I. Introduction:

Problem Statement

You are a recently graduated student and are employed as a database developer for a large technology company. The company has signed a contract with FPT University with the scenario gave requires a database designed for an Attendance System to connect students and schools. The system is divided into classes, Attendance, Student, Enrollment. This site provides data such as class search and class information time, location, scores, and information exchanged between the school and students.

User Requirement

- Students require the attendance system to display the class schedule, clearly see whether the status is present or absent during the class. See the schedule in the near future as the start and end dates of the course and the study location. And students require a detailed report for each subject
- Teachers require the attendance system such as displaying the student's name, student ID number of each class and each subject. And there is a comment section for each student who is late or absent from the allowed percentage, present or absent for each student in that class.

System Requirement

- Maximum user requirements for a system attendance that lists 40 students in a class,600 students, 20 teachers, and 5 employees
- The service quality of the attendance system is always good despite the addition of entities such as students and classrooms. The system must always protect the island and are not blocked activity data

ERD:

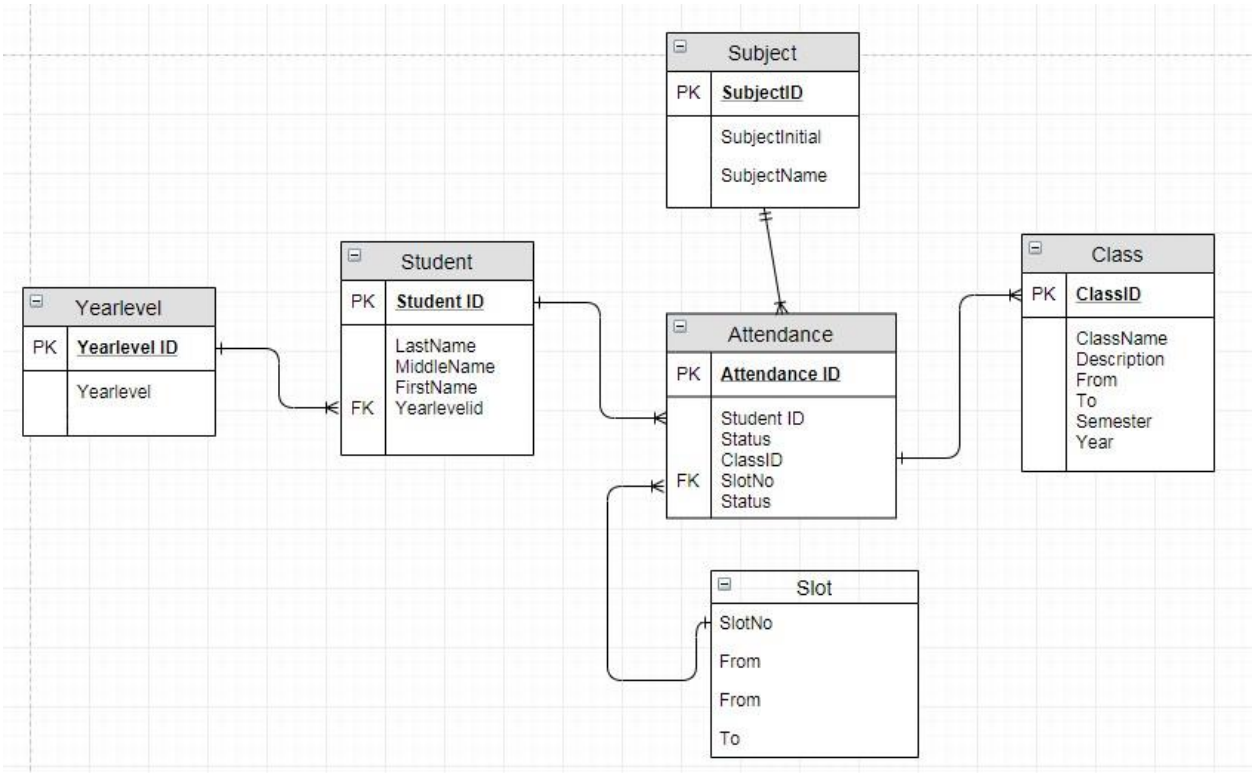

Figure 1 ERD

II. Input Interface :

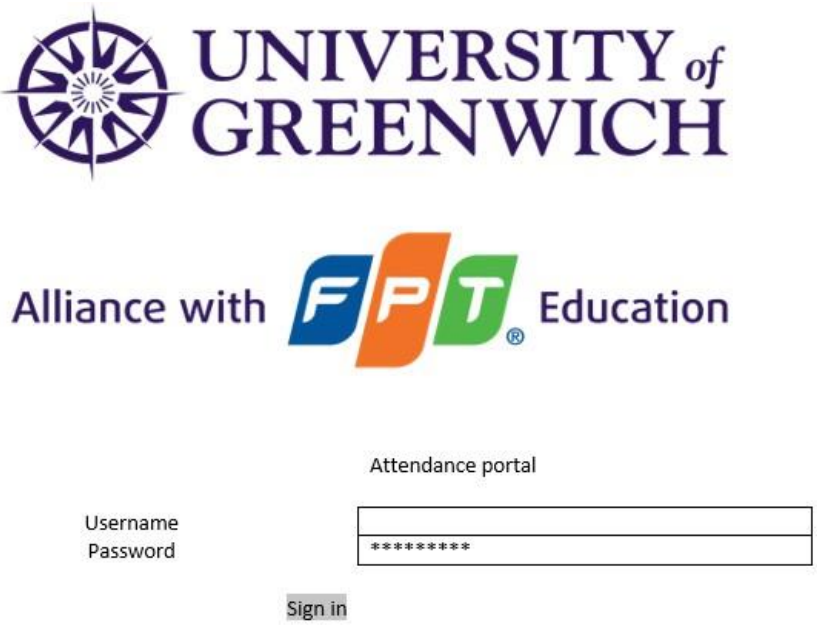

Figure 2 login interface

After logging in at the website, students and teachers will have different interfaces and functions


UNIVERSITY of GREENWICH
Alliance with FPT Education
Student Login
View Attendance Details

Figure 3

Teacher Login

View Attendance Details View Session Details View Subject Details

Take Attendance Update Attendance

Figure 4

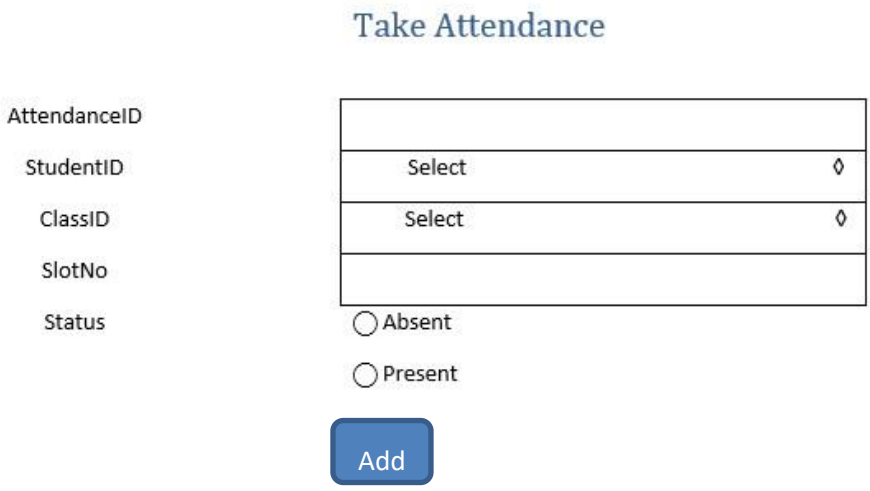

Figure 5 Take Attendance interface

After filling in the information and clicking add and run the query:

```
insert into Attendance values(1,'GCH17230','Absent','GCH0712','SLOT5')
```

