

UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI



## **Group Project Final Report**

Group 1 – ICT

### **Web Platform for Checking attendance system by using GPS**

#### *Members*

BI11-110	Phùng Gia Huy
BI11-195	Nguyễn Hoài Nam
BI11-198	Hoàng Đức Nghĩa
BI11-236	Bùi Hồng Sơn
BI11-273	Nguyễn Đăng Trung
BI11-286	Nguyễn Xuân Vinh

#### *Supervisor*

Dr. Đoàn Nhật Quang

**December, 2022**

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

# Content

<b>I. INTRODUCTION</b>	5
1. Context and Motivation	5
2. Thesis Structure	6
<b>II. OBJECTIVES</b>	7
1. Desired Features	8
2. Expected Outcome	9
<b>III. REQUIREMENT ANALYSIS</b>	10
1. General System Requirements	10
2. Use Cases	11
2.1. Use case Diagram	11
2.2. Users Characteristics	12
2.3. Use case Description	12
2.4. Use Case and Scenario Description	13
<b>IV. METHODOLOGY</b>	17
1. Tools and techniques	17
1.1. Figma	17
1.2. HTML	17
1.3. CSS	18
1.4. JavaScript	18
1.5. MySQL Workbench	18
1.6. PHP	19
2. System Architecture	19
2.1. Client Side	20
2.2. Server Side	20
3. Database Design	21
4. Use Cases Implementation	27
4.1. Login	27
4.2. Edit profile	29
4.3. Register	31
4.4. Report	33
4.5. Attendance Information	35
4.6. Course – Student	36
4.7. Course – Professor	39
4.8. Generate QR – Scan QR	42
<b>V. RESULT AND DISCUSSION</b>	45
1. Result	45
2. Discussion	46
<b>VI. CONCLUSION AND FUTURE WORK</b>	48
1. Conclusion	48
2. Future Work	48
<b>VII. APPENDIX – USER INTERFACE</b>	49
<b>VIII. REFERENCES</b>	69

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

## List of Figures

1	Use Case Diagram	11
2	System Architecture Diagram	19
3	Front-end Component Design	20
4	Overall Database Design	21
5	Login Sequence Diagram	27
6	Login Activity Diagram	28
7	Edit Profile Sequence Diagram	29
8	Edit Profile Activity Diagram	30
9	Register Sequence Diagram	31
10	Register Activity Diagram	32
11	Report Sequence Diagram	33
12	Report Activity Diagram	34
13	Attendance Information Sequence Diagram	35
14	Attendance Information Activity Diagram	35
15	Course-Student Sequence Diagram	37
16	Course-Student Activity Diagram	38
17	Course-Professor Sequence Diagram	40
18	Course-Professor Activity Diagram	41
19	Generate QR – Scan QR Sequence Diagram	43
20	Generate QR – Scan QR Activity Diagram	44
21	Login User Interface (both Student and Professor)	49
22	Forget Password User Interface (both Student and Professor)	50
23	Sign Up Student User Interface	51
24	Welcome Student User Interface	52
25	Edit Information Student User Interface	53
26	Attendance Information Student User Interface	54
27	Report Student User Interface	55
28	Course User Interface (both Student and Professor)	56
29	Course User Interface (after choosing Year & Major) (both Student & Professor)	57
30	Subject Student User Interface	58
31	Join Class Student User Interface	59
32	Class User Interface (both Student and Professor)	60
33	Attendance Checking User Interface (both Student and Professor)	61
34	List Participants User Interface (both Student and Professor)	62
35	Welcome Professor User Interface	63
36	View Information professor User Interface	64
37	Subject Professor User Interface	65
38	Create Class Professor User Interface	66
39	Generate QR Code Professor User Interface	67
40	Checking Attendance Form Student User Interface (after scanning QR)	68

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

## List of Tables

1	Use Case Table	12
2	Scenario Description of UC.1	13
3	Scenario Description of UC.2	13
4	Scenario Description of UC.3	14
5	Scenario Description of UC.4	14
6	Scenario Description of UC.5	14
7	Scenario Description of UC.6	15
8	Scenario Description of UC.7	16
9	Training Program Database Design	22
10	Major Database Design	22
11	Subject Database Design	23
12	Subject Class Attendance Database Design	23
13	Class Database Design	24
14	Class Students Database Design	24
15	Attendance Database Design	25
16	Students Database Design	25
17	Report Database Design	26
18	Teachers Database Design	26

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

## **I. INTRODUCTION**

### *1. Context and Motivation*

Checking attendance is a daily activity in university to supervise the learning process of students. In recent years, the number of students is gradually increasing, especially ICT students; and this can cause enormous pressure on the administrations or managers to monitor and arrange the process manually. It is so hard to govern all the information related to attendance accurately and on time.

In the University of Science and Technology of Hanoi (USTH), that the entire admission process is handled manually is time-consuming and exhausting. Each year, hundreds of students stand in queue waiting for their turn to check attendance, which may be many times in the class. In each class, the majority of professors have to check attendance by calling each of the students' names. It could lead to some problematic matters. Firstly, if the list of students of each class is too long, the professor will spend too much time on checking attendance. Secondly, this method could make it difficult for the professors or university to manage all the students' attendance in the class. Fingerprint checkers can be a solution, but from my point of view, this type of device could cost an arm and a leg. Therefore, at least at the moment, we should find a solution to solve this issue in an effective and cheap way.

In this project, a Web Platform for Checking attendance system by using GPS is anticipated to resolve the mentioned problem. With this system, the university or the professor can easily manage the information and the data could become more precise. The professor could generate the QR code and show it on the slide. If students want to check attendance, they just scan this QR, with the account registered before and turning on the GPS to get the precise location. By this way, the professor can find the checking attendance less time, more convenient and precise. Also, professors can check attendance many times in the class to monitor the attendance of the class more closely, but it is still convenient, fast and not time-consuming.

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

## *2. Thesis Structure*

The thesis is likely to contain all the information to reproduce the result. Firstly, we are anticipated to introduce the interesting issue in monitoring attendance information in USTH and the solution proposed in **section I: Introduction**. In **section II: Objectives**, we are projected to define expected requirements of the project, provide a concise overview of the function of the system and the reasons for its development and improvement. Then, we will illustrate the scenarios, use cases, object model, and dynamic models for the system in **section III: Requirement Analysis**. This section contains the complete functional particularization, including navigational paths representing the sequence of screens. In **section IV: Methodology**, we are expected to list all the techniques and tools utilized in the project, the reasons why they are opted and the detailed use cases implementation. In **section V: Result**, we are going to list all the functionalities which are implemented in the system. There is **Conclusion and Future Work** in **section VI**. Finally is **section VII: Appendix**, which displays all the user interfaces of our system.

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

## **II. OBJECTIVES**

In this section, we will define the expected essential demand of the project, provide a concise overview of the function of the system and the reasons for its development.

First of all, the website is used for helping professors manage the attendance of the class faster and precisely. For students, they need to register, then log into the system, then join the class of the subject and they can check attendance by scanning QR and turning on the GPS. For professors, with accounts created before, they can create the class and generate QR for students to scan to check their attendance.

Secondly, we make a website platform instead of a mobile application. Our system is required to run on the background of IOS, Android, Windows,.... In addition, our system can create QR Code for each subject, allow students to scan QR Code for checking attendance, and allow Professors to manage the list of students. We really want users who utilize our system to feel satisfied with checking attendance by scanning QR Code and turning on the GPS.

Thirdly, we determine there are two main kinds of users of the system: Student and Professor. Besides, we also determine the main functions of our system below.

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

## *1. Desired Features*

The main aim of this project is to develop a System for students being able to check attendance by scanning QR Code. Hence, it has the basic following services:

- **Feature 1: Authenticate**

- Sub-Feature 1.1: Register:** allows students to create their own accounts.

- Sub-Feature 1.2: Login:** allows professors or students to access the system.

- **Feature 2: Edit Profile:** allows students or professors to edit their private profiles.

- **Feature 3: Attendance Information:** allows students to see their information about their own attendance.

- **Feature 4: Course:** allows students or teachers to know more information about the course.

- Sub-Feature 4.1:** For students, they can choose a year and major relative to the course they want to know more information about, then they can see the student’s list of the subject they have enrolled before and they could join the class of this subject and view attendance of all students in the class.

- Sub-Feature 4.2:** For teachers, they can choose year, major and subject relatively to the course. In one particular subject they manage, they could create a class, check attendance of all students in the class and generate QR Code for students to scan for checking their attendance by turning on the GPS.

- **Feature 5: Report:** allows students to write feedback or report about what issue or problem happened in students’ application and they are sent to the database.



\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

## *2. Expected Outcome*

Our system aims at helping university or professor control of students’ attendance in an effective and convenient way. Professors can create QR Code and Students can check their attendance by scanning QR Code and turning on the GPS. Therefore, the particular goals include:

- Develop a backend engine for monitoring attendance data like time checked and location checked.
- Develop a user interface that could help students and professors interact easily, comfortably and conveniently.
- The application should have all the features mentioned in the part of Desired Feature and make users feel satisfied.
- The web page should be able to run in various types of browsers such as Chrome, Safari,...
- Any kind of mobile phone can scan QR and run this web page.

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

### **III. REQUIREMENT ANALYSIS**

In this section, we are likely to describe a brief overview of the functions in the project, the prospect and use cases for the system. This part contains the complete functional particularization, including navigational paths representing the sequence of screens. Our system focuses on generating & scanning the QR Code and requires turning on the GPS as well.

#### *1. General System Requirements*

Overall, our application relating to creating and scanning QR Code has to satisfy the following main requirements:

- A login and register system with authentication.
- The system could limit users with some features:

For Student:

- + Edit profile information.
- + See attendance information.
- + Write a report or feedback and they are sent to the database.
- + See the list of students in class.
- + Join class.
- + Receive a checking attendance form when scanning QR Code and be required to turn on the GPS.

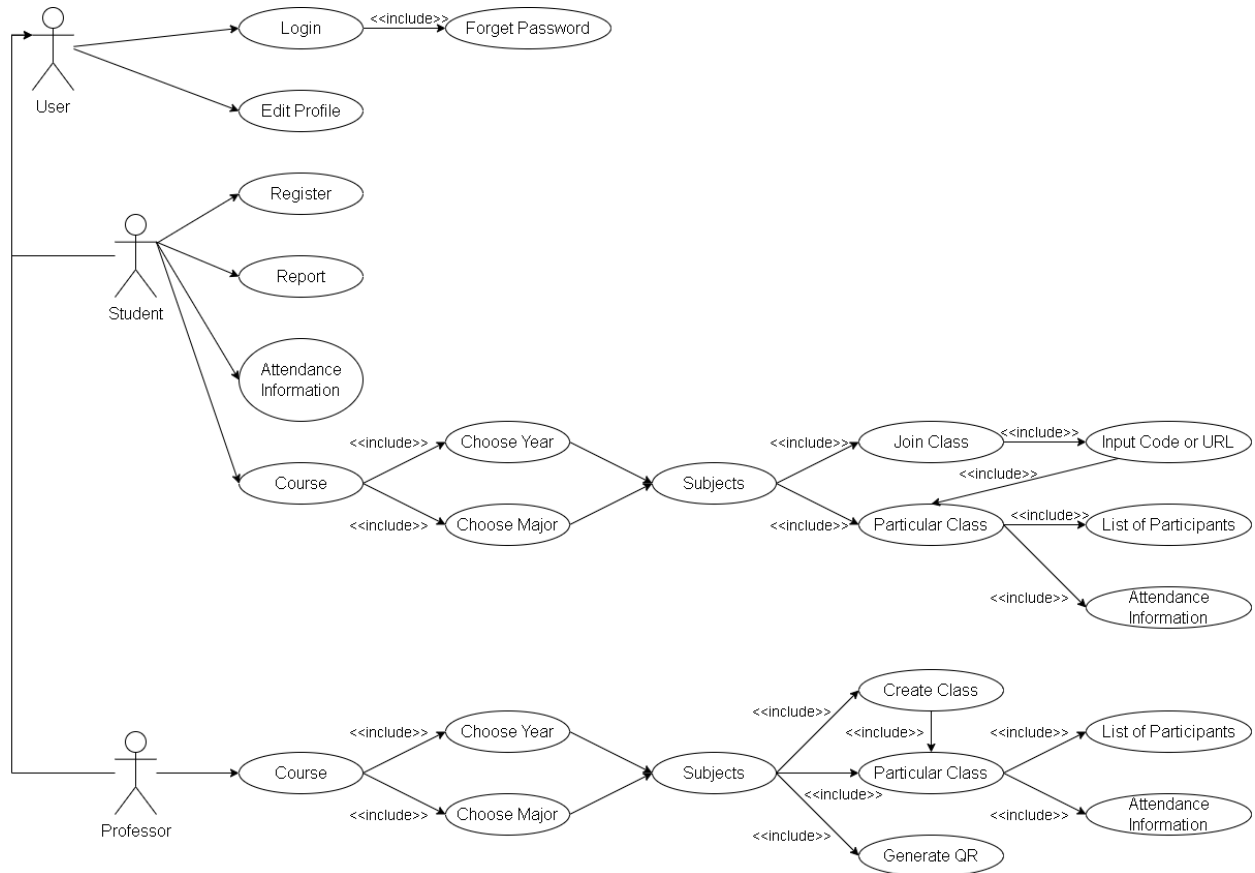
For Professors:

- + Edit profile information.
- + Create class.
- + View attendance of all students.
- + Create QR Code for students to scan for checking attendance.

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

## 2. Use Cases

### 2.1. Use case Diagram



**Figure 1. Use Case Diagram**

Figure 1 illustrates the Use-case diagram that contains all features/sub-features of the system.

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

## 2.2. Users Characteristics

There are two kinds of users that mostly interact with the system: **Professor** and **Student**. Each kind of user is expected to use the system in different ways so each of them has their own requirements.

- **Student**: a kind of user who utilizes the system to check and monitor their own attendance, view list students of the class and report to the Administrator if some problems happen.
- **Professor**: a kind of user who utilizes the system to create class and generate QR to check and monitor attendance for all students of class.

## 2.3. Use case Description

No.	Use Case Name	Note
UC.1	Login	None
UC.2	Edit Profile	None
UC.3	Register	None
UC.4	Report	None
UC.5	Attendance Information	None
UC.6	Course	From actor “Student”
UC.7	Course	From actor “Professor”
UC.8	Forget Password	Include case of UC.1
UC.9	Choose Year	Include case of UC.6
UC.10	Choose Major	Include case of UC.6
UC.11	Choose Year	Include case of UC.7
UC.12	Choose Major	Include case of UC.7
UC.13	Subjects	Summarized of UC.9 and UC.10
UC.14	Subjects	Summarized of UC.11 and UC.12
UC.15	Join Class	Include case of UC.13
UC.16	Particular Class	Include case of UC.13 and UC.20
UC.17	Create Class	Include case of UC.14
UC.18	Particular Class	Include case of UC.14 and UC.17
UC.19	Generate QR	Include case of UC.14
UC.20	Input Code or URL	Include case of UC.15
UC.21	List of Participants	Include case of UC.16
UC.22	Attendance Information	Include case of UC.16
UC.23	List of Participants	Include case of UC.18
UC.24	Attendance Information	Include case of UC.18

**Table 1.** Use Case Table

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

## 2.4. Use Case and Scenario Description

Use Case Name	Login	UC.1
User/Actor	Both student and professor	Relationship Include: UC.8
Brief Definition	User logs in the web with username and password.	
Basic Flow	1) User opens the web page. 2) User inputs his/her username and password. 3) User is redirected into the main page view.	
Alternative Flow	- If the user inputs the wrong username or password, the website will display an error alert or message to notify him/her. - If the user forgets his/her password, he/she could press button “Forget Password” to perform UC.8	
Pre-condition	User hasn’t logged into the website before or has logged out with the previous login session.	

**Table 2.** Scenario Description of UC.1

Use Case Name	Edit Profile	UC.2
User/Actor	Both student and professor	Relationship Include: None
Brief Definition	User can edit their private profile if there is something wrong.	
Basic Flow	1) User clicks the button “Edit Private Profile”. 2) User can edit his/her information related to the profile by inputting the exact profile he/she wants. 3) User clicks the button “Submit” when done.	
Alternative Flow	None	
Pre-condition	Successfully perform UC.1	

**Table 3.** Scenario Description of UC.2

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

Use Case Name	Register			UC.3
User/Actor	Student	Relationship	Include: None	
Brief Definition	Student create his/her own account with his/her own information.			
Basic Flow	1) Student opens the web page. 2) Student clicks “Signup”. 3) Student inputs his/her own information to register. 4) Student clicks the button “Register”. 5) Student is redirected into the Login page view.			
Alternative Flow	If the student inputs the wrong type of information (like the email does not have “@”, or the student ID is not true), the website will display an error alert or message to notify them.			
Pre-condition	Student hasn’t created his/her own account before.			

**Table 4.** Scenario Description of UC.3

Use Case Name	Report			UC.4
User/Actor	Student	Relationship	Include: None	
Brief Definition	Student can write feedback or report about the website if there are some errors in the website.			
Basic Flow	1) Student clicks the button “Report” 2) Student write his/her feedback or report if they want to offer suggestions or some errors happen in the website. 3) Student clicks the button “Submit”			
Alternative Flow	None			
Pre-condition	Successfully perform UC.1.			

**Table 5.** Scenario Description of UC.4

Use Case Name	Attendance Information			UC.5
User/Actor	Student	Relationship	Include: None	
Brief Definition	Student can know detailed information about his/her attendance			
Basic Flow	1) Student clicks the button “Attendance Information”. 2) Student can see information related to his/her attendance here.			
Alternative Flow	None			
Pre-condition	Successfully perform UC.1.			

**Table 6.** Scenario Description of UC.5

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

Use Case Name	Course			UC.6
User/Actor	Student	Relationship	Include: UC.10, UC.11, UC.14, UC.16, UC.17, UC21, UC.22, UC.23	
Brief Definition	Student chooses a year and major to find the relative class of the subject he/she is learning, then he/she could see the list of all students in the class and join the class.			
Basic Flow	<div>1) Student clicks the button “Course”.</div> <div>2) Student chooses Year and Major.</div> <div>3) Student chooses the Subject relative to the Year and Major he/she has chosen before.</div> <div>4) Student can join the new class by clicking the button “Add Class” and inputting the class Code or URL.</div> <div>5) Student clicks the name of the class in the list below to see the list of participants in the class and view attendance information of all students in the class.</div> <div>6) Student clicks the button “List” to see the list of participants in the class. Otherwise, student clicks the button “Attendance Check” to see his/her own attendance information.</div>			
Alternative Flow	None			
Pre-condition	Successfully perform UC.1			

**Table 7.** Scenario Description of UC.6

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

Use Case Name	Course			UC.7
User/Actor	Professor	Relationship	Include: UC.10, UC.11, UC.14, UC.16, UC.17, UC21, UC.22, UC.23	
Brief Definition	Professor chooses a year and major to find the relative class of the subject he/she is teaching, then he/she creates a class and generates QR for students to scan QR for checking attendance.			
Basic Flow	<ol style="list-style-type: none"><li>1) Professor clicks the button “Course”.</li><li>2) Professor chooses Year and Major.</li><li>3) Professor chooses the Subject relative to the Year and Major he/she has chosen before.</li><li>4) Professor is directed into the main page view of Course.</li><li>5) Professor can create a new class by clicking the button “Add here” and typing the name of the class.</li><li>6) Professor can generate a QR code by clicking the button “Generate QR”; then clicking the button “Click Me” to get exact location by turning GPS, inputting the time and then click the button “Generate QR Code” to create a QR code and professor can show it on the slide for students to scan to check their own attendance.</li><li>7) Professor clicks the button “List” to see the list of participants in the class. Otherwise, professor clicks the button “Attendance Check” to see the attendance information of all the classes.</li></ol>			
Alternative Flow	None			
Pre-condition	Successfully perform UC.1			

**Table 8.** Scenario Description of UC.7



\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

## IV. METHODOLOGY

In this section, all the tools and techniques used in the project, the reasons why they are opted, the detailed use cases implementation and how they help us to make the user be able to create & scan QR Code and turn on the GPS as well are projected to be listed.

### *1. Tools and Techniques*

#### **1.1. Figma**

Figma is a browser-based, collaborative user interface design tool that lets users work together to create vibrant and interactive prototypes. Since its release in 2016, Figma has become a popular tool both in the web design industry and in online communities. Users can collaborate and share templates, designs, and widgets with millions of users across the globe.

We use Figma to design our Web User Interface.

#### **1.2. HTML**

HTML (the Hypertext Markup Language) is one of the core technologies for building Web pages. HTML provides the *structure* of the page.

HTML is one of the basis of building Web pages and Web applications.

HTML is the language for describing the structure of the Web pages. HTML gives authors the means to:

- Publish online documents with headings, text, tables, lists, photos, etc.
- Retrieve online information via hypertext links, at the click of a button.
- Design forms for conducting transactions with removed services, for use in searching for information, making reservations, ordering products, etc.

