THE UNIVERSITY OF DANANG
DANANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

GRADUATION PROJECT THESIS

MAJOR: INFORMATION TECHNOLOGY

SPECIALTY: SOFTWARE ENGINEERING

PROJECT TITLE:

BUILD AN ONLINE LEARNING SUPPORT SYSTEM FOR PRIMARY AND SECONDARY SCHOOL

Instructor: **MSc. NGUYEN THI MINH HY**
Student: **NGUYEN THI THUY TRINH**
Student ID: **102170128**
Class: **17T2**

Da Nang, 03/2022

INSTRUCTOR'S COMMENTS

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

REVIEWER'S COMMENTS

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

SUMMARY

Project name: Build an online learning support system for primary and secondary school.

Student name : Nguyen Thi Thuy Trinh

Student ID: 102170128

Class: 17T2

Summary: Nowadays, in the situation of the COVID-19 epidemic, online learning has become more and more popular and necessary. For high school and university students, online learning is quite convenient and effective, however, for primary and secondary school students, there are still many obstacles. It is difficult for students to concentrate and understand the lesson. This project aims to create a system where teachers can manage student information, create homework, assignment to help students understand the lesson better after school. In addition, the system helps manage and statistics student learning results, helping students practice the knowledge they have learned through assignments. The system also helps to connect parents in the learning of their children through notification and reminders. The questions in the system are built in the form of muiltichoice with a variety of expressions such as images, audio, and video to make it easier for students to grasp.

I have participated sufficiently in the conception and design of this work, as well as the analysis and implementation of the project. I worked on the website includes building API, Web client for teacher and Mobile app for student.

THE UNIVERSITY OF DANANG
UNIVERSITY OF SCIENCE AND TECHNOLOGY
INFORMATION TECHNOLOGY FACULTY

SOCIALIST REPUBLIC OF VIET NAM
Independence – Freedom – Happiness

GRADUATION PROJECT REQUIREMENTS

Student Name: Nguyen Thi Thuy Trinh Student No.: 102170128

Class: 17T2 Faculty: Information Technology Major: Software Engineering

1. *Topic title:*

   Build an online learning support system for primary and secondary school.

2. *Project topic:* ☐ *has signed intellectual property agreement for final result.*
3. *Initial figure and data: No data.*
4. *Content of the explanations and calculations:*

   The content of the thesis includes:

   INTRODUCTION – This chapter gives information about the context and purpose of the project as well as giving the scope of the problems which will be focused on the thesis.

   Chapter 1: THEORIES AND TECHNOLOGIES – This chapter introduces about all knowledge theories and technologies used in this project.

   Chapter 2: ANALYSIS AND DESIGN – This chapter covers the main features, software requirement specifications and database design of the project.

   Chapter 3: IMPLEMENTATION AND EVALUATION– This chapter shows an implementation of this project, including pictures and a brief explanation for each main function.

   CONCLUSION – The concluding section of the project simultaneously emphasizes the problem solved, as well as presenting issues still unresolved and provides recommendations and suggestions.

   REFERENCES – Presentation about detail of referenced information used in this thesis.

5. *Drawings, charts (specify the types and sizes of drawings):* No drawings, no charts.
6. *Supervisor(s):* MSc. Nguyen Thi Minh Hy
7. *Date of assignment: 18/10/2021*
8. *Date of completion: 28/02/2022*

Da Nang, date month year 2022

Head of Division……. **Instructor**

ACKNOWLEDGEMENTS

After an intensive period of over four months, today is the day: writing this note of thanks is the finishing touch on my senior project. It has been a period of intense learning for me. I would like to reflect on the people who have supported and helped me so much throughout this period.

I would first like to thank my supervisor, **MSc. Nguyen Thi Minh Hy** for her continuous support, supervision, motivation, and guidance throughout the tenure of my project in spite of her hectic schedule. She remained a driving spirit in my project and her experience gave me the understanding in handling research projects as well as helping me to clarify the abstruse concepts, requiring knowledge and perception, handling critical situations and in understanding the objective of my work.

I also want to thank my families and friends, who gave me the strength and confidence during my time of learning and during the implementation phase of this project. They have given a lot of love and encouragement for me which helped pass over the difficulties and fatigues.

Without your generous help, my senior year would not have been successful.

Sincerely,

Nguyen Thi Thuy Trinh

ASSURANCE

I guarantee:

1. *The contents of this senior project are performed by myself following the guidance of supervisors MSc. Nguyen Thi Minh Hy.*
2. *All references used in this senior project thesis, are quoted with the author's name, project name, time and location to publish clearly and faithfully.*
3. *All invalid copies, educated statute violation or cheating will be borne the full responsibility by myself.*

Students,

Nguyen Thi Thuy Trinh

TABLE OF CONTENT

LIST OF TABLES AND FIGURES

1. List of figures

2. List of tables

LIST OF ACRONYMS

No.	Items	Description
1	API	Application Programming Interface
2	JSON	JavaScript Object Notation
3	HTML	Hypertext Markup Language
4	CSS	Cascading Style Sheet
5	REST	Representational State Transfer
6	URL	Uniform Resource Locator
7	UX	User Experience
8	UI	User Interface
9	SMS	Short Messaging Service
10	O/RM	Object-Relational Mapper
11	HTTP	Hyper Text Transfer Protocol
12	EF	Entity Framework

INTRODUCTION

1. Project overview

1.1. Context

"Online learning" has become the most popular topic recently, since the beginning of the COVID-19 pandemic.

In order to ensure the safety of students as well as lecturers during a prolonged epidemic situation without disrupting the training program, online learning is necessary and become a top concern. Schools started implementing online learning plans, first universities, then high schools, and finally middle schools and elementary schools. For college students and high school students, online learning is less daunting and more effective. There are many online support tools used such as Microsoft Teams, Google Meet, Zoom, etc. for video calls, Kahoot, Bamboo for practicing exercises. At the same time, teaching productivity and student management are less of an obstacle than in elementary and middle schools because there are assistive technology tools and easier use of those tools for students. vien and high school.

For primary school students, online teaching faces many difficulties such as student management, student concentration in class and the use of complex technology tools. As an information technology student, I want to contribute what I have learned to create a system that enables teachers in middle and elementary schools to create assignments, tests, questions during class time to help improve teaching productivity, and manage and track student learning outcomes. Therefore, I come up with the idea "**Build an online learning support system for primary and secondary school.**". Besides, the system helps students better grasp the lesson through assignments, the interaction and monitoring of parents learning situation is also enhanced.

1.2. Statement of problem

The danger and impact of the epidemic makes it difficult for students and teachers to study in person at school, so online learning is the most effective alternative. Information technology plays a key role in online training.

Online teaching for primary and secondary school students faces many difficulties because it is difficult for students to focus on online lectures, as well as the lack of technology equipment, the connection between teachers, parents and parents

Therefore, a system that allows to manage students and improve the lesson through exercises and notify the results to parents is necessary and effective.

1.3. Purpose

My purpose is to build an system where teacher can post assignment and manage student, students can do assignment and view their learning result. It also help to send notification through SMS to parent about assignment schedule and the result of examination.

1.4. Main content to build the system

Stages of project implementation

- Research about the difficulties and problems faced with online learning for elementary and middle school students, analyze the influence of technology on online learning.
- Learn and analyze UML.
- Analysis and design of information systems.
- Design database for the system.
- Identify the actors and build the database.
- Learn and research, apply .NET programming languages and tools, Javascript with React Native and ReactJS frameworks, how to build WebAPI according to RESTful mechanism.
- Researching and applying SQL Server as a database platform.
- Build apps with main functions.
- Do the report.

2. Theories

2.1. Technologies

.NET Entity Framework.

ReactJS.

React Native

RESTful programming.

HTML, CSS, Javascript.

2.2. Tools and environment development

Microsoft Visual Studio Code.

Microsoft Visual Studio.

SQL Server.

Postman.

3. Structure of the thesis

INTRODUCTION – This chapter gives information about the context and purpose of the project as well as giving the scope of the problems which will be focused on the thesis.

Chapter 1: THEORIES AND TECHNOLOGIES – This chapter introduces about all knowledge theories and technologies used in this project.

Chapter 2: ANALYSIS AND DESIGN – This chapter covers the main features, software requirement specifications and database design of the project.

Chapter 3: IMPLEMENTATION AND EVALUATION– This chapter shows an implementation of this project, including pictures and a brief explanation for each main function.

CONCLUSION – The concluding section of the project simultaneously emphasizes the problem solved, as well as presenting issues still unresolved and provides recommendations and suggestions.

REFERENCES – Presentation about detail of referenced information used in this thesis.

THEORIES AND TECHNOLOGY

This system applies the current popular technologies and platforms that combine to create a system that responds to the services that users desire.

To learn more about the technologies, I would like to present the concept of some key technologies that I used in this project.

1.1. .NET Entity Framework

Definition

Entity Framework is an object-relational mapper (O/RM) that enables .NET developers to work with a database using .NET objects. It studied the need for most of the data-access code that developers usually need to write.

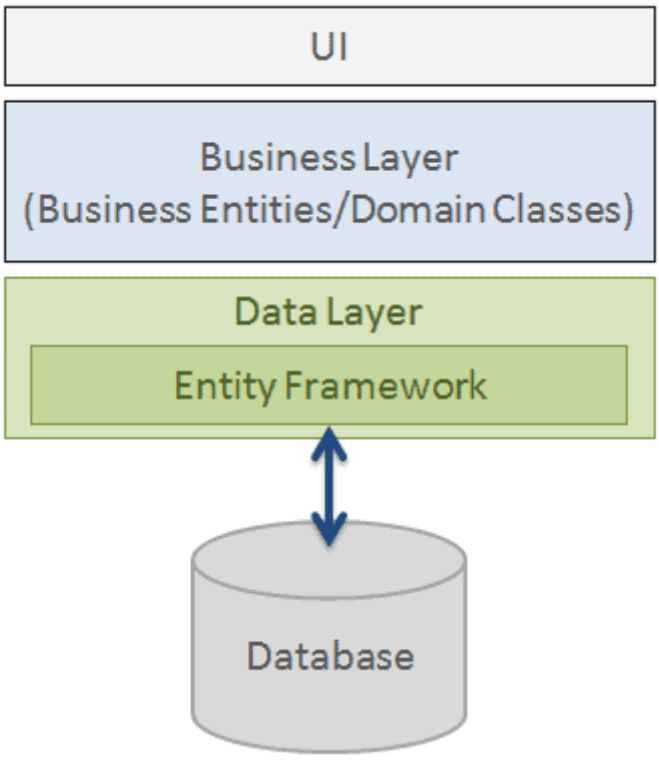


Figure 1.1. Entity Framework workflow overview.

As per the above figure, Entity Framework fits between the business entities (domain classes) and the database. It saves data stored in the properties of business entities and also retrieves data from the database and converts it to business entities objects automatically.

Entity Framework Features

- **Cross-platform**: EF Core is a cross-platform framework which can run on Windows, Linux and Mac.
- **Modeling**: EF (Entity Framework) creates an EDM (Entity Data Model) based on POCO (Plain Old CLR Object) entities with get/set properties of different data types. It uses this model when querying or saving entity data to the underlying database.
- **Querying**: EF allows us to use LINQ queries (C#/VB.NET) to retrieve data from the underlying database. The database provider will translate this LINQ queries to the database-specific query language (e.g. SQL for a relational database). EF also allows us to execute raw SQL queries directly to the database.
- **Change Tracking**: EF keeps track of changes occurred to instances of your entities (Property values) which need to be submitted to the database.
- **Saving**: EF executes INSERT, UPDATE, and DELETE commands to the database based on the changes occurred to your entities when you call the SaveChanges() method.
- **Concurrency**: EF uses Optimistic Concurrency by default to protect overwriting changes made by another user since data was fetched from the database.
- **Transactions**: EF performs automatic transaction management while querying or saving data. It also provides options to customize transaction management.
- **Caching**: EF includes first level of caching out of the box. So, repeated querying will return data from the cache instead of hitting the database.
- **Built-in Conventions**: EF follows conventions over the configuration programming pattern, and includes a set of default rules which automatically configure the EF model.
- **Configurations**: EF allows us to configure the EF model by using data annotation attributes or Fluent API to override default conventions.
- **Migrations**: EF provides a set of migration commands that can be executed on the NuGet Package Manager Console or the Command Line Interface to create or manage underlying database Schema.

1.2. ReactJS

Definition

React JS is JavaScript library used for building reusable UI components. According to React officicial documentation, following is the definition:

React is a library for bulding composable user interfaces. It encourages the creation of reusable UI components, which present data that changes over time. Lots of people use React as the V in MVC. React abstracts away the DOM from you, offering a simpler programming model and better performance. React can also render on the server using Node, and it can power native apps using React Native. React implements one-way reactive data flow, which reduces the boilerplate and is easier to reason about than traditional data binding.

JSX

JSX is a language that allows writing HTML code in Javascript. JSX performs optimization while compiling to Javacscript code. These codes give a much faster execution time than an equivalent code written directly in Javascript. In contrast to Javascript, JSX is statically-typed, meaning it is compiled before runtime, just like Java and C++. So errors will be detected during compilation. In addition, it also provides very good debug when compiling. Easier: Easier. Legacy JSX is based on Javascript, so it's very easy for Javascripts programmers to use.

Single-way data flow

ReactJS does not have modules to do dedicated tasks to process data. So ReactJS will break down the view into small components, they have a close relationship. On the question of why should we care about the structure and relationships between components in ReactJS? Answer by data flows in ReactJS: Data flow is one-way from parent to child. It can be said that ReactJS using one-way data flow will create difficulties. However, this will make ReactJS will promote its use as well as its role.

Virtual DOM

Frameworks using Virtual-DOM are typical ReactJS at the time of Virtual-DOM changes, the programmer does not need to manipulate the DOM directly in the view, but the change is still reflected. Because Virtual-DOM both acts as a Model and a View, so a change on the Model entails a change on the View, of course and vice versa. Although it does not directly affect the DOM elements in the View view, the programmer can still implement the mechanism of Data-binding. This makes the speed increase in a better way.

Currently, programmers or enterprise companies are always looking for the best technology within reach to beat the competition, one of them is ReactJS. In a word, ReactJS will improve user experience, higher click and conversion rates. It's not just DOM updates that make apps faster and load better. Businesses using ReactJS are guaranteed to have a better interface than those using conventional frameworks.

Components

React is built around components, not templates like other frameworks. In React, we build websites using small components. We can reuse a component in many places, with different states or properties, in a component that can contain other components. Each component in React has its own, mutable state, and React will perform component updates based on state changes. Everything React is a component. They help maintain code when working on large projects. A simple react component just needs a render method. There are many other methods available, but render is the dominant method.

Props and State

Props: helps components interact with each other, the component takes input called props, and returns properties that describe what the child component should render. Prop is immutable. State: represents the state of the application, when the state changes, the component also re-renders to update the UI.

1.3. React Native

Definition

React native is a tool that helps us cross-platform programming to create applications on the native environment. It is an open source framework developed by Facebook that allows you to use Java scripts to develop software on Android and IOS mobile phones. Instagram, Facebook, Skype, etc. are featured apps using React Native. React native is like React in that they use native components instead of web components. So to understand the structure of React native we need to have basic knowledge with basic React concepts such as JSX, components, props or state.

Operation

React Native works by integrating 2 threads, Main Thread and JS Thread for mobile applications. In there:

- Main Thread: updates the user interface (UI) and handles user interaction.
- JS Thread: executes and processes Javascript code.

These two Main Thread and JS Thread work independently of each other. Two Threads will interact with each other through a Bridge. This bridge will convert data back and forth between Threads.

Advantages

- **Reusable code**: React Native allows developers to reuse code while developing cross-platform apps. In particular, developers can reuse almost 80-90% of the code instead of having to write and create separate applications for different platforms.

- **Large user community**: React Native is rated as one of the most loved frameworks (stack overflow survey in 2019). Thanks to the huge user community around the world, we can get help if we encounter bugs.
- **Stability and optimization**: Developed by Facebook, React Native has high stable performance.

Disadvantages

- Security is not really good due to the use of JavaScript. By using JavaScript, users will also be affected by the characteristics of JavaScript: easy to do wrong, which makes it difficult to maintain later.
- Memory management.
- Customization is not really good in some modules.
- Not suitable for applications that need high computing power (hash, crypto, etc).

1.4. RESTful programming

Definition

REST stands for **Re**presentational **S**tate **T**ransfer. REST is a web standards-based architecture and uses HTTP Protocol for data communication. It revolves around resources where every component is a resource and a resource is accessed by a common interface using HTTP standard methods.

In REST architecture, a REST Server simply provides access to resources and the REST client accesses and presents the resources. REST uses various representations to represent a resource like Text, JSON and XML. JSON is now the most popular format being used in Web Services.

RESTful Web Service

A web service is a collection of open protocols and standards used for exchanging data between applications or systems.

Web services based on REST Architecture are known as RESTful Web Services. These web services use HTTP methods to implement the concept of REST architecture. A RESTful web service usually defines a URI (Uniform Resource Identifier), which is a service that provides resource representation such as JSON and a set of HTTP Methods.

HTTP methods

- **GET** − Provides a read only access to a resource.
- **PUT** − Used to create a new resource.
- **DELETE** − Used to remove a resource.

- **POST** – Used to update an existing resource or create a new resource.
- **OPTIONS** – Used to get the supported operations on a resource.

RESTful Resources

REST architecture treats every content as a resource. REST uses various representations to represent a resource where Text, JSON, XML. The most popular representations of resources are XML and JSON.

A resource in REST is a similar Object in Object Oriented Programming or is like an Entity in a Database.

RESTful Message

RESTful Web Services make use of HTTP protocols as a medium of communication between client and server. A client sends a message in form of a HTTP Request and the server responds in the form of an HTTP Response.

1.5. Tools to support during project implementation

In the process of analyzing and designing UML, draw.io tool (accessed link on website: https://app.diagrams.net/) is used. This tool makes it easy for us to manipulate and use to create use case diagrams, activity diagrams, ...

