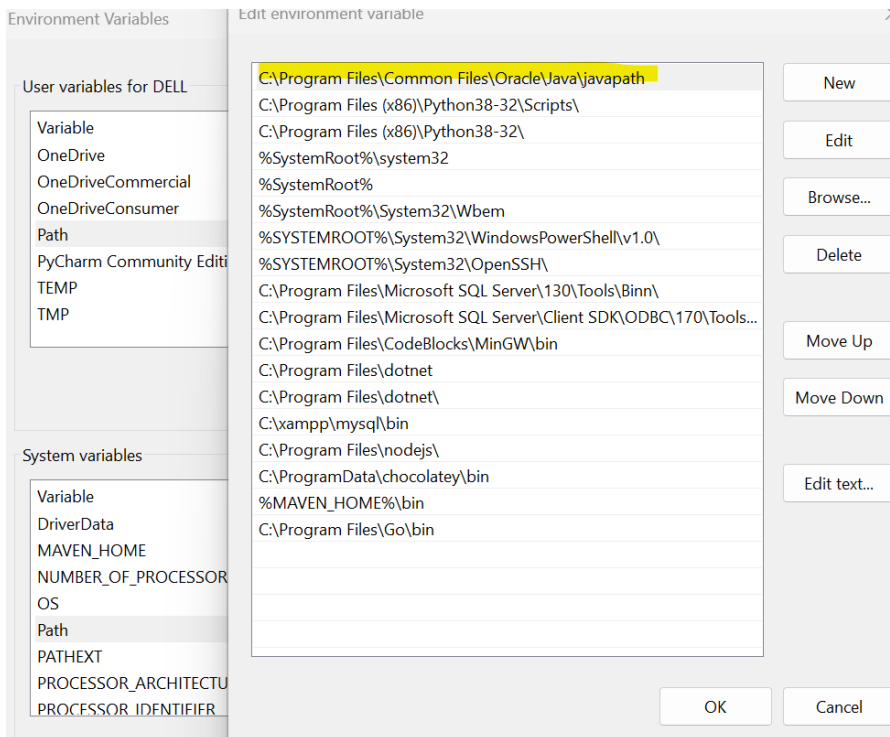
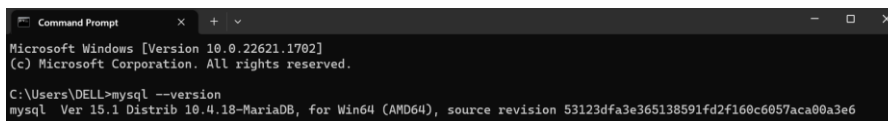


Figure 34. Checkout JDK path

#### 4.1.2.6. MySQL

- A Visit the official MySQL website: Go to the official MySQL website at <https://dev.mysql.com/downloads/workbench/>
- Choose the MySQL edition: On the MySQL, you can choose the "MySQL Community Server," which is the open-source version.
- . Accept the License Agreement by clicking on the corresponding checkbox.
- Choose the operating system: Choose the appropriate version for your system.
- Download the MySQL installer: Click on the download link for the MySQL installer for your operating system.
- Run the MySQL installer: Locate the downloaded installer file on your computer and run it.
- Start MySQL server: After the installation is complete, you can start the MySQL server.
- Verify the installation: Open a terminal or command prompt and execute the command ``mysql -v`` or ``mysql --version``. You should see the installed MySQL version information if the installation was successful.



```
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL>mysql --version
mysql Ver 15.1 Distrib 10.4.18-MariaDB, for Win64 (AMD64), source revision 53123dfa3e365138591fd2f160c6057aca00a3e6
```

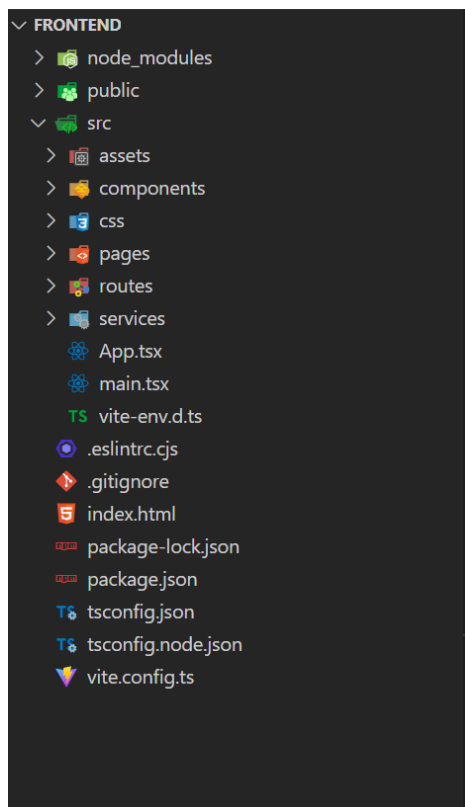
Figure 35. Verification MySQL server

## 4.2. Implementations

In this section, I will provide readers with an overview of the code structure and the functionality of each module or component mentioned in this thesis paper. While I won't delve into the detailed source code explanations, I aim to give a high-level understanding of how the system is organized and the purpose of each module or component [\[11\]](#).