With HTML, authors describe the structure of pages using *markup*. The *elements* of the language label pieces of content such as “paragraph”, “list”, “table”, and so on. HTML helps to form the structure of our website, from basic to advanced levels and make our web a complete system.

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

### 1.3. CSS

CSS (Cascading Style Sheet) is one of the core technologies for building Web pages. CSS provides the (visual and aural) *layout*, for a variety of devices.

CSS is one of the basis of building Web pages and Web applications.

CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments. This is referred to as *the separation of structure (or: content) from presentation*.

CSS is likely to make it possible for us to add a little “style” to HTML elements in our system.

### 1.4. JavaScript

JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles.

JavaScript is forecast to help our front-end web development a lot.

### 1.5. MySQL Workbench

With the amount of data in our system, it is important to choose the suitable Database to manage data.

SQL is suitable for a project with a fixed schema, high transaction, low maintenance, data security with a limited budget.

In our project, the data has been fully predefined and it may be expanded in size but should be constrained in its structure and relation. In addition, the student and professor application information, USTH administrator department must be strictly connected.

Hence, one of the best solutions of our system is using SQL Database.

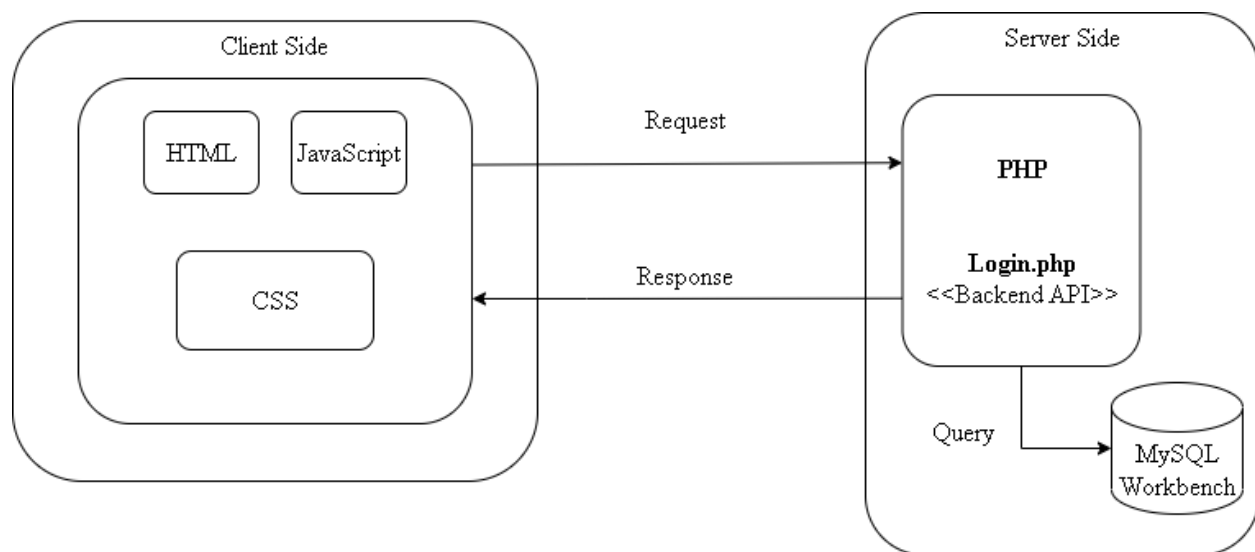
\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

## 1.6. PHP

PHP - Hypertext Preprocessor is used for server-side programming for many reasons. Firstly, PHP is compatible with MySQL which is our database so it makes sense for us to use it. This language also has very good online documents with a huge community and that makes it much more convenient to search and learn. Secondly, PHP is open source so the cost of using it is minimal.

With PHP, our system is anticipated to be helped with some issues related to perfect integration with SQL databases like MySQL. Besides, PHP could handle text, string and numeric data easily.

## 2. System Architecture



**Figure 2.** System Architecture Diagram

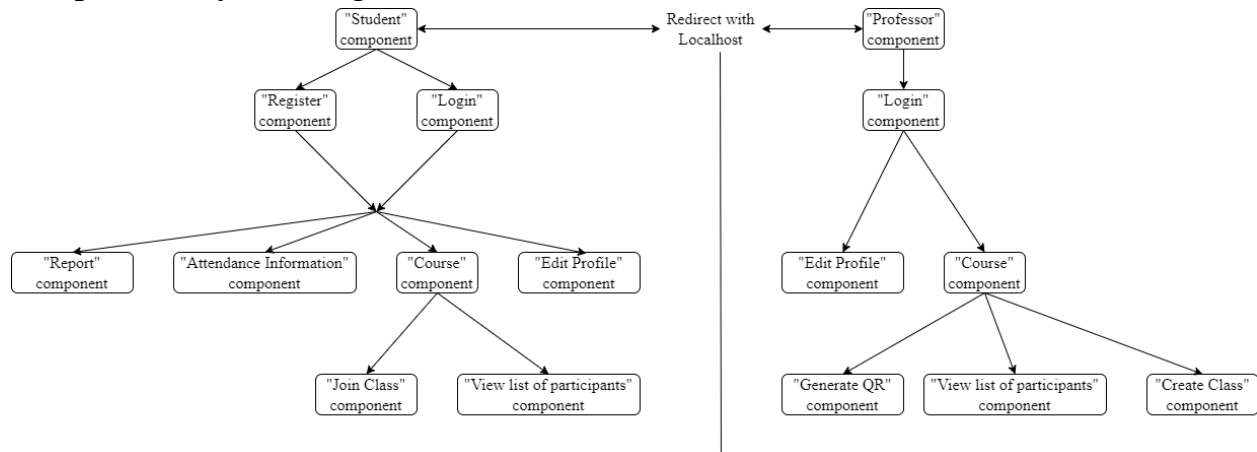
This Web Platform for Checking attendance system by using GPS is a full side project with a completely developed Client side (Front-end) and Server side (Back-end):

- The Client side interacts with the user, retrieves the user quest, sends it to the Server side, receives the response from the Server then response data back and displays it to the Client side.
- The Server side receives the request from the Client side, gets data from the Database and processes then returns a response.

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

## 2.1. Client Side

Many components with the task to handle the created and scanned QR Code are included in the web front-end. These components are arranged as in the figure below. Sharing data between components is divided into two kinds: pass data from parent component to child component (using prop) and from child component to parent component (by emitting event).



**Figure 3.** Front-end Component Design

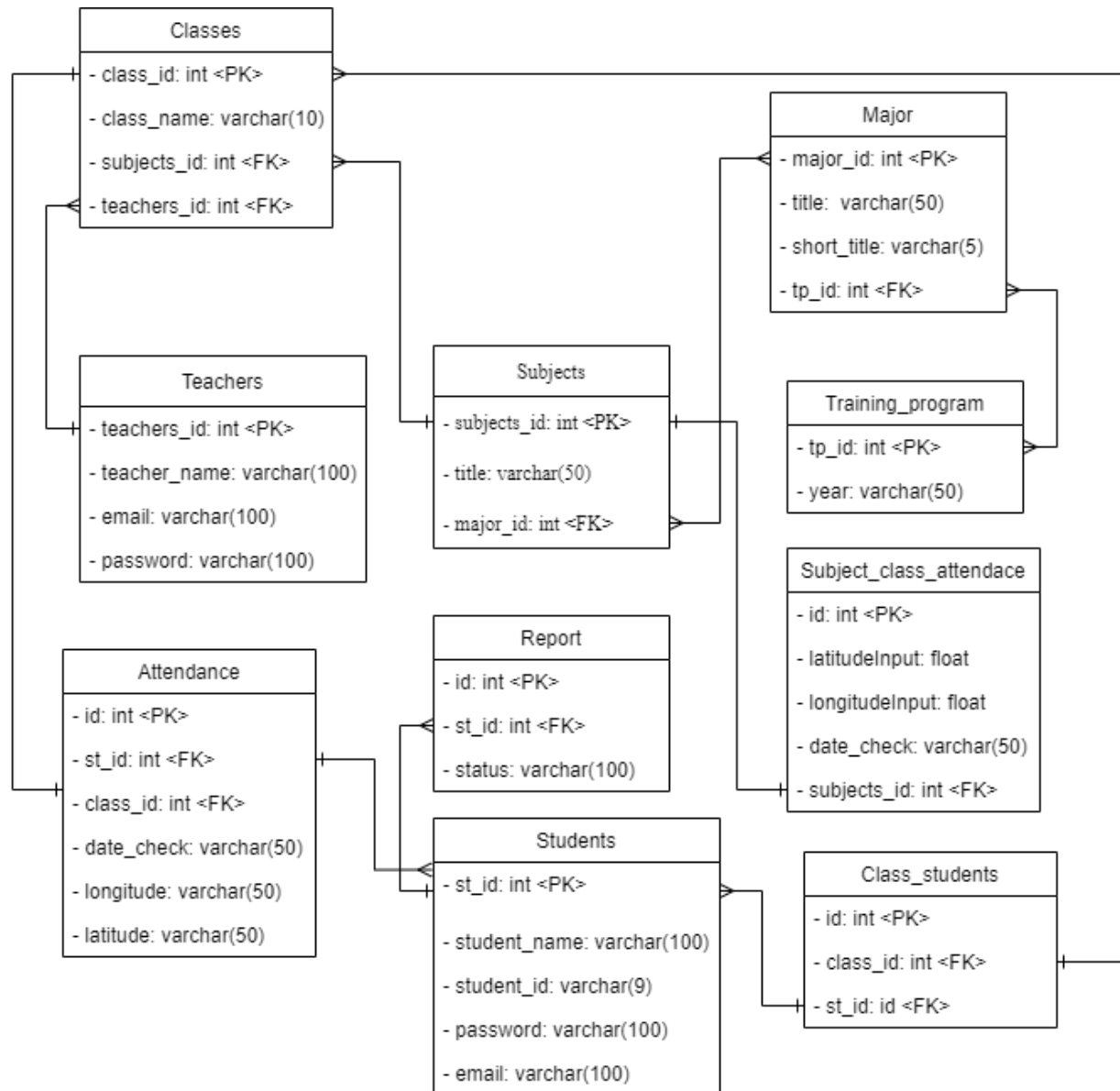
## 2.2. Server Side

This is a Client side connecting to web application, the Back-end takes the responsibility of creating a secure connection between the web browser and system's database. The main responsibility may send and receive transferred data from the QR Code. This side includes two main parts:

- MySQL: A database management system to manage, monitor and administer all the system data.
- PHP for API processing (including receiving requests, authenticating requests, communicating with databases, return response). Structure of an API contains:
  - + Define action.
  - + Request verification with token.
  - + Query from Database.
  - + Response.

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

### 3. Database Design



**Figure 4.** Overall Database Design

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

Training_program
- tp_id: int <PK> - year: varchar(50)

This table shows the training programs in a period of the year. For example, we have 2022-2023. One program includes many majors and a major can be displayed in many training programs. Therefore, it has a many-to-many relationship with table Major.

**Table 9.** Training Program Database Design

Major
- major_id: int <PK> - title: varchar(50) - short_title: varchar(5) - tp_id: int <FK>

This table displays all information about titles for each major, including full and short titles. The users (including students and teachers) can see all information about the major.

**Table 10.** Major Database Design

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

Subjects
<ul style="list-style-type: none"> <li>- subjects_id: int &lt;PK&gt;</li> <li>- title: varchar(50)</li> <li>- major_id: int &lt;FK&gt;</li> </ul>

This table displays all information about titles for each subject, including full and short titles. One major includes many subjects and a subject can be displayed in many majors. Hence, it has a many-to-many relationship with table Major.

**Table 11.** Subject Database Design

Subject_class_attendace
<ul style="list-style-type: none"> <li>- id: int &lt;PK&gt;</li> <li>- latitudeInput: float</li> <li>- longitudeInput: float</li> <li>- date_check: varchar(50)</li> <li>- subjects_id: int &lt;FK&gt;</li> </ul>

This table is a kind of form that helps teachers to make a Uniform Resource Locator (URL) for students in class. The teacher is going to create his/her own URL including class positions by taking longitude & latitude. They also set up dates & time for class and subject. This table has only one subject and it can not have more than one subject in this form. Therefore, it has a one-to-one relationship with the Subjects table.

**Table 12.** Subject Class Attendance Database Design

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

Classes
<ul style="list-style-type: none"> <li>- class_id: int &lt;PK&gt;</li> <li>- class_name: varchar(10)</li> <li>- subjects_id: int &lt;FK&gt;</li> <li>- teachers_id: int &lt;FK&gt;</li> </ul>

This table shows information about a class. In a class there is just one attendance list and an attendance list for just one class. Hence, this table has the class name and it has a one-to-one relationship with table Attendance. Moreover, it also has a one-to-many relationship with three tables: Subjects, Class Students and Teacher. In a class there are many students but just one teacher and one subject.

**Table 13.** Classes Database Design

Class_students
<ul style="list-style-type: none"> <li>- id: int &lt;PK&gt;</li> <li>- class_id: int &lt;FK&gt;</li> <li>- st_id: id &lt;FK&gt;</li> </ul>

This table helps Table Students be associated with Table Classes. Therefore, the Table Class Students has a one-to-many relationship with two of them. The users can see information about students and teachers in their class.

**Table 14.** Class Students Database Design



\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

Attendance
<ul style="list-style-type: none"><li>- id: int &lt;PK&gt;</li><li>- st_id: int &lt;FK&gt;</li><li>- class_id: int &lt;FK&gt;</li><li>- date_check: varchar(50)</li><li>- longitude: varchar(50)</li><li>- latitude: varchar(50)</li></ul>

This table shows information about students' attendance: their exact position (longitude and latitude), their classes and the date that they have checked before. It has a one-to-many relationship with table Students because there are many students in an attendance list. The teacher can check their students' attendance information so that they can know who is present or absent.

**Table 15.** Attendance Database Design

Students
<ul style="list-style-type: none"><li>- st_id: int &lt;PK&gt;</li><li>- student_name: varchar(100)</li><li>- student_id: varchar(9)</li><li>- password: varchar(100)</li><li>- email: varchar(100)</li></ul>

This table shows information about students: name, student ID, email and password for accessing this website. It has a one-to-many relationship with the table Report because a student can send many reports to the system. In addition, a one-to-many relationship with the table Class Students because a class includes many students.

**Table 16.** Students Database Design

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

Report
<ul style="list-style-type: none"><li>- id: int &lt;PK&gt;</li><li>- st_id: int &lt;FK&gt;</li><li>- status: varchar(100)</li></ul>

This table includes reports that have been posted by a student before.

**Table 17.** Report Database Design

Teachers
<ul style="list-style-type: none"><li>- teachers_id: int &lt;PK&gt;</li><li>- teacher_name: varchar(100)</li><li>- email: varchar(100)</li><li>- password: varchar(100)</li></ul>

This table shows information about teachers: name, email and password for accessing this website. It has a one-to-many relationship with table Classes because there is just one teacher in a class.

**Table 18.** Teachers Database Design

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

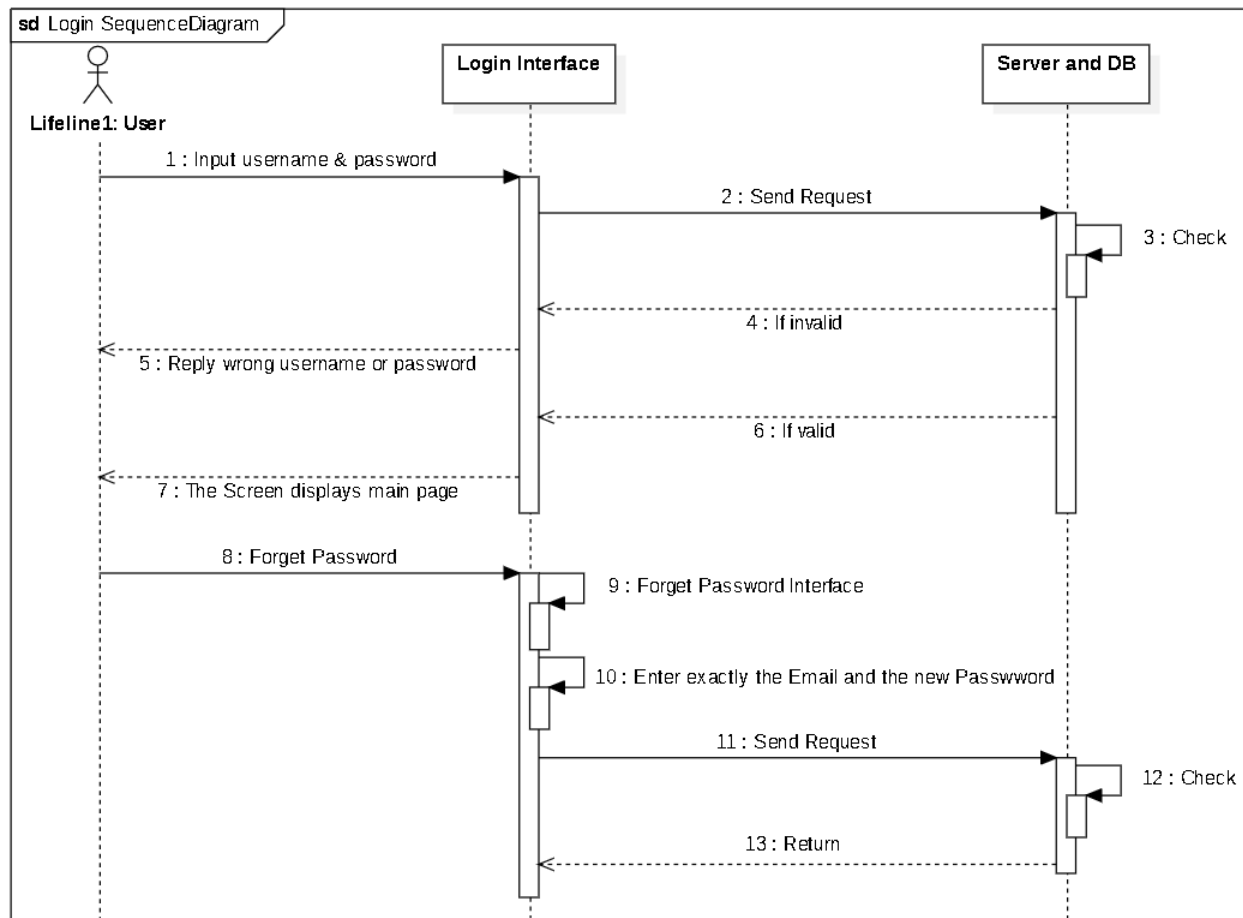
## 4. Use Cases Implementation

### 4.1. Login

This use case illustrates how a user can log into the system.

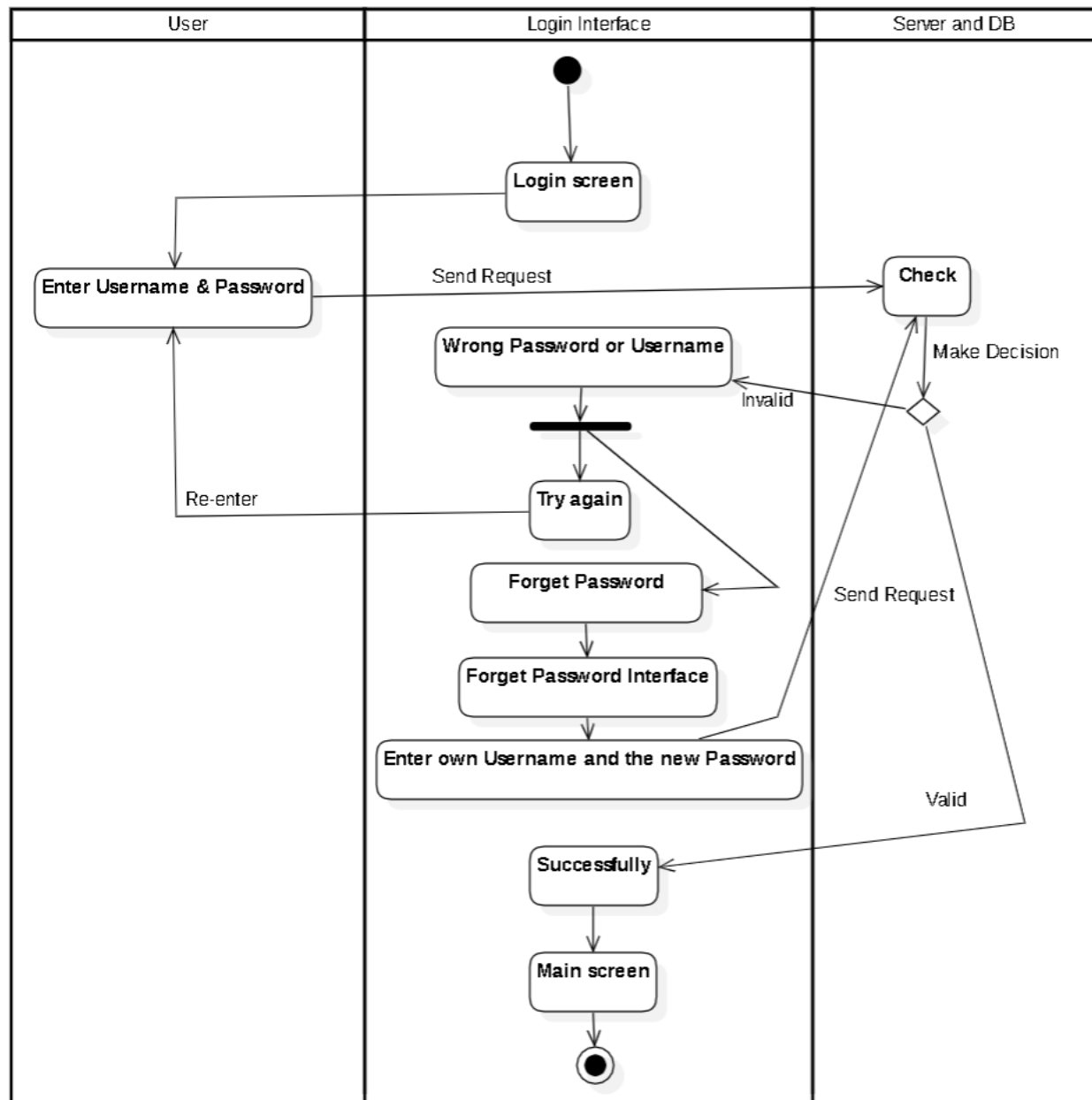
Professor or Student could be a user.