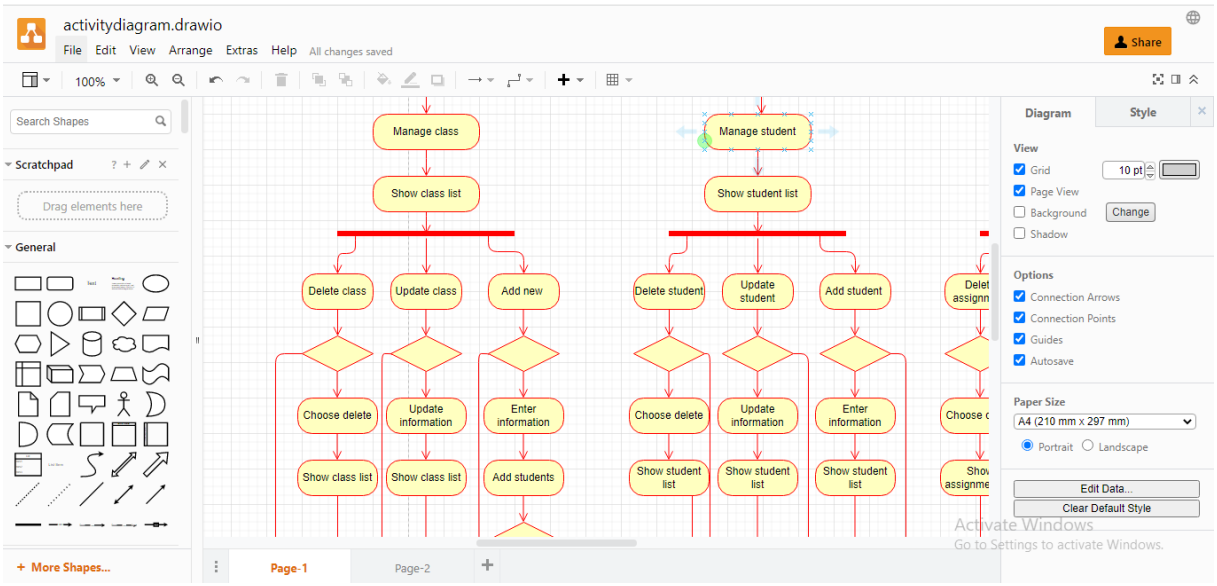


Figure 1.2. Draw.io tool

The process of building program code tools editor is VSCode, the outstanding feature is simple, lightweight, easy to install, supports many operating systems as well as many different programming languages.

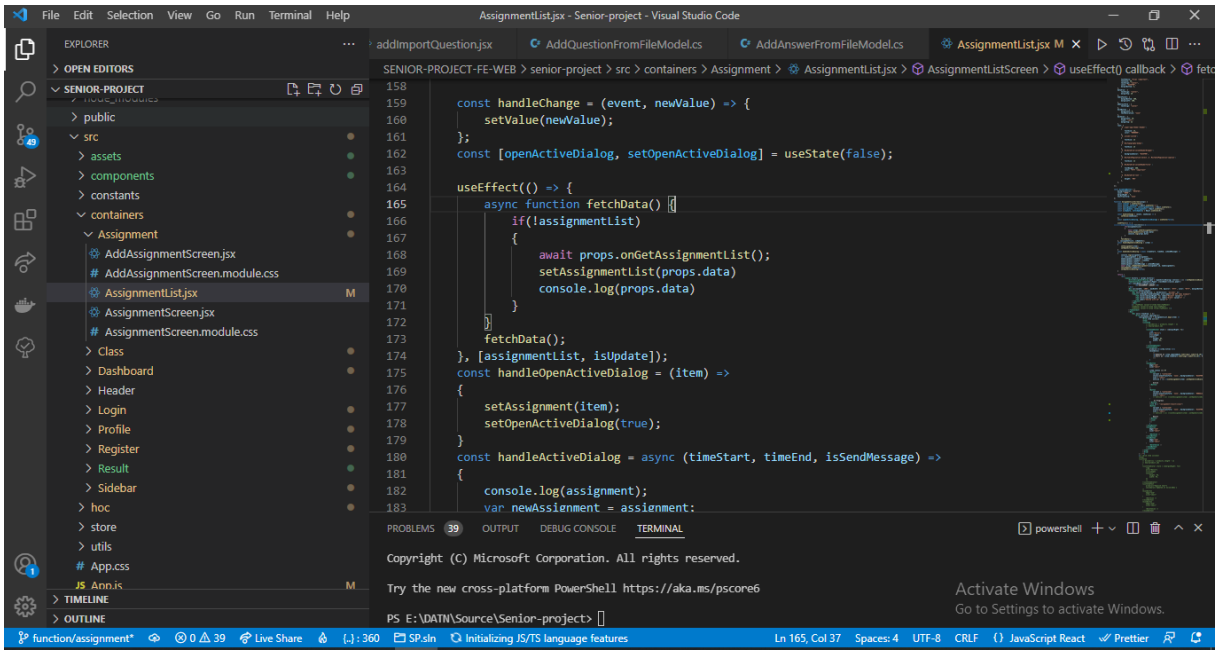


Figure 1.3. Editor VSCode tool

The process of storing and querying data, the tool used is SQL server Management Studio, the advantage is quite simple, easy to use and connect, support the process of editing, querying and viewing data.

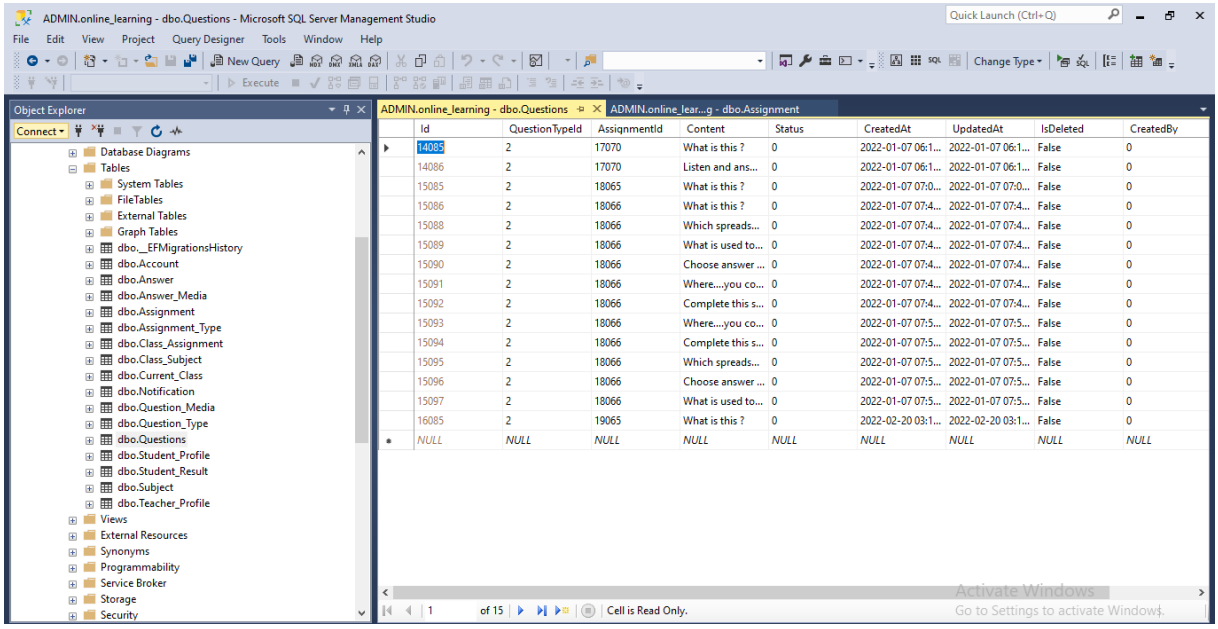


Figure 1.4. SQL server Management Studio tool

The process of checking, testing and testing APIs that have been programmed before being applied to call from website requests, the tool used is Postman.

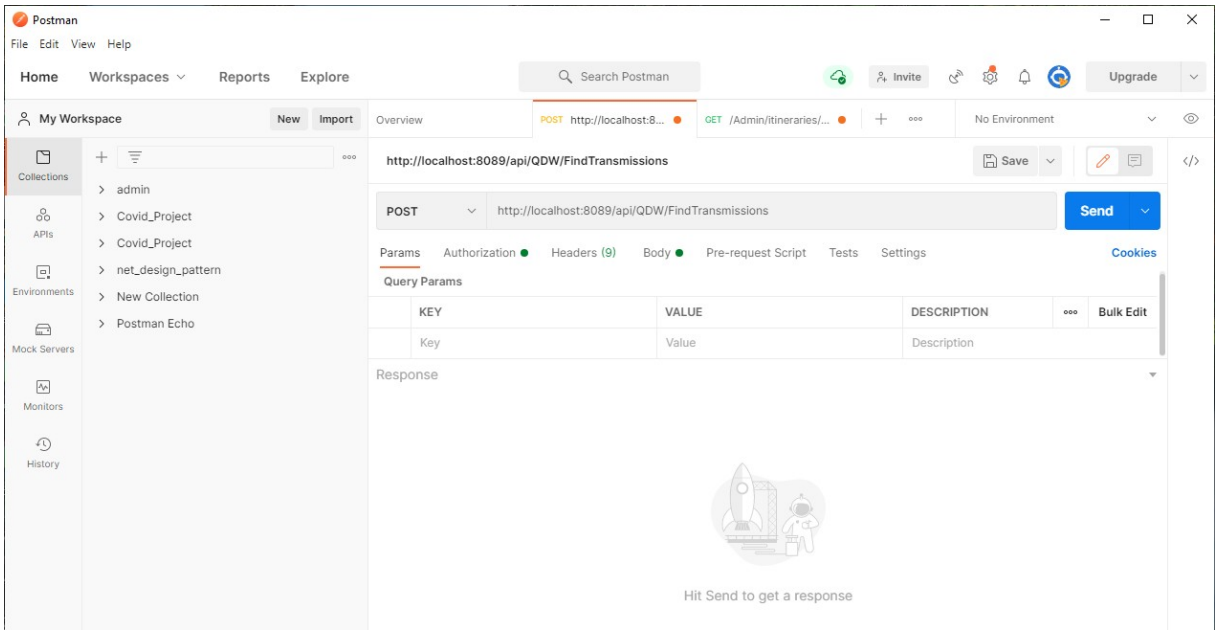


Figure 1.5. Postman tool

1.6. Conclusion

By studying and learning about the above technologies, I successfully applied the concepts and their mechanism operating in this project to create an online learning support system for primaru and secondary school.

Some of these technologies are not new, but they are widely using and a trend for the software development industry. Therefore, understanding the concept is very important, help to apply properly technology for each project, in order to improve the efficiency and usability.

ANALYSIS AND DESIGN

This chapter will go into detail about the requirements, describing nonfunctional requirements, design constraints and other factors necessary to provide a complete and comprehensive description of the requirements for the application. This consists of a package containing Requirements Specification, Use-Cases of the use-case model, Use Case Specifications and Activity Diagram. Shows an overview of what functions the application can satisfy. In addition, it defines the architecture, modules, and data for a system to satisfy specified requirements. System design is intended to be the link between the system architecture and the implementation of technological system elements that compose the physical architecture model of the system. It could be seen as the application of systems theory to product development.

The System Design process is to provide sufficiently detailed data and information about the system and it is a system element to enable the implementation consistent with architectural entities as defined in models and views of the system architecture. It shows the components of the application, the structure of data tables, the relationship the elements that make up the system.

2.1. Requirement analysis and design

2.1.1. User interaction

This system has four three actors:

Teacher:

This actor plays an important role in this system.

They can access the system to manage student information and classes. They can also create and manage assignment, monitor student learning result.

Student:

This actor plays an important role in this system.

They are those who can access the system to do the assignment. They can use the system to join assignment and see their learning result.

Parent:

They can receive the notification through SMS about new assignment and their children’s learning result.

2.1.2. Decomposition diagram

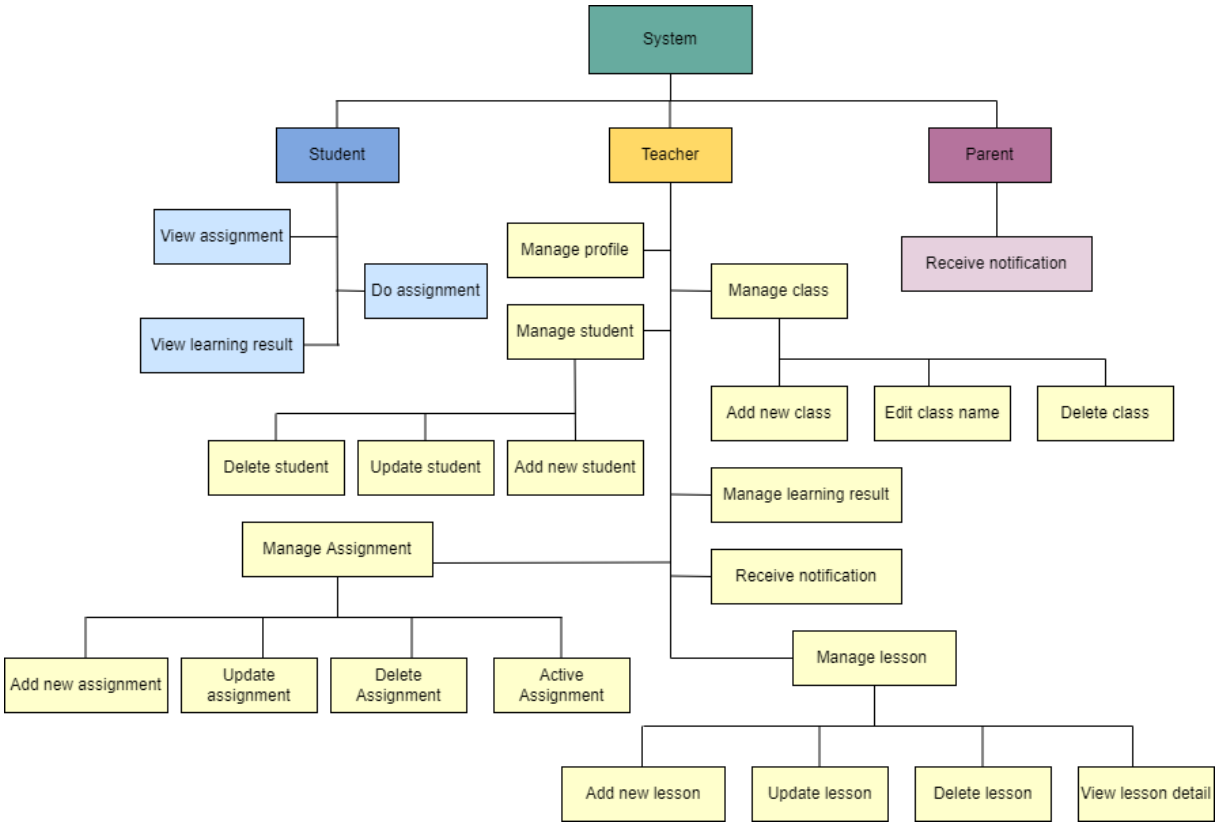


Figure 2.6. The Online Learning Support System Decomposition Diagram

This is the decompositon diagram that descript all main functions of the system.

2.1.3. Main features

Teachers

- Manage account: teacher can login, logout and register new account. Besides, the can also update their profile and password.
- Manage class: teacher can add new class, update and delete class. Student can be add to class manually or import from excel file.
- Manage student: teacher can update student information, delete or add new students to class.
- Manage assignment: teacher can add new assignmet to class, question in assignment can be text,image, audio or video. Question can be add manually or import from template. Answer can be text or image.
- Manage student result: teacher can view assignment result and student learning result.
- Receive notification: Teacher can receive notification about assignment result when it is completed.
- Manage lesson: teacher can add new lesson, update and delete lesson in class.

Student: Student can access to mobile to do assignment and view their learing result.

Parent: Parent can receive notification through SMS about new assignment.