### 4.2.1. Front-end / Client-side

To begin with, the figure below shows the front-end structure.



**Figure 36. Front-end structure folder**

In this project structure, aside from the configuration files automatically generated by ReactJS and Vite, the remaining folders serve specific purposes. Here is a brief explanation of the mission of each folder [\[8\]](#):

- `src/components`: This folder contains reusable UI components that can be used throughout the application. These components encapsulate specific functionality and can be composed together to build complex user interfaces.
- `src/pages`: The pages folder contains individual page components that represent different routes or views in the application. Each page component typically corresponds to a specific URL or route and acts as a container for organizing related components and managing the page's logic.



- `src/assets`: This folder is used to store static assets such as images, fonts, or other media files that are required by the application.
- `src/css`: Here, you can find CSS or SCSS files that define the styling and visual presentation of the application. This includes global styles, utility classes, or custom styles specific to certain components or pages.
- `src/services`: This folder may contain service modules that handle communication with external APIs, perform data fetching, or manage state-related operations. These services encapsulate the logic for interacting with external resources and provide a clean abstraction layer for data handling.
- `src/routes`: In this folder, you may find route configuration files or modules that define the application's routing logic. These files specify the mapping between URLs and corresponding page components, allowing the application to navigate between different views based on the user's interactions.

## 4.2.2. Back-end / Server-side

### 4.2.2.1 Source code

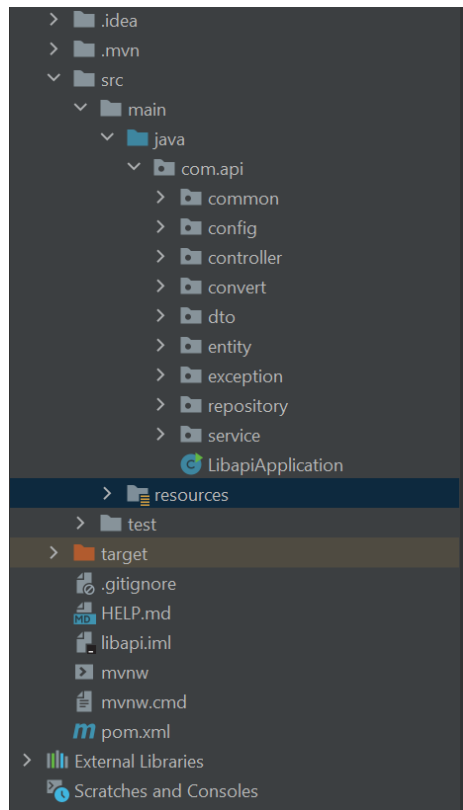


Figure 37. Back-end structure folder

With this project structure, I will provide an explanation for the purpose of each of the following folders [\[9\]](#):

- **common**: This folder is intended for storing utility classes, helper methods, or shared constants that are utilized across different components or modules within the application. It serves as a central location to store commonly used functionality for easy access and reuse.
- **config**: The `"config"` folder is dedicated to housing configuration files that specify settings, properties, or behavior for various components or modules within the application. It contains files that define the configuration for databases, security, logging, and other third-party integrations.

- **dto:** The `"dto"` folder is specifically designated for storing Data Transfer Object (DTO) classes. These classes act as plain Java objects used for transferring data between different layers or components of the application. DTOs provide a standardized format for exchanging data and facilitate communication between the client and server.
- **controller:** The `"controller"` folder contains classes responsible for defining and handling REST API endpoints or web request handlers. These classes receive incoming requests, process the data, and generate appropriate responses to be sent back to the client. Controllers act as the entry point for request handling and typically invoke corresponding service methods to execute the required operations.
- **service:** The `"service"` folder is intended for storing service classes that encapsulate the business logic of the application. These classes provide higher-level operations and coordinate the interaction between different components, such as repositories, external APIs, or other services. Services are responsible for implementing the core functionality of the application and perform tasks like data processing, applying business rules, and executing complex operations.
- **convert:** The `"convert"` folder houses classes responsible for converting or transforming data between different representations or formats. These classes facilitate the seamless conversion and manipulation of data between various layers of the application. Common conversions include transforming entities to DTOs and vice versa, ensuring smooth data interchange.
- **entity:** The `"entity"` folder is dedicated to storing entity classes that represent the persistent data models within the application. These classes typically map to database tables or collections and encapsulate the data and behavior associated with them. Entities are often annotated with JPA (Java Persistence API) annotations to define their mappings and relationships with other entities.
- **Repository:** The `"repository"` folder is used for storing interfaces or classes that define the data access layer of the application. These classes provide methods and operations for interacting with the database or other data sources. Repositories often leverage ORM (Object-Relational Mapping) frameworks, such as Hibernate or Spring Data JPA [\[10\]](#), to simplify database operations and querying.