Add

Add Yearlevel	Add Class	Add Slot
Add Student	Subject	

Figure 6 Add interface

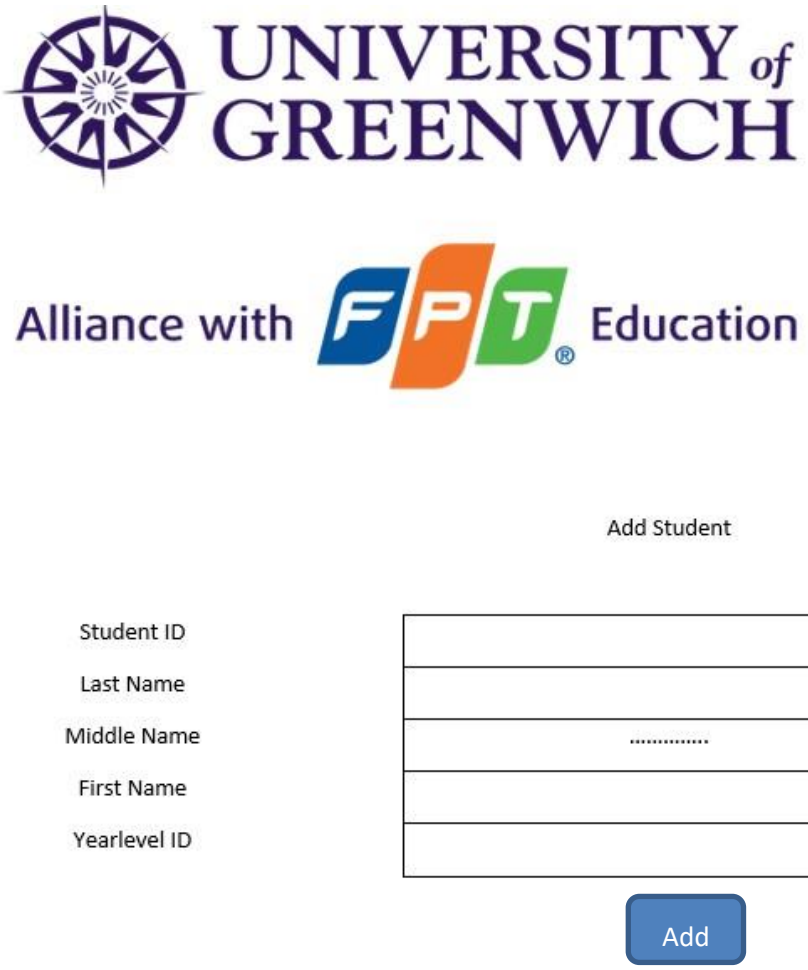

Figure 7

After filling in the information and clicking add and run the query:

```
insert into Student values('GCH17230','VINH','THE','DO',1)
```

After filling in the information and clicking add and run the query:

After filling in the information and clicking add and run the query:

Add Yearlevel

Yearlevel ID	
Yearlevel	

Add

Figure 8

After filling in the information and clicking add and run the query:

```
insert into Yearlevel values(2,'beginer')
```

Add Class

Class ID	Select ◊
Descriptions	
From	12:50 P.M
To	17:30 P.m
Semester	
Year	2019

Add

Figure 9

After filling in the information and clicking add and run the query:

```
insert INTO Class VALUES('GCH0712','Top1dsadsad','12:30',1,2,3)
```

Figure 10

After filling in the information and clicking add and run the query:

```
insert into Subject values(1,'entertainment','Professional Practice (1620)',1)
```

Add Slot

Slot Number	Slot5,Slot6
From	
To	

Add

Figure 11

After filling in the information and clicking add and run the query:

```
insert into Slot values('slot5',2,4)
```

UNIVERSITY of GREENWICH

Alliance with FPT Education

Update

Update Year level	Update Attendance	Update Subject
Update Student	Update Class	Update Slot

Update Interface

Figure 12

Figure 13

After filling in the information and clicking Update and run the query:

```
update Yearlevel set Yearlevel = 'beginer' where Yearlevel_ID = 4
```