2.2. Use case diagram

2.2.1. Use case diagram for the system

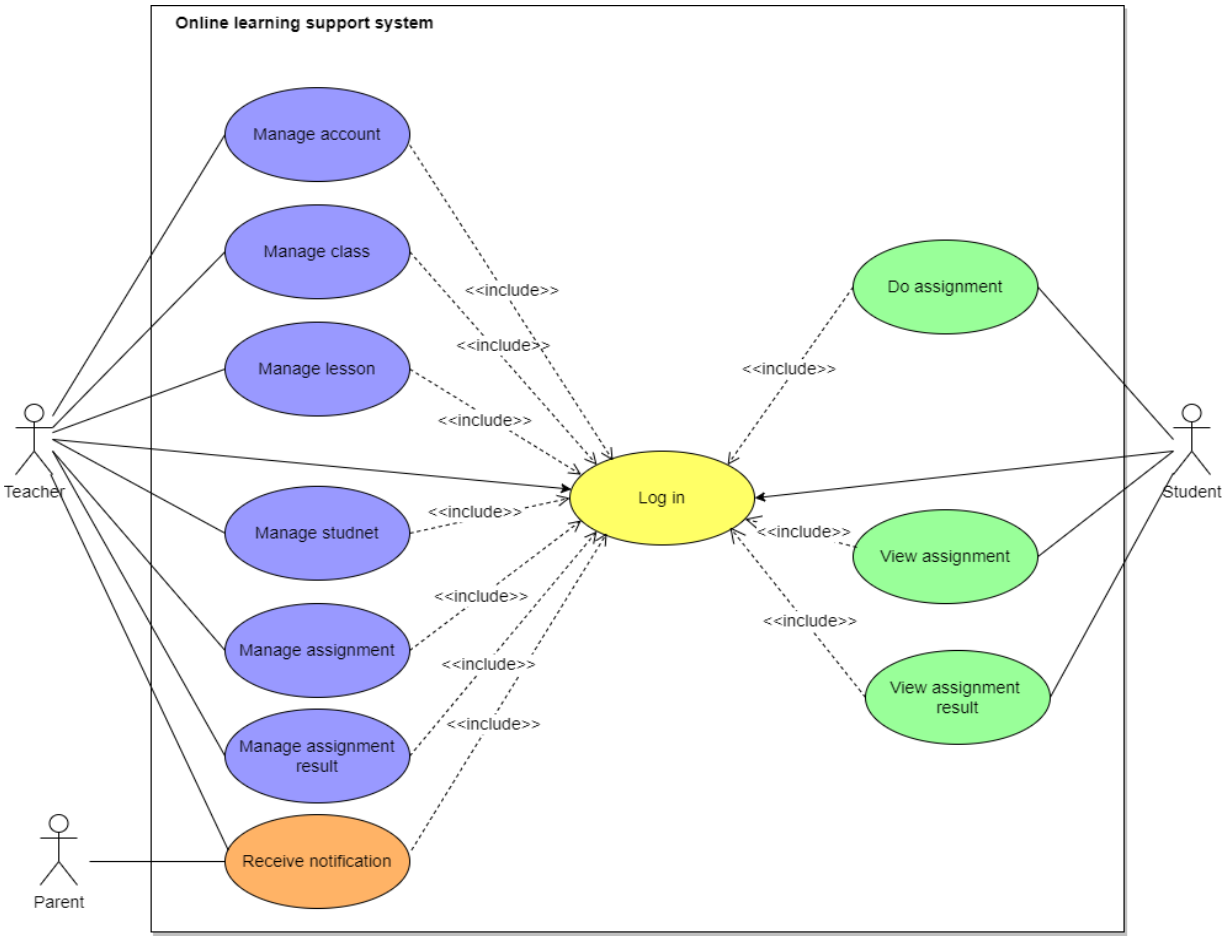


Figure 2.7. Use-case diagram for the system

2.2.2. Use case diagram for teacher to manage account

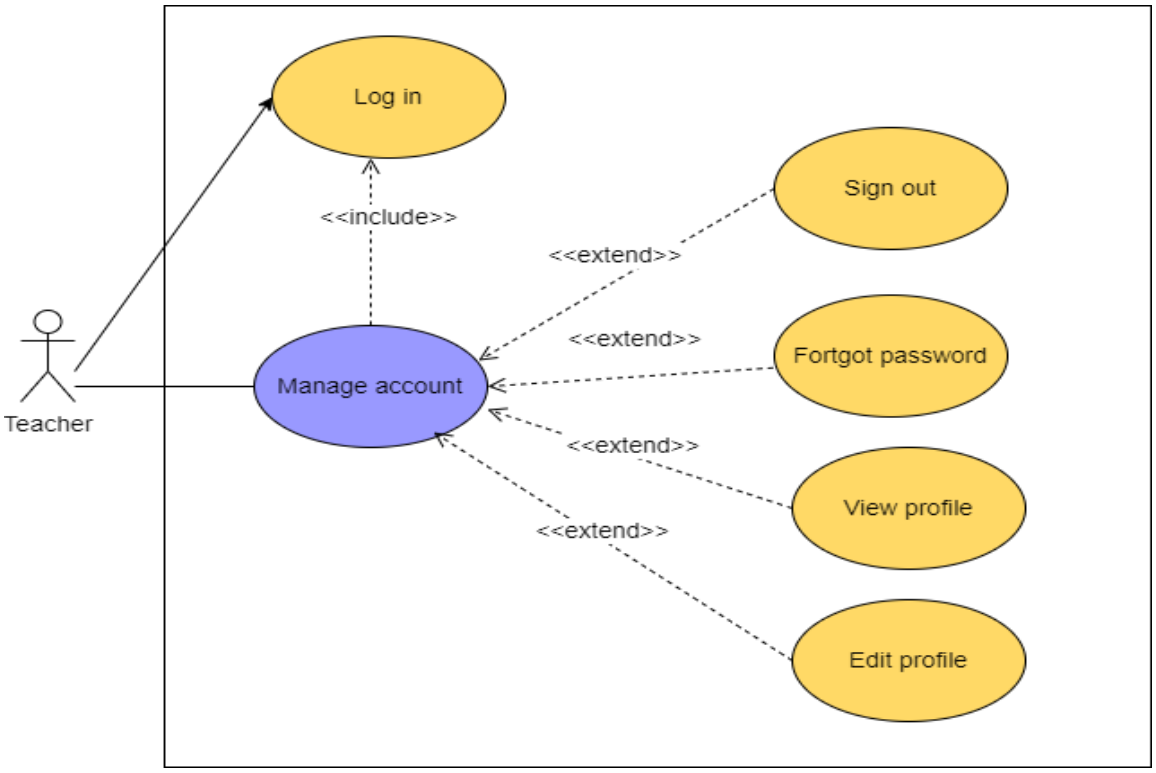


Figure 2.8. Use-case diagram for teacher to manage account

2.2.3. Use case diagram for teacher to manage class

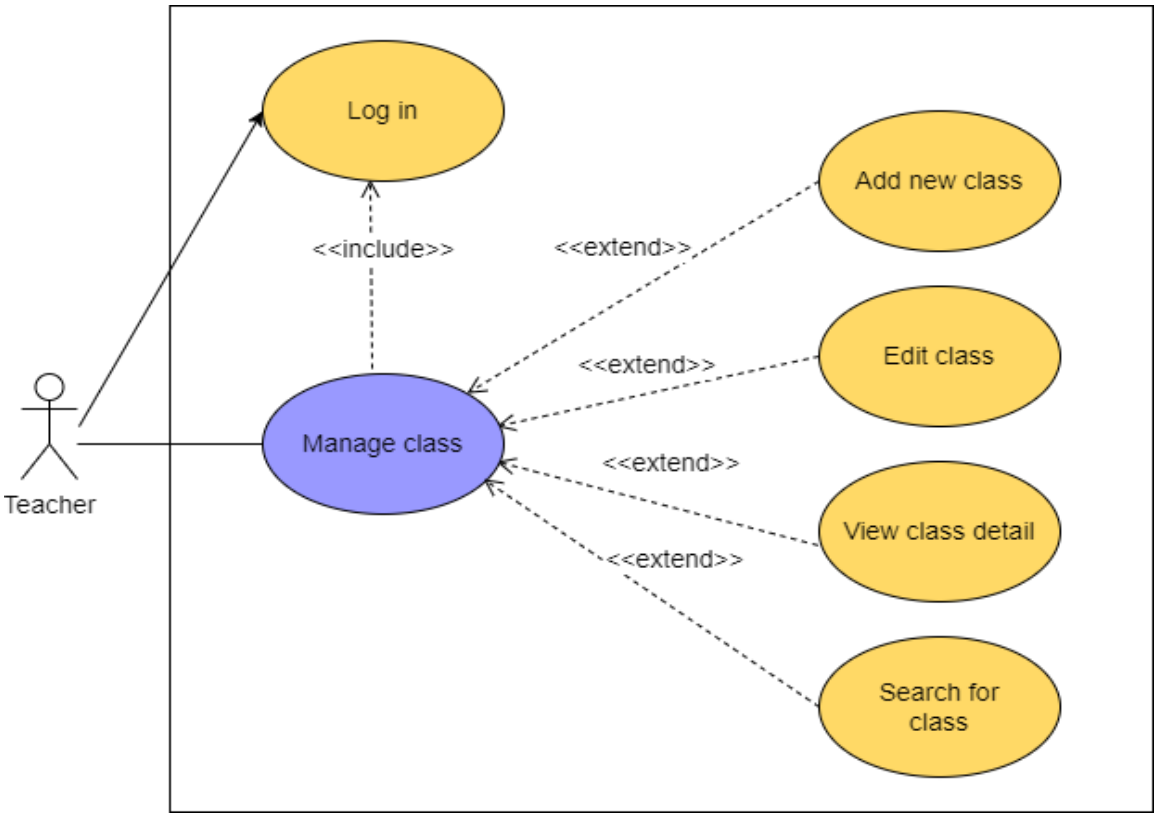


Figure 2.9. Use-case diagram for teacher to manage class

2.2.4. Use case diagram for teacher to manage student

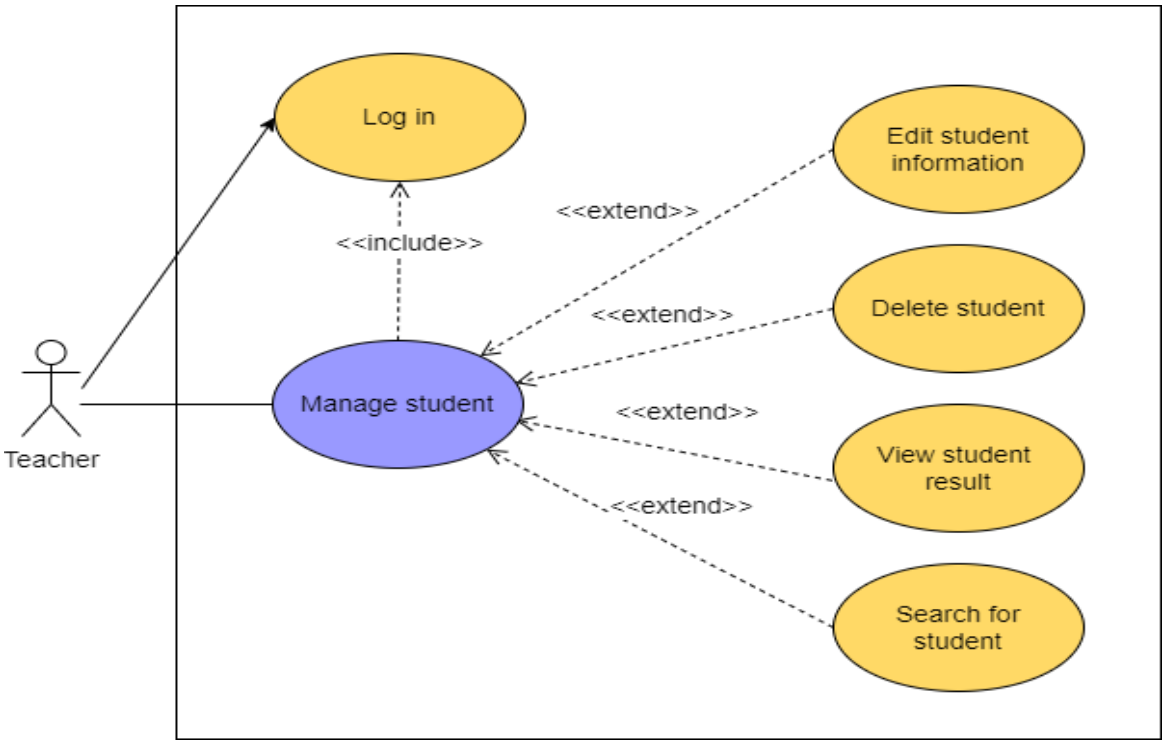


Figure 2.10. Use-case diagram for teacher to manage student

2.2.5. Use case diagram for teacher to manage assignment

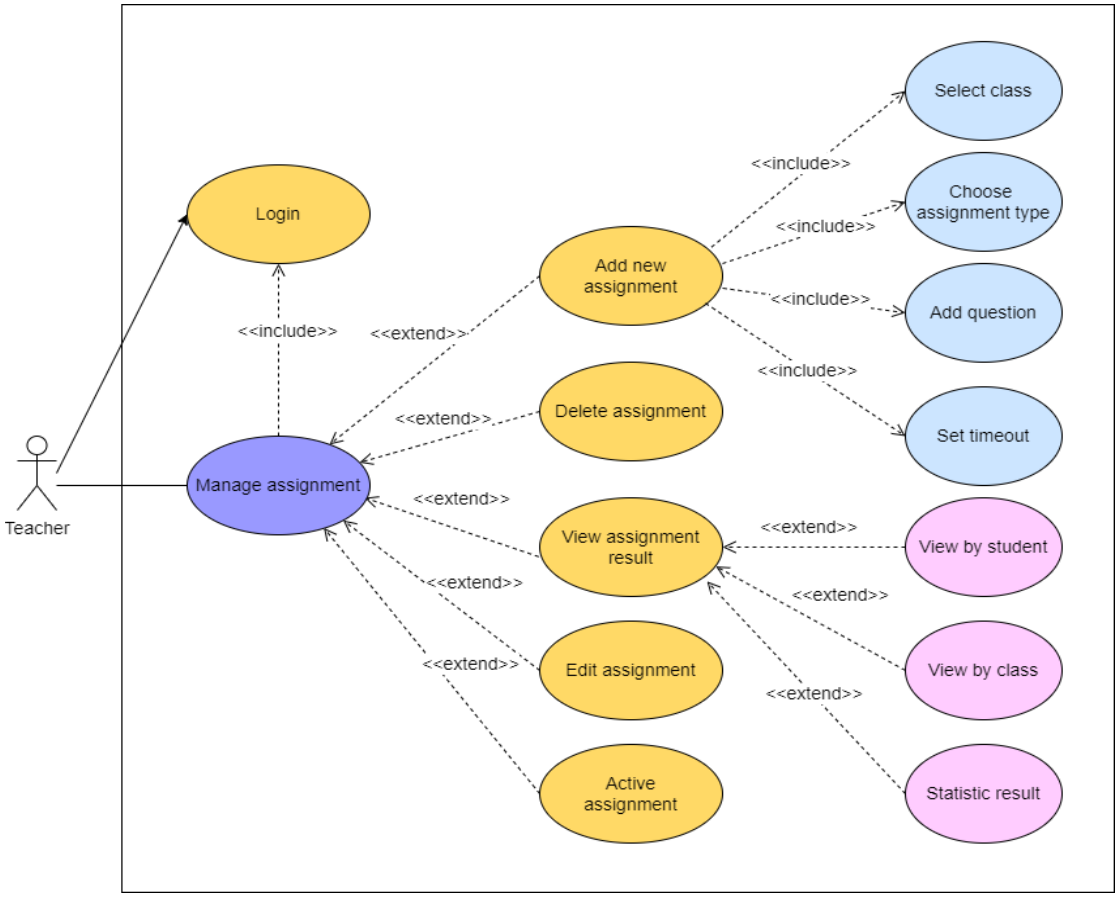


Figure 2.11. Use-case diagram for teacher to manage assignment

2.2.6. Use case diagram for teacher to manage assignment result

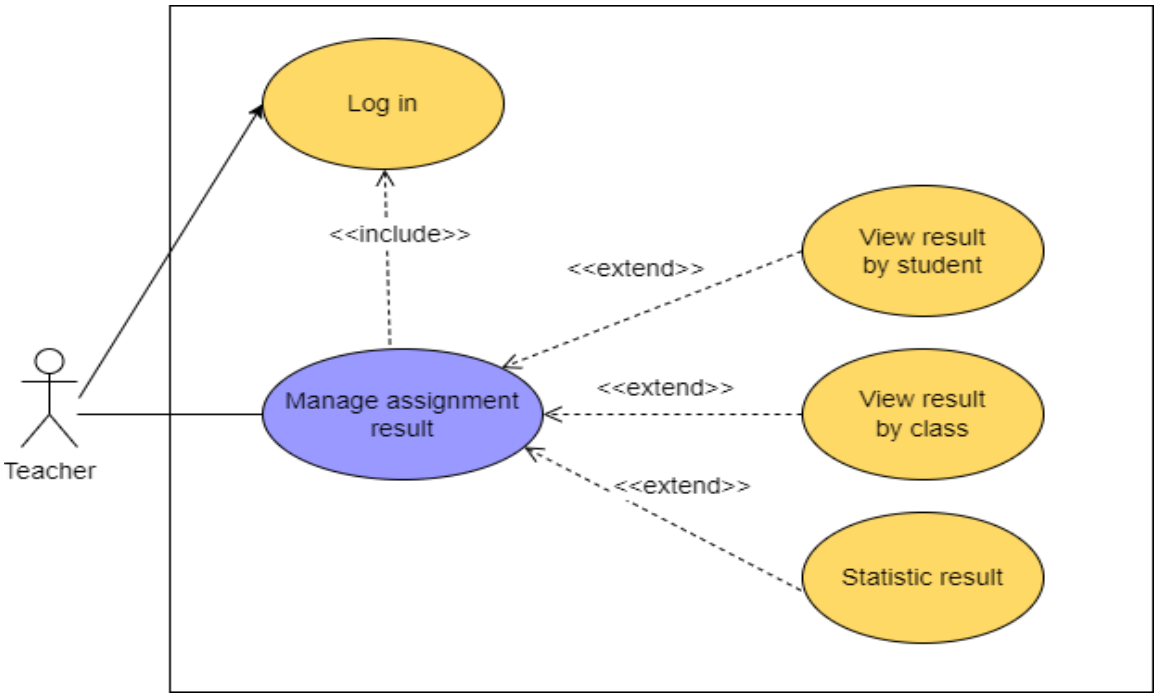


Figure 2.12. Use-case diagram for teacher manage assignment result

2.2.7. Use case diagram for teacher to manage lesson

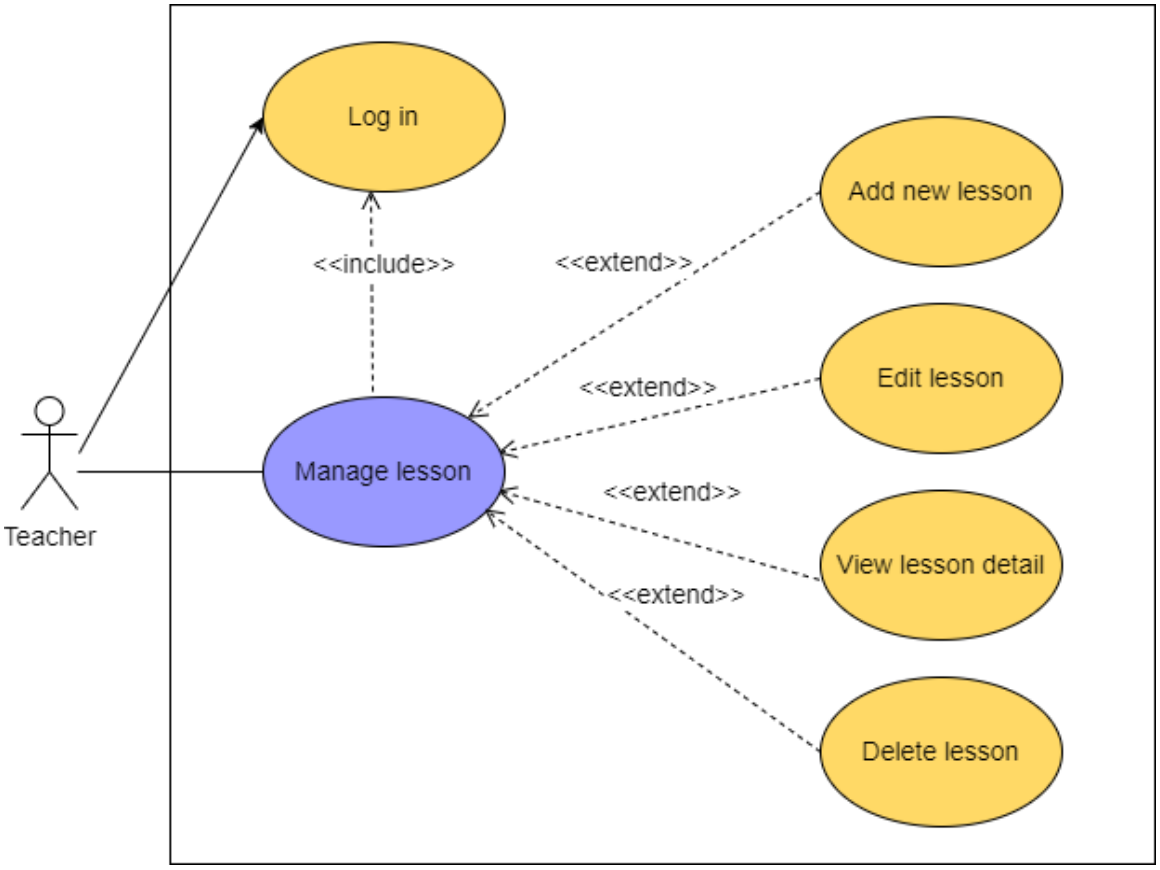


Figure 2.13. Use-case diagram for teacher to manage lesson

2.3. Use case description

Table 2.1. Use-case description for Login

Use Case ID	**UC-01**		
Actor	Teacher		
Brief description	This use-case for logging in.		
Pre-conditions	User who has account to access to application.		
Post-conditions	User logged in successfully.		
Flow of events		**Actor Input**	**System Response**
	1	User opens the website and click Login button.	
	2		System shows login form.
	3	User fills the email and password. Click Login button.	
	4		System validates account that user filled and submit. Show home screen if user's account is a trainer or trainee.