- exception: The "exception" folder contains classes related to exception handling and error management within the application. It may include custom exception classes, exception handlers, or global exception configuration. These classes assist in handling and processing various types of exceptions that may occur during runtime, enabling meaningful error messages and specific error-related actions.

#### 4.2.2.2. Excel template format

In this thesis paper, there are 2 tabs that import excel file, so I need to create the rule for excel template format.

- With Subject filter tabs:

	A	B	C	D	E	F	G	H
1	STT	Mã SV	In-class	Mid-term	Final	Note		
2	1	ITITI19070	100	92	96			
3	2	ITITI19081	100	90	95			
4	3	ITITI19008	100	0	50			
5	4	ITITI19094	100	0	50			
6	5	ITITI18270	70	92	81			
7	6	ITITI19097	30	92	61			
8	7	ITITI19115	100	0	50			
9	8	ITITI18043	0	0	0	Permitted		
10	9	ITITI19125	70	92	81			
11	10	ITITI20222	100	90	95			
12	11	ITITI19022	100	70	85			
13	12	ITITI19153	100	85	93			
14	13	ITITI19026	70	75	73			
15	14	ITITI19025	100	92	96			
16	15	ITITI18293	70	75	73			
17	16	ITITI19157	100	85	93			
18	17	ITITI19031	100	90	95			
19	18	ITITI19161	100	85	93			
20	19	ITITI19163	100	85	93			

Figure 38. Excel template with Subject Filter

There are 6 columns with “STT”, “Mã SV”, “In-class”, “Mid-term”, “Final”, and “Note”.

When importing an excel file I will read the “Final” column and “Note” column, depending on that I will create conditions to validate and calculate the number of PASSED, FAILED, PROHIBITED students in that course.

- With Student Management tabs

	A	B	C	D	E	F	G	H
1	STT	Name	Mã SV	Status				
2	1	A	ITTTIU19070	STUDYING				
3	2	B	ITTTIU19081	STUDYING				
4	3	C	ITTTIU19008	STUDYING				
5	4	D	ITTTIU19094	STUDYING				
6	5	E	ITTTIU18270	STUDYING				
7	6	F	ITTTIU19097	STUDYING				
8	7	G	ITTTIU19115	STUDYING				
9	8	H	ITTTIU18043	STUDYING				
10	9	I	ITTTIU19125	STUDYING				
11	10	J	ITTTIU20222	STUDYING				
12	11	K	ITTTIU19022	STUDYING				
13	12	L	ITTTIU19153	STUDYING				
14	13	M	ITTTIU19026	STUDYING				
15	14	N	ITTTIU19025	STUDYING				
16	15	O	ITTTIU18293	STUDYING				
17	16	P	ITTTIU19157	STUDYING				
18	17	Q	ITTTIU19031	STUDYING				
19	18	R	ITTTIU19161	STUDYING				

Figure 39. Excel template with Student tracking

There are 4 rows “STT”, “Name”, “Mã SV” and “Status”. After importing excel file, I will collect name, student ID, status to insert to the system at that semester. Depending on the number of statuses I will count total of each type of status and generated the chart.

### 4.3 Result

- Users use Task management to handle the status of the dispatch that they are going to update.

Task management's Table

ID	Task name	Created Date	Updated Date	Status	Comment
1	Task 1	5/15/2022		In Progress	A is doing it
2	Task 2	5/12/2022	5/14/2022	Done	B finished it
3	Task 3	5/16/2022		Pending	Suspended, waiting for A fi...
4	Task 4	5/11/2022	5/15/2022	Done	D finished it
5	Task 5	5/15/2022		In Progress	E is doing it

1 - 5 of 5

Figure 40. Task Management UI

This thesis project is designed as a Single Page Application (SPA) with multiple tabs, the first of which is the Task Management tab. Upon accessing this tab, users will immediately view all existing tasks stored in the database.