If a user wants to generate QR Code or check attendance by scanning QR Code, he/she has to login first. When a user wants to login, the system will request the user to input his/her username & password. Then the request will be sent to the Server and DB. The Server and DB will check and make decisions. If the username and password are all valid, user will login successfully and the main screen will be displayed. If the username and password are invalid, the Login Interface will display the wrong password or username. Then, the user is made to try again to re-enter the username or password. If a user forgets his/her password, the forget password screen will be displayed. After that, the user has to input his/her exact email and the new password. Then, the request is sent to the Server and DB again. The Server and DB will check and make decisions. If it works successfully, the user returns to the Login Interface again.



**Figure 5.** Login Sequence Diagram

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.



**Figure 6.** Login Activity Diagram

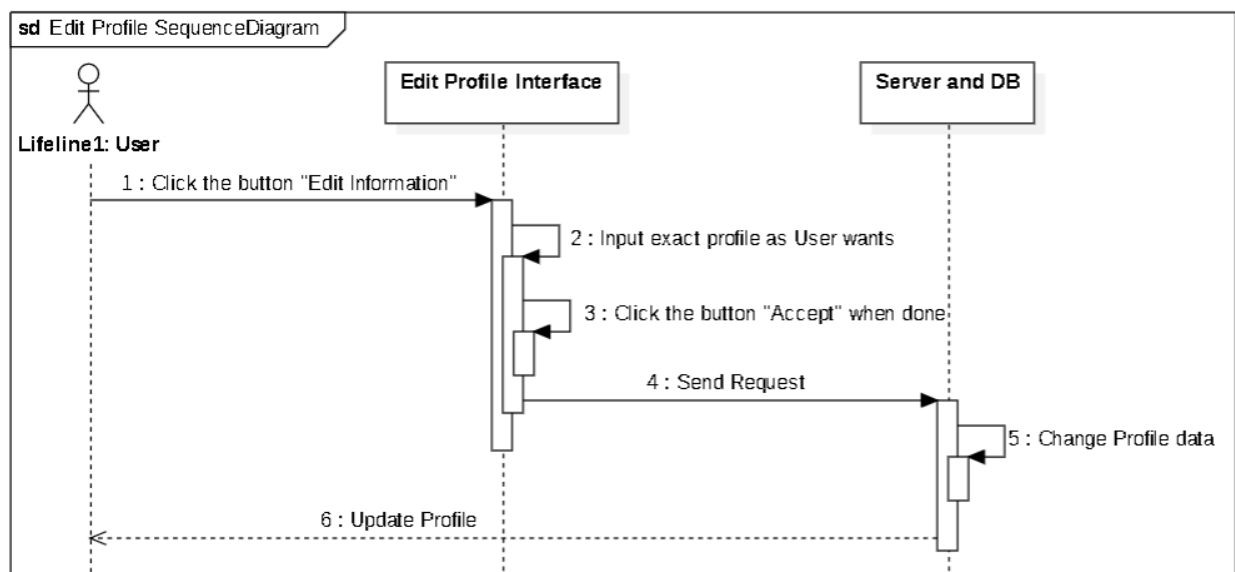
\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

## 4.2. Edit Profile

This use case illustrates how a user could edit his/her private profile.

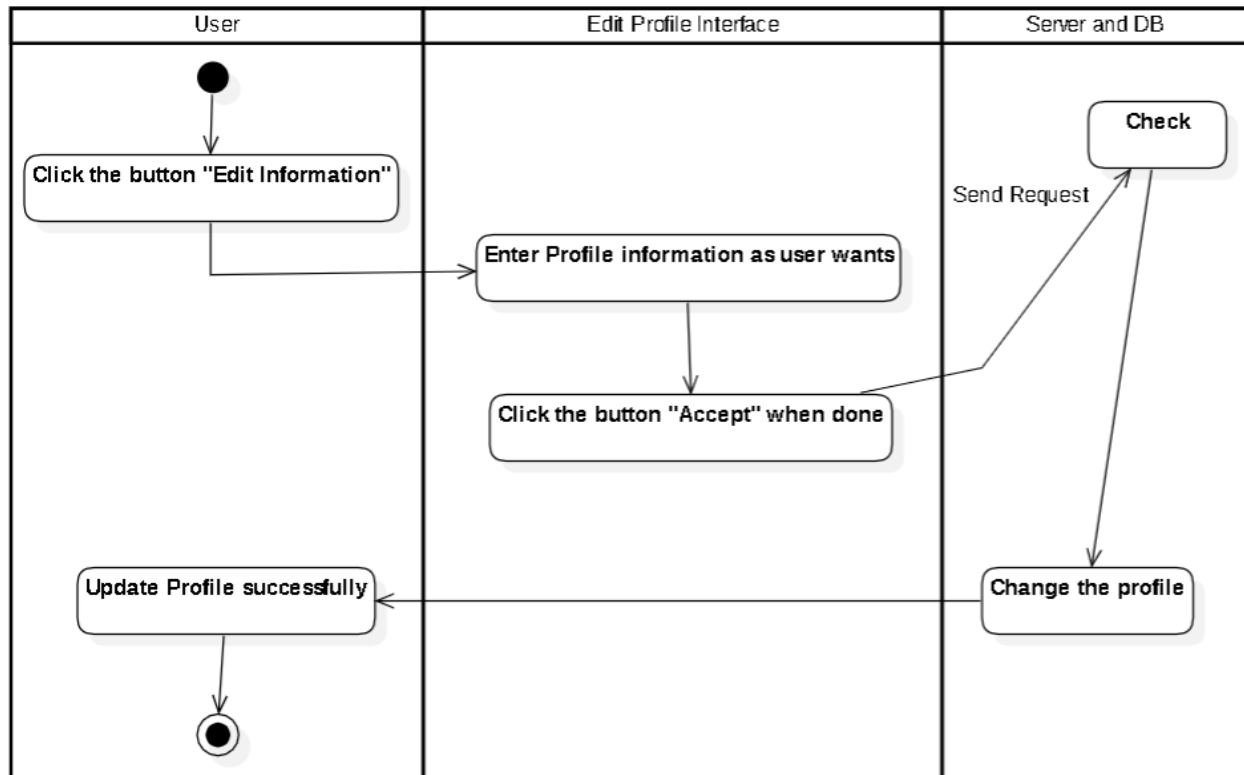
Professor or Student could be a user.

The data or profile information plays an important role while using the generating or scanning QR Code system. Whereas a user wants to edit his/her own private profile, he/she can click the button “Edit Information”. Then he/she can enter exact profile details as he/she wants. When done, he/she could click the button “Accept”. After that, the request will be sent to Server and DB to change his/her profile data. Finally, User’s profile can be updated successfully.



**Figure 7.** Edit Profile Sequence Diagram

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.



**Figure 8.** Private Profile Activity Diagram

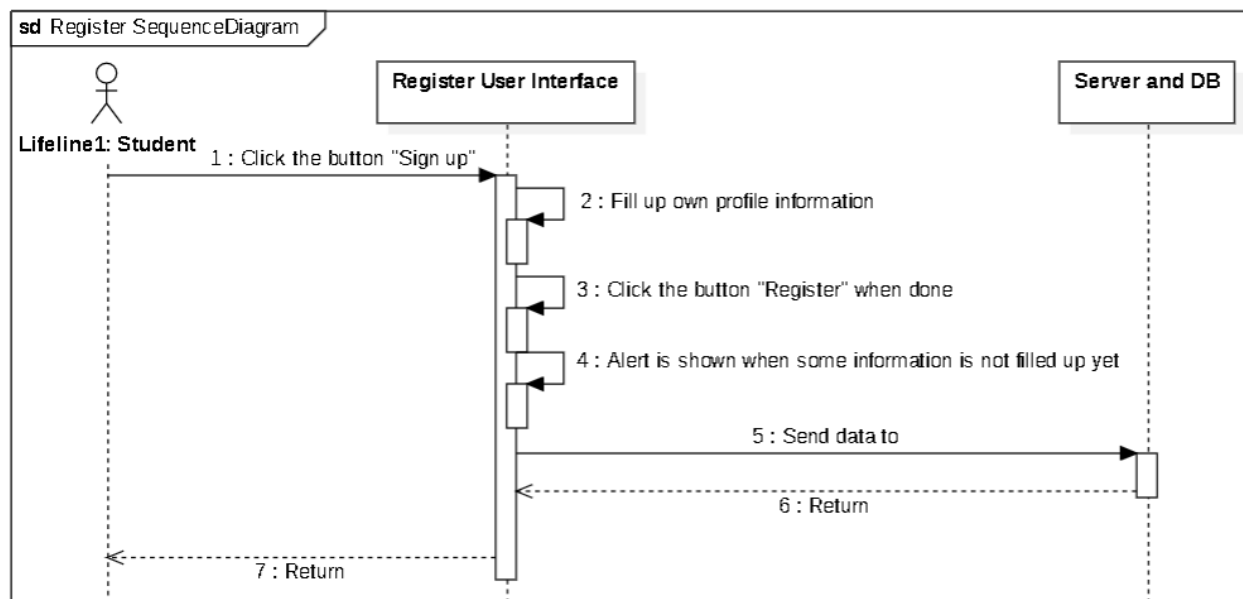
\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

### 4.3. Register

This use case illustrates how a user creates his/her own account in the system.

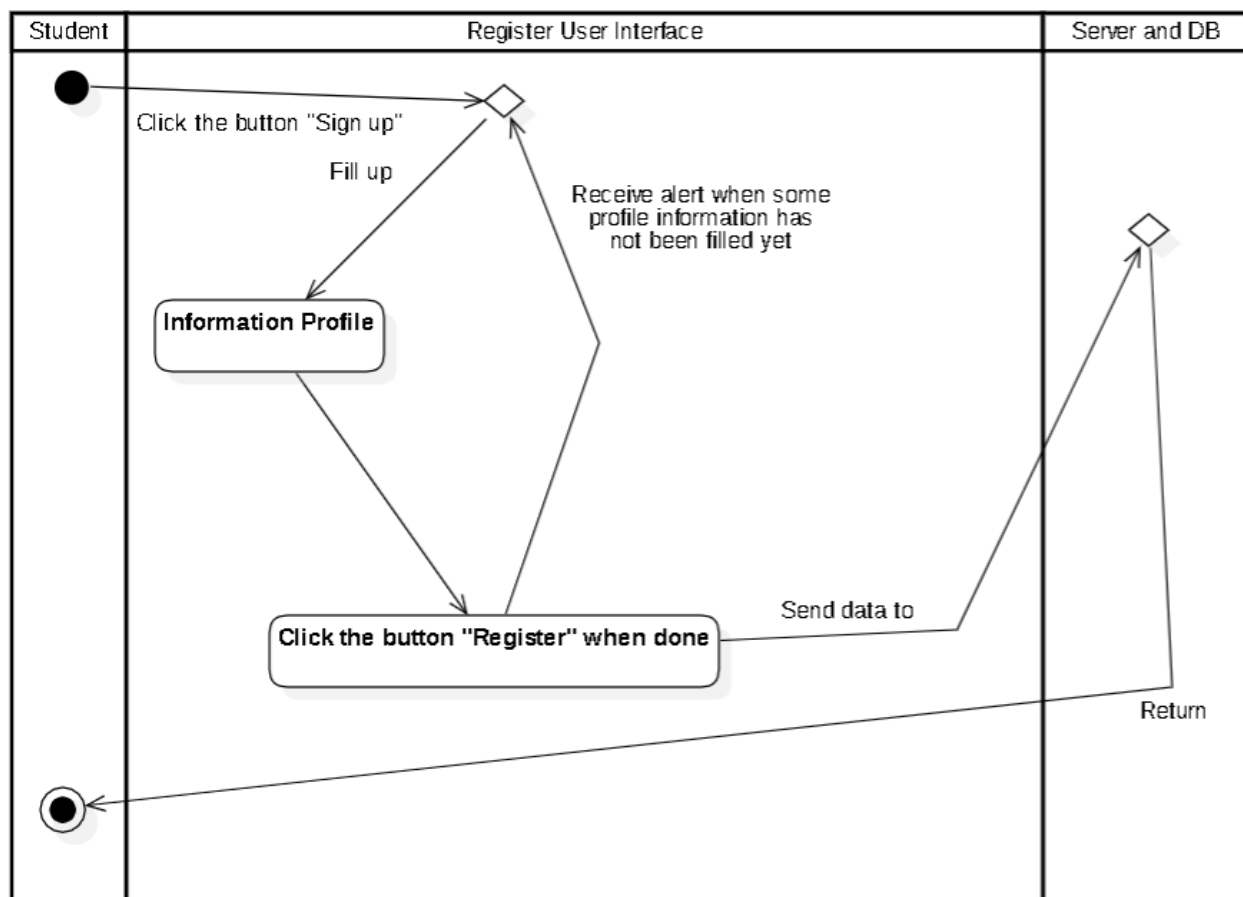
Just Student is the user.

If a user wants to check their own attendance, he/she needs to scan the QR Code created by the Professors. If he/she is able to do this, he/she needs to have his/her own account. And if a user is able to possess his/her own account, he/she must register an account at first. When a user wants to register, he/she clicks the button “Sign up”. Then, he/she has to fill up their own profile information. If there are some blanks not filled up yet when a user clicks the button “Register”, an alert is shown “You have to fill up all the information”. When you register successfully, the data is sent to the Server and DB. Finally, the user could think that he/she has registered completely.



**Figure 9.** Register Sequence Diagram

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.



**Figure 10.** Register Activity Diagram



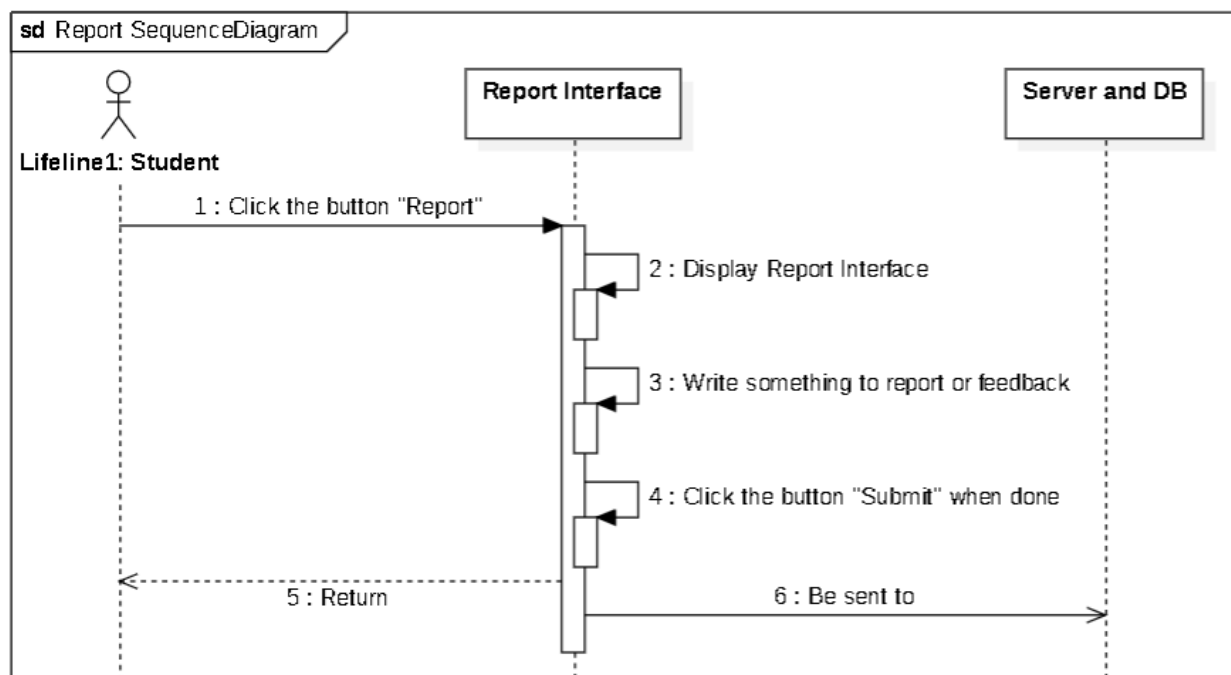
\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

#### 4.4. Report

This use case illustrates how a user writes his/her own report or feedback to give to the administrator.

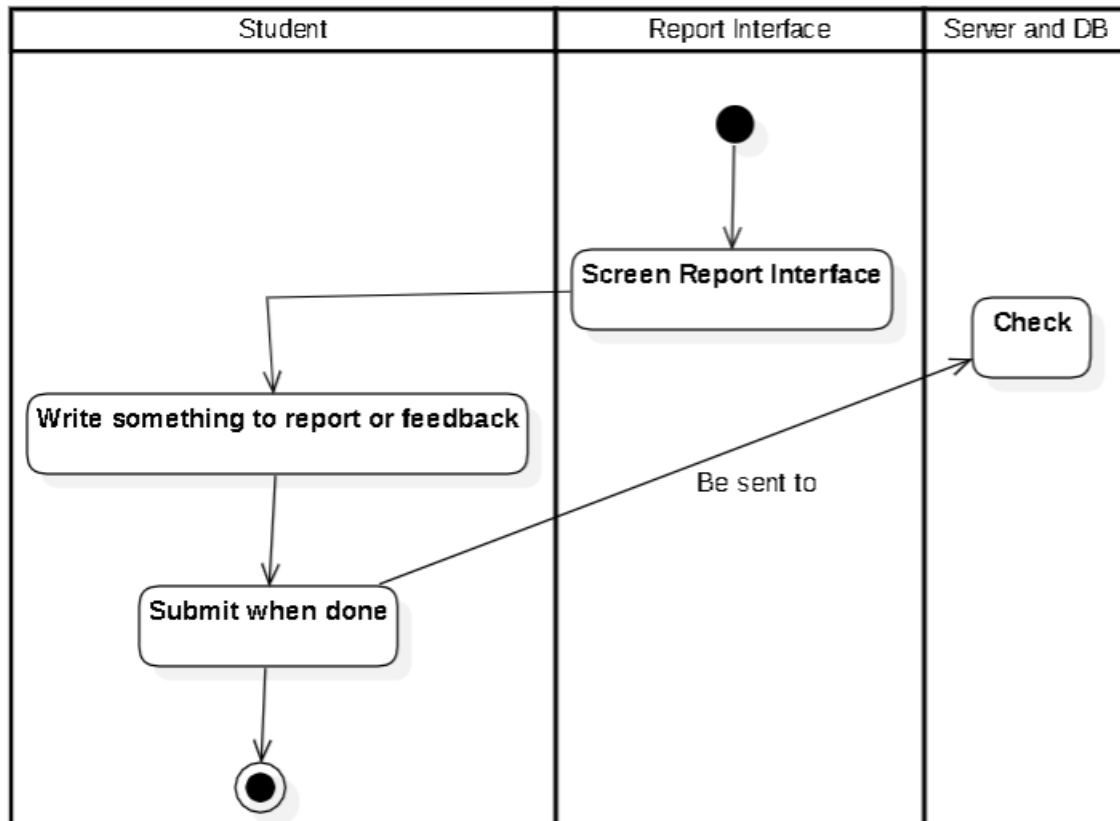
Just Student is the user.