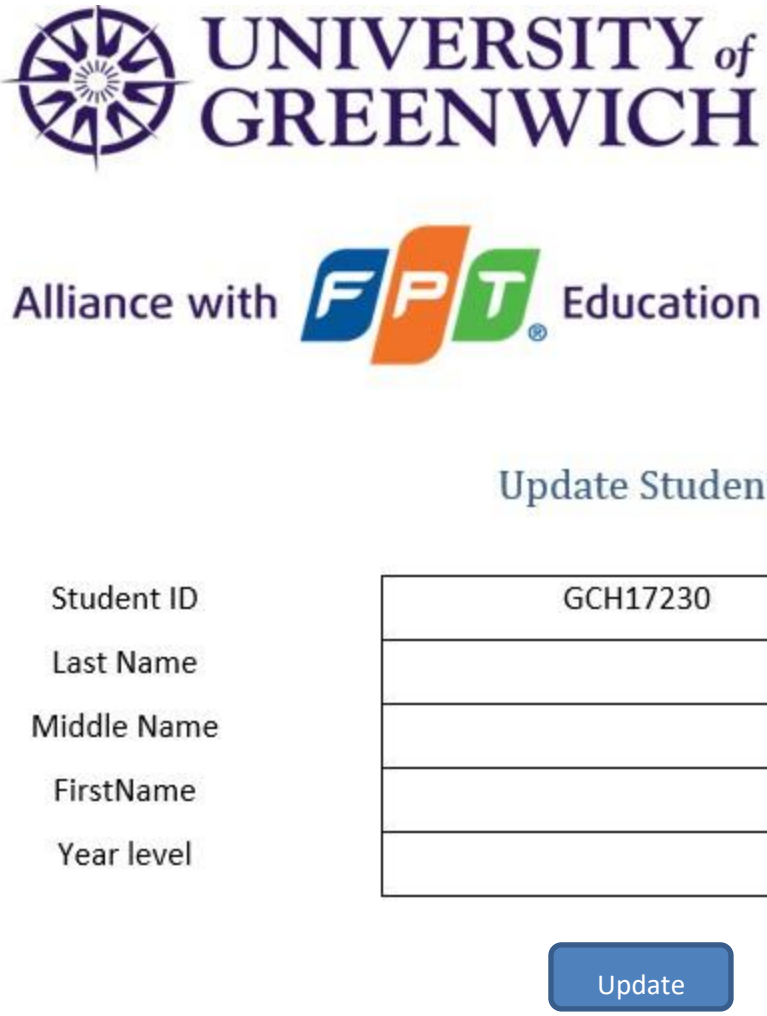

Figure 14

After filling in the information and clicking Update and run the query:

```
update Student set LastName = 'TUAN', FirstName = 'ANH', MiddleName = 'LƯU' where Student_ID = 'GCH17230';
```

Update Attendance

Attendance ID	GCH17230
Student ID	
Status	
Class ID	
Slot Number	

Update

Figure 15

After filling in the information and clicking Update and run the query:

```
update Attendance set Student_ID = 'GCH17231', Statuss = 'Absent' where Attendance_ID = 2
```

Update Class

Class ID	GCH0712
Descriptions	
From	
To	
semester	
Year	

Update

Figure 16

After filling in the information and clicking Update and run the query:

```
update Class set descriptions = 'top1234', DFrom = '1900-01-01'  where Class_ID = 'GCH0712'
```

Update Subject

Subject ID	1622
Subject Initial	
Subject Name	
Attendance	

Update

Figure 17

After filling in the information and clicking Update and run the query:

```
update Subject set SubjectInitial = 'gender', SubjectName = 'database developer(1622)' where Subject_ID = 2
```

Update Slot

Slot Number	
From	
To	

Update

Figure 18

After filling in the information and clicking Update and run the query:

```
update Slot set Dfrom = 5, Dto = 6 where Slot_No = 'slot6'
```