To facilitate CRUD functionality, the interface includes Add, Update, and Delete buttons. These buttons enable users to create new tasks, update existing ones, and remove tasks from the system.

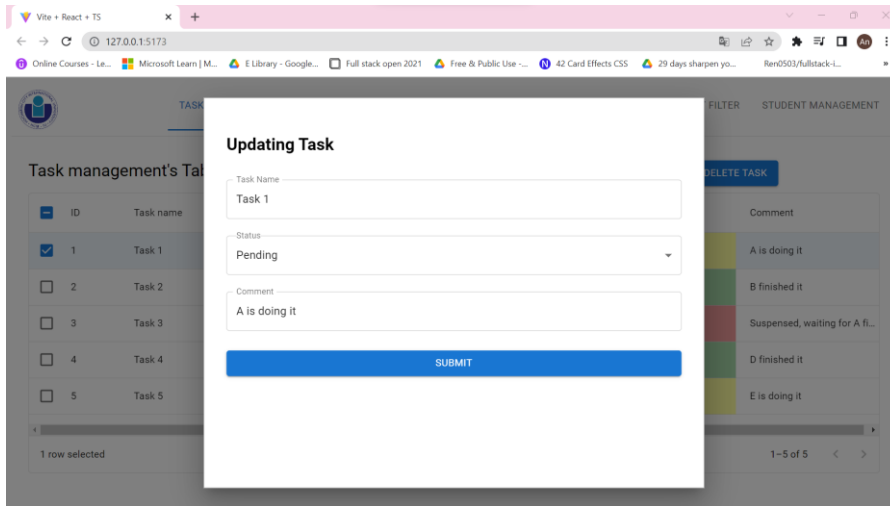
The table within the Task Management tab displays various details related to each task, including the task index, task name, creation date, last update date, status, and comments. This tab provides users with an organized and comprehensive view of the tasks, allowing them to easily track and manage task-related information.

**Figure 41. Adding Task form**

When adding a new task, users will be required to fill out a form that includes the task name field. They can select the status of the task from available options, and there is also an optional field to add additional comments or notes related to the task.

The values for the "created at" and "updated at" fields will be automatically generated on the back-end side. These timestamps will be added to the TaskDTO object, which is then sent to the front-end for display purposes. This ensures that the front-end accurately reflects the creation and last update times of the task.

By automatically generating the timestamps and including them in the TaskDTO, users will have visibility into when each task was created and last updated, providing valuable information for tracking the lifecycle of tasks within the system.



**Figure 42. Updating Task form**

When users select a row in the task table and click on the Update button to modify the task, the value of the "created at" field will remain unchanged. This is because the creation timestamp represents the original creation time of the task and should not be altered during updates.

However, the "updated at" field will be regenerated on the back-end side when the task is updated. This ensures that the "updated at" timestamp accurately reflects the most recent modification of the task. By regenerating the timestamp upon updating, users can easily track when the task was last modified, providing them with up-to-date information on its status.

By preserving the original "created at" value and generating a new "updated at" value upon each update, the system maintains a clear distinction between the initial creation of the task and subsequent modifications, enabling users to track the timeline of changes made to the task.

Same with Update function, when users select a row in the task table and click on the Delete button, it will delete the record from UI and database in the back-end.

- Users interact with Exam management to handle the status of the exam time that they are going to update.

Functionalities are the same as Task Management, so in this case I will explain the purpose of this tab.

The users will see the list of exam time which front-end call API to get from database. From the details, user can know the status of the exam in that day.

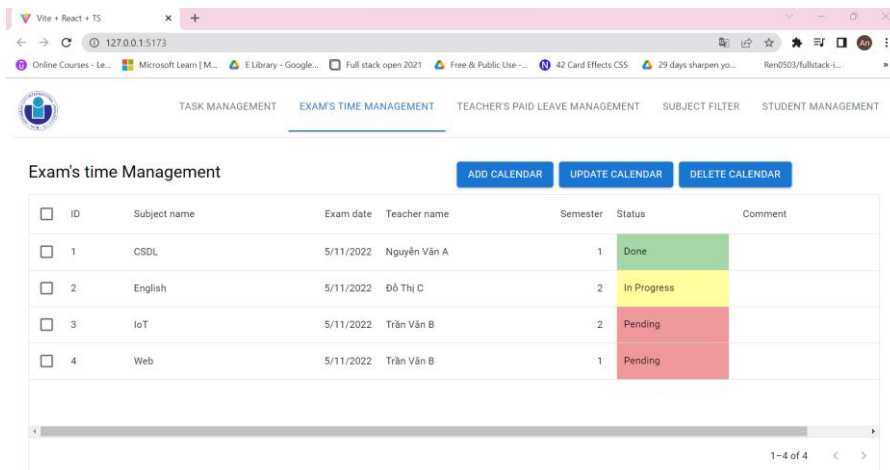


Figure 43. Exam time Management UI

Filling this form to add new exam time to tracking status in that semester. It required fields to give the user enough information to monitor.