While a user is using the system, he/she can meet some problematic matters or issues. Hence, he/she may need a place to report to the Administrator. When a user wants to report or feedback something, he/she can click the button “Report”. Then the system displays the Report Interface. Here, he/she can write whatever thing he/she wants to report or feedback to the administrator. If he/she has done so, he/she can click the button “Submit”. User will be directed to the browser. And the report or feedback written already will be sent to server and DB.



**Figure 11.** Report Sequence Diagram

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.



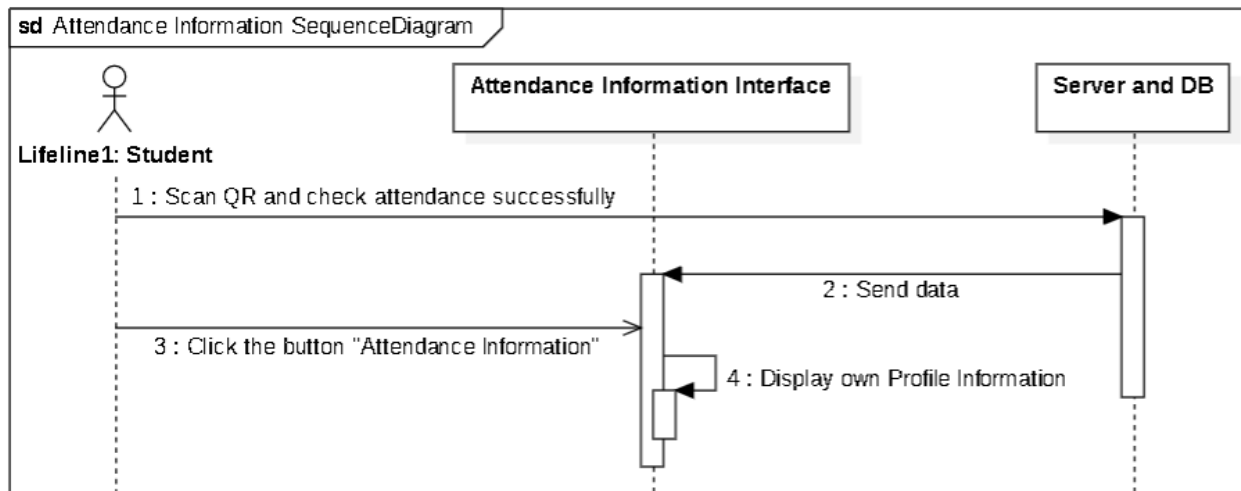
**Figure 12.** Report Activity Diagram

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

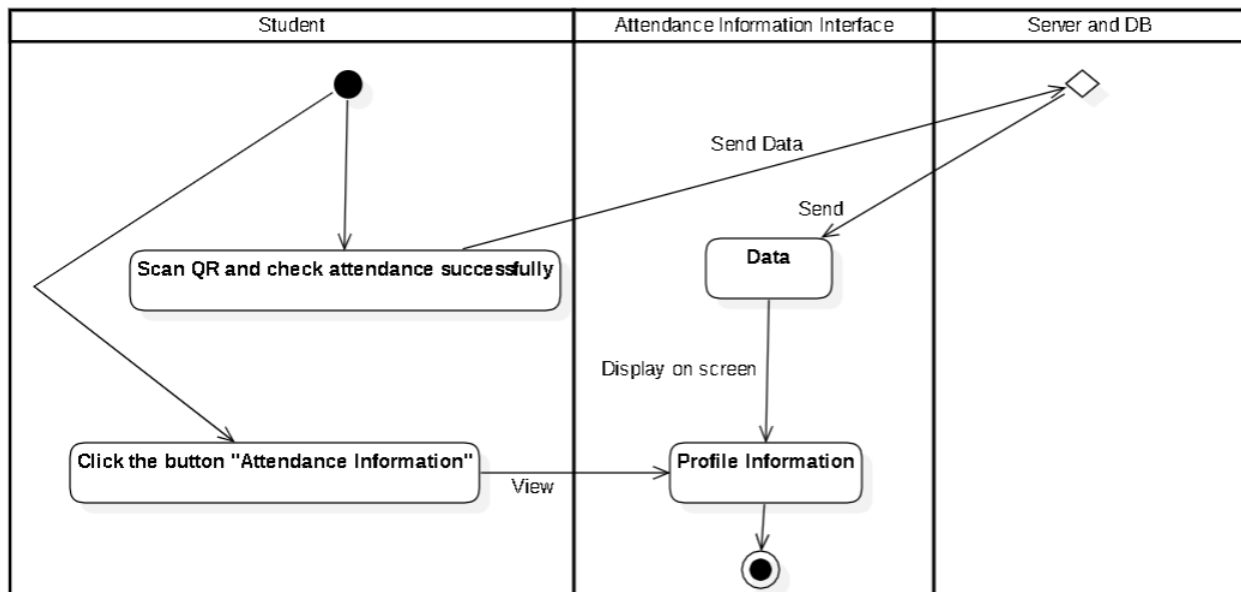
#### 4.5. Attendance Information

This use case illustrates how a user views or checks his/her attendance information. Just Student is a user who wants to check attendance after scanning QR Code.

When a user wants to view or check his/her attendance information, maybe after the user scans the QR code and checks attendance successfully, then data is stored into Server and DB and sent to Attendance Information to display on screen; user can click the button “Attendance Information. After that, he/she can view information of his/her attendance shown on the screen.



**Figure 13.** Attendance Information Sequence Diagram



**Figure 14.** Attendance Information Activity Diagram

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

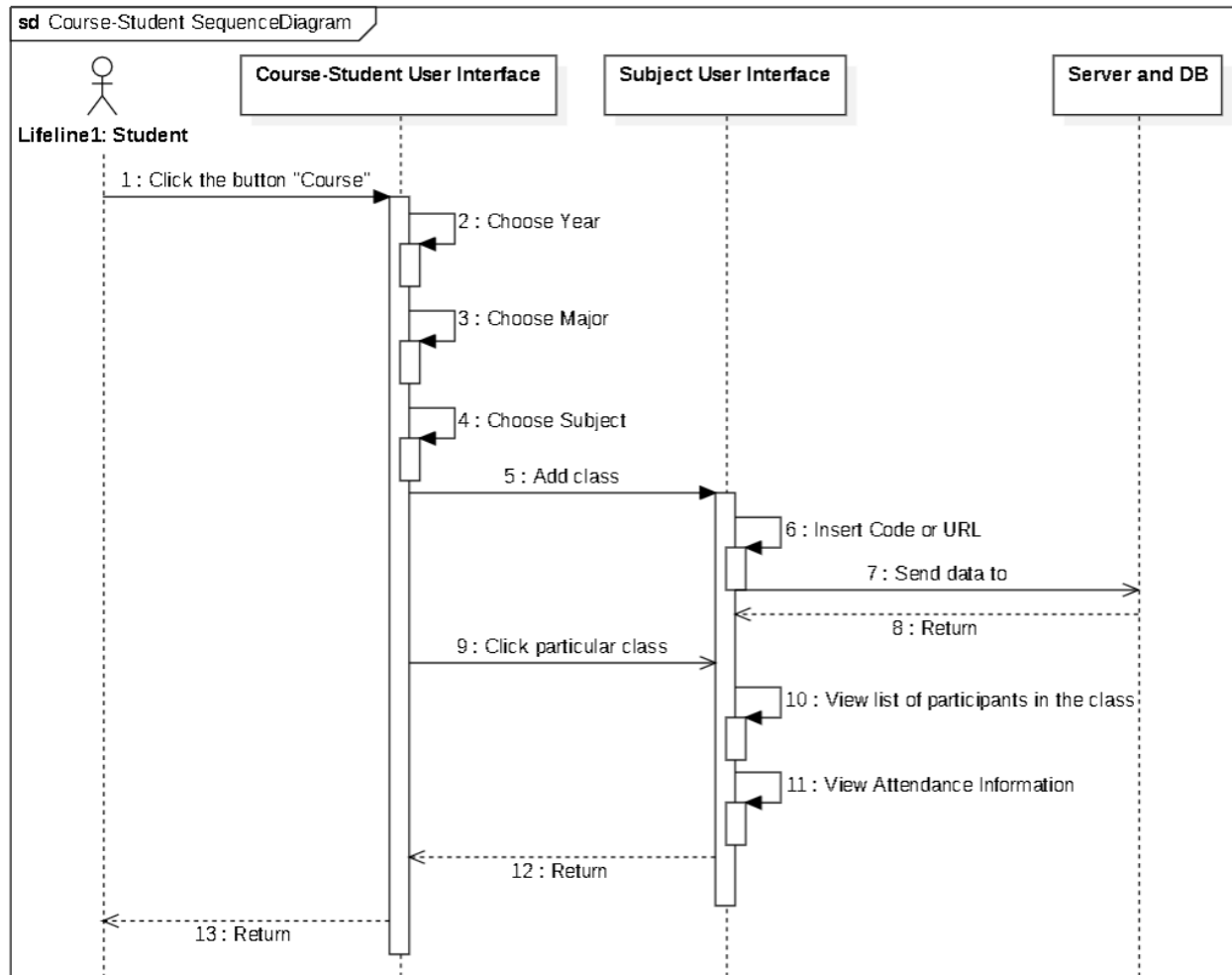
#### 4.6. Course - Student

This use case illustrates how a user interacts with the Course-Student of the system. The user can join class for manager or administrator finding monitoring easier and he/she can see the list of participants in the class and he/she also could see his/her attendance information here.

Just Student is a user.

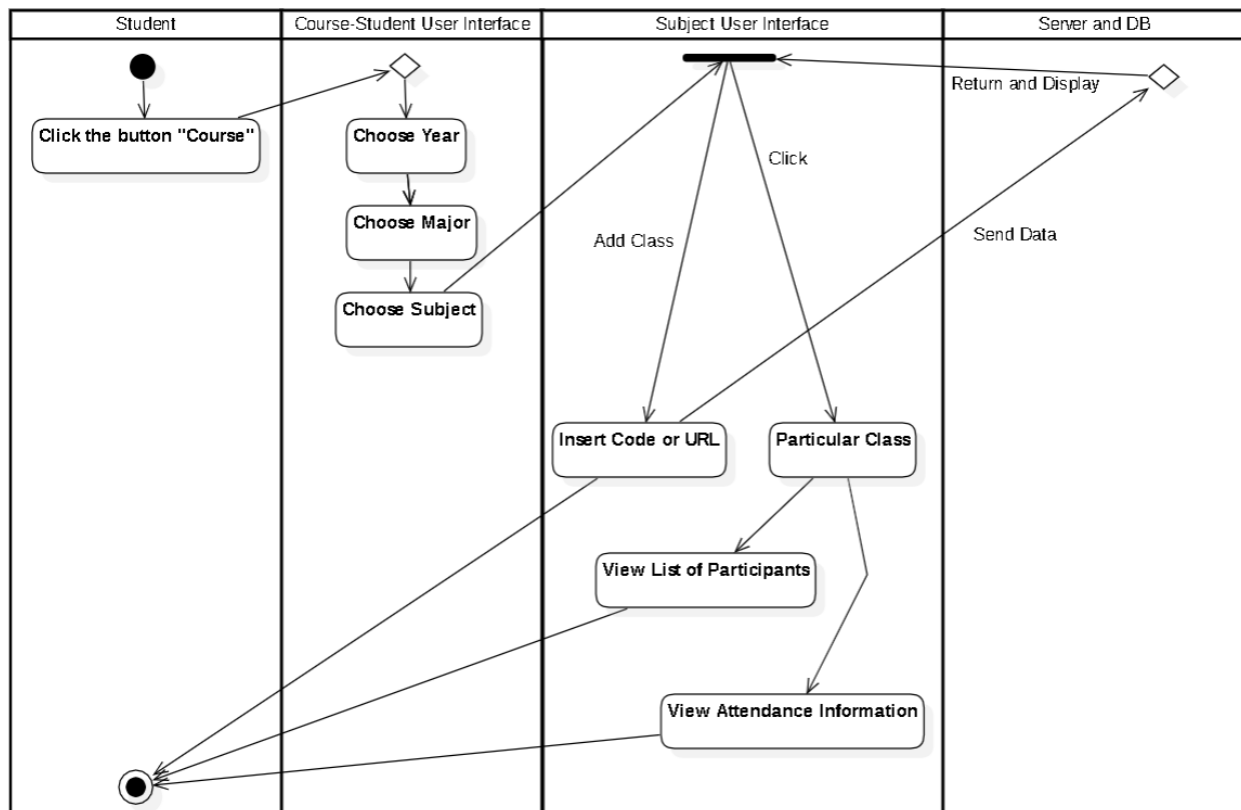
A user can do many things in the Course, especially something relating to scanning the QR Code created by the Professor. First of all, whereas he/she chooses Year, the relative Major appears. After that, when he/she chooses Major, the relative Subject appears. After he/she clicks one particular subject, user is directed to the Subject User Interface. User can join the class by clicking the button “Add class” and inserting Code or URL of the class (maybe created by professor or administrator). Then, the data is sent to Server and DB. In Server and DB, we can understand that this student has been in this class. In addition, if user clicks one particular class in Course-Student User Interface, he/she could see the list of participants in the class (including Students and Professors) by clicking the button “List”. User also can view information about attendance of all students in the class by clicking the button “Attendance Check”.

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.



**Figure 15.** Course-Student Sequence Diagram

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.



**Figure 16.** Course-Student Activity Diagram

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

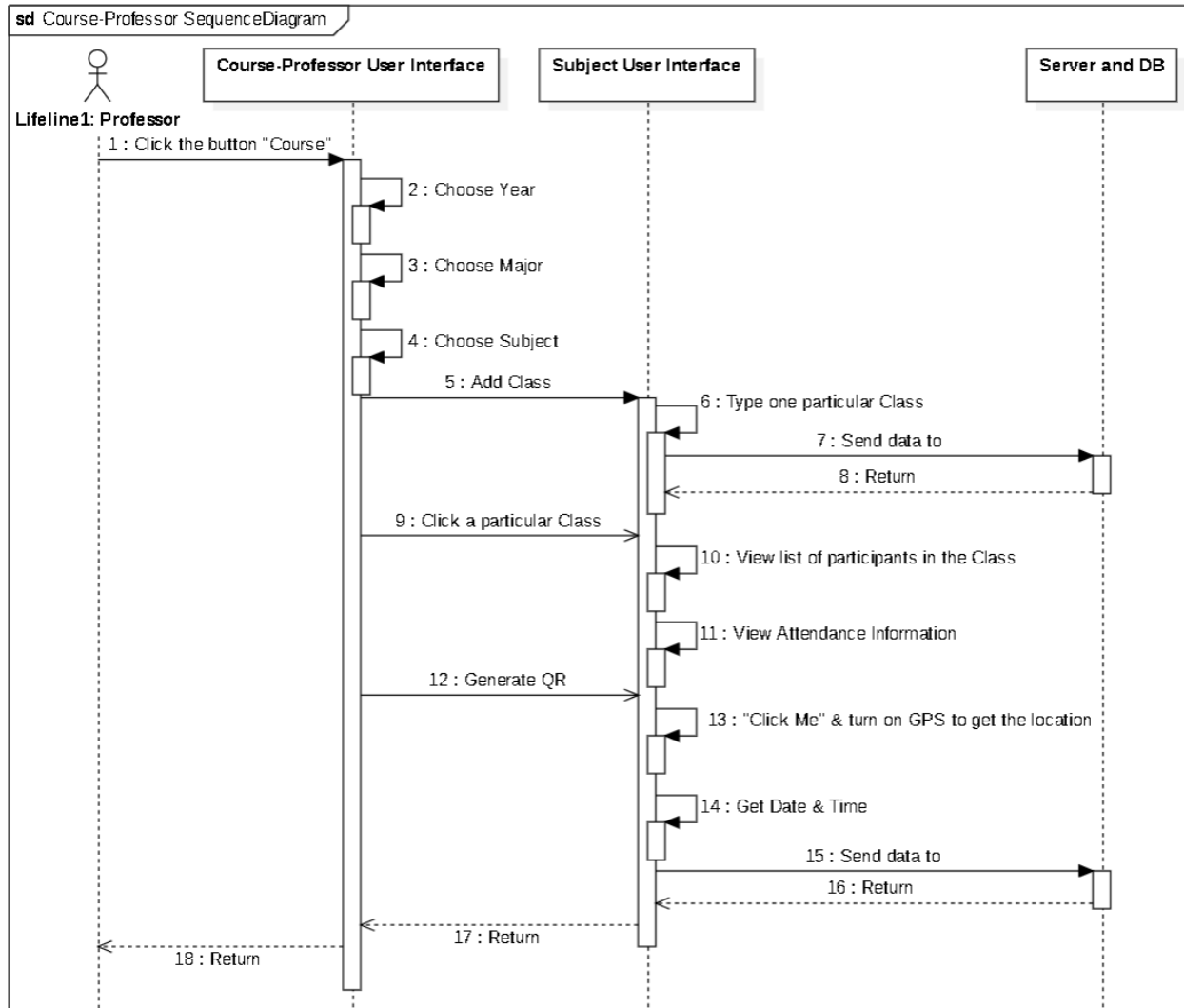
#### 4.7. Course - Professor

This use case illustrates how a user interacts with the Course-Professor of the system. The user can create class(es) to feel more comfortable and convenient in monitoring a lot of students. Furthermore, the user also can generate QR Code. Students can scan them for checking their own attendance.

Just Professor is a user.

A user can do many things in the Course, especially something relating to generating the QR Code for the Students to scan for checking their own attendance. First of all, whereas he/she chooses Year, the relative Major appears. After that, when he/she chooses Major, the relative Subject appears. After he/she clicks one particular subject, user is directed to the Subject User Interface. User can create a class by clicking the button “Add here” and typing one particular class. Then, the data is sent to the Server and DB. We can understand that there has been a class created in the Server and DB. In addition, if user clicks one particular class in Course-Professor User Interface, he/she could see the list of participants in the class (including Students and Professors) by clicking the button “List”. User also can view information about attendance of all students in the class by clicking the button “Attendance Check”. Furthermore, user can generate QR by clicking the button “Generate QR” at first. Then, user clicks the button “Click Me” and turns on GPS to get the exact longitude and latitude of location. After that, user sets the Date and Time. Ultimately, user clicks the button “Generate QR Code” below to create a QR Code for students to scan for checking their attendance.

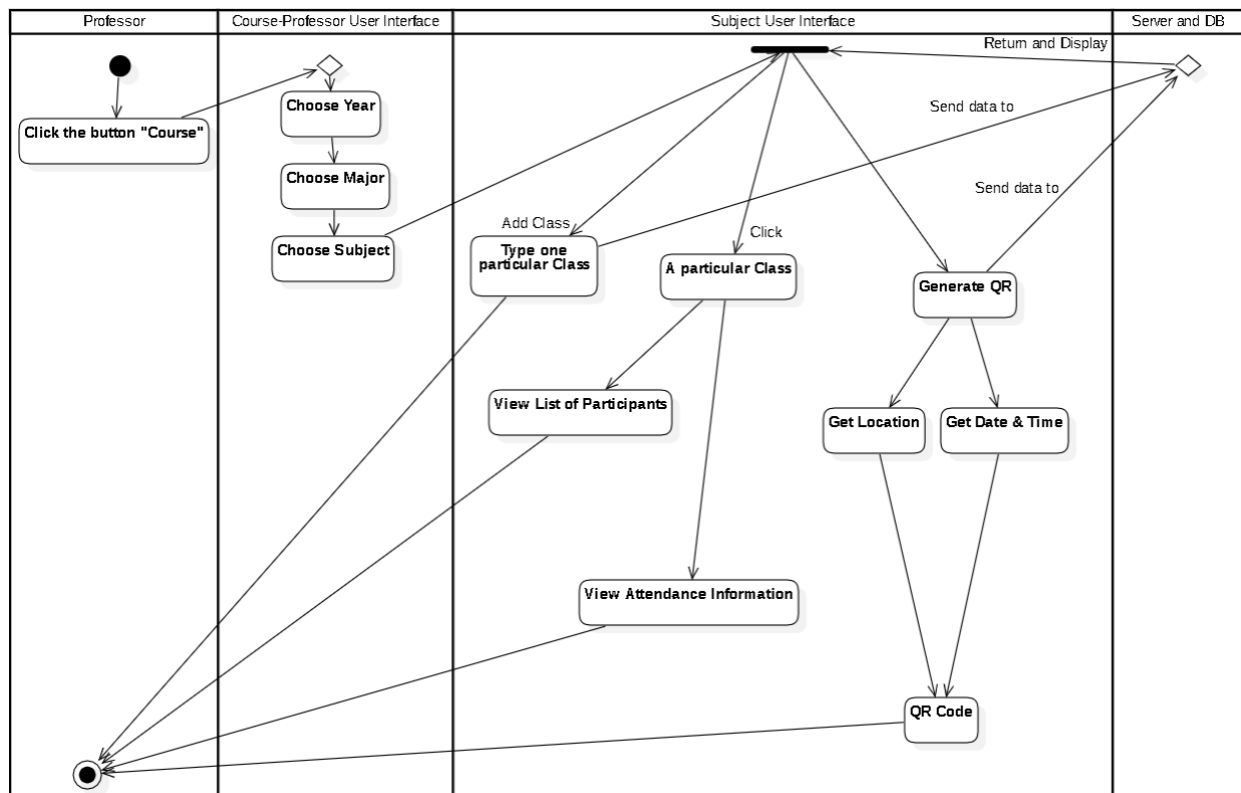
\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.