Delete

Delete Year level	Delete Subject	Delete Slot
Delete Student	Delete Class	

Figure 19 Delete Interface

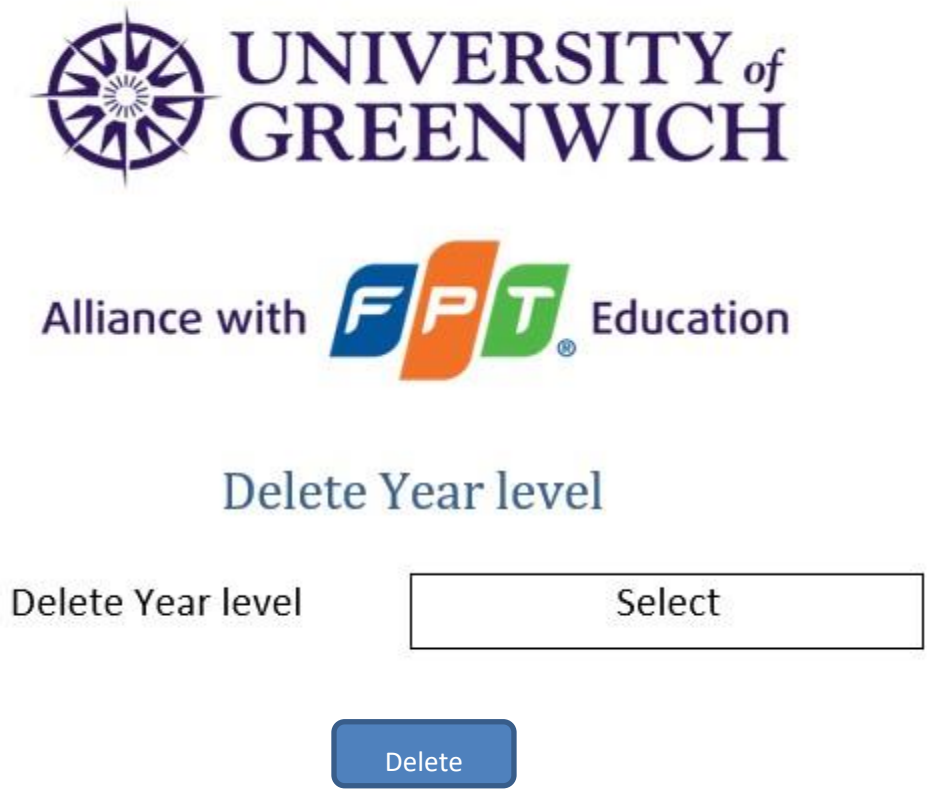

Figure 20

After filling in the information and clicking Delete and run the query:

```
delete Yearlevel where Yearlevel_ID = 'Yearlevel'
```