**Adding New Exam's time**

Subject Name

Exam date

Teacher Name

Semester

Status

Comment

SUBMIT

ID	Subject name
1	CSDL
2	English
3	IoT
4	Web

**Figure 44. Add Exam time form**

For example, when the subject A was created but not start yet, the update function will help the users to change the status which is suitable in that moment.

**Updating New Exam's time**

Subject Name

Exam date

Teacher Name

Semester

Status

Comment

SUBMIT

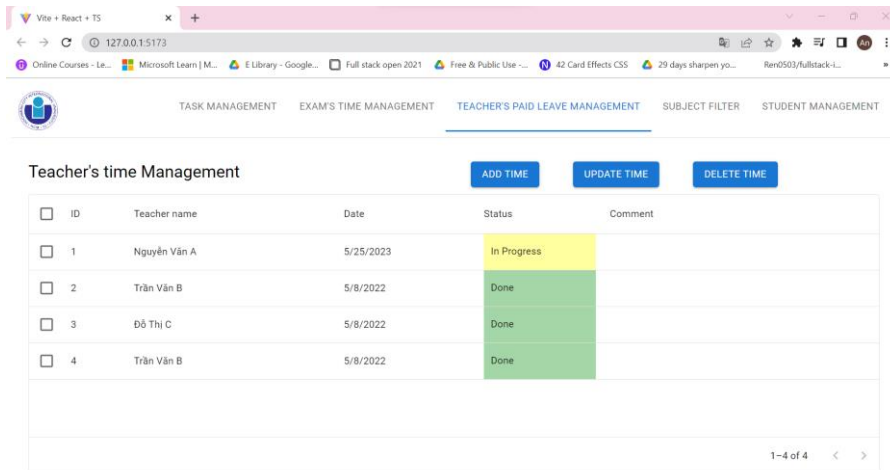
ID	Subject name
1	CSDL
2	English
3	IoT
4	Web

1 row selected

**Figure 45. Update Exam time form**

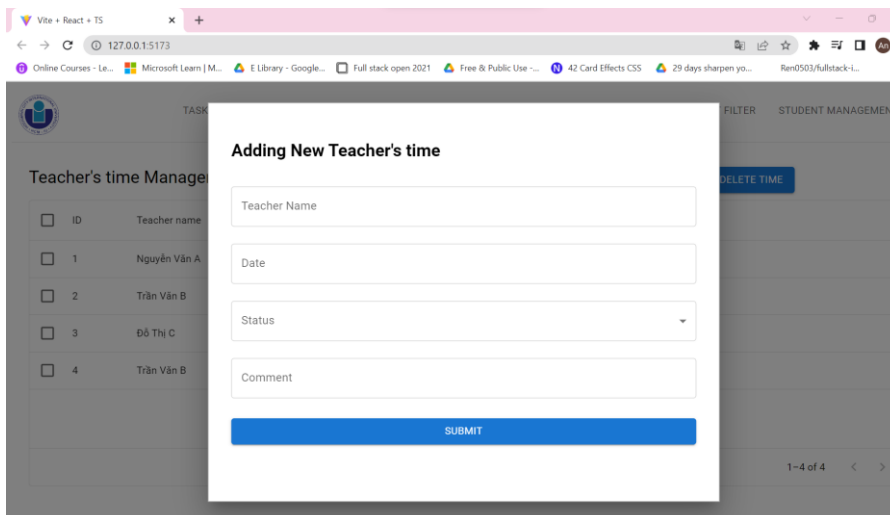
- Users interact with Teacher's time management to handle the status of the exam time that they are going to update.

Access this tab, UI will show all the list of teacher's day off for the users. When a teacher takes a day off, the system records and keeps track of this absence. It then checks later to confirm that the teacher has completed their required teaching compensation for the missed day.



**Figure 46. Teacher's paid leave Management UI**

When lecturer have day off and cannot take course in that day, secretary will add and record it in order to keep track of this absence and status of solving that case.



**Figure 47. Add Teacher's time form**

If the issues were solved, secretary will update the status of the time.