**Figure 17.** Course-Professor Sequence Diagram



\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.



**Figure 18.** Course-Professor Activity Diagram

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

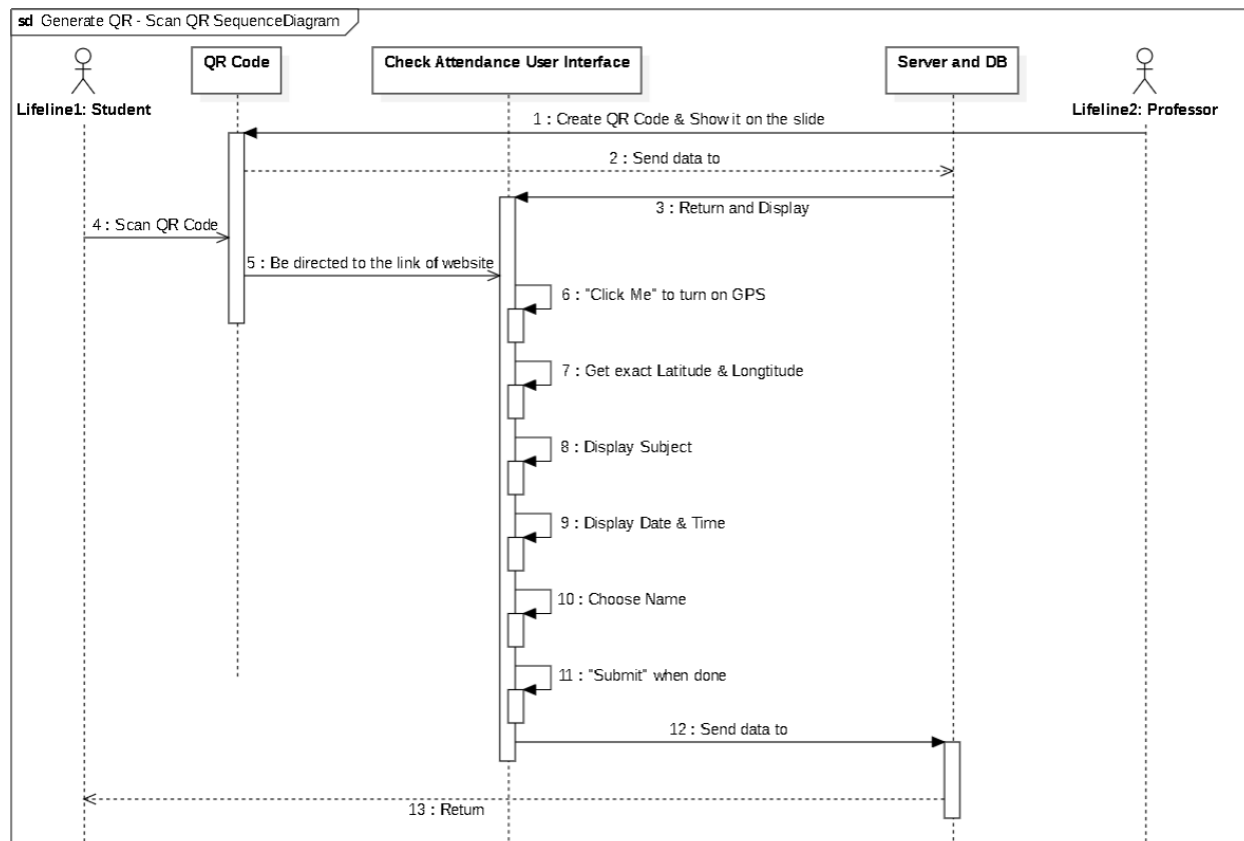
#### 4.8. Generate QR - Scan QR

We do not put this part in the Use Case because there are not any scanning QR parts in the website. If students want to check their attendance when the professor shows a QR Code on the slide, they have to use the Scanning QR Code Application being available on their devices.

Although, the process of scanning the QR Code generated by the Professor still plays a remarkable and important role in checking attendance. Hence, here, we will illustrate the process of scanning the QR Code of a student from a pre-existing QR Code generated by a professor and how the data is processed in the database.

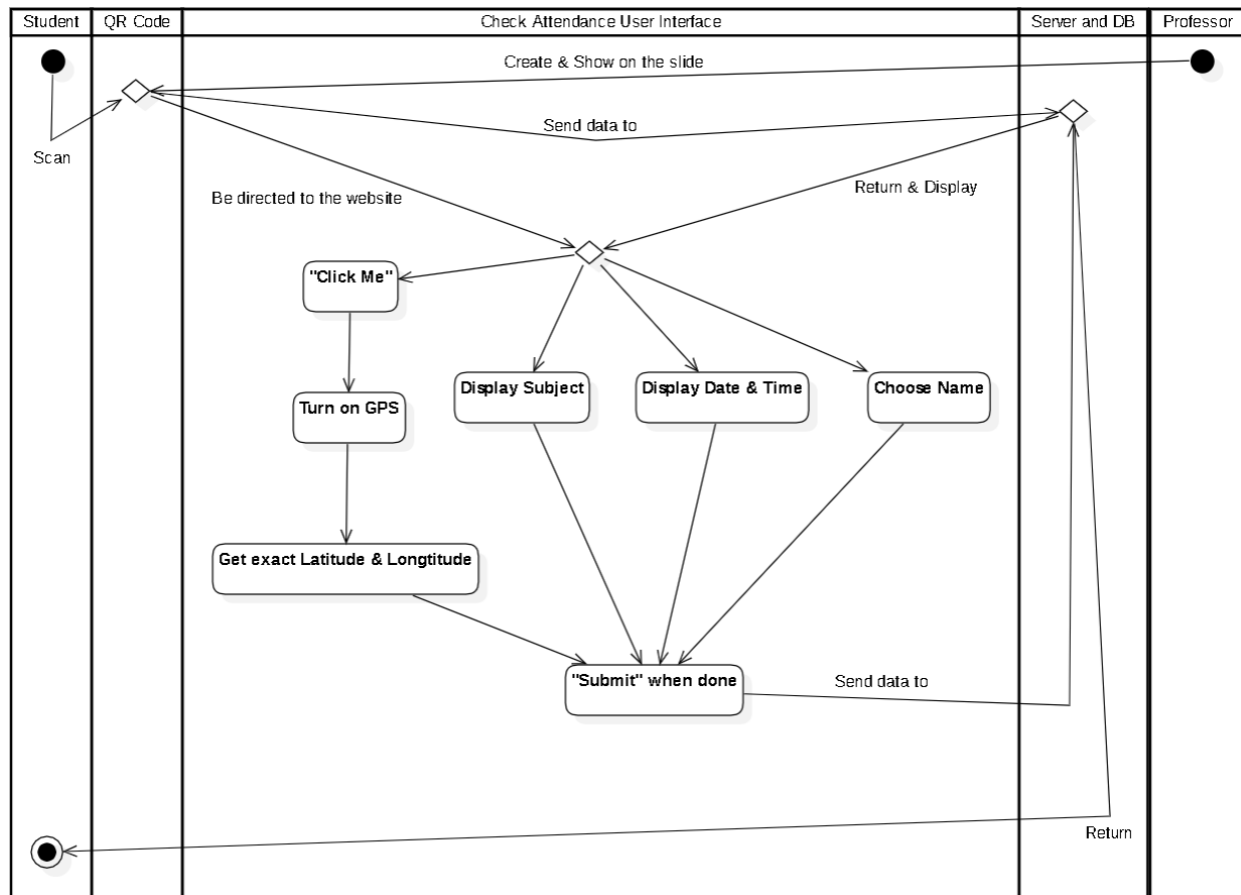
Initially, the Professor creates a QR Code and shows it on the slide. The data in this QR Code is stored in the Server and DB. When a Student scans the QR Code, a link to the website will be displayed. If he/she clicks this link, he/she will be directed to the website of Check Attendance User Interface. The Server and DB transfers data there to display the information about Subject and Date & Time. In order to finish checking attendance, a Student has to click the button “Click Me” and turn on the GPS to get the exact Latitude & Longitude of the location; and choose his/her exact Name. Whereas he/she has done, he/she could click the button “Submit”. All the information about his/her checking attendance is sent to the Server and DB. Finally, the Student will receive notification when he/she checks his/her own attendance completely.

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.



**Figure 19.** Generate QR - Scan QR Sequence Diagram

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.



**Figure 20.** Generate QR - Scan QR Activity Diagram

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

## V. RESULT AND DISCUSSION

### *1. Result*

After three months of development and upgradation, our checking attendance system by using GPS to get as exact location as possible on a mobile website has basically been completed. We have almost achieved our goals we have proposed at first. To be more specific, the website has been designed with the following functions:

- **Authentication: Register & Login:**

+ Be able to allow students to create an account and access into the system.

+ Be able to allow professors to access based on the account that has been provided before.

- **Edit Profile:** be able to allow students or professors to edit and change their personal private information.

- **Attendance Information:** be able to allow students to view their own attendance information.

- **Course:**

+ Be able to allow students to join the class for a particular subject, view list of all participants in the class they have already enrolled in, and view attendance data of all students in the class by selecting a year, a major and a subject relevant to the course about which they want to know further information.

+ Be able to allow professors to choose a year and a major relative to a subject they are teaching, and manage it by creating a class for students to be able to join, monitoring student’s attendance and generating QR codes for students to scan and check their own attendance.

- **Report:** Be able to allow students to write feedback or report about what issues they encounter while using the website and then send it to the database of the Administrator.

Moreover, we succeed in making a system that allows the Students to scan QR Code for checking their own attendance and allows the Professors to generate the QR Code.

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

## *2. Discussion*

When we begin to do the project, we move up step by step slowly. Our aims or objectives are always the desire to succeed in the function of scanning the QR Code generated by the user. First of all, we meet our supervisor and agree on how to work together. We might have a few difficulties in the way of contacting or working because we have not worked together before.

Secondly, we could have difficulty in the orientation of our project. All the things like requirements, use case, sequence and activity diagrams, database design, user interface make us so hard. We have to determine, draw or design them many times. Besides, we are also helped to orient by our supervisor a lot of times. Thanks to him, at least at the moment, we might get the right direction.

Thirdly, we create too many tables in SQL and we might not manage and monitor our database so well. Later, we are guided to optimize the quantity of tables in the database and try to connect them together to administer more easily and conveniently.

Fourthly, when we are working, especially web programming, we have to change a lot of things. At first, we intend to use React and link it to Python or other good techniques. But later, we do not know how to connect React to the database, particularly PHP. Hence, we come back to use HTML & CSS and connect them to SQL and PHP.

Fifthly, so many functions we intend to add to the system initially have to be changed or altered. This is because we have difficulty continuing this part or we do not think that it is possible or reasonable. For instance, the part of the notification displaying how the user interacts with the system is removed by us because we can not complete it in time.

Sixthly, basically, our system is quite complete. Besides, there are still some bugs or errors we have to fix or debug. For example, there are some errors in some accounts registered by our system. Therefore, when the demo is shown, maybe we can not show everything in the demo yet.

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

Seventhly, it is the problem about the domain. Firstly, in the part of authentication, we can not get and connect it to the domain of gmail. Like we can not make the domain “@usth.edu.vn” validate yet. Secondly, we have not bought the domain or the server yet so our website system can not be used on the mobile phone.

Lastly, we may have some other problematic matters or issues like too much gasoline to move, our computer running out of battery fast or broken,... or we may not work well as a real team or group, but we always try to overcome difficulties anytime, anywhere.

To conclude, while we are doing the project, some problems happen and we may have difficulties. They are invaluable experiences we have ever had and we try to do better next time.

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

## **VI. CONCLUSION AND FUTURE WORK**

### *1. Conclusion*

In conclusion, the objectives and purposes of the Web Platform for Checking attendance system by using GPS is achieved. The Student could scan the QR Code for checking their own attendance and the Professor is able to create the QR Code. By providing a friendly, comfortable and simple user interface on the website, the interaction between the system and users like students or professors can become more convenient. Managing data or databases could become easier and more efficient.

The main functions of the Web Platform for Checking attendance system by using GPS have already been implemented:

- Users can log into the system and edit their personal private information.
- Students can create accounts, send reports, view attendance information, join class to view all participants in the class and check their own attendance by scanning QR Codes.
- Professor can create classes and generate QR Codes for students to scan to check their own attendance.

### *2. Future Work*

To further improve this web application in the future, the following tasks need to be done:

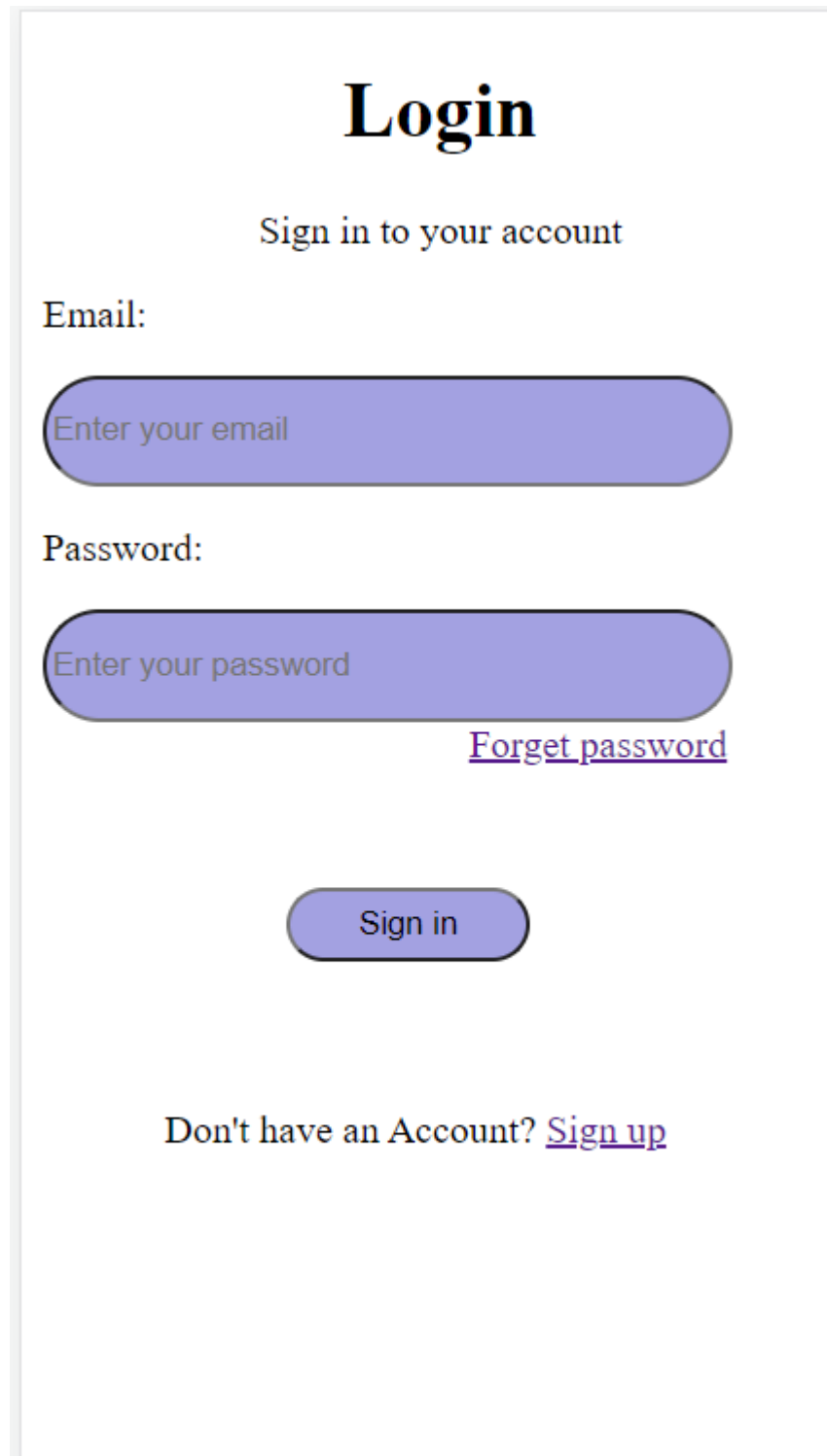
- Fix all errors and debug all bugs.
- Improve and develop website security.
- Have the DNS servers for our system.
- Develop our system to be able to be used on mobile phones.
- Make a Guideline for using the system.
- Improve and develop the performance or the user interface of the system to be more responsive and friendly.
- Design the system more beautifully and eye-catchingly.

If our system is forecast to become more complete and perfect in the future, I think our system could be used widely and become more ubiquitous as a cheap, fast, convenient and effective method.



\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

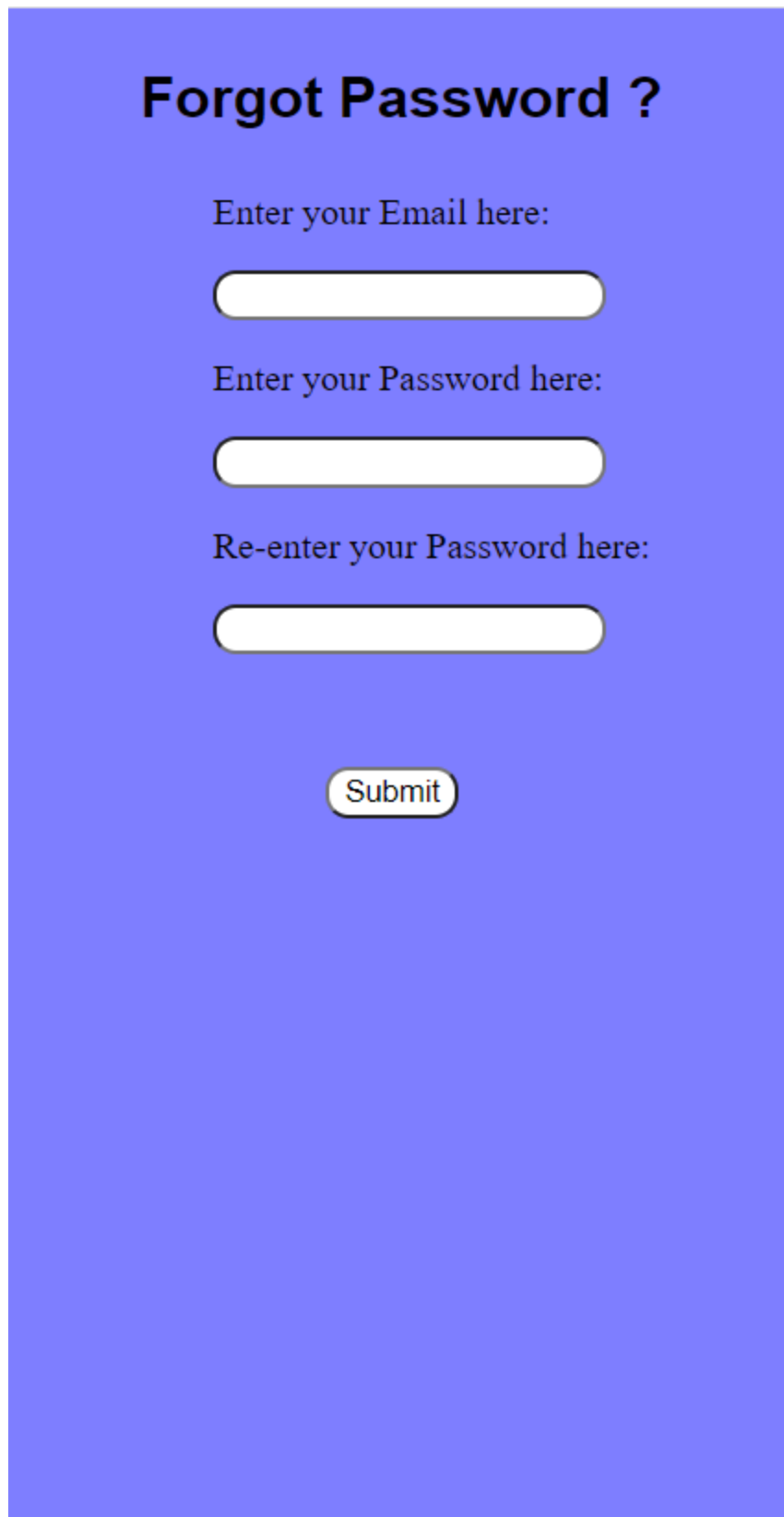
## VII. APPENDIX - USER INTERFACE



The image shows a login user interface. At the top, the word "Login" is displayed in a large, bold, black serif font. Below it, the text "Sign in to your account" is centered in a smaller, black serif font. The form consists of two main input fields, each with a light purple rounded rectangular background and a thin black border. The first field is preceded by the label "Email:" and contains the placeholder text "Enter your email". The second field is preceded by the label "Password:" and contains the placeholder text "Enter your password". To the right of the password field, the text "[Forget password](#)" is displayed in a purple serif font. Below the password field, there is a light purple rounded rectangular button with the text "Sign in" in a black serif font. At the bottom of the form, the text "Don't have an Account? [Sign up](#)" is displayed, with "Sign up" being a purple underlined link in a serif font.