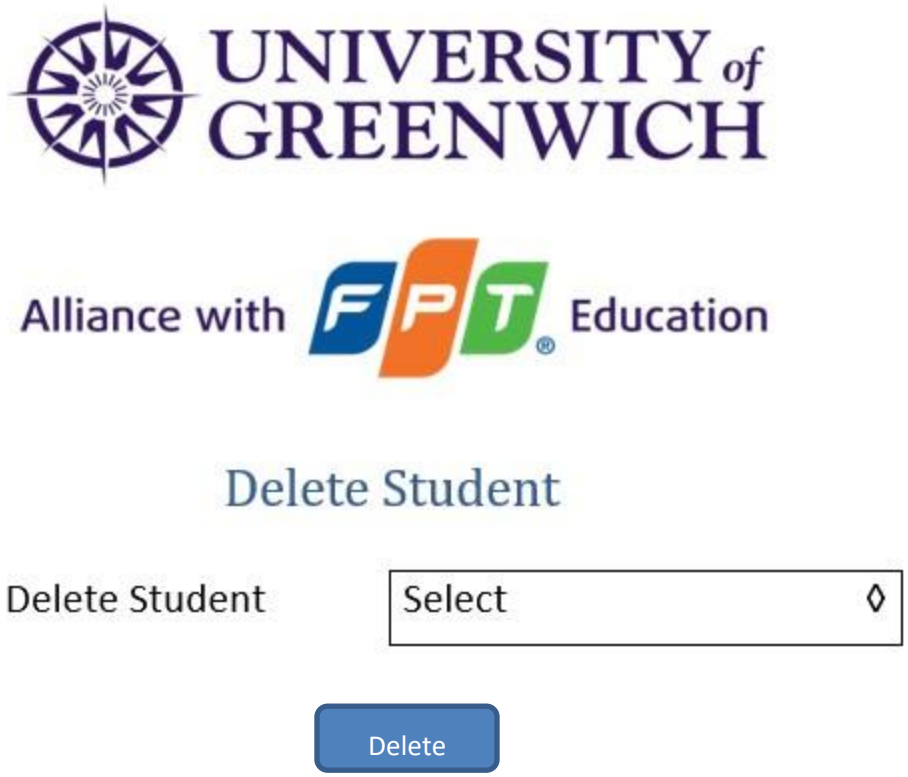

Figure 21

After filling in the information and clicking Delete and run the query:

```
delete Student where Student_ID  = 'StudentID'
```

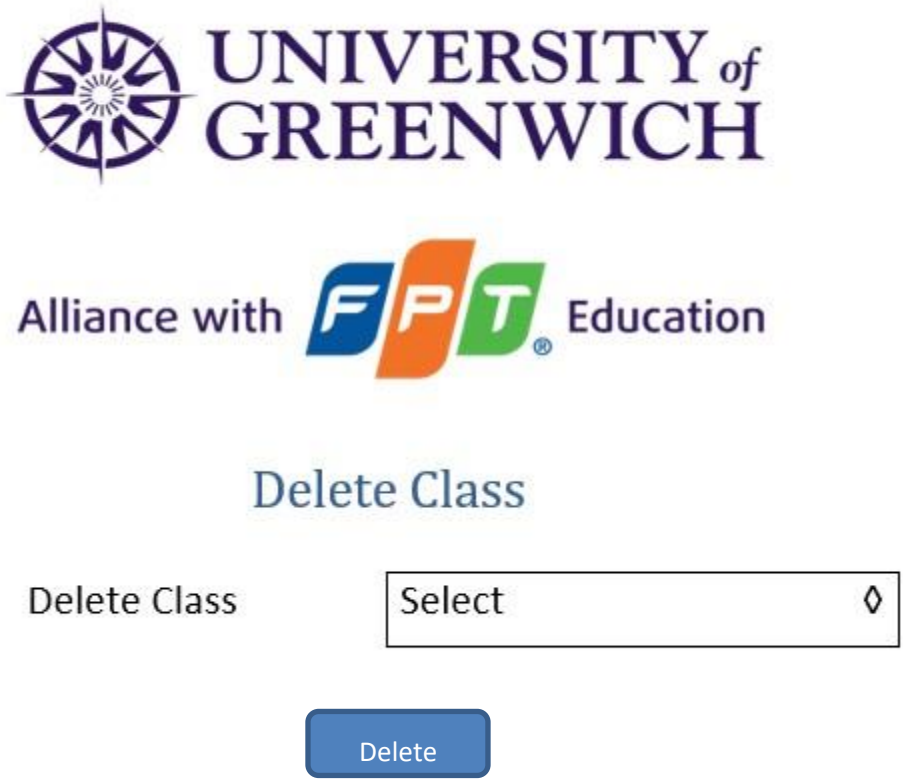

Figure 22

After filling in the information and clicking Delete and run the query:

```
delete Class where Class_ID = 'ClassID'
```