**Updating Teacher's time**

Teacher Name  
Nguyễn Văn A

Date  
5/25/2023

Status  
Done

Comment  
Have make-up class

SUBMIT

Figure 48. Update Teacher's time form

- Users interact with Subject Filter to statistics the quality of that course through each semester.

The UI will show the course and the statistic chart to evaluate the quality of that course. In the chart, there will be 4 column which present number of passed, failed, prohibited, and total of students.

**Subject Filter**

ADD NEW UPDATE

IMPORT FROM EXCEL

ID	Subject name	Teacher name	Semester	School year
1	CSDL	Nguyễn Văn A	HK1	2020
2	Web	Trần Văn B	HK1	2021
3	IoT	Trần Văn B	HK1	2022
4	English	Đỗ Thị C	HK2	2022

1-4 of 4

Bảng đồ thống kê sinh viên

th sinh viên

Figure 49. Subject Filter UI

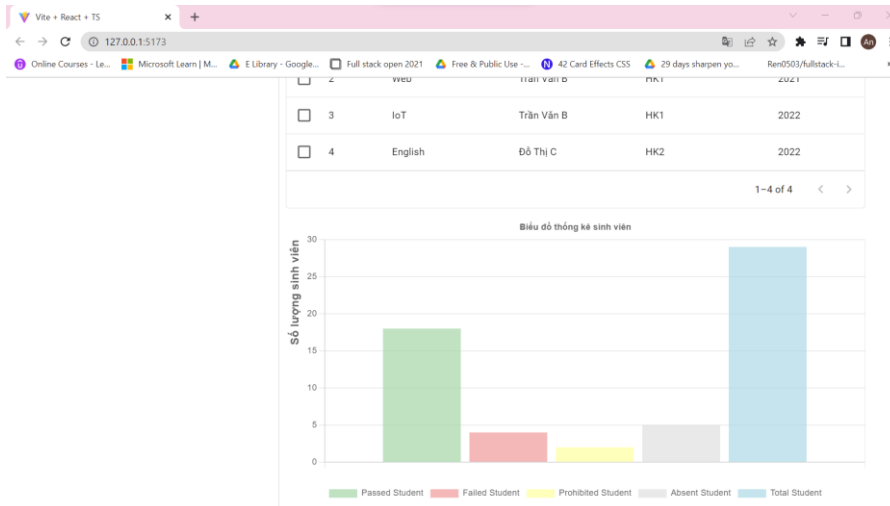


Figure 50. Statistic Chart for subject

When users select the other course, the UI will automatically get the data of that course to re-render the chart without refreshing the page.

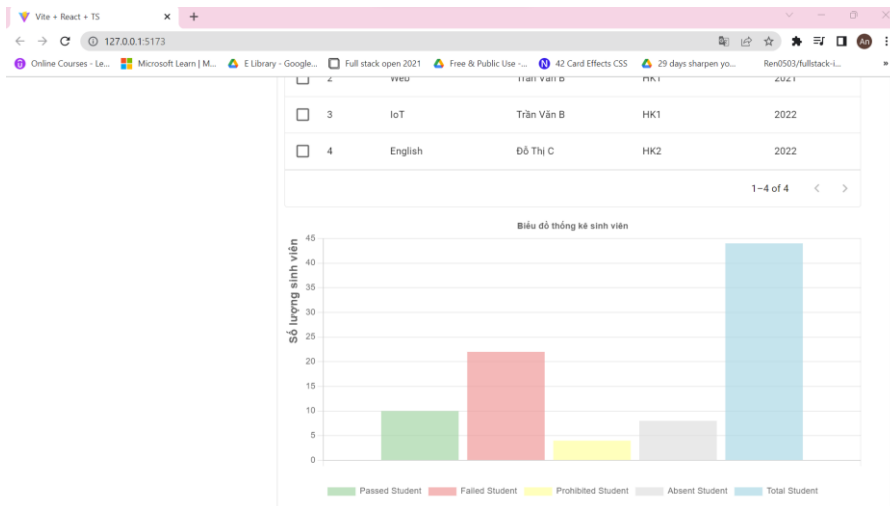


Figure 51. Re-rendering chart when select other subject

Filling this form with semester, year and importing Excel file to insert data to database.

From the excel file, the system will count the course by priority.

- First, it will calculate the column “Final” column, if score  $\geq 50$ , the pass\_count will increase by 1. On the other hand, the failed\_count will increase by 1.
- Next it will read the string value for “Note” column, if it has value “Prohibited”, the prohibited\_count will increase by 1.