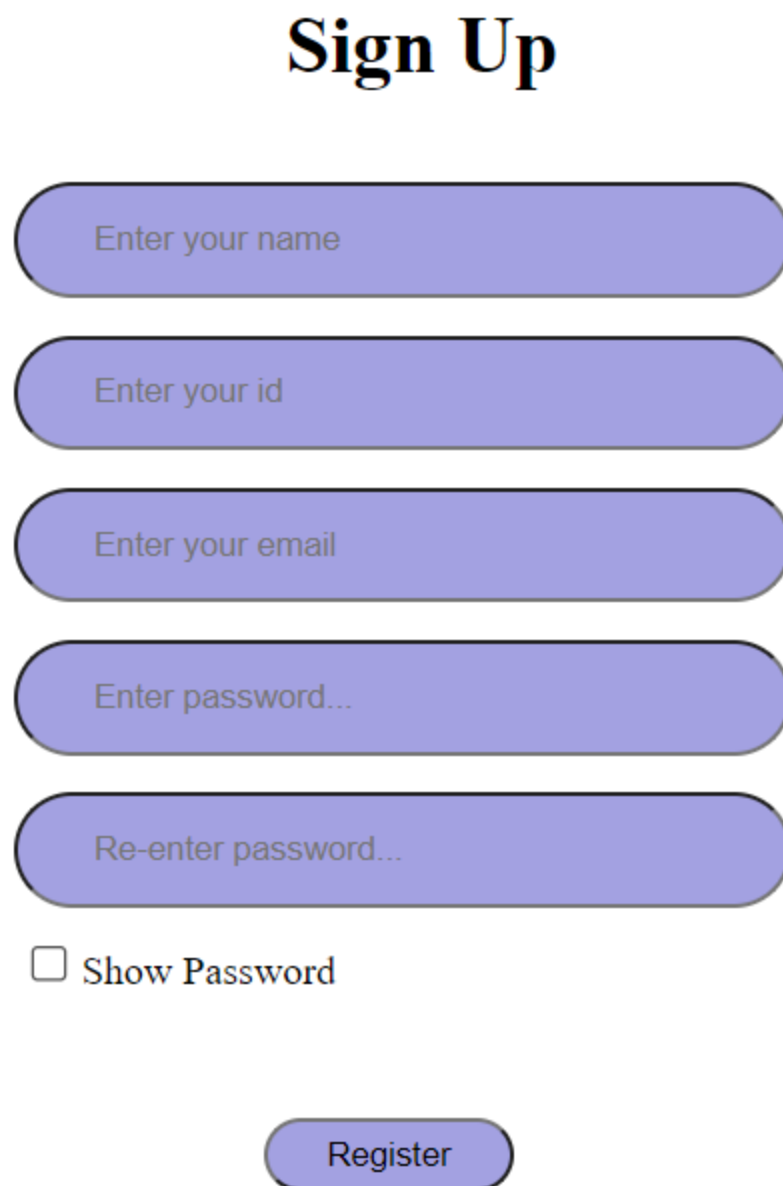
**Figure 21.** Login User Interface (both Student and Professor)

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

The image shows a user interface for a 'Forgot Password' function. It has a solid blue background. At the top, the text 'Forgot Password ?' is displayed in a large, bold, black font. Below this, there are three input fields, each preceded by a label in a black serif font. The first label is 'Enter your Email here:', followed by a white rounded rectangular input field. The second label is 'Enter your Password here:', followed by a white rounded rectangular input field. The third label is 'Re-enter your Password here:', followed by a white rounded rectangular input field. At the bottom of the form, there is a white rounded rectangular button with the word 'Submit' in a black serif font.

**Figure 22.** Forget Password User Interface (both Student and Professor)

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.



The image displays a 'Sign Up' form for a student user interface. The form is centered on a light gray background and consists of several rounded rectangular input fields and a button. The title 'Sign Up' is prominently displayed at the top in a large, bold, black serif font. Below the title, there are five stacked input fields, each with a light purple background and a thin black border. The labels for these fields are 'Enter your name', 'Enter your id', 'Enter your email', 'Enter password...', and 'Re-enter password...', all in a light gray sans-serif font. Below the input fields, there is a checkbox labeled 'Show Password' in a black sans-serif font. At the bottom of the form, there is a single rounded button with a light purple background and a thin black border, labeled 'Register' in a black sans-serif font.

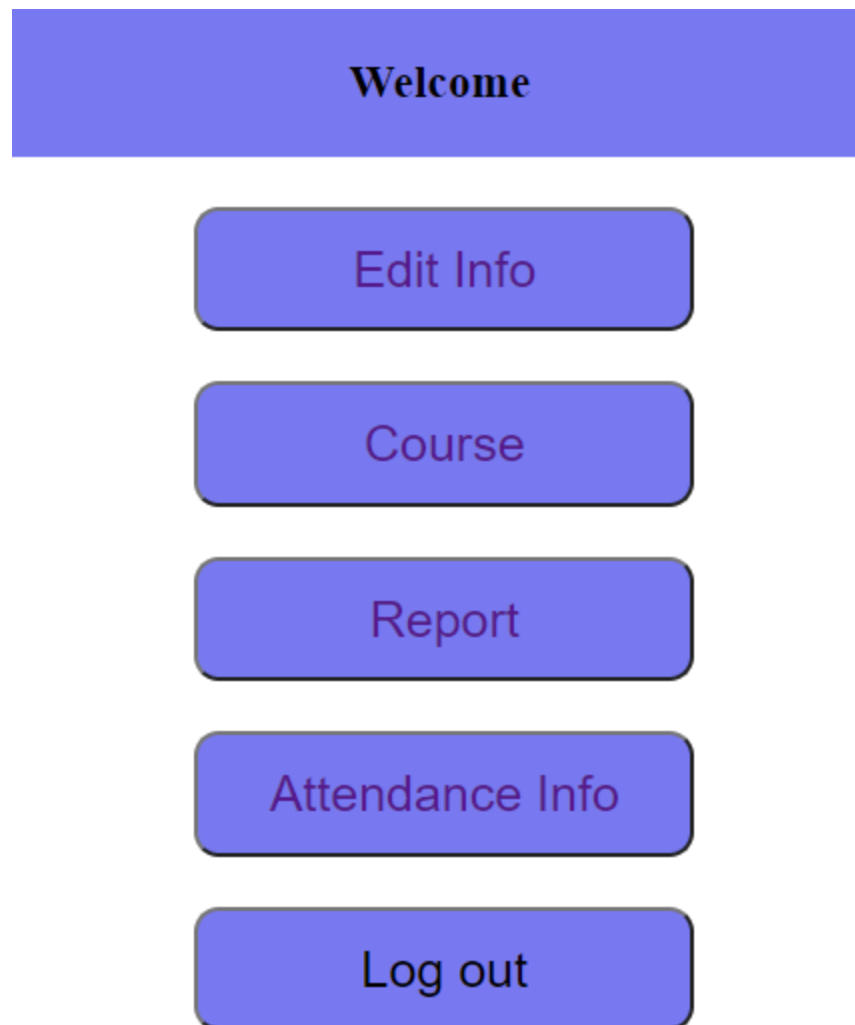
# Sign Up

☐ Show Password

Register

**Figure 23.** Sign Up Student User Interface

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.



---

**Figure 24.** Welcome Student User Interface

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

The image shows a user interface for editing student information. It features a blue header bar with the text "Please enter your information". Below this, there are three input fields: "Full name:" with the value "Hoang Quang Huy", "Email:" with the value "phunggiahuy17@gmail.com", and "Student ID:" with the value "BI11-112". At the bottom, there is a blue bar containing two buttons: "Cancel" and "Accept".

Please enter your information

Full name:

Hoang Quang Huy

Email:

phunggiahuy17@gmail.com

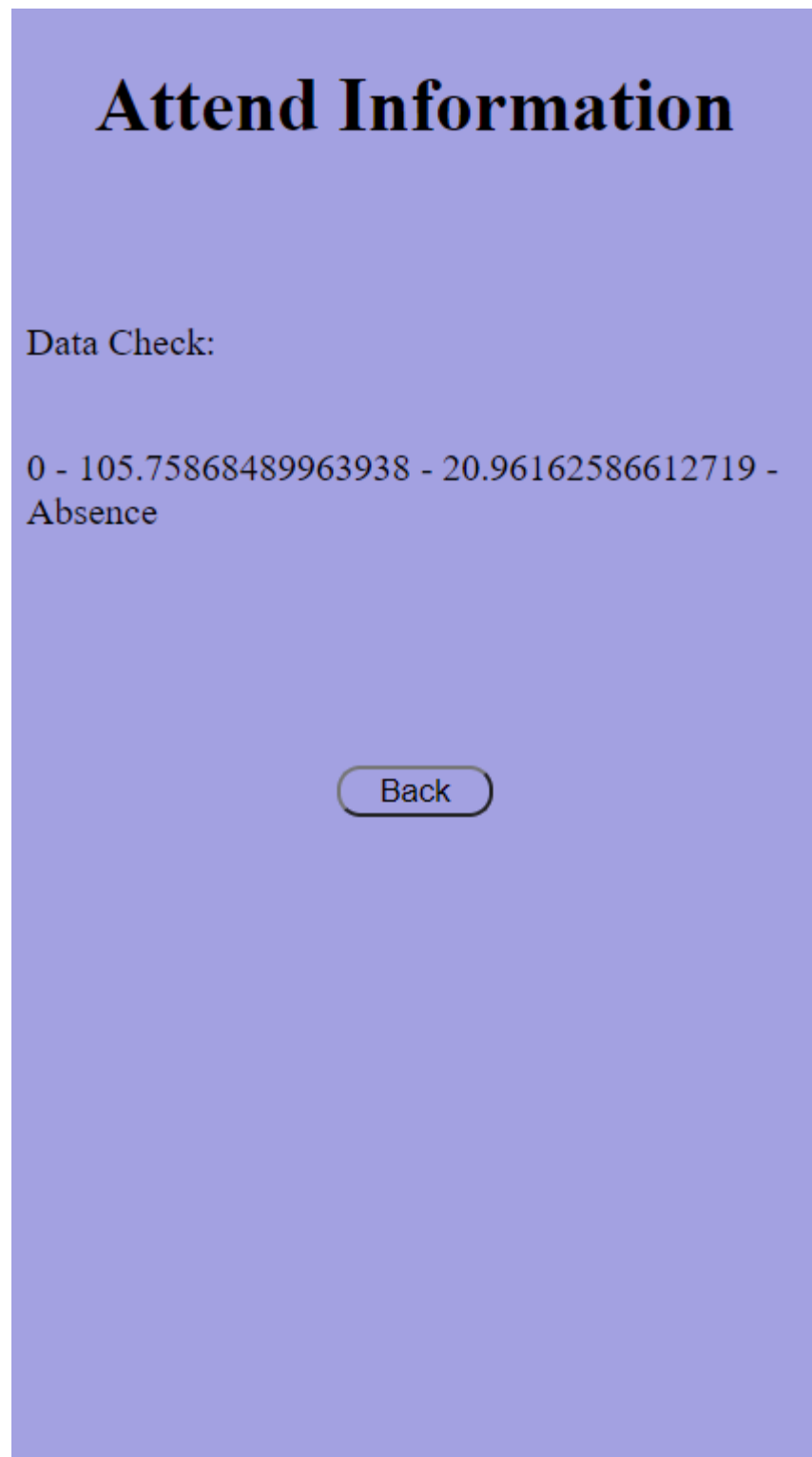
Student ID:

BI11-112

Cancel Accept

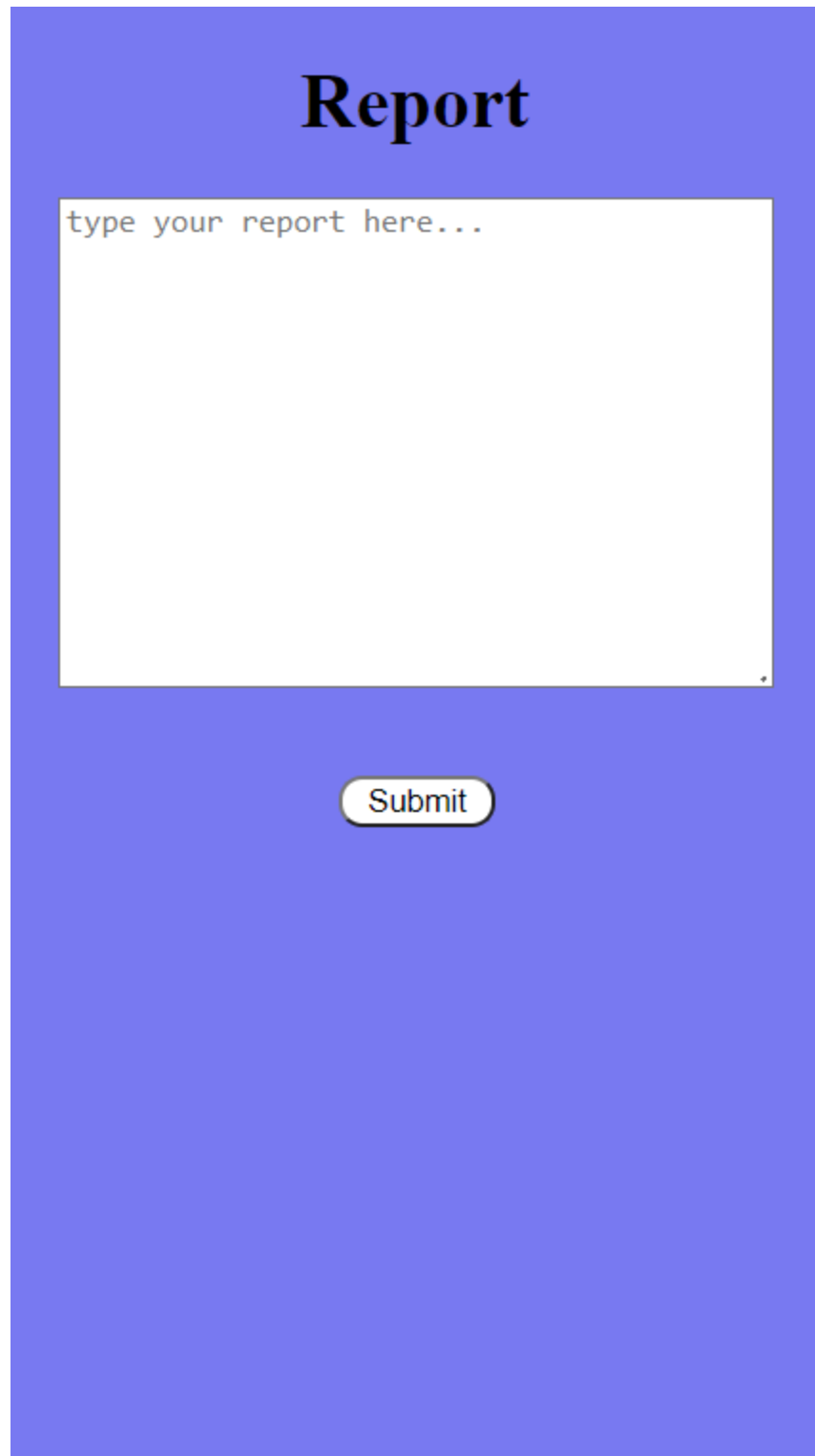
**Figure 25.** Edit Information Student User Interface

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.



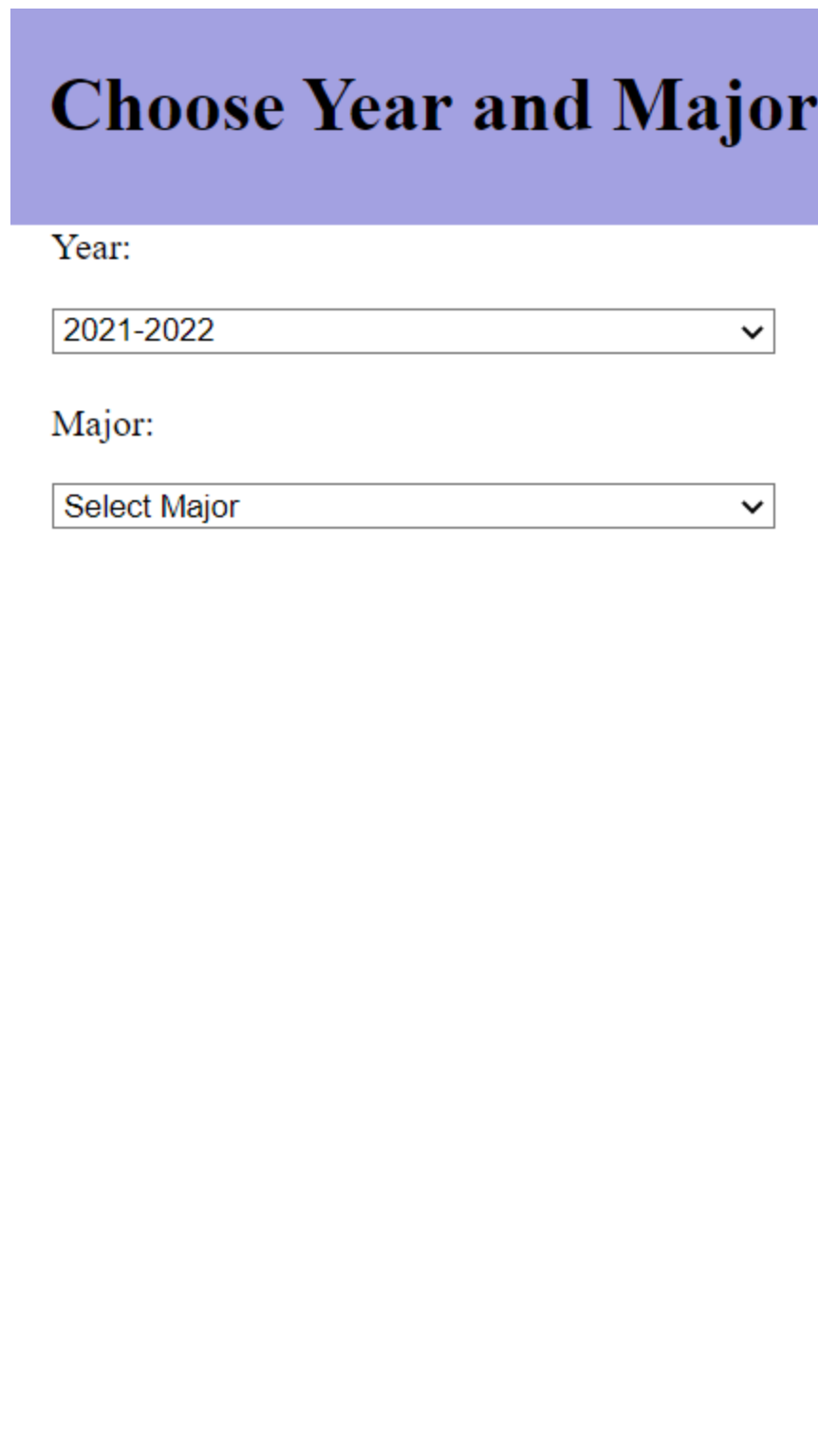
**Figure 26.** Attendance Information Student User Interface

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

The image shows a user interface for submitting a report. It features a solid blue background. At the top, the word "Report" is written in a large, bold, black serif font. Below this, there is a large white rectangular text area with a thin black border. Inside the top-left corner of this area, the text "type your report here..." is written in a small, grey, sans-serif font. At the bottom-right corner of the white area, there is a small, faint asterisk (\*). Centered below the white text area is a white, rounded rectangular button with a thin black border. The button contains the word "Submit" in a black, sans-serif font.

**Figure 27.** Report Student User Interface

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.



The image shows a web interface for selecting a course. It features a purple header with the text "Choose Year and Major" in a large, bold, black serif font. Below the header, there are two dropdown menus. The first is labeled "Year:" and has a white box with the text "2021-2022" and a small black downward arrow on the right. The second is labeled "Major:" and has a white box with the text "Select Major" and a small black downward arrow on the right. The entire form is set against a light gray background.