Table 2.2. Use-case description for Register

Use Case ID	**UC-02**		
Actor	Teacher		
Brief description	This use-case for teacher to register new account.		
Pre-conditions	User who want to access to application.		
Flow of events		**Actor Input**	**System Response**
	1	User opens the website and click Login button.	
	2		System shows login form.
	3	User click Sign Up in Login screen.	
	4		System shows Register form.
	5	User fills the username, email, phone number and password. Click Sign Up button.	
	6		System validates account that user filled and submit. Show home screen if user's

			account is a trainer or trainee.

Table 2.3. Use-case description for Show profile

Use Case ID	**UC-03**		
Actor	Teacher		
Brief description	This use-case for show personal information.		
Pre-conditions	User already logged into the system.		
Post-conditions	Server received the request from Applicant then gets all information on database.		
Flow of events		Actor Input	System Response
	1	User clicks to Icon “Avatar” in navigation bar and click “Profile” in dropdown.	
	2		System shows all personal information on a form.

Table 2.4. Use-case description for Update personal information

Use Case ID	**UC-04**		
Actor	Teacher		
Brief description	This use-case for update personal information.		
Pre-conditions	User already logged into the system.		
Post-conditions	Server received the request from Applicant then update new information into database.		
Flow of events		**Actor Input**	**System Response**
	1	User clicks to Icon “Avatar” in navigation bar and click “Profile” in dropdown.	
	2		System shows update personal information form.
	3	User fills the necessary information to the form.	
	4		System validates data and update data to the database.

Table 2.5. Use-case description for Update password

Use Case ID	**UC-05**
Actor	Teacher
Brief description	This use-case for update password.

Pre-conditions	User already logged into the system.		
Post-conditions	Server received the request from Applicant then update new password into database.		
Flow of events		**Actor Input**	**System Response**
	1	User clicks to Icon “Avatar” in navigation bar and click “Profile” in dropdown.	
	2		System shows update personal information form.
	3	User clicks “Change Password” Button.	
	4		System shows update password form.
	5	User fills the necessary new password to the form and submit form.	
	6		System validates data and update data to the database.

Table 2.6. Use-case description for Register to become a user for teacher

Use Case ID	**UC-06**		
Actor	Teacher		
Brief description	This use case for teacher registration from system.		
Pre-conditions	Trainee already logged into the system		
Flow of events		**Actor Input**	**System Response**
	1	User clicks to “Sign in” in login screen.	
	2		System shows sign in page.
	5	User fills the necessary information to the form and press “Register” button.	
	6		System validates data and update the role to the database.

Table 2.7. Use-case description for view class list

Use Case ID	**UC-07**
Actor	Teacher
Brief description	This use-case for view class list of current teacher in the system.

Pre-conditions	Teacher already logged into the system and went to the Homepage screen.		
Flow of events		**Actor Input**	**System Response**
	1	At the Homepage screen, press “Access” button in Your class card.	
	2		System will show class list of current teacher.

Table 2.8. Use-case description for creating new class

Use Case ID	**UC-08**		
Actor	Teacher		
Brief description	This use case is for teacher to create new class.		
Pre-conditions	Teacher already logged into the system and went to class list page.		
Post-conditions	Server received the request from teacher then update new information into database.		
Flow of events		**Actor Input**	**System Response**
	1	At the screen for showing the class list, press “Add new class”.	
			System will navigate to add class page
		User fills the necessary information to the form and press “Save” button.	
	2		System will validate data and update it in the database.

Table 2.9. Use-case description for adding students to class by importing excel file

Use Case ID	**UC-09**		
Actor	Teacher		
Brief description	This use case for adding student list to class by importing excel file		
Pre-conditions	Teacher already logged into the system and went to add class page.		
Flow of events		**Actor Input**	**System Response**
	1	User press “Import students”	

		in add class page and choose excel file.	
	2		System will validate and show student list.

Table 2.10. Use-case description for adding students to class by add manually

Use Case ID	**UC-10**		
Actor	Teacher		
Brief description	This use case for adding student list to class by add manually through excell sheet		
Pre-conditions	Teacher already logged into the system and went to add class page.		
Flow of events		**Actor Input**	**System Response**
	1	User press “Add manually” in add class page.	
	2		System shows the add student manually dialog.
	3	User fills the necessary information to the form and press “Save” button.	
	4		System will validate and show student list.

Table 2.11. Use-case description for Delete classes

Use Case ID	**UC-11**		
Actor	Teacher		
Brief description	This use case for deleting classes from the system.		
Pre-conditions	Users already logged into the system and went to class list page.		
Flow of events		**Actor Input**	**System Response**
	1	User choose classes to delete in the grid.	
	2		System shows the selected classes.
	3	User clicks to button “Delete”.	
	4		System shows a dialog that confirm before deleting classes.

	5	User clicks to button “Delete” in the dialog.	
	6		System will validate data and remove classes in database, the grid will update with new class list.

Table 2.12. Use-case description for teacher to view students in class.

Use Case ID	**UC-12**		
Actor	Teacher		
Brief description	This use case for showing students in class.		
Pre-conditions	Users already logged into the system and went to class list page.		
Flow of events		**Actor Input**	**System Response**
	1	User clicks to “Folder” icon in column “Student list” of the grid.	
	2		System navigate to students page and shows students in this class.

Table 2.13. Use-case description for teacher to view assignments in class.

Use Case ID	**UC-12**		
Actor	Teacher		
Brief description	This use case for showing assignments in class.		
Pre-conditions	Users already logged into the system and went to class list page.		
Flow of events		**Actor Input**	**System Response**
	1	User clicks to “Folder” icon in column “Assignment list” of the grid.	
	2		System will navigate to assignment library page and show assignments in this class.

Table 2.14. Use-case description for add new student to class

Use Case ID	**UC-13**
Actor	Teacher
Brief description	This use case for add new student to class in the system.
Pre-conditions	Users already logged into the system and went to students page.

	Server received the request to create student from the teacher then save data into the database.		
Flow of events		**Actor Input**	**System Response**
	1	User clicks ”Add student” button.	
	2		System shows the add student form in a dialog.
	3	User fills the necessary information to the form and press Save”.	
	4		System will validate data and save data to the database. The grid of student list will update with new result.

Table 2.15. Use-case description for update student information in class

Use Case ID	**UC-13**		
Actor	Teacher		
Brief description	This use case for update student information in class in the system.		
Pre-conditions	Users already logged into the system and went to students page. Server received the request to update student from the teacher then save data into the database.		
Flow of events		**Actor Input**	**System Response**
	1	User select student to update in the grid.	
	2		System shows the selected student.
		User click “Edit” button.	
			System shows the update student form filled with old information in a dialog
	3	User fills the necessary information to the form and press Save” in the dialog.	
	4		System will validate data and save data to the

			database. The grid of student list will update with new result.

Table 2.16. Use-case description for delete students in class

Use Case ID	**UC-13**		
Actor	Teacher		
Brief description	This use case for delete students in class.		
Pre-conditions	Users already logged into the system and went to students page.		
Flow of events		**Actor Input**	**System Response**
	1	User select students to delete in the grid.	
	2		System shows the selected students.
	3	User click “Delete button”.	
	4		System will show a dialog to confirm before removing students.
		User click “Delete” Button in the dialog.	
			System save data to the database. The grid of student list will update with new result.

Table 2.17. Use-case description for add new assignment to class.

Use Case ID	**UC-13**		
Actor	Teacher		
Brief description	This use case for add new assignment to class in the system.		
Pre-conditions	Users already logged into the system and went to class list page. Server received the request to create assignment from the teacher then save data into the database.		
Flow of events		**Actor Input**	**System Response**
	1	User select classes to add assignment in the grid.	
	2		System shows the selected classes.
	3	User click “Add assignment” button.	
	4		System will naviagte to add

			assignment page.
		User fills the necessary information to the form and press Save” button.	
			System will validate data and save data to the database then navigate to assignment library page.

Table 2.18. Use-case description for import questions from excel file.

Use Case ID	**UC-13**		
Actor	Teacher		
Brief description	This use case for teacher to import questins from excel file.		
Pre-conditions	Users already logged into the system and went to add assignment page.		
Flow of events		**Actor Input**	**System Response**
	1	User clicks ”Add questions” button.	
	2		System shows the toolip to add questions options.
	3	User clicks ”Import SpreedSheet” button.	
	4		System will show the import question file dialog.
		User clicks “Download here” to down the question template and fill information.	
			System will download the template file.
		User click “Upload or drag file”.	
			System will will validate file format.
		User click “Upload” button.	
			System will read file to question list and display in the screen.

Table 2.19. Use-case description for active assignment.

Use Case ID	**UC-13**		
Actor	Teacher		
Brief description	This use case for teacher to active assignment in class.		
Pre-conditions	Users already logged into the system and went to assignment library page.		
Post-conditions	Server will send SMS to all parent that related to this assignment.		
Flow of events		**Actor Input**	**System Response**
	1	User clicks ”Active” button.	
	2		System shows the active assignment dialog.
	3	User fills the necessary information to the form and press Save”.	
	4		System will validate data and save data to the database.

Table 2.20. Use-case description for manage student learning result.

Use Case ID	**UC-13**		
Actor	Teacher		
Brief description	This use case for teacher to active assignment in class.		
Pre-conditions	Users already logged into the system and went to assignment library page.		
Post-conditions	Server will send SMS to all parent that related to this assignment.		
Flow of events		**Actor Input**	**System Response**
	1	User clicks ”Active” button.	
	2		System shows the active assignment dialog.
	3	User fills the necessary information to the form and press Save”.	
	4		System will validate data and save data to the database.

Table 2.21. Use-case description for student to login.

Use Case ID	**UC-13**		
Actor	Student		
Brief description	This use case fors student to login the mobile app.		
Pre-conditions	User who has account to access to application.		
Post-conditions	User logged in successfully.		
Flow of events		**Actor Input**	**System Response**
	1	User open the mobile app.	
	2		System shows the login form.
	3	User fills the necessary information to the form and press “Log in”.	
	4		System will validate data and redirect to Home screen.

Table 2.22. Use-case description for student to do the assignment.

Use Case ID	**UC-13**		
Actor	Student		
Brief description	This use case fors student todo the assignment.		
Pre-conditions	User who has account to access to application and went to home page.		
Flow of events		**Actor Input**	**System Response**
	1	User choose the assignment to do.	
	2		System shows the assignment question.
	3	User do each question and press “Next”.	
	4		System will validate data and display the result.