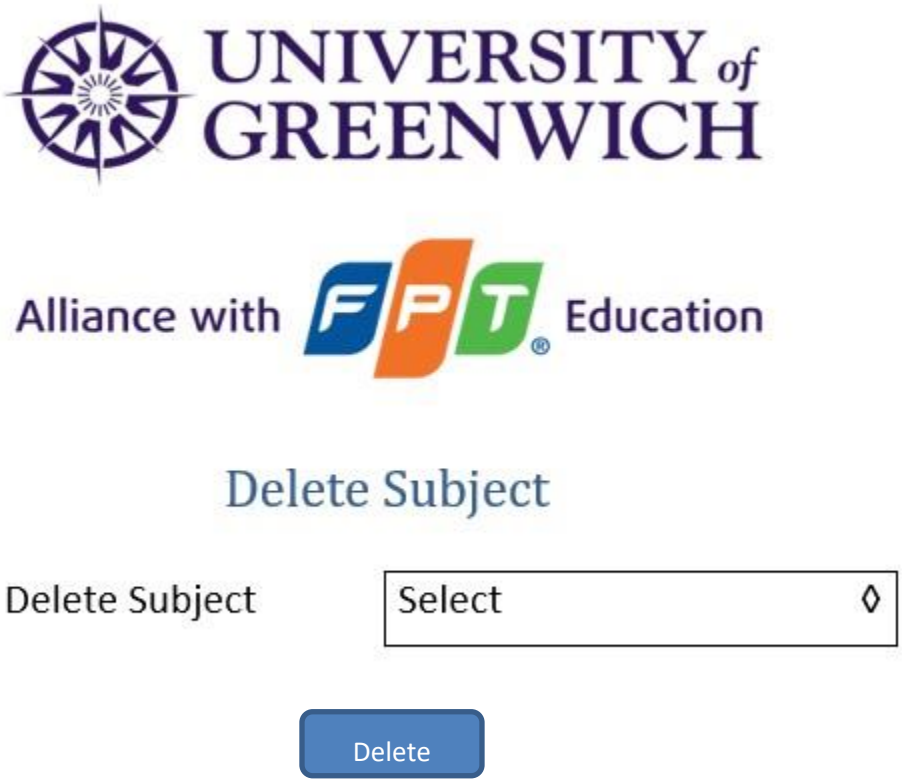

Figure 23

After filling in the information and clicking Delete and run the query:

```
delete Subject where Subject_ID = 'SubjectID'
```

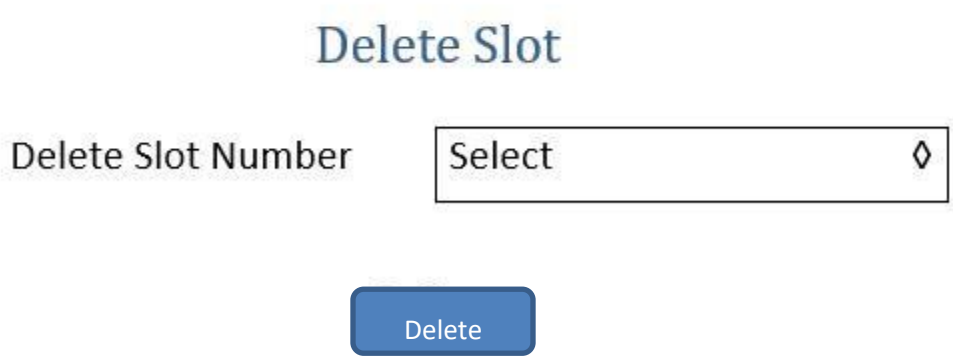

Figure 24

After filling in the information and clicking Delete and run the query:

```
delete Slot where Slot_No = 'Slot_No'
```

View

View Year level Details	View Attendance Details	View Subject Details
View Student Details	View Class Details	View Slot Details

Figure 25

III. Code SQL.

- Create tables.

• Create Database

```
create database attendanceSystem
use attendanceSystem go
```

• Create Yearlevel tables.

```
create table Yearlevel
(
Yearlevel_ID int primary key not null,
Yearlevel varchar(50) not null,
)
```

• Create Student tables.

```
create table Student
(
Student_ID varchar(8) primary key not null,
LastName varchar(50) not null,
MiddleName varchar(50) not null,
FirstName varchar(50) not null,
Yearlevel_ID int foreign key references Yearlevel(Yearlevel_ID)
)
```