**Figure 28.** Course User Interface (both Student and Professor)



\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

The image shows a web interface for selecting a course. It features a purple header with the text "Choose Year and Major". Below this, there are three dropdown menus: "Year:" with "2021-2022" selected, "Major:" with "ICT" selected, and "Subject:". Below the "Subject:" label is a link labeled "Informatic". At the bottom right, there is a purple rectangular area containing a "Back" button.

Choose Year and Major

Year:

2021-2022

Major:

ICT

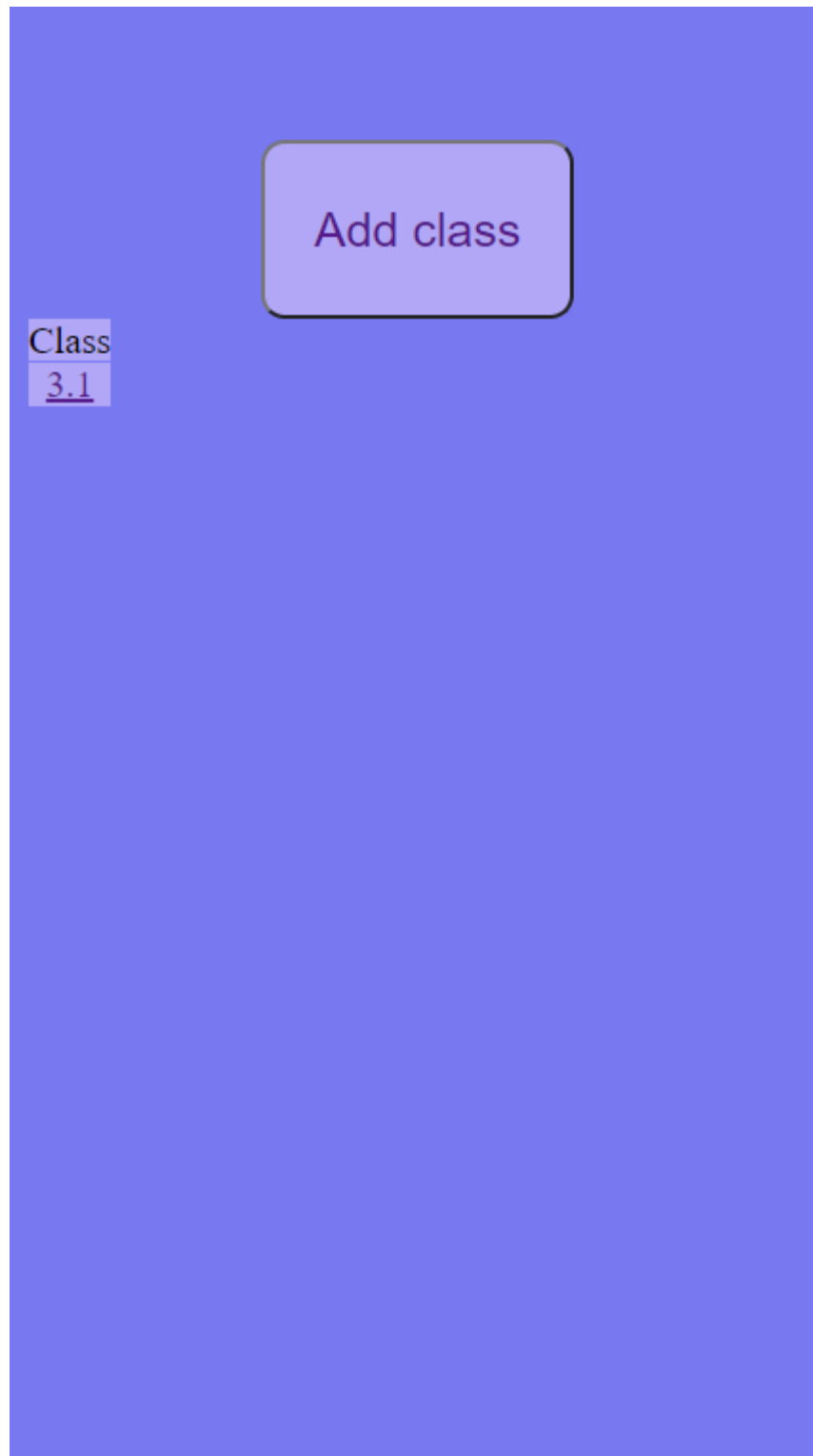
Subject:

[Informatic](#)

Back

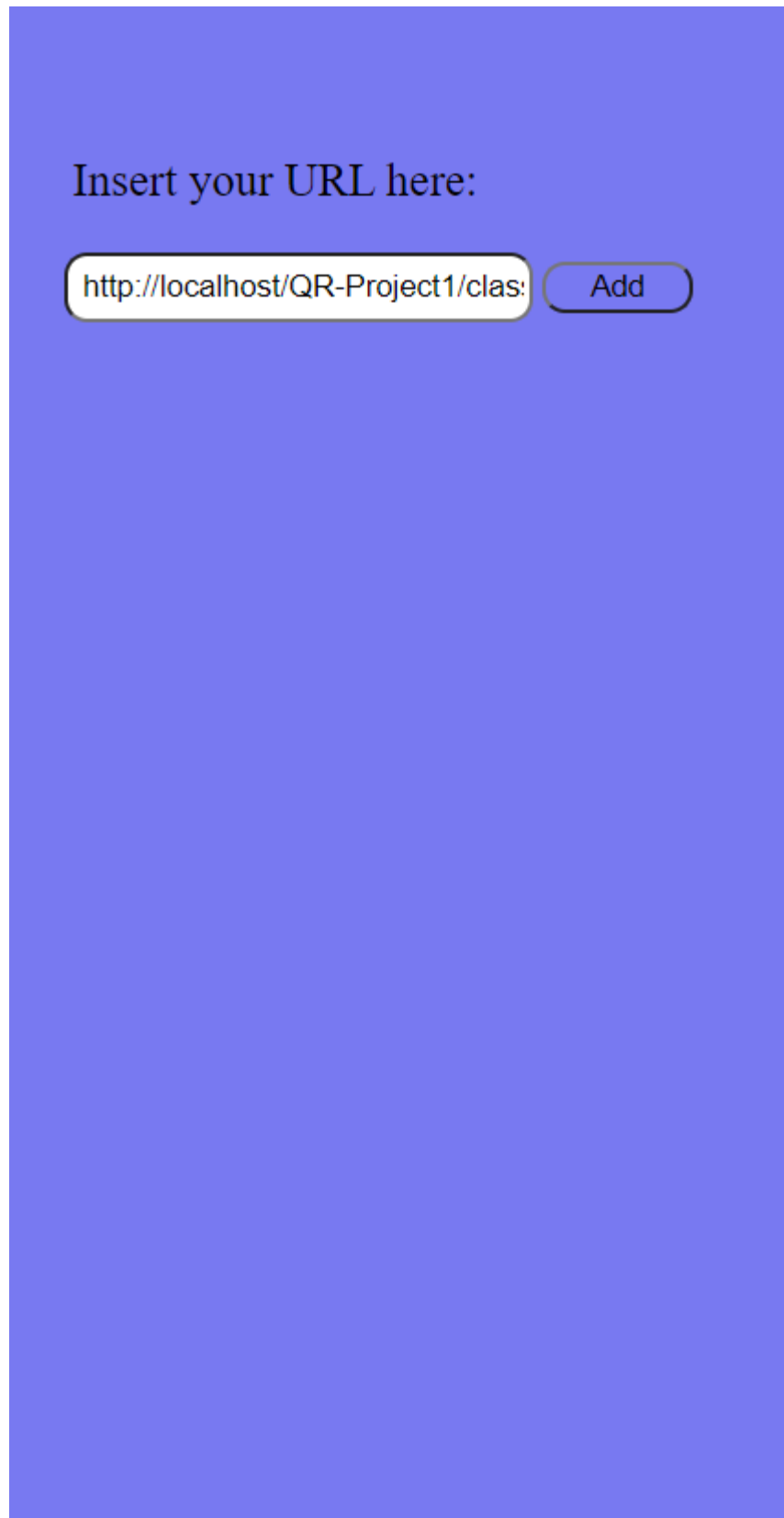
**Figure 29.** Course User Interface (after choose Year and Major) (both Student and Professor)

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.



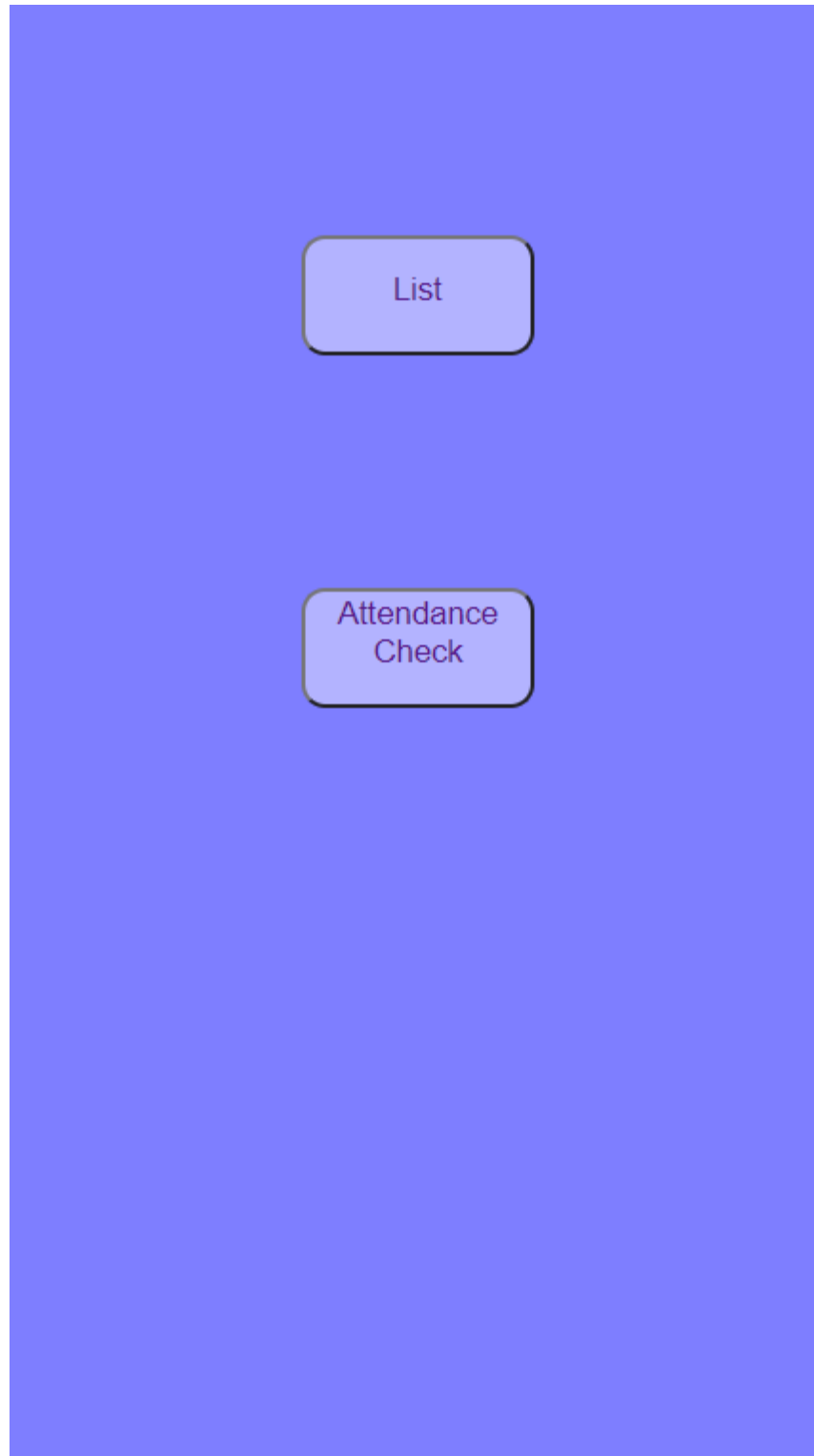
**Figure 30.** Subject Student User Interface

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

The image shows a user interface for joining a class. It features a solid blue background. At the top, the text "Insert your URL here:" is displayed in a white, serif font. Below this text is a white input field with rounded corners, containing the text "http://localhost/QR-Project1/clas:". To the right of the input field is a white button with rounded corners and a blue border, labeled "Add" in blue text.

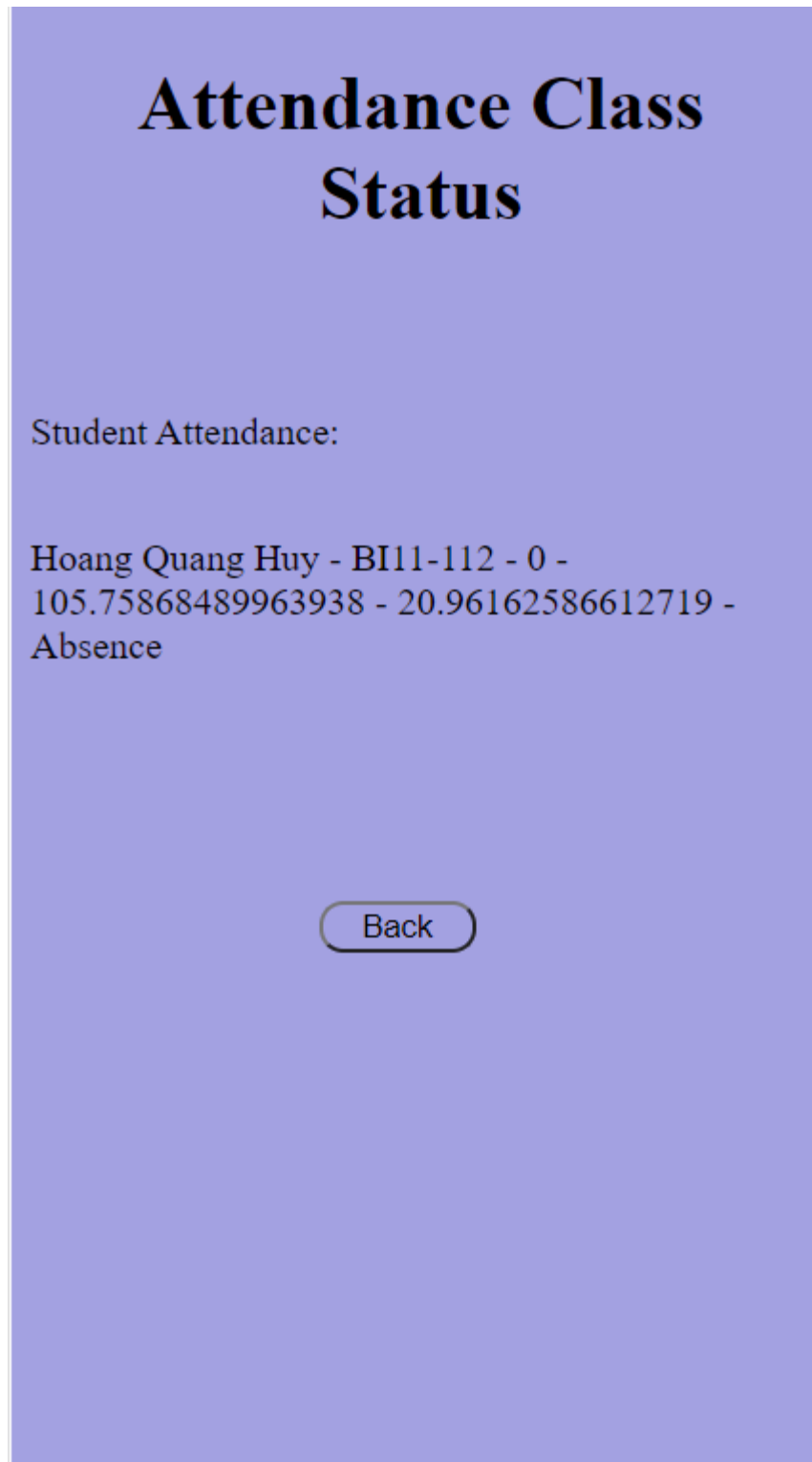
**Figure 31.** Join Class Student User Interface

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.



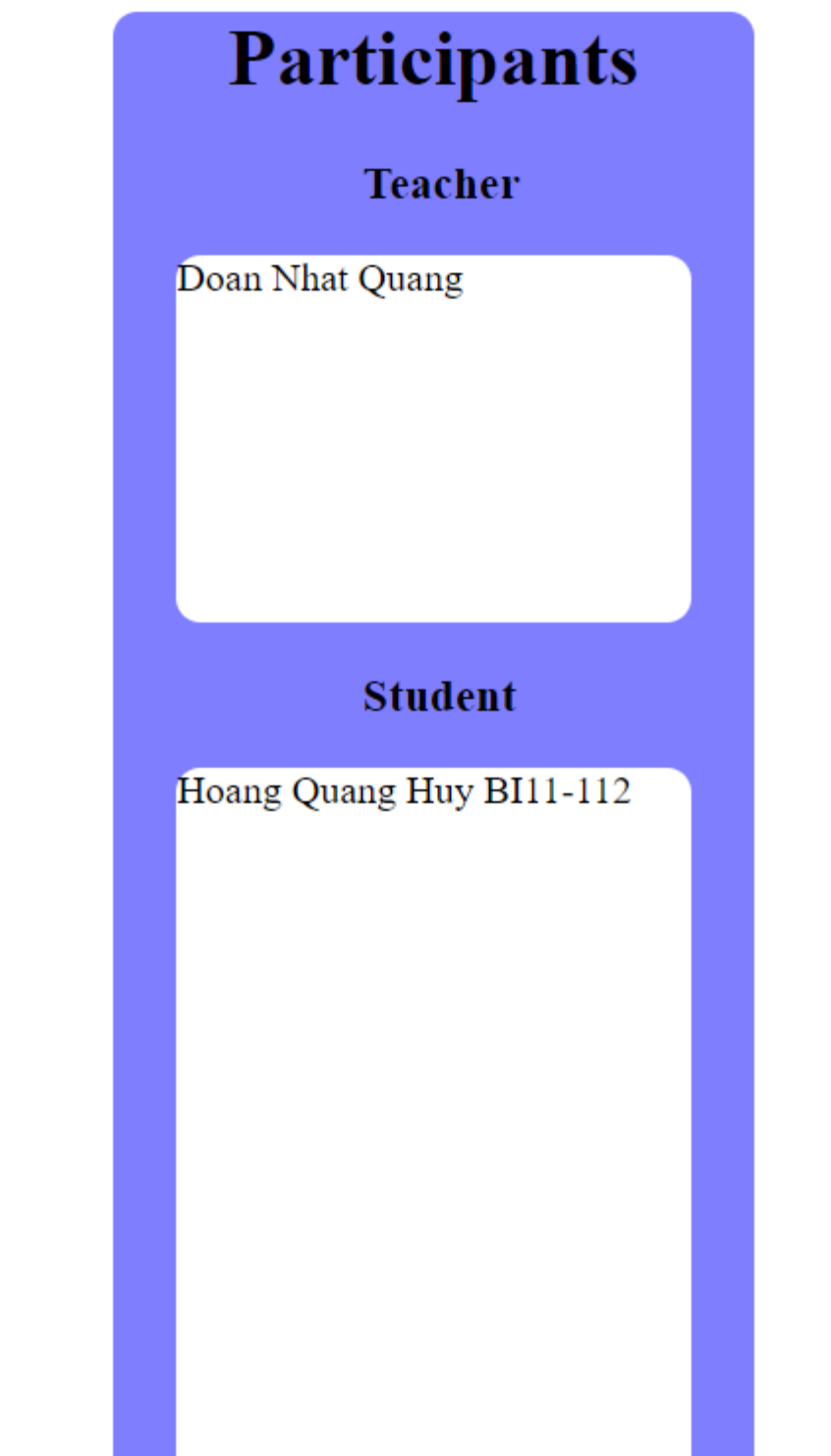
**Figure 32.** Class User Interface (both Student and Professor)

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.