2.4. Class diagram

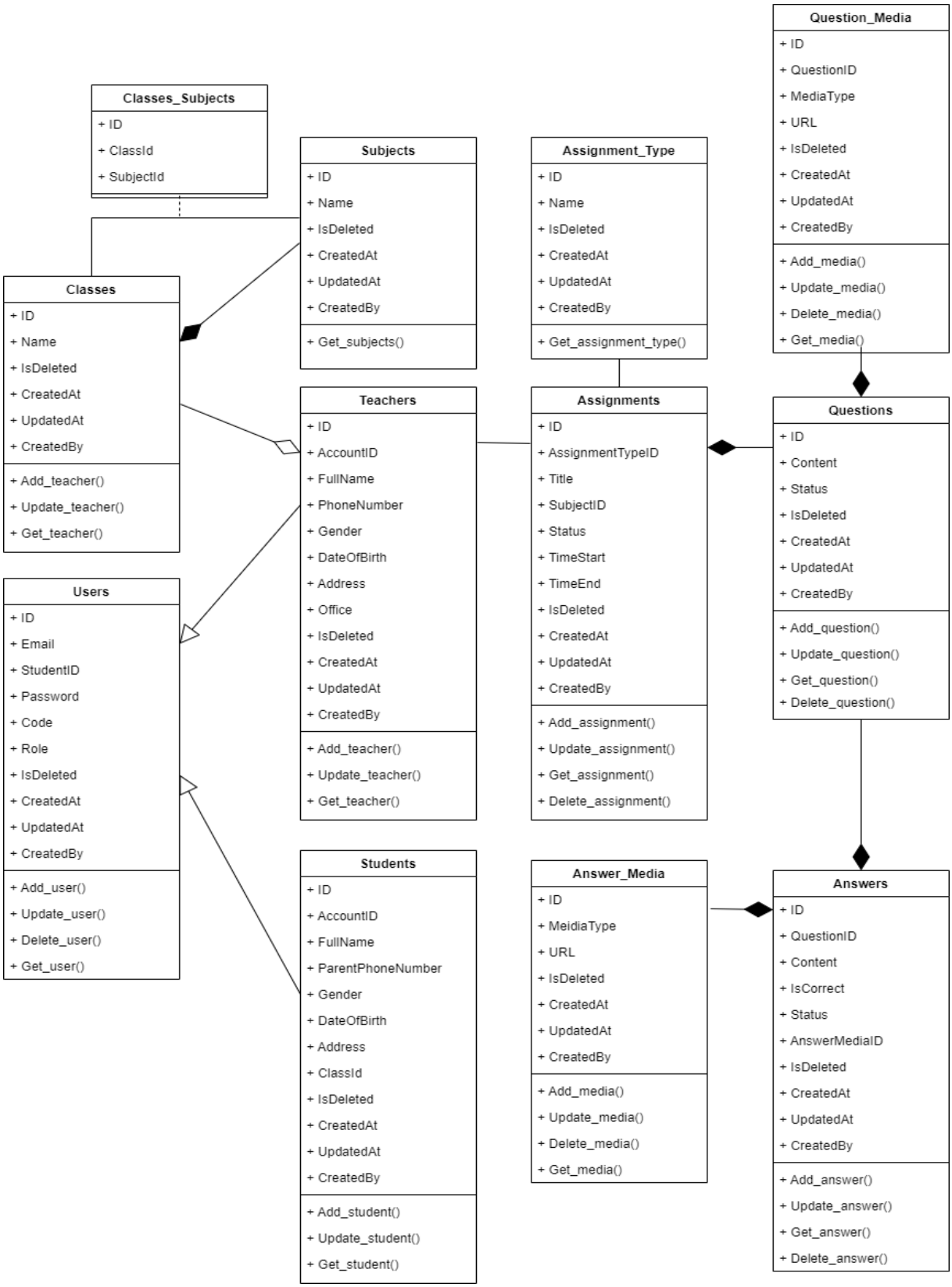


Figure 2.14. General class diagram

2.5. Sequence diagrams

2.5.1. Sequence diagrams for teacher to login to system

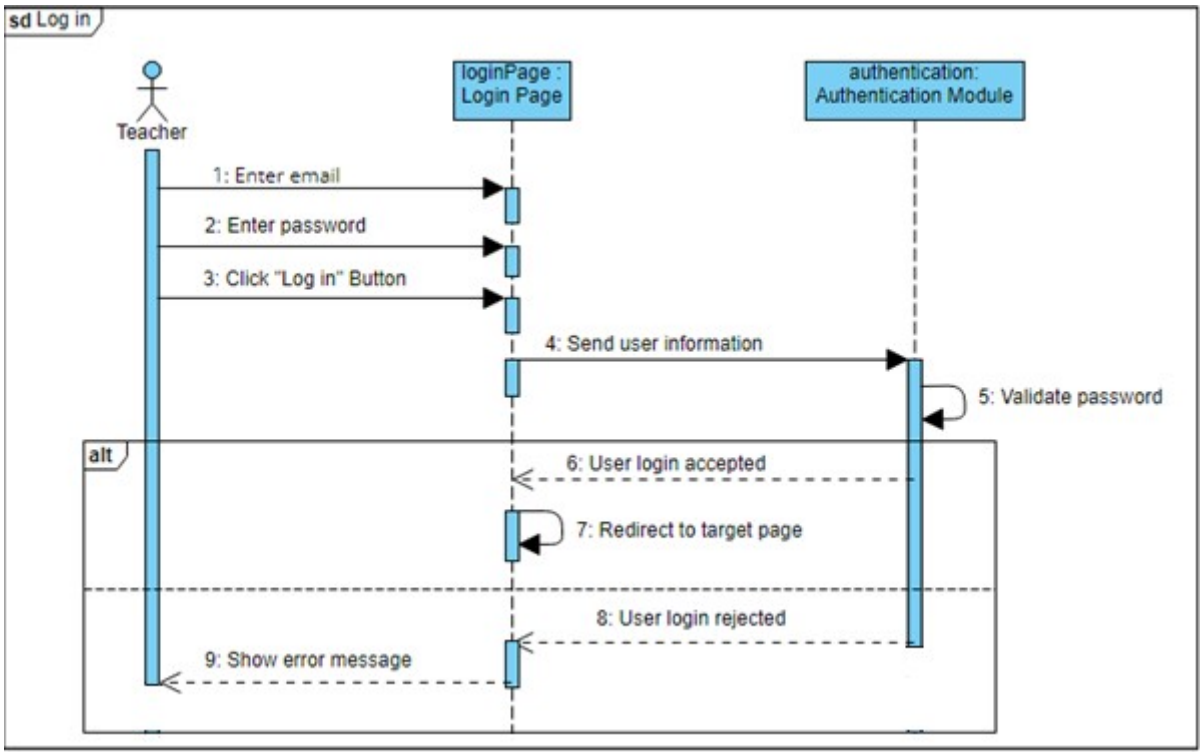


Figure 2.15. Sequence diagram for login

2.5.2. Sequence diagram for teacher to create class

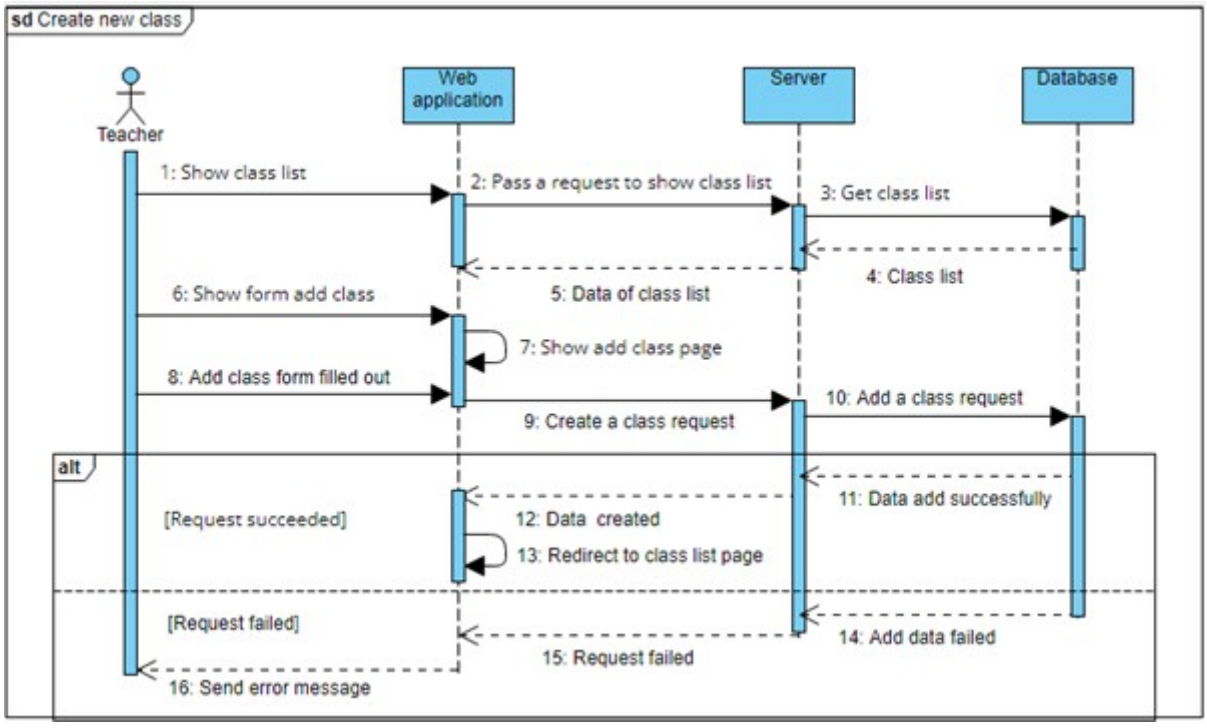


Figure 2.16. Sequence diagram for creating class

2.5.3. Sequence diagram for teacher to create student

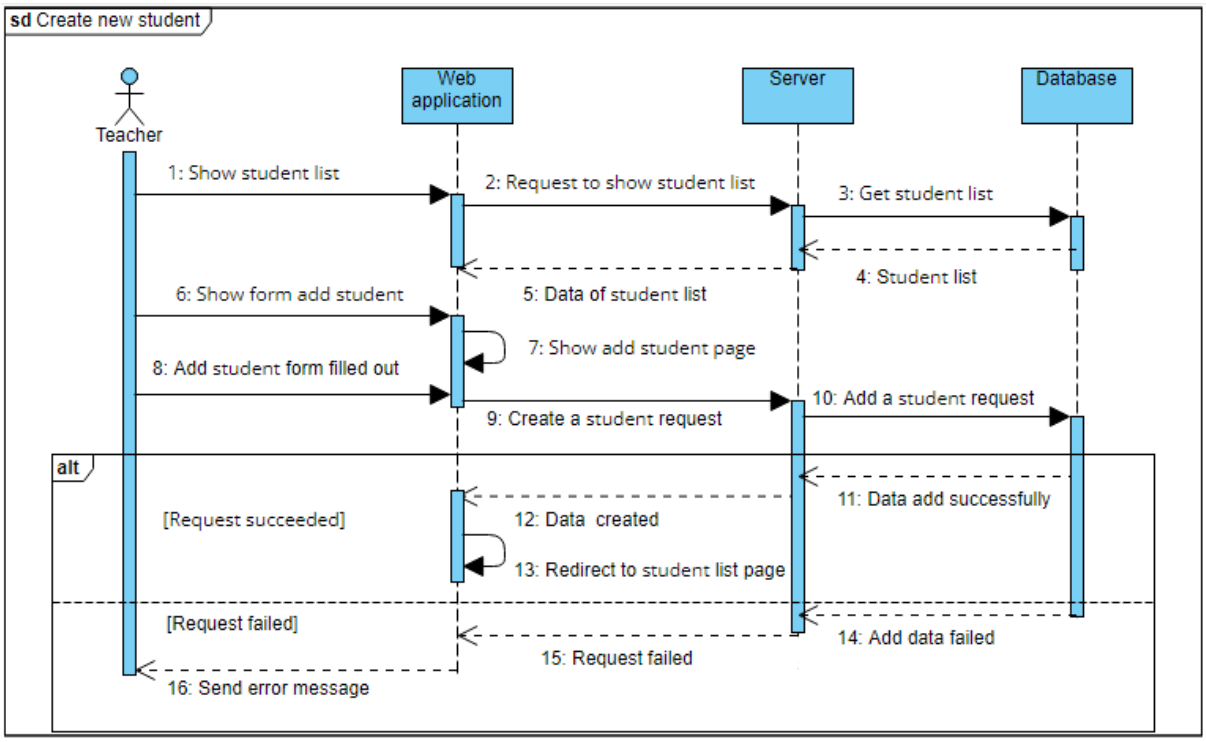


Figure 2.17. Sequence diagram for creating student

2.5.4. Sequence Diagram for teacher to create assignment

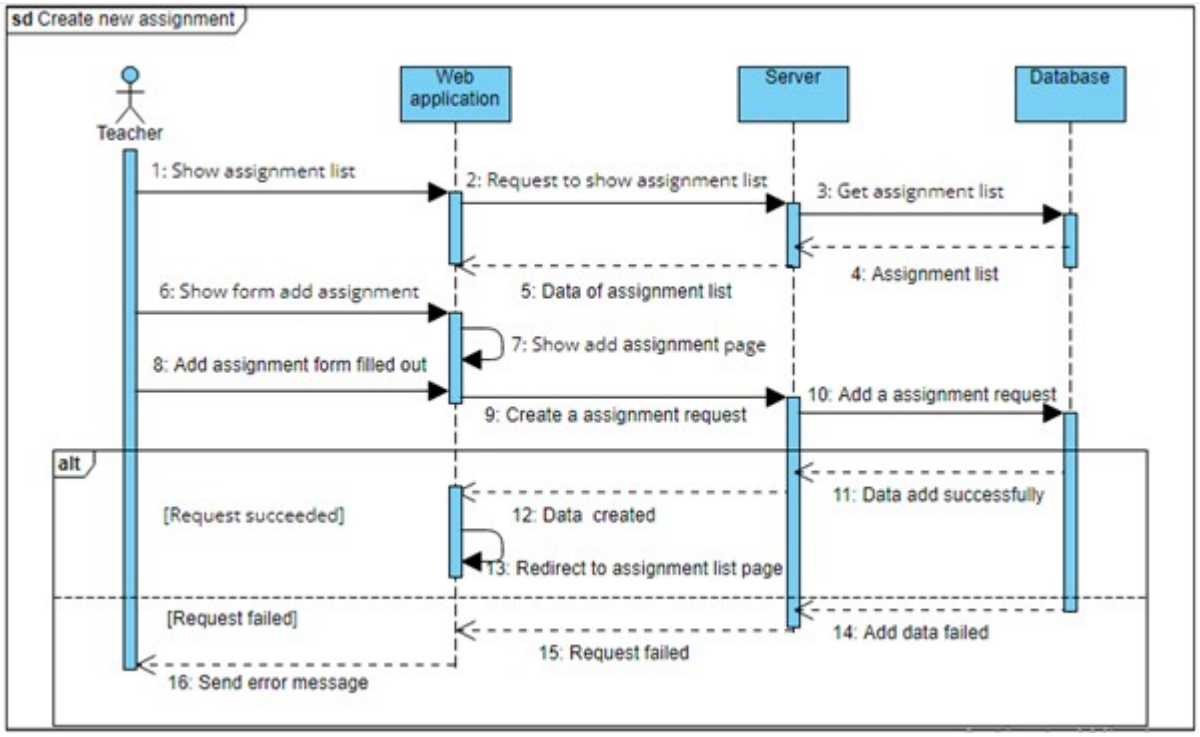


Figure 2.18. Sequence diagram for creating assignment.

2.6. Database design

2.6.1. Database design model

The system database will include the following tables:

- Account: table to save user account in system.
- Teacher_profile: table to save teacher profile information in system.
- Student_profile: table to save student profile information in system.
- Current_class: table to save class in formation in system.
- Subject: table to save subject in class.
- Assignment_type: table to save assignment type in system.
- Assignment: table to save assignment in system.
- Question: table to save question in system.
- Question_media: table to save question media information in system.
- Answer: table to save answer information in system.
- Answer_media: table to save answer media information in system.
- Class_assignment: table to save relationship of assignment and class.
- Lesson: table to save lesson information in system.
- Student_result: table to save student learning result.

2.6.2. Relationship between table

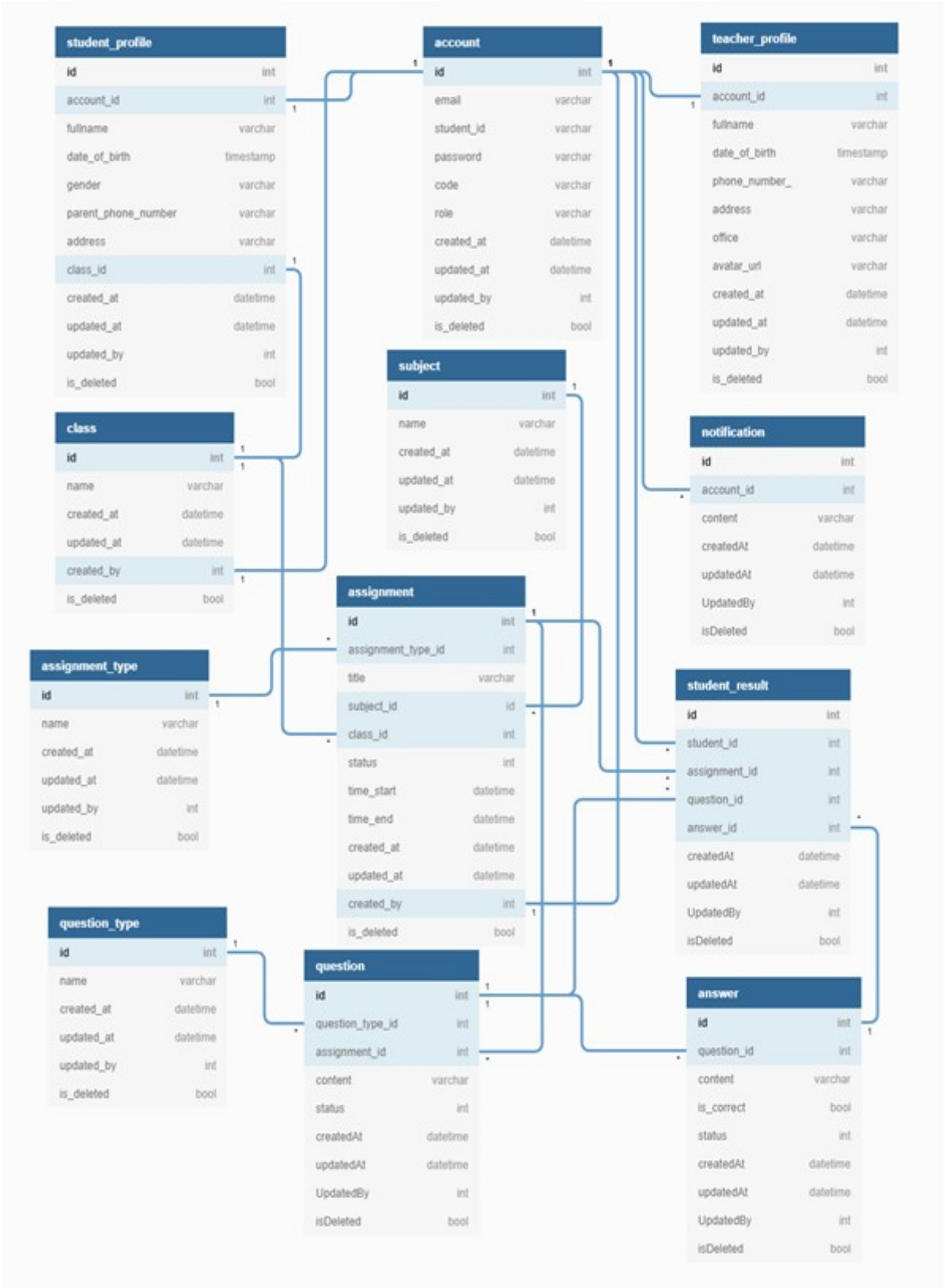


Figure 2.19. Relationship between tables

2.6.3. Description of the table

Table 2.23. Description of table: Account

No.	Name	Type	Length	Description	Key
1	id	uuid	255	Table's primary key	**PK**
2	email	text	255	Email	
3	studentId	text	10	Student ID	
4	password	text		Password	
5	role	text		Role of user	
6	isDeleted	boolean		Record is deleted or not	
7	createdAt	timestamp		Date created	
8	updatedAt	timestamp		Date updated	
9	createdBy	uuid		Created by someone	

Table 2.24. Description of table: teacher_profile

No.	Name	Type	Length	Description	Key
1	id	uuid	255	Table's primary key	**PK**
2	accountId	uuid	255	Account Id map to this profile	**FK**
3	fullName	text		FullName	
4	phoneNumber	text		Phone number	
5	gender	int		Gender	
6	dateOfBirth	timestamp		Date of birth	
7	address	text		Address	
8	office	text		Office	
9	createdAt	timestamp		Date created	
10	updatedAt	timestamp		Date updated	
11	isDeleted	boolean		Record is deleted or not	
12	createdBy	uuid		Created by someone	

Table 2.25. Description of table: student_profile

No.	Name	Type	Length	Description	Key
1	id	uuid	255	Table's primary key	**PK**
2	accountId	uuid	255	Account Id map to this profile	**FK**
3	fullName	text		FullName	
4	parentPhoneNumber	text		Parent’s phone number	
5	gender	integer		Gender	
6	dateOfBirth	timestamp		Date of birth	
7	address	text		Address	

8	currentClassId	uuid		Class code	**FK**
9	createdAt	timestamp		Date created	
10	updatedAt	timestamp		Date updated	
11	isDeleted	boolean		Record is deleted or not	
12	createdBy	uuid		Created by someone	

Table 2.26. Description of table: current_class

No.	Name	Type	Length	Description	Key
1	id	uuid	255	Table's primary key	**PK**
2	name	text		Name	
3	createdAt	timestamp		Date created	
4	updatedAt	timestamp		Date updated	
5	isDeleted	boolean		Record is deleted or not	
6	createdBy	uuid		Created by someone	

Table 2.27. Description of table: subject

No.	Name	Type	Length	Description	Key
1	id	uuid	255	Table's primary key	**PK**
2	name	text		Name	
3	createdAt	timestamp		Date created	
4	updatedAt	timestamp		Date updated	
5	isDeleted	boolean		Record is deleted or not	
6	createdBy	uuid		Created by someone	

Table 2.28. Description of table: class_subject

No.	Name	Type	Length	Description	Key
1	id	uuid	255	Table's primary key	**PK**
2	currentClassId	uuid	255	Class code	**FK**
3	subjectId	uuid	255	Subject code	**FK**
4	createdAt	timestamp		Date created	
5	updatedAt	timestamp		Date updated	
6	isDeleted	boolean		Record is deleted or not	
7	createdBy	uuid		Created by someone	

Table 2.29. Description of table: assignment

No	Name	Type	Length	Description	Ke

.					y
1	id	uuid	255	Table's primary key	**PK**
2	assignmentTypeId	uuid	255	Assignment type	**FK**
3	title	text		Title	
4	subjectId	uuid	255	Subject code	
5	status	integer		Status of assignment	
6	timeStart	timestamp		Start time	
7	timeEnd	timestamp		End time	
8	createdAt	timestamp		Date created	
9	updatedAt	timestamp		Date updated	
10	isDeleted	boolean		Record is deleted or not	
11	createdBy	uuid		Created by someone	

Table 2.30. Description of table: assignment_type

No.	Name	Type	Length	Description	Key
1	id	uuid	255	Table's primary key	**PK**
2	name	text		Name	
3	createdAt	timestamp		Date created	
4	updatedAt	timestamp		Date updated	
5	isDeleted	boolean		Record is deleted or not	
6	createdBy	uuid		Created by someone	

Table 2.31. Description of table: class_assignment

No.	Name	Type	Length	Description	Key
1	id	uuid	255	Table's primary key	**PK**
2	classId	uuid	255	Class code	**FK**
3	assignmentId	uuid	255	Assignment code	**FK**
4	createdAt	timestamp		Date created	
5	updatedAt	timestamp		Date updated	
6	isDeleted	boolean		Record is deleted or not	
7	createdBy	uuid		Created by someone	