**Figure 52. Add new Subject form and import excel file**

- Users interact with Student Management to statistics the number of students through each semester.

The UI provides 2 tables, the first one shows the number of students in semester, year and the second one show the details of students.

When importing excel file, the system will record all the status of students (STUDYING, RESERVED, ABSENT) and the second table will display it.

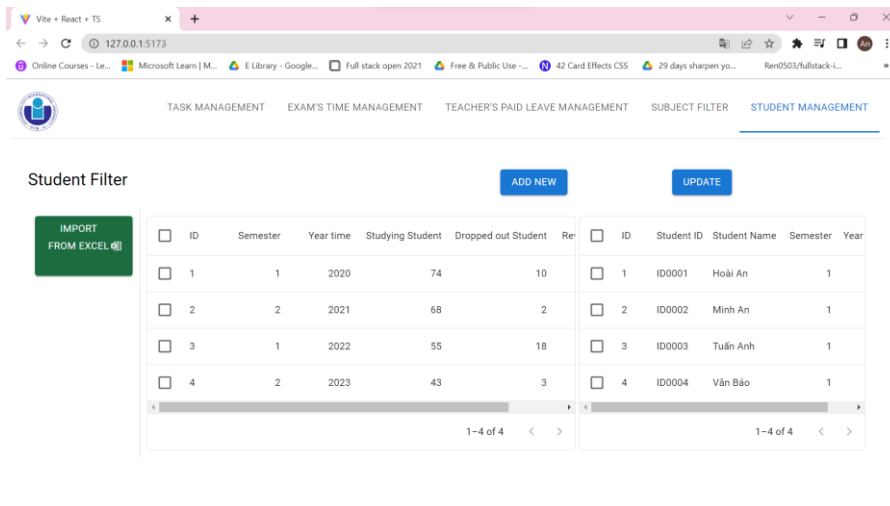


Figure 53. Student Management UI

When users select the other filter, it will re-generate the table without refreshing the page.

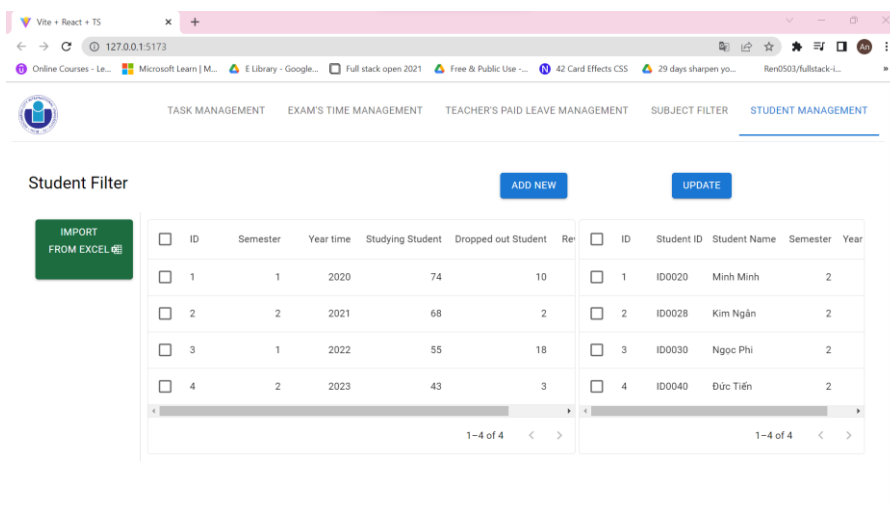


Figure 54. Table filter will change when select other semester

## **CHAPTER 5. DISCUSSION AND EVALUATION**

### **5.1. Discussion**

This thesis has fulfilled some of the fundamental requirements outlined by the instructors. However, in my view there are still some problems:

Can the application extend and interact with other applications that the school is using (Edusoft, OAA Online Request,)? I think the answer is yes, for example in Edusoft when students log in and access the educational program page. It will show you all the subjects the student has studied or passed.

Keeping the data of faculty students through each semester is too much. Through each semester, there will be only a few students with academic problems (dropout, reservation). The repeated saving affects the performance and effort of faculty staff to track status. In my opinion, I will apply solutions make each information of students is a file and import into the data center from the beginning. When the data center receives a file, it will send notification to the service of application to update the information. It will run automatically.

### **5.2. Evaluation**

The Administrative Support System for Faculty Staff provides valuable assistance in managing various administrative tasks within the faculty. The system offers several features, such as task management, exam time management, teacher's day off management, subject results management, and student management, which contribute to the efficient functioning of the faculty.