**Figure 33.** Attendance Checking User Interface (both Student and Professor)

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

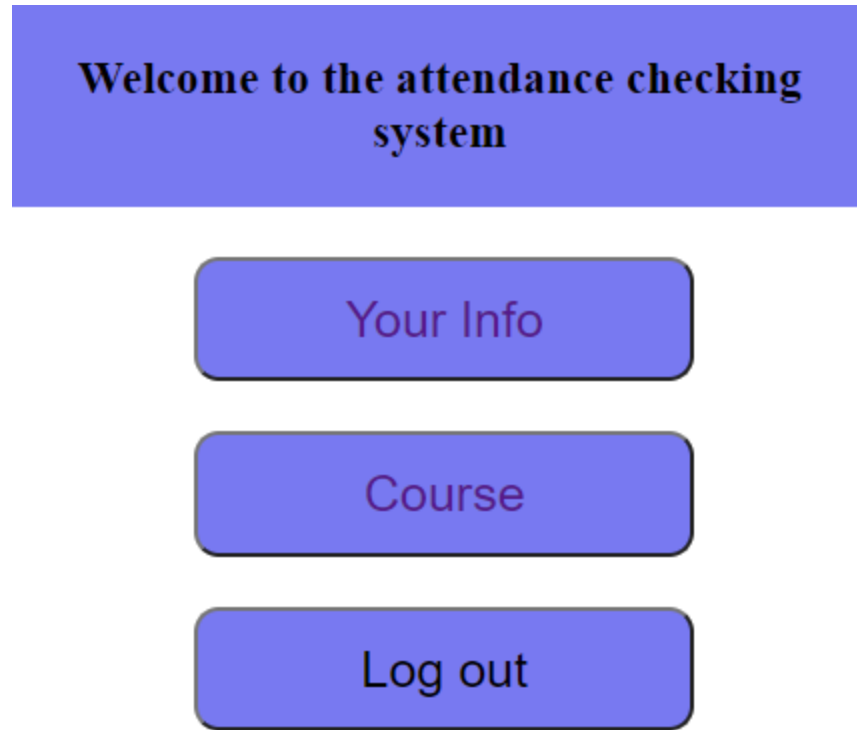


The image shows a user interface for listing participants, enclosed in a blue rounded rectangle. At the top, the word "Participants" is written in a large, bold, black serif font. Below this, the word "Teacher" is centered in a bold, black serif font. Underneath "Teacher" is a white rounded rectangle containing the text "Doan Nhat Quang" in a black serif font. Below the "Teacher" section, the word "Student" is centered in a bold, black serif font. Underneath "Student" is a white rounded rectangle containing the text "Hoang Quang Huy BI11-112" in a black serif font.

Teacher	Student
Doan Nhat Quang	Hoang Quang Huy BI11-112

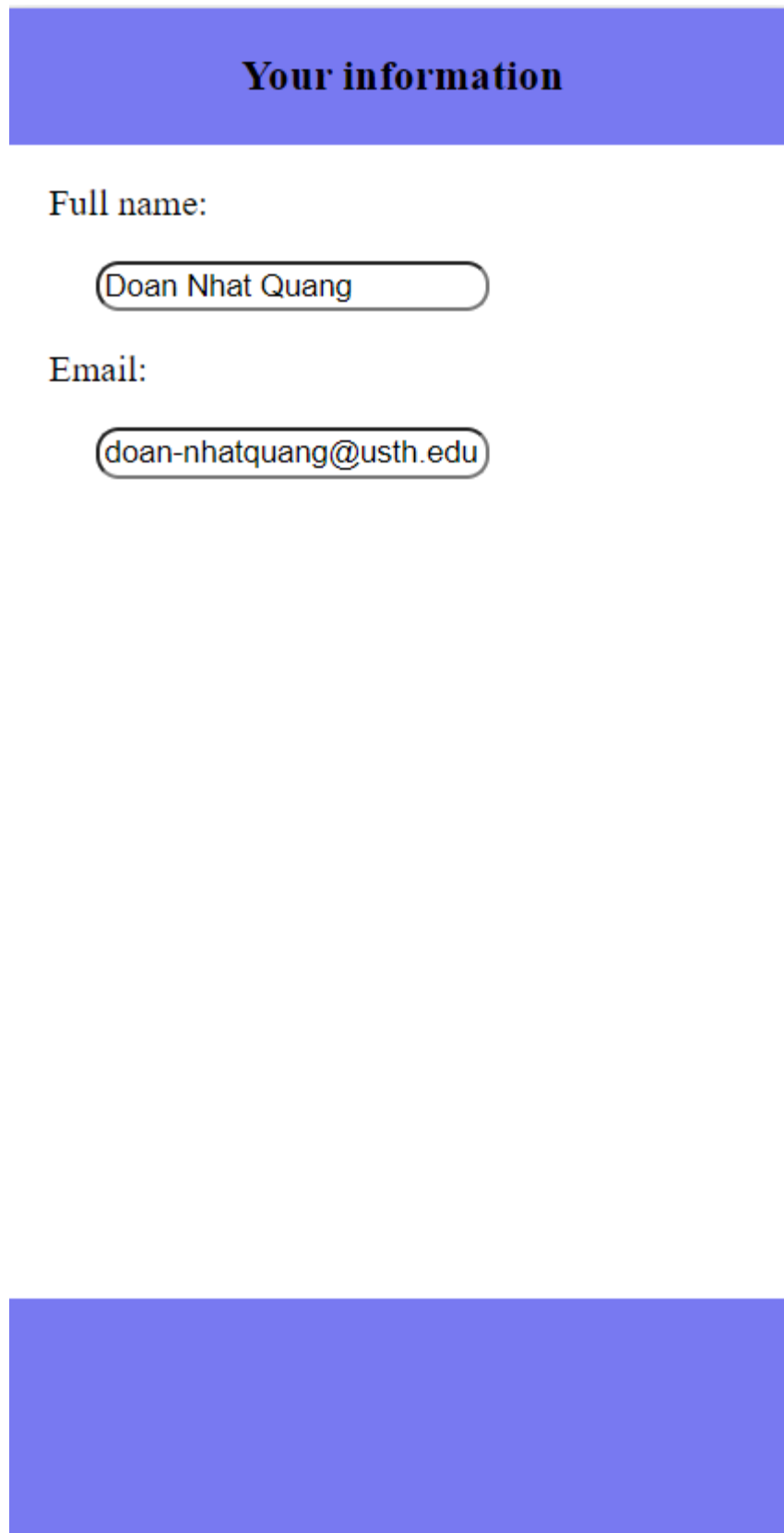
**Figure 34.** List Participants User Interface (both Student and Professor)

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.



**Figure 35.** Welcome Professor User Interface

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.



**Your information**

Full name:

Doan Nhat Quang

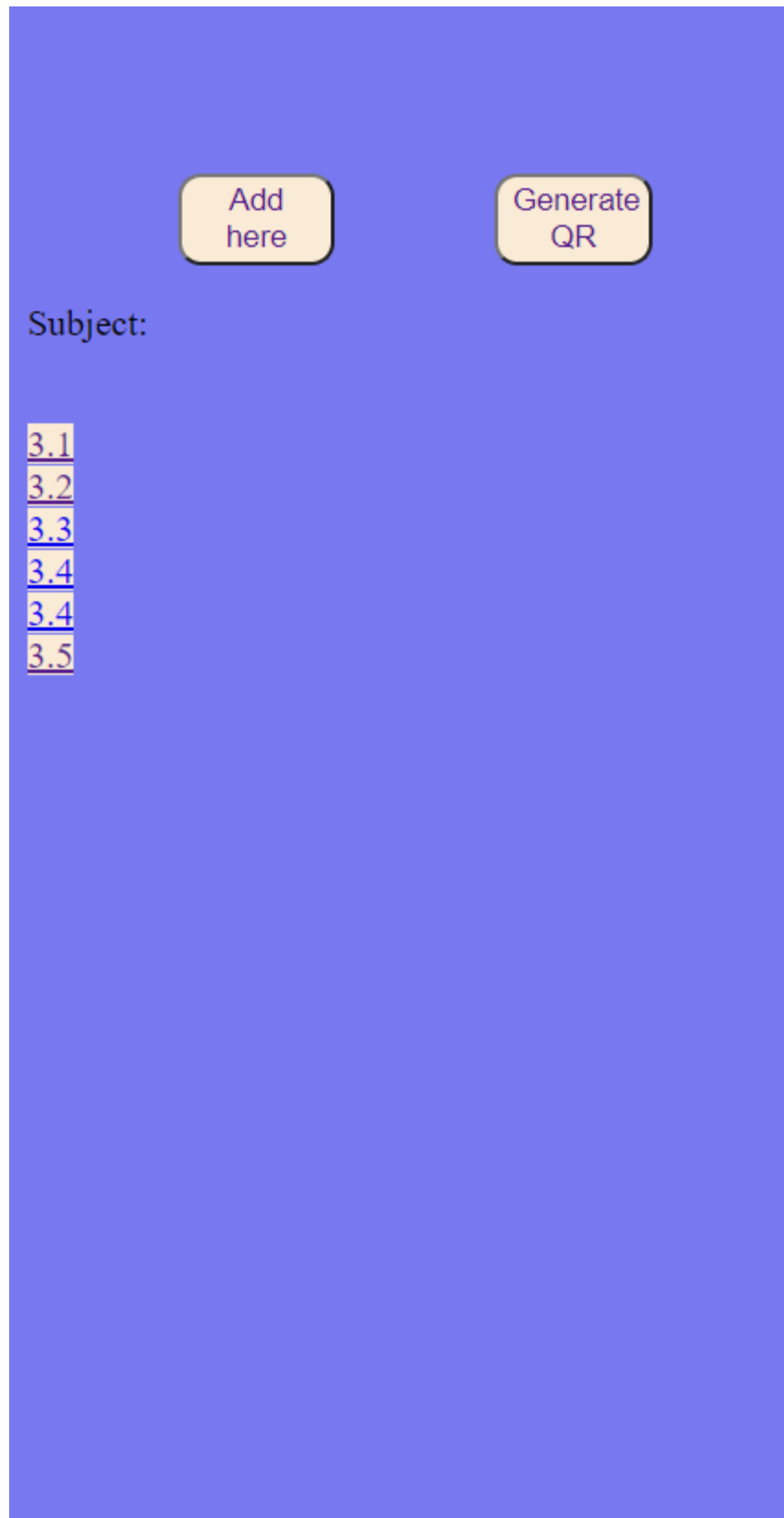
Email:

doan-nhatquang@usth.edu

**Figure 36.** View Information Professor User Interface

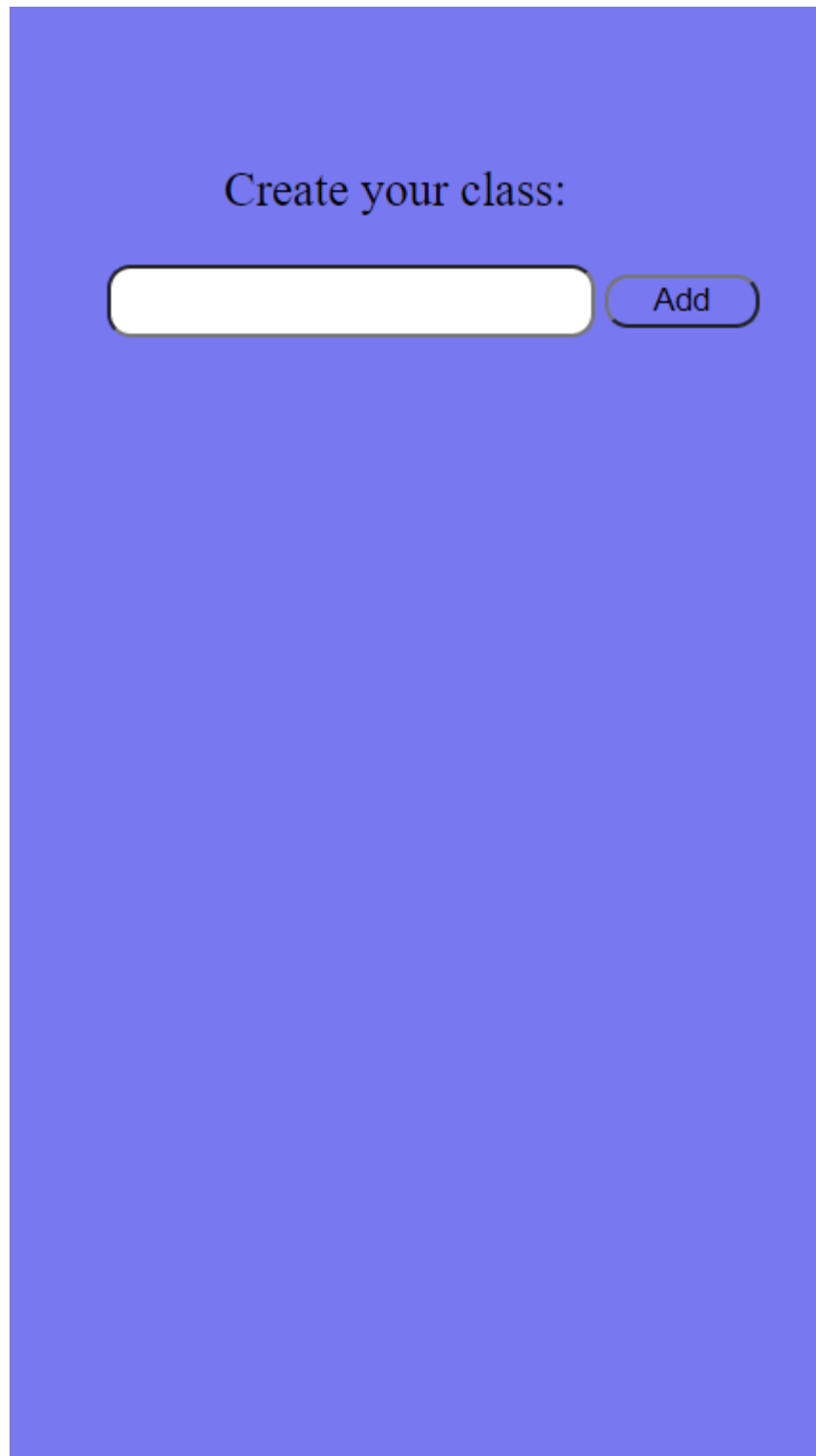


\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.



**Figure 37.** Subject Professor User Interface

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

The image shows a user interface for creating a class. It features a solid blue background. At the top, the text "Create your class:" is centered in a black, serif font. Below this text is a horizontal input field with a white background and a thin black border. To the right of the input field is a rounded rectangular button with a black border and the word "Add" in a black, serif font.

**Figure 38.** Create Class Professor User Interface

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

Displaying location using Latitude and Longitude

Click Me

Latitude:

Longitude:

Date:

dd/mm/yyyy

Subject:

Informatic

Submit

QR Code Generator

Paste a url to create QR code

http://localhost/QR-Project

Longitude:

Date:

dd/mm/yyyy

Subject:

Informatic

Submit

QR Code Generator

Paste a url to create QR code

http://localhost/QR-Project

Generate QR Code

**Figure 39.** Generate QR Code Professor User Interface

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

Click the button to display location with Latitude and Longitude

Click Me

Subject:

Informatic

Date:

2023-01-06

Your name:

Hoang Quang Huy ▼

Latitude:

Longitude:

Submit

**Figure 40.** Checking Attendance Form Student User Interface (after scanning QR)

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

## VIII. REFERENCES

1. Vũ Khánh Huyền. (2020). *AN ONLINE ADMISSION APPLICATION AND INTERVIEW IN UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI* [Bachelor's Thesis, University of Science and Technology of Hanoi].
2. Đỗ Thị Minh Ngọc. (2020). *Develop an online management system for enrollment in University of Science and Technology of Ha Noi* [Bachelor's Thesis, University of Science and Technology of Hanoi].
3. Github. Retrieved 31/10/2022, from [https://github.com/?fbclid=IwAR0W\\_DV1ZG1-tqwG30XmVdTwRwYiesEKmSgGSAJ00nPHZyTtjlzjp1a0fv0](https://github.com/?fbclid=IwAR0W_DV1ZG1-tqwG30XmVdTwRwYiesEKmSgGSAJ00nPHZyTtjlzjp1a0fv0)
4. Stack Overflow. Retrieved 31/10/2022, from <https://stackoverflow.com/?fbclid=IwAR1dns6BbPOKRPTxzRsFFf2bgNf7cLKHwvO3Tb1lwtWPfW3Ik6ubsHIxgiM>
5. OpenJavaScript. OpenJavaScript.info. Retrieved 1/11/2022, from <https://www.youtube.com/@OpenJavaScript>
6. Bootstrap. Retrieved 1/11/2022, from [https://getbootstrap.com/?fbclid=IwAR2o9\\_-u8qSN05ZqEomcO3XQQTmIY9M4-r24z38LvUxDNVwkcigTFbtx6A](https://getbootstrap.com/?fbclid=IwAR2o9_-u8qSN05ZqEomcO3XQQTmIY9M4-r24z38LvUxDNVwkcigTFbtx6A)
7. *Giới thiệu chương trình đại học tại USTH.* (2022). TRƯỜNG ĐẠI HỌC KHOA HỌC VÀ CÔNG NGHỆ HÀ NỘI. Retrieved 1/11/2021, from [https://tuyensinh.usth.edu.vn/dai-hoc/?fbclid=IwAR0FglYSbbjn7uTw-qzoCW\\_ial34NwnydfOiiJiTFNz\\_BmSQM\\_uQaaZlkNA](https://tuyensinh.usth.edu.vn/dai-hoc/?fbclid=IwAR0FglYSbbjn7uTw-qzoCW_ial34NwnydfOiiJiTFNz_BmSQM_uQaaZlkNA)
8. *Entity-Relationship Diagram Arrow Types.* SOFTWARE IDEAS MODELER. Retrieved 3/11/2022, from [https://www.softwareideas.net/erd-relation-arrows?fbclid=IwAR2o9\\_-u8qSN05ZqEomcO3XQQTmIY9M4-r24z38LvUxDNVwkcigTFbtx6A](https://www.softwareideas.net/erd-relation-arrows?fbclid=IwAR2o9_-u8qSN05ZqEomcO3XQQTmIY9M4-r24z38LvUxDNVwkcigTFbtx6A)
9. USTH Moodle. Retrieved 4/11/2022, from <https://moodle.usth.edu.vn/?fbclid=IwAR2WlnjNkQdQuLxviJIQukLA9mnUBMllk-ModXXCSEiMx5gvw5Hh8H6MzdM>
10. *Database Structure and Design Tutorial.* Lucidchart. Retrieved 5/11/2022, from [https://www.lucidchart.com/pages/database-diagram/database-design?fbclid=IwAR0oASXJLt34j\\_0EQMsZwUd4ZuAcMYgJzQcGidQYV-2QrEyuybWE3UffV6k](https://www.lucidchart.com/pages/database-diagram/database-design?fbclid=IwAR0oASXJLt34j_0EQMsZwUd4ZuAcMYgJzQcGidQYV-2QrEyuybWE3UffV6k)

\* In this report, professor(s) ~ teacher(s). Particularly, if somewhere in the report is “teacher(s)”, it also means “professor(s)” and vice versa. Sorry for the dissimilarity, two different words but just one thing or one meaning.

11. *MySQL Tutorial*. W3schools. Retrieved 5/11/2022, from  
<https://www.w3schools.com/MySQL/default.asp?fbclid=IwAR0HjLyqKwOeVbuE9Xl-ShlkgY82u0MmwcJYPfIyeExZhSz5XfwhScbmQsA>
12. *PHP Tutorial*. W3schools. Retrieved 5/11/2022, from  
<https://www.w3schools.com/php/?fbclid=IwAR3W59GUcQuGCrhod5qsqaV7CPGelneDYxebSaKfcgln4xtBPCE1LizjeE>
13. CodingNepal. (2022). *QR Code Generator in HTML CSS & JavaScript / QR Code Generator in JavaScript*. YouTube<sup>VN</sup>. Retrieved 6/11/2022, from  
<https://www.youtube.com/watch?v=pv5K28zVepE>
14. *HTML Geolocation API*. W3schools. Retrived 6/11/2022, from  
[https://www.w3schools.com/html/html5\\_geolocation.asp?fbclid=IwAR2b-ql6clrWOa9AWzLXhuxXkjp1LpwMHGZHeTEpnJ9EBwiNDnXwIpxzdfM](https://www.w3schools.com/html/html5_geolocation.asp?fbclid=IwAR2b-ql6clrWOa9AWzLXhuxXkjp1LpwMHGZHeTEpnJ9EBwiNDnXwIpxzdfM)
15. The Code Knowledge. (2019). *Dependent Drop Down List in Php and Mysql (Part 1)*. YouTube<sup>VN</sup>. Retrieved 6/11/2022, from  
<https://www.youtube.com/watch?v=fE2O6ngJhzA>
16. The Code Knowledge. (2019). *Dependent Drop Down List in Php and Mysql (Part 2)*. YouTube<sup>VN</sup>. Retrieved 6/11/2022, from  
<https://www.youtube.com/watch?v=uWae39OrlzA>
17. The Code Knowledge. (2019). *Dependent Drop Down List in Php and Mysql (Part 3)*. YouTube<sup>VN</sup>. Retrieved 6/11/2022, from  
<https://www.youtube.com/watch?v=OLfA3SDlca8>
18. Ken Swartwout. (2017). *How to Create Dynamic URLs with PHP and MySQL*. YouTube<sup>VN</sup>. Retrieved 7/11/2022, from  
<https://www.youtube.com/watch?v=39Kd1UCQu8M>