Table 2.32. Description of table: question

No.	Name	Type	Length	Description	Key
1	id	uuid	255	Table's primary key	**PK**
2	questionTypeId	uuid	255	Question type	**FK**
3	assignmentId	uuid	255	Assignment code	**FK**
4	content	text		Content	
5	status	integer		Status	
6	createdAt	timestamp		Date created	
7	updatedAt	timestamp		Date updated	
8	isDeleted	boolean		Record is deleted or not	
9	createdBy	uuid		Created by someone	

Table 2.33. Description of table: question_type

No.	Name	Type	Length	Description	Key
1	id	uuid	255	Table's primary key	**PK**
2	name	text		Name	
3	createdAt	timestamp		Date created	
4	updatedAt	timestamp		Date updated	
5	isDeleted	boolean		Record is deleted or not	
6	createdBy	uuid		Created by someone	

Table 2.34. Description of table: question_media

No.	Name	Type	Length	Description	Key
1	id	uuid	255	Table's primary key	**PK**
2	mediaType	text	255	Media Type	
3	url	text	255	Media URL	
4	createdAt	timestamp		Date created	
5	updatedAt	timestamp		Date updated	
6	isDeleted	boolean		Record is deleted or not	
7	createdBy	uuid		Created by someone	

Table 2.35. Description of table: answer

No.	Name	Type	Length	Description	Key
1	id	uuid	255	Table's primary key	**PK**
2	questionId	uuid	255	Question code	**FK**
3	content	text		Content	
4	isCorrect	boolean		Answer is correct or not	
5	status	integer		Status	

6	answerMediaId	uuid	255	Answer media	**FK**
7	createdAt	timestamp		Date created	
8	updatedAt	timestamp		Date updated	
9	isDeleted	boolean		Record is deleted or not	
10	createdBy	uuid		Created by someone	

Table 2.36. Description of table: answer_media

No.	Name	Type	Length	Description	Key
1	id	uuid	255	Table's primary key	**PK**
2	mediaType	text		Media type	
3	url	text		Meida URL	
4	createdAt	timestamp		Date created	
5	updatedAt	timestamp		Date updated	
6	isDeleted	boolean		Record is deleted or not	
7	createdBy	uuid		Created by someone	

Table 2.37. Description of table: student_result

No.	Name	Type	Length	Description	Key
1	id	uuid	255	Table's primary key	**PK**
2	AccountId	uuid	255	Account code	**FK**
3	AssignmentId	uuid	255	Assignment code	**FK**
4	Result	float		Result	
5	createdAt	timestamp		Date created	
6	updatedAt	timestamp		Date updated	

Table 2.38. Description of table: lesson

No.	Name	Type	Length	Description	Key
1	id	uuid	255	Table's primary key	**PK**
2	AssignmentId	uuid	255	Assignment code	**FK**
3	Classes	text		Classes contain lesson	**FK**
4	Subject	text		Subject of lesson	
5	Title	text		Title of lesson	
6	LessonTime	text		Lesson time	
7	AttachmentURL			Attachment file	
8	VideoURL			Video of lesson	
9	createdAt	timestamp		Date created	
10	updatedAt	timestamp		Date updated	
11	isDeleted	boolean		Record is deleted or not	
12	createdBy	uuid		Created by someone	

2.7. Activity diagram for teacher

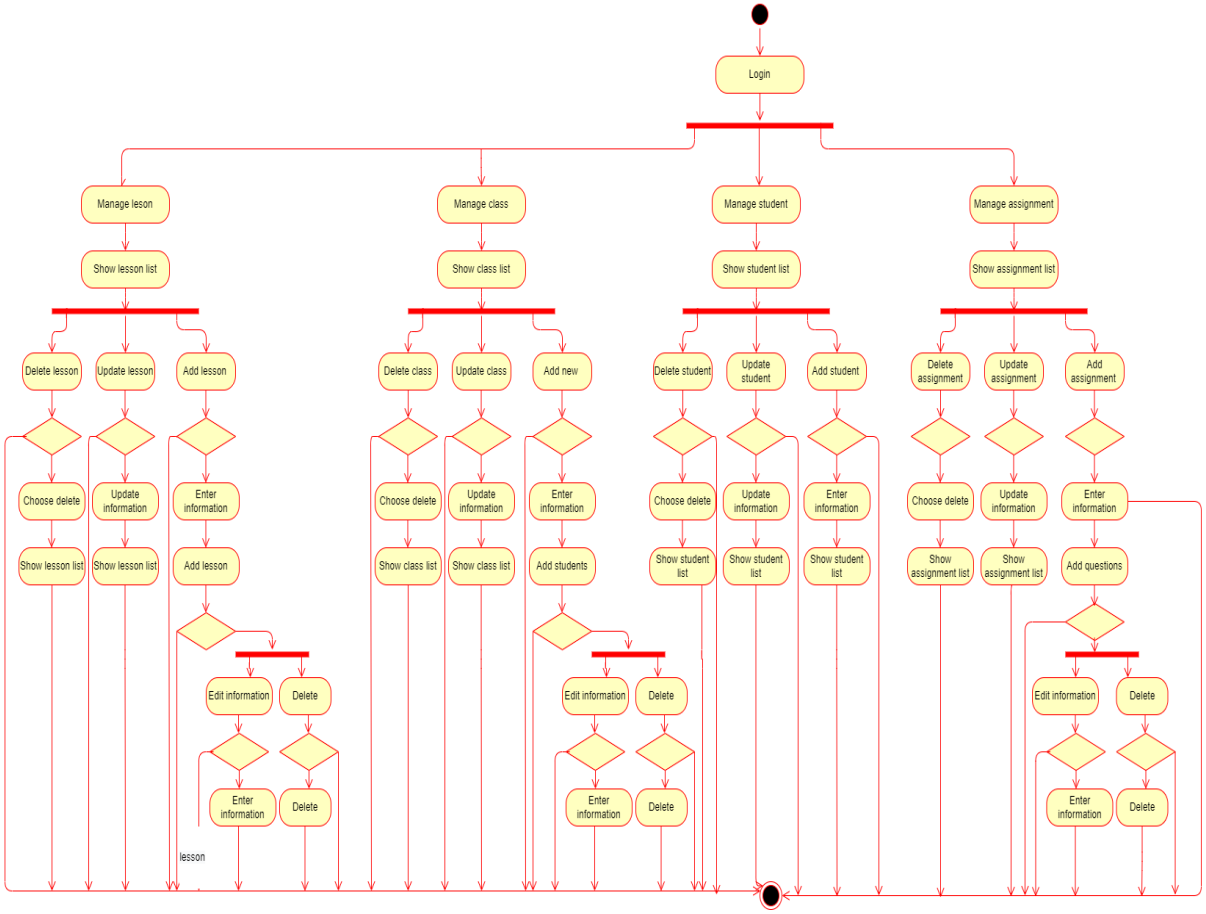


Figure 2.20. Activity diagram for teacher

2.8. Conclusion

This chapter presented the requirements specification that the system could meet the user' demands. Follow the requirements, the use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. The action sequences as well as the interactions between user and user, user and system. Thereby, the overview and the activity streams of the system are fully presented.

Besides, it also describes the system structure, as well as the action sequences for each function. By that, it facilitates the testing phase, the tester can go back the sequence diagrams to follow the action sequences and create the function tests and the input data as well. Furthermore, it shows the database and some UI application designs to help the reader have clearer views of the system. By that, we can consider and evaluate the complexity of this system.

IMPLEMENTATION AND EVALUATION

3.1. Development environment

Web service

The web service is a RESTful API. It is built by .NET Entity Framework, basing on representational state transfer (REST) technology. I use SQLServer to store data.

Software Development Tools

We use below tools: Visual Code, Visual Studio Code, SQL Server Management Studio and Postman.

3.2. Demo main feature

3.3.1 Web application for teacher

Login

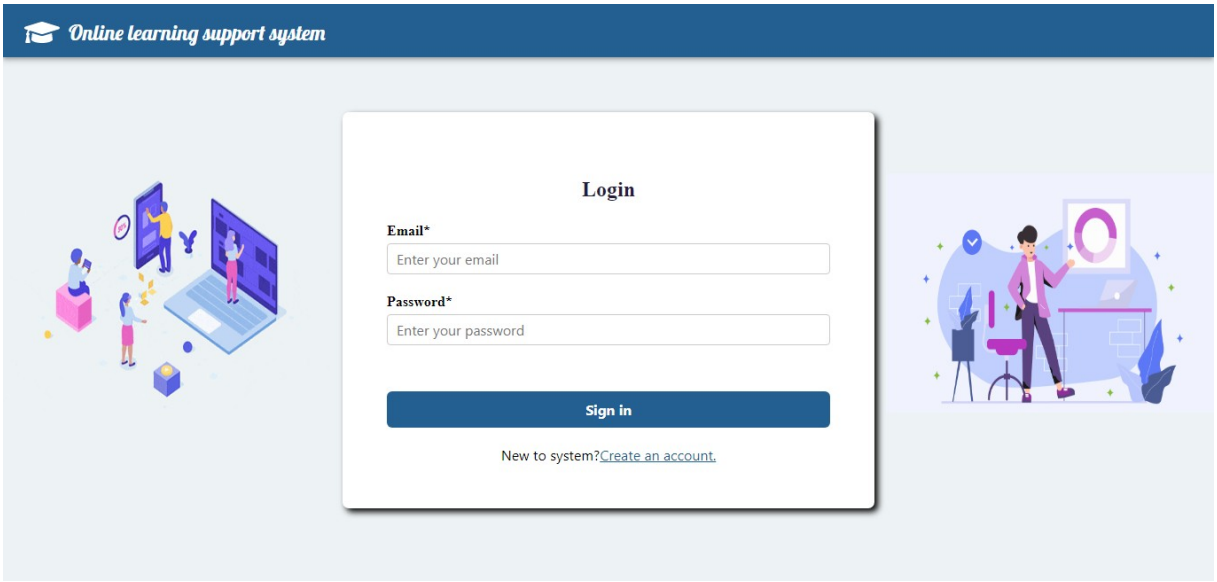


Figure 3.21. Login page

The website uses email and password to log in. After login successful, sytem will direct to home page.

Register page

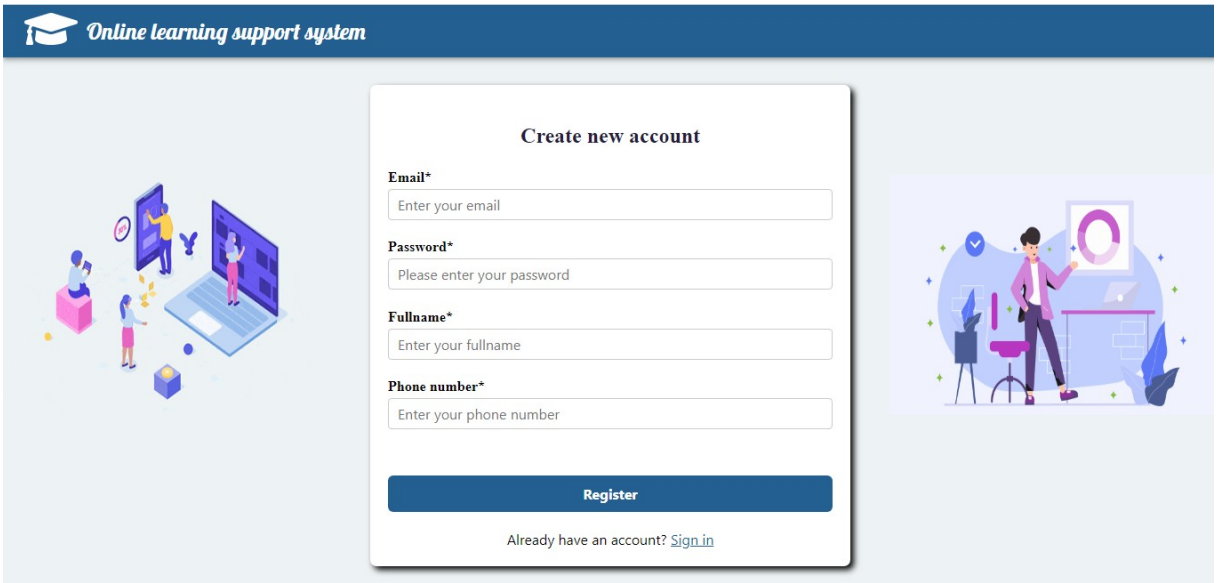


Figure 3.22. Register page

If users do not have an account, the user clicks "Create an account" on the login page, the registration form is displayed, the user fills in the information and registers an account. After successful registration, the user will go to the login page, enter the newly created account and log in to the system.

Home page

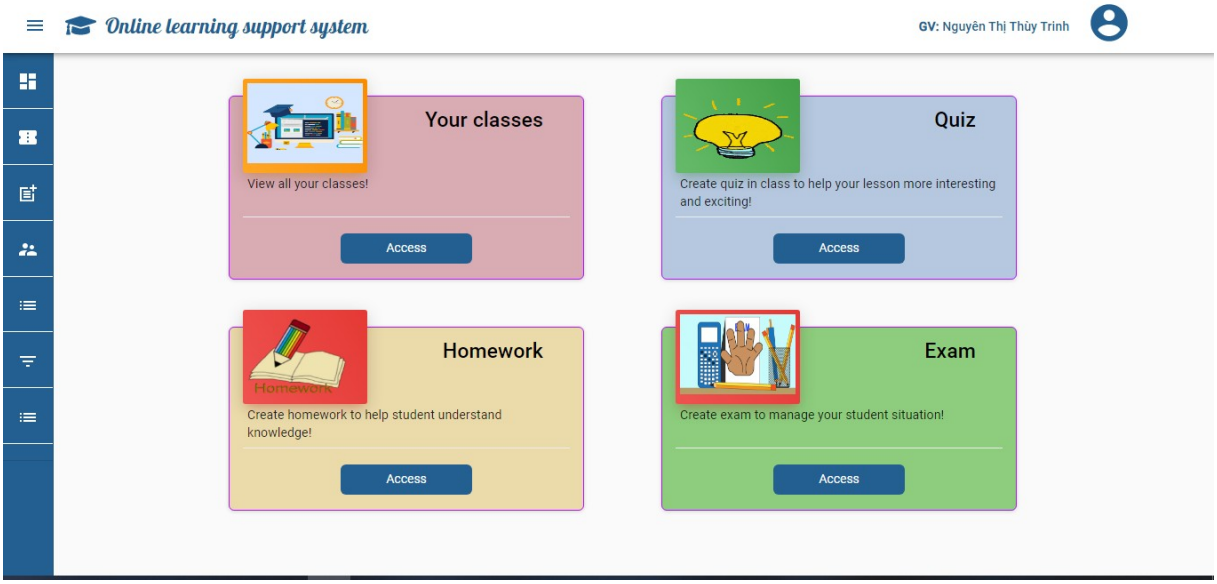


Figure 3.23. The overview about the home page

The home page is acceptable for teacher who logged in. It incluses four fast access card to class list of this teacher, quiz, homework and examination in the assignment library.

Teacher profile page

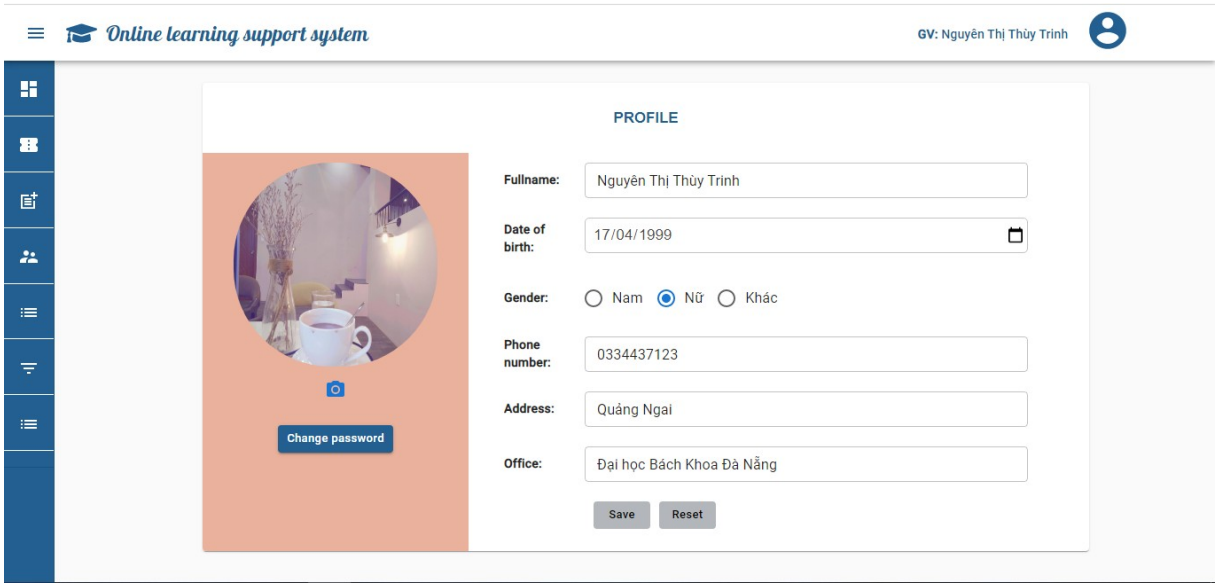


Figure 3.24. Teacher profile page.

Teacher can access to profile by click on the avatar icon and select “Profile”. Teacher can update personal information and save it.