One notable strength of the system is its ability to handle CRUD operations effectively. The inclusion of CRUD functions allows users to easily manage and manipulate data related to tasks, exam schedules, teacher availability, subject results, and student information. This promotes streamlined data management and ensures accurate and up-to-date records.

Moreover, the system addresses specific needs of faculty staff, providing them with tools to effectively manage their workload and responsibilities. By facilitating task

management, instructors can organize and prioritize their duties, enhancing productivity and time management. The system's support for exam time management helps in scheduling and coordinating exams, reducing conflicts and ensuring a smooth examination process. The teacher's day off management feature enables staff members to request and track their time off, fostering a balanced work-life schedule.

Additionally, the subject results management feature proves valuable for instructors in tracking and managing student performance. It allows for efficient recording and analysis of grades, supporting informed decision-making and providing timely feedback to students.

While the system meets several essential requirements set by the faculty staff, it's important to consider areas for potential improvement. User experience and interface design should be user-friendly and intuitive, enabling easy navigation and minimizing the learning curve for faculty members. Additionally, ensuring the system's scalability and adaptability to evolving needs is crucial to cater to future growth and changing administrative processes.

In summary, the Administrative Support System for Faculty Staff is an effective tool that partly fulfills the fundamental requirements of faculty members. It streamlines administrative tasks, promotes efficient data management, and provides necessary features to enhance productivity and support decision-making. Continued development and refinement of the system can further enhance its usefulness and overall satisfaction among faculty staff.



## CHAPTER 6. CONCLUSION AND FUTURE WORK

### 6.1. Conclusion

In conclusion, the development of the Administrative Support System for Faculty Staff has addressed crucial administrative needs within the faculty. The system has successfully incorporated essential features such as task management, exam time management, teacher's day off management, subject results management, and student management. These functionalities have proven to be valuable in streamlining administrative processes and facilitating efficient data management. While the thesis has successfully met a significant portion of the faculty staff's requirements, there are opportunities for further improvement. Enhancements in user experience and interface design will contribute to a more intuitive and user-friendly system. Additionally, ensuring scalability and adaptability to future administrative needs will be crucial for sustained success. Overall, the Administrative Support System for Faculty Staff has demonstrated its value in addressing key administrative challenges. Its implementation has enhanced efficiency, data management, and decision-making processes within the faculty. Continued development and refinement will undoubtedly contribute to even greater effectiveness and satisfaction among faculty members, ultimately promoting a more streamlined and productive academic environment.

### 6.2. Future work

Based on the result, all the basic requirements have been done but there is also some point to improve in the future:

- Implement user management services to manage the account of the system. Create more role and function for lecturer in order to interact with faculty staff.
- Design database again. Because data is like record so I refer move to MongoDB.
- Try to apply API from Edusoft to connect and get data, therefore, we can avoid importing excel file.
- Processing solution with scalability and repetitive data saving.

## REFERENCES

- [1]. *SpringBoot documentation.*  
URL: <https://spring.io/projects/spring-boot>
- [2]. *Spring security documentation.*  
URL: <https://www.baeldung.com/security-spring>
- [3]. *Maven dependency.*  
URL: <https://mvnrepository.com/>
- [4]. *[vi](#)JVM definition.*  
URL: <https://www.javatpoint.com/jvm-java-virtual-machine>
- [5]. *Vite development guideline.*  
URL: <https://vitejs.dev/guide/>
- [6]. *ReactJS and Vite development guideline.*  
URL: <https://blog.openreplay.com/how-to-build-your-react-app-using-vite/>
- [7]. *ChartJS documentation.*  
URL: <https://www.chartjs.org/>
- [8]. *Material UI Component in ReactJS.*  
URL: <https://mui.com/material-ui/>
- [9]. *Building RESTful Web service.*  
URL: <https://spring.io/guides/gs/rest-service/>
- [10]. *Differences between JPA and JDBC.*  
URL: <https://www.baeldung.com/jpa-vs-jdbc>
- [11]. *Fullstack SpringBoot and ReactJS tutorial*  
URL: [https://www.youtube.com/watch?v=O\\_XL9oQ1\\_To](https://www.youtube.com/watch?v=O_XL9oQ1_To)
- [12]. *Workflow instructions*  
URL: <https://www.smartdraw.com/workflow-diagram/>

**Commented [A2]:** Em nên tham chiếu những tài liệu này được sử dụng ở nội dung nào của phần nội dung các chương ở trên. Ngay sau nội dung liên quan, em mở ngoặc chèn số tài liệu tham khảo tương ứng nhé.

.....[1]