Teacher change password word page

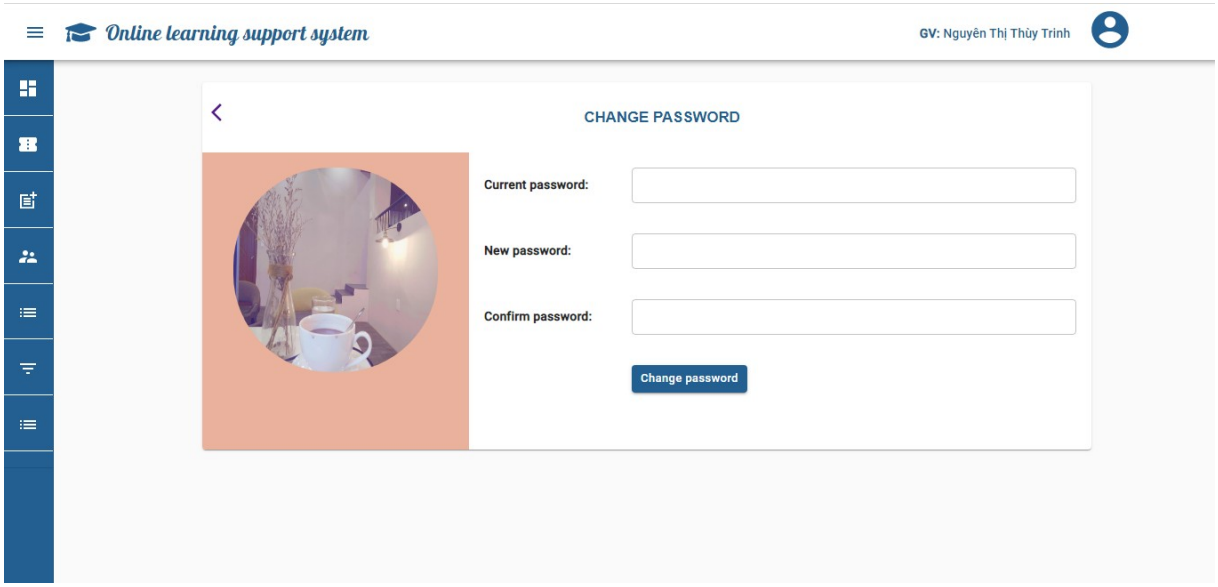
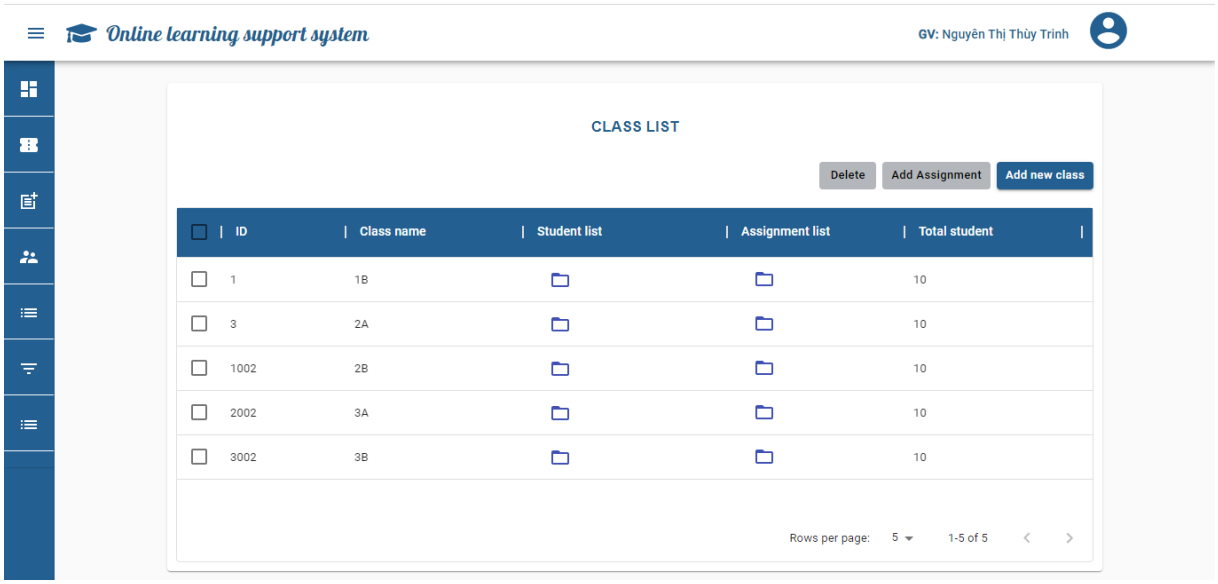


Figure 3.25. Change password page

Teacher can change password by click on “Change password” button in profile screen.

Classes page



This is class list page, teacher can select classes and delete, update, add assignment to classes. Teacher can see the students and assignments in class by click folder icon in student list and assignment list column.

Figure 3.26. Class list page

Create class page.

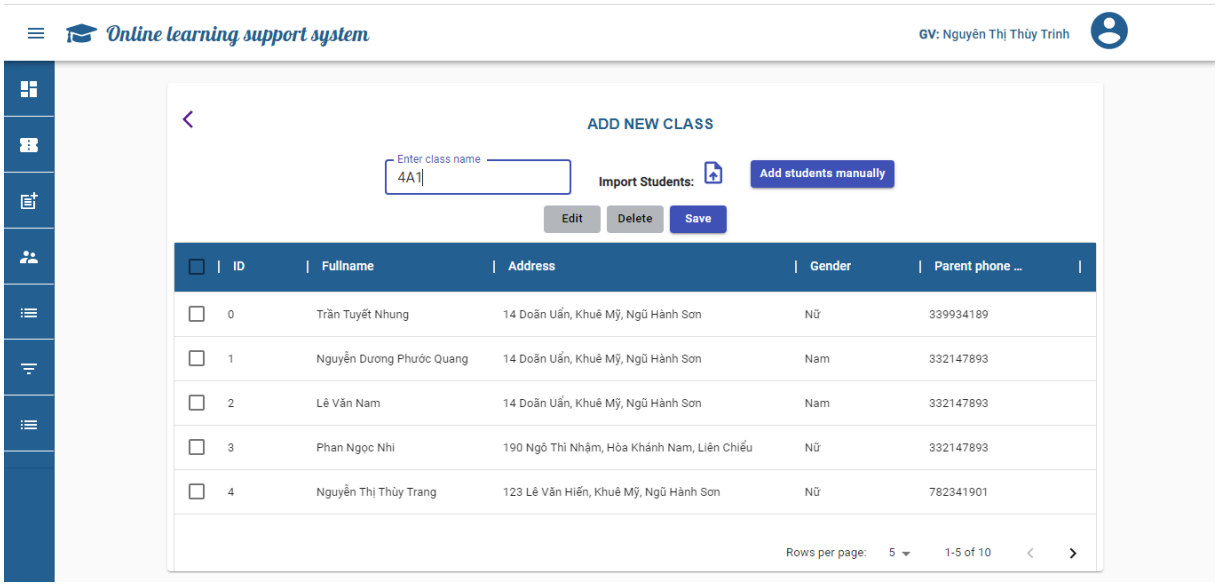


Figure 3.27. Creating class page

This is add new class page. To add new class, teacher enter class name and add students by two ways: import student from excel file or add manually by the spreedsheet.

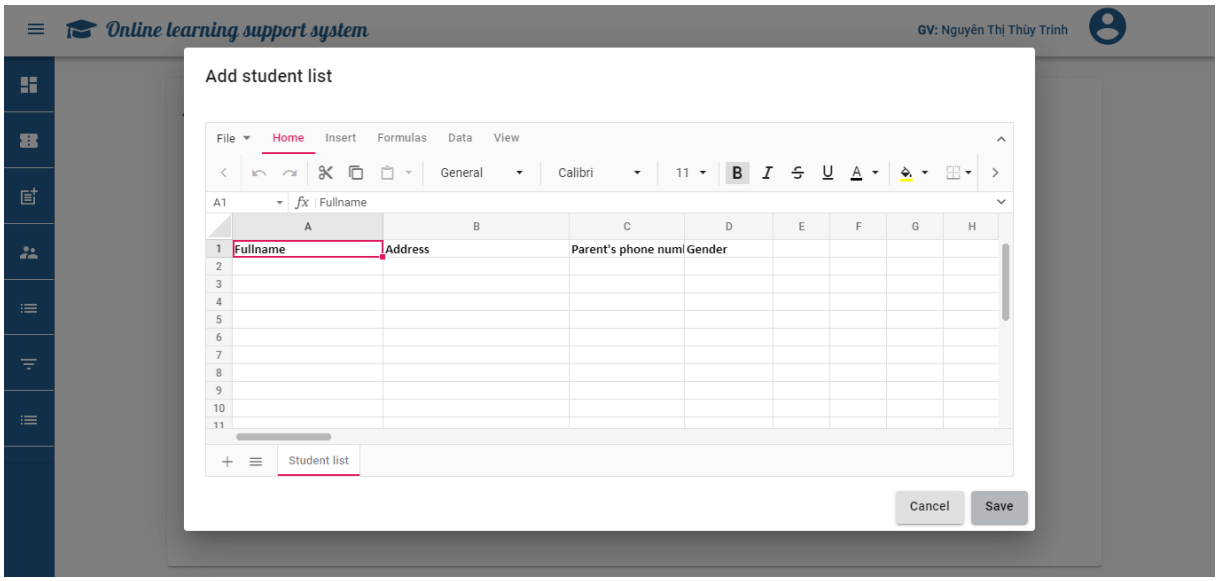


Figure 3.28. Create students with spreedsheet

Student list page

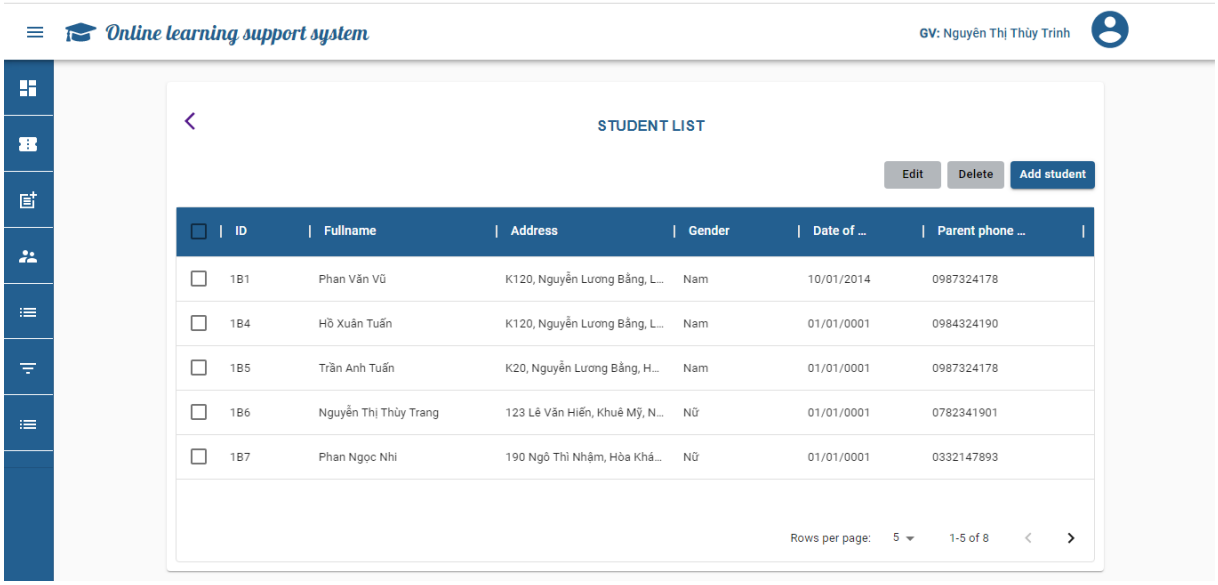


Figure 3.29. Student list page.

This is student list page, teacher can select student to update, detele and add new student to class. Below is the form of update student, the add student form is the same as update form.

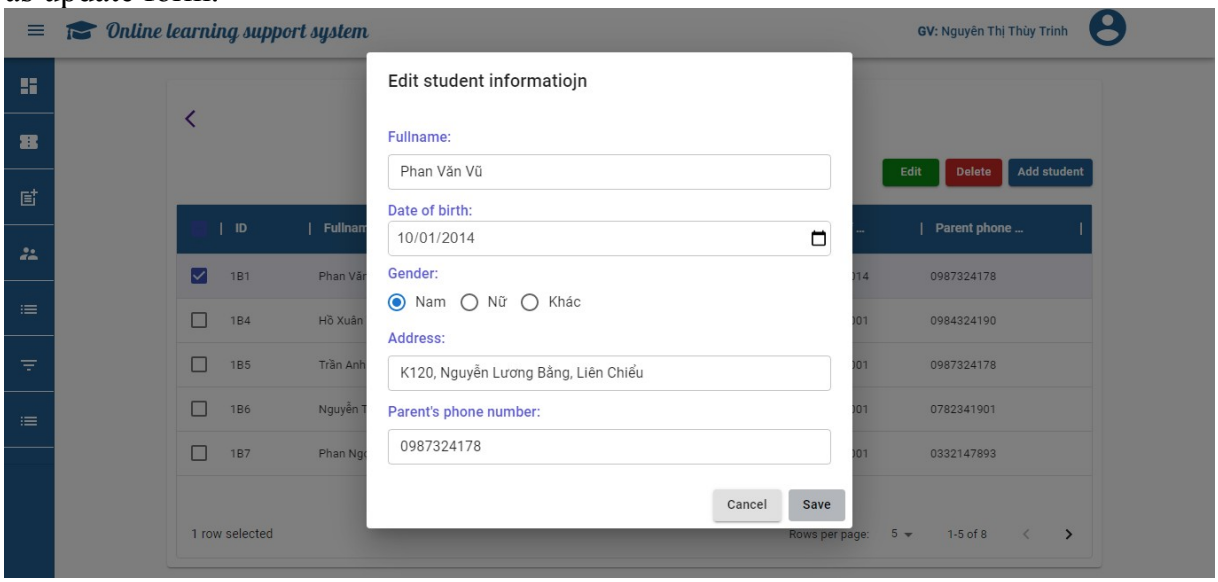


Figure 3.30. Update student information form

This is update student form, the save button will enable when user change information in form.

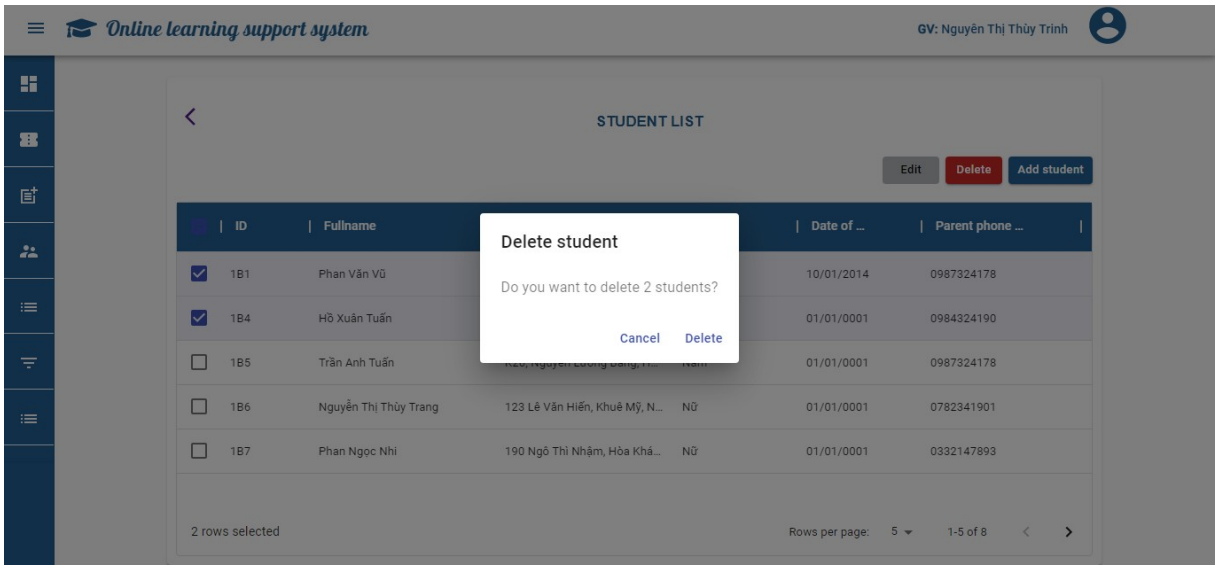


Figure 3.31. Delete student confim dialog

This is confirm dialog before deleting student.

Add assignment page

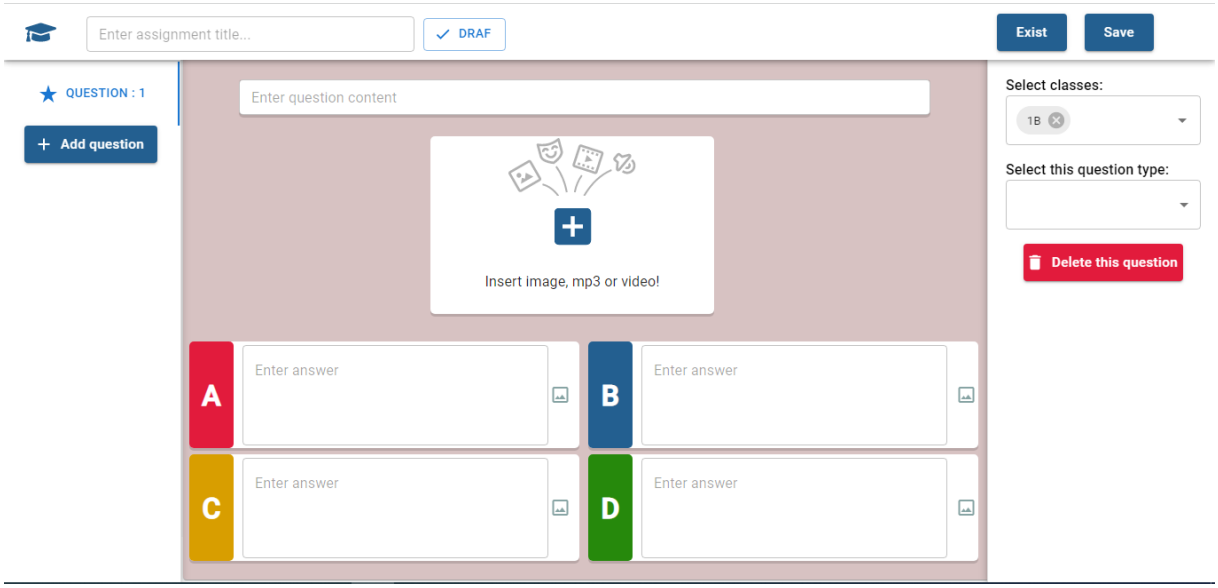


Figure 3.32. Add assignment page

This is add assignment page, teacher can add new question in the left sidebar, choose class and question type in the right sidebar. Question can be text, image, audio or video. Answer can be text or image.

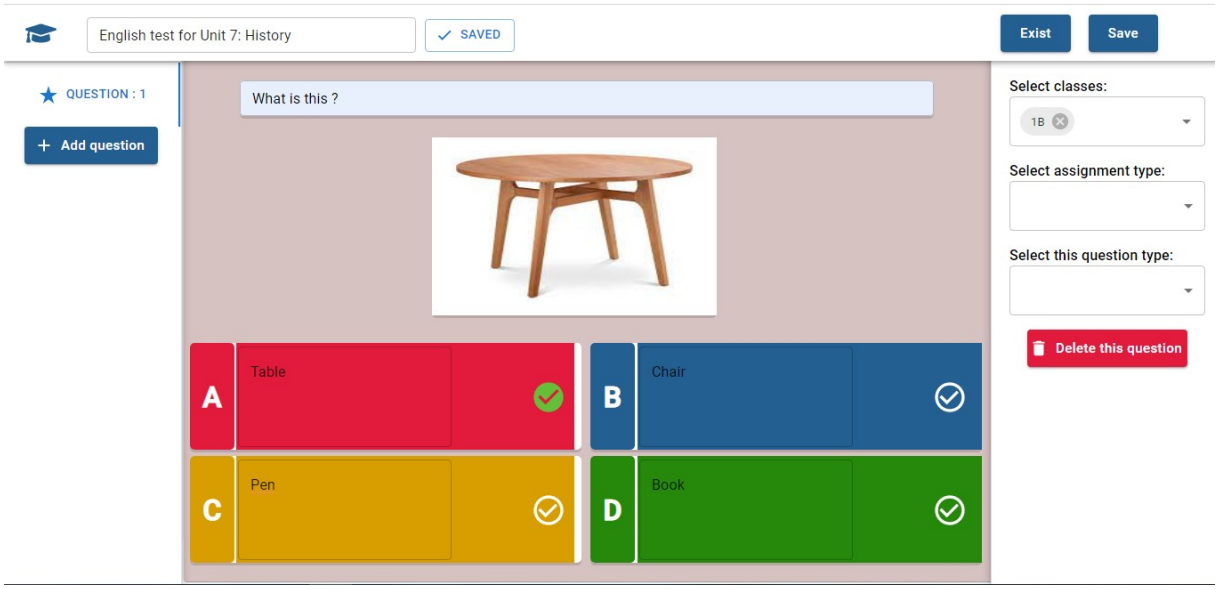


Figure 3.33. Example of questions and answers.

This is example of questions and answers.

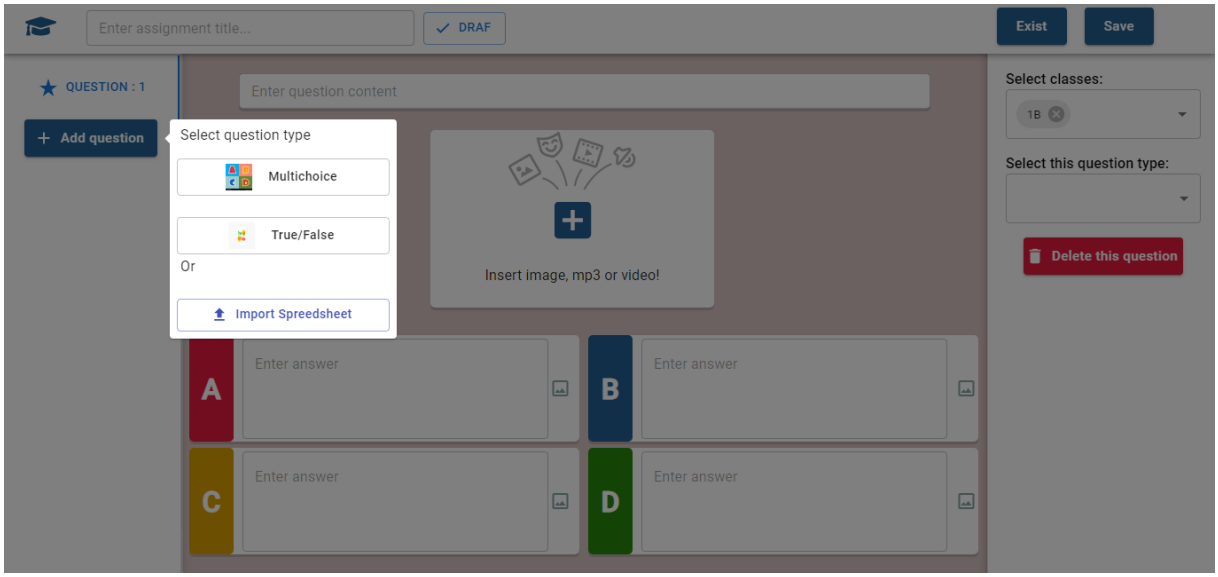


Figure 3.34. Add questions options.

When add question, teacher can choose the question type to add one by one or import questions from spreedsheet. Below is the steps to add questions from spreedsheet.

Import question dialog

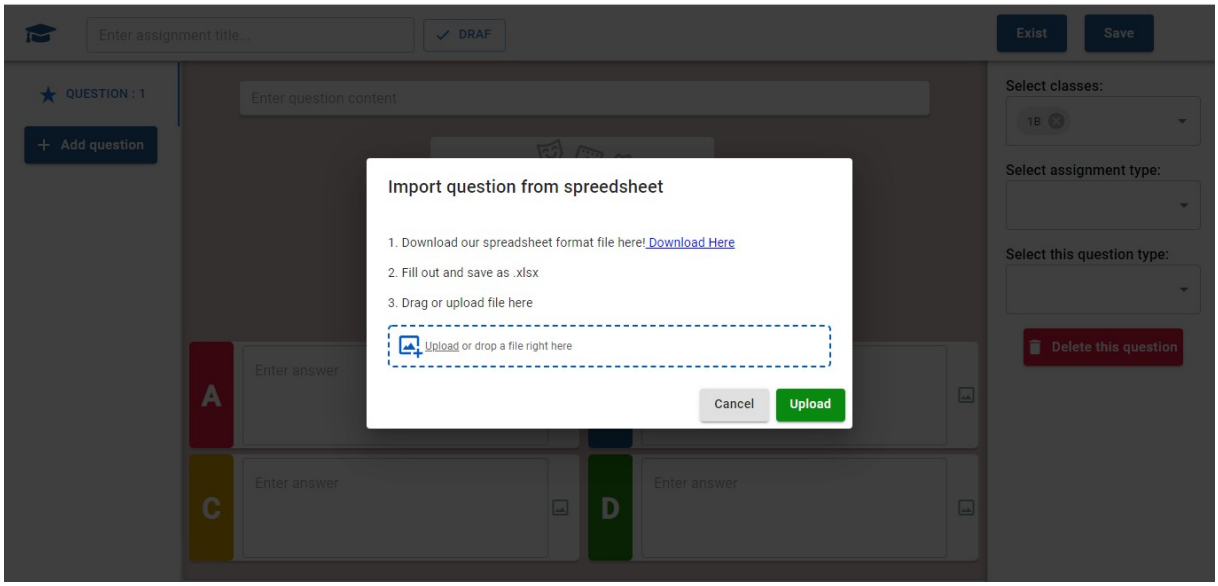


Figure 3.35. Import question dialog.

Assignment library page

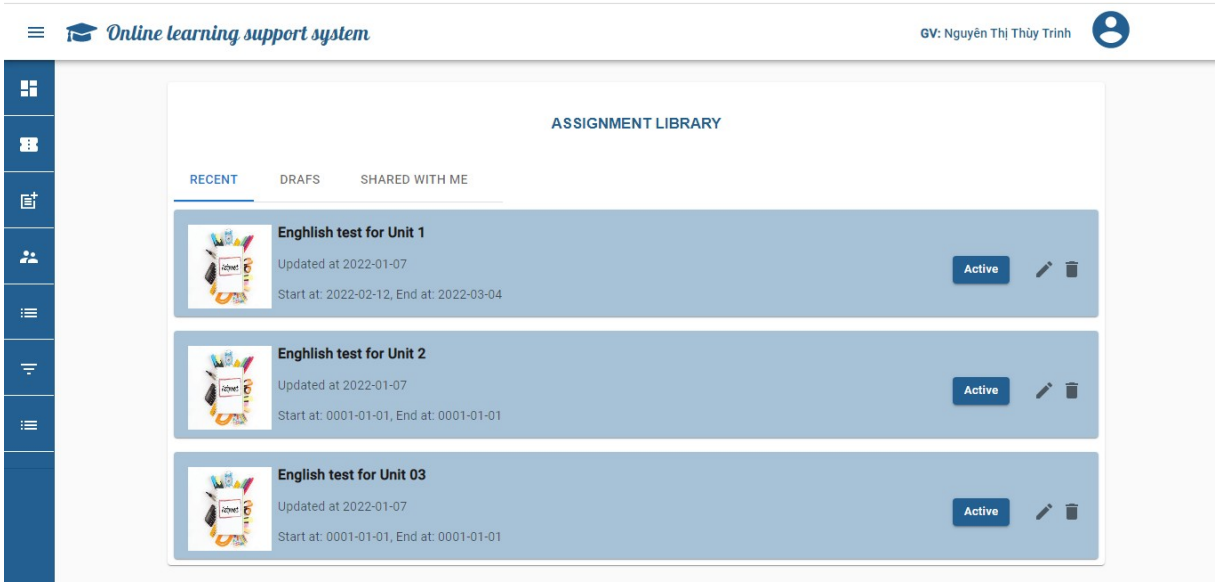


Figure 3.36. Assignment library page.

Teacher can access to assignment library in the sidbar and view all assignment, active assignment. When active assignment, if teacher choose to send SMS to parent, a message to notify about new assignment will be sent to parent.

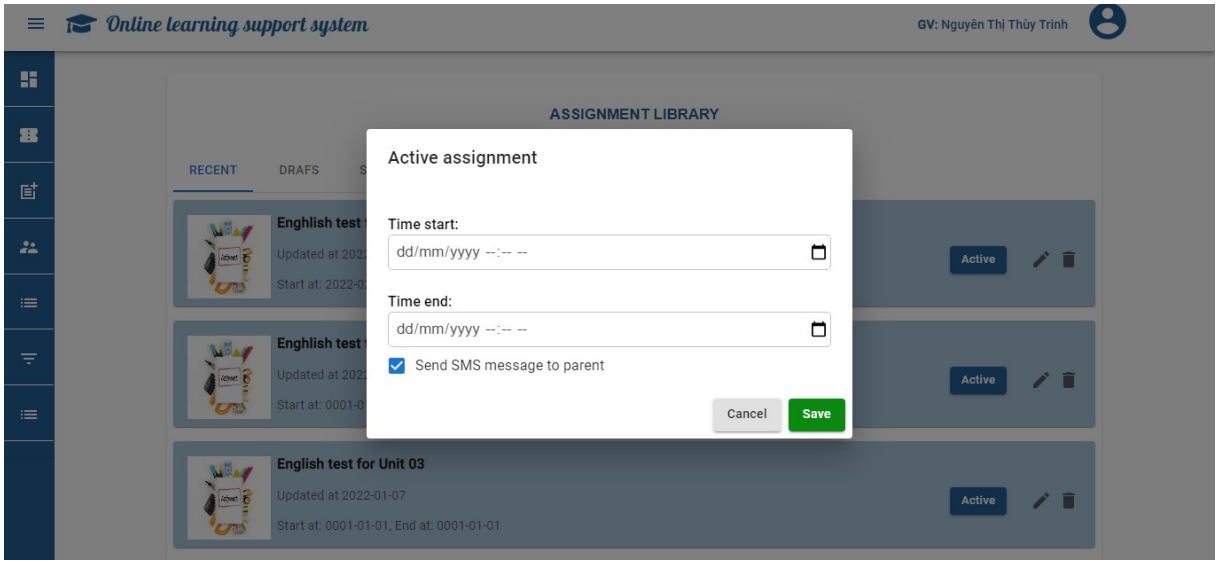


Figure 3.37. Active assignment dialog

3.3.2. Mobile application for student

Figure 3.38. Login and Home screen in mobile application.

For student, to access to mobile app to do the assignment, enter student ID and password to login, after login, the assignment list will display in home page and student choose assignment to do.

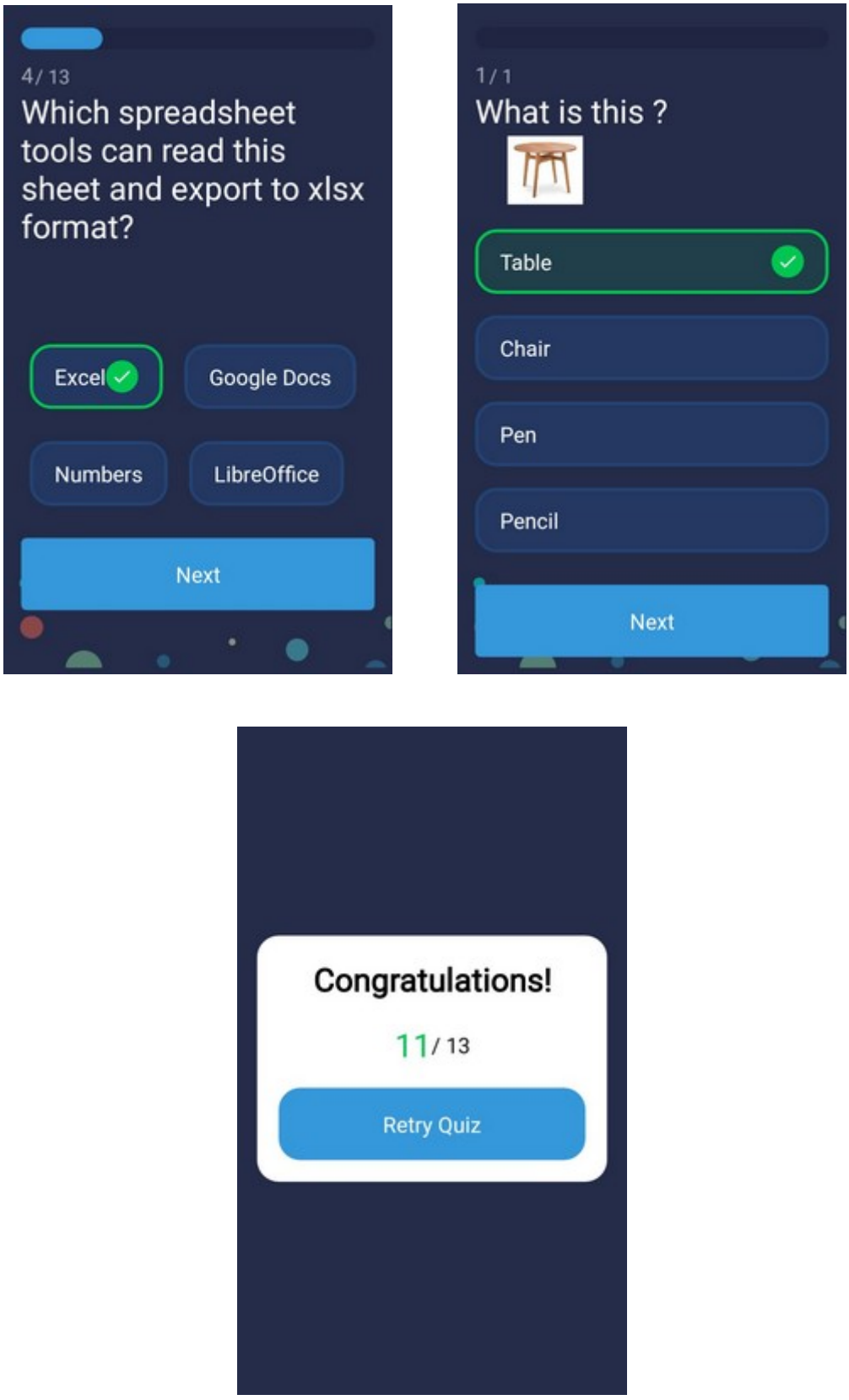


Figure 3.39. Access assignment in mobile application

These are the steps to do question in assignment. Students choose one answer and press “Next” button to do the next question, total result will display when complete all question.

3.3. Evaluation

3.4.1. Advantages

Basically, *"The online learning support system for primary and secondary school*" meets the needs of teacher and student with online learning.

In terms of UX/UI, our system provides simple and familiar UIs for users to interact and get information through a website.

In my system, As a teacher you can manage class, student, create assignment with question in variety type and manage student learning result. As a student, you can access to do assignment and view learning result. As a parent, you can receive notification about new assignment of your children.

3.4.2. Disadvantages

Besides the advantages mentioned above, my system still has some disadvantages:

No real-time questions.

Some functions are not yet fully developed.

CONCLUSION

1. Achievement

During the time of researching, researching the theoretical basis and deploying technology application, the project has achieved the following results:

Theoretical side: After building this system, I could understand how to work with some frameworks and libraries such as .NET Entity Framework, ReacJS and React Native. I learned about the structure of the frameworks and how to apply them to my project. Besides, it also helps me to update technology trend and my future career.

Application: Built a system that allows teachers to manage student information, add assignments and track student learning results, besides that students can access and do assignments, notification system for parents.

The user interface does not have too many complicated operations, making it easy for users to use. The system has functions for users who are teachers, students and parent.

During the process of making the project, I have learned and improved many skills such as English skills, self-study and research, planning, presentation skills,...

2. Future work

With some disadvantage I have already mentioned in evaluation part, the future work of this project is as follow:

Firstly, integrate functions to add real-time question during the lessons that student can access and do it to understand the lesson.

Secondly, integrate share assignment function between teachers so that they can save time to create assignment.

Finally, develop function for classification of student learning outcomes according to learning result.

REFERENCES

[1] Le Thi My Hanh, Textbook of software engineering, Faculty of Information Technology, University of Science and Technology, University of Danang.

[2] .NET: https://www.entityframeworktutorial.net/what-is-entityframework.aspx , access on October 01st, 2021.

[3] Material UI: https://mui.com/getting-started/installation , access on October 01st, 2021.

[4] ReactJS: https://reactjs.org/ , access on December 01st, 2021.

[5] React Native: https://reactnative.dev/ , access on January 01st, 2021.

[6] RESTful APIs: https://techtalk.vn/thiet-ke-restful-apis.html, access on October 01st, 2021.