- Create Attendance tables. create table Attendance
  (
  Attendance_ID int primary key,
  Student_ID varchar(8) foreign key references Student(Student_ID),
  Statuss varchar(8) not null,
  Class_ID varchar(7) foreign key references Class(Class_ID),
  Slot_No varchar(8) foreign key references Slot(Slot_No),
)
- Create Class tables.
  (
  Class_ID varchar(7) primary key, descriptions
  varchar(50) not null,
  DFrom date not null, DTo
  int not null, semester int
  not null,
  SYear int not null
)
- Create Subject tables.
  (
  Subject_ID int primary key not null,
  SubjectInitial varchar(30) not null,
  SubjectName varchar(30) not null,
  Attendance int foreign key references Attendance(Attendance_ID),
)
- Create Slot tables. create table Slot
  (
  Slot_No varchar(8) primary key not null,
  Dfrom int not null,
  Dto int not null,
)

- Insert sample data.

- Insert Yearlevel insert into Yearlevel values(1,'Fresher') insert into Yearlevel values(2,'beginer') insert into Yearlevel values(3,'INTERMEDIATE') insert into Yearlevel values(4,'ADVANCED')
- Insert Student
  insert into Student values('GCH17230','VINH','THE','DO',1) insert into Student values('GCH17231','DUONG','THUY','PHAM',2) insert into Student values('GCH17232','HIEN','VAN','LE',3) insert into Student values('GCH17233','NGUYEN','ANH','TRAN',1) insert into Student values('GCH17234','GIANG','TRUONG','DUC',1)
- Insert Attendance insert into Attendance values(1,'GCH17230','Absent','GCH0712','SLOT5') insert into Attendance values(2,'GCH17231','Present','GCH0713','SLOT5') insert

into Attendance values(3,'GCH17232','Present','GCH0714','SLOT5')
insert into Attendance
values(4,'GCH17233','Absent','GCH0715','SLOT5')

- Insert Class insert INTO Class
  VALUES('GCH0712','Top1dsadsad','12:30',1,2,3) insert INTO Class
  VALUES('GCH0713','Top1dsadsaắ','12:32',2,4,5) insert INTO Class
  VALUES('GCH0714','Top1dsadsaắ','12:32',2,4,5) insert INTO Class
  VALUES('GCH0715','Top1dsadsaắ','12:32',2,4,5)
- Insert Subject insert into Subject
  values(1,'entertainment','Professional Practice (1620)',1) insert into
  Subject values(2,'education','Professional Practice (1620)',1) insert
  into Subject values(3,'gender','Professional Practice (1620)',1)
- Insert Slot insert into Slot values('slot5',2,4) insert into Slot
  values('slot6',2,4) insert into Slot values('slot7',2,3) - **Output interface and query.**

  - Output interface and query.

  Student's Interface:

  After click View Attendance Details Button Students will receive their attendance information output according to the StudentID of the logged-in account with the student whose StudentID is GCH17230

Attendance Details

	Attendance_ID	Student_ID	Statuss	Class_ID	Slot_No
1	1	GCH17230	Absent	GCH0712	SLOT5
2	2	GCH17231	Present	GCH0713	SLOT5
3	3	GCH17232	Present	GCH0714	SLOT5
4	4	GCH17233	Absent	GCH0715	SLOT5

Figure 26

Query:

```
select Attendance_ID,Student_ID,Statuss,Class_ID,Slot_No from Attendance
```

View Year level Details

	Yearlevel_ID	Yearlevel
1	1	Fresher
2	2	beginer
3	3	INTERMEDIATE
4	4	ADVANCED

Figure 27

Query: select Yearlevel_ID,Yearlevel From Yearlevel

View Student Details

	Student_ID	LastName	FirstName	MiddleName	Yearlevel_ID
1	GCH17230	VINH	DO	THE	1
2	GCH17231	DUONG	PHAM	THUY	2
3	GCH17232	HIEN	LE	VAN	3
4	GCH17233	NGUYEN	TRAN	ANH	1
5	GCH17234	GIANG	DUC	TRUONG	1

Figure 28

Query: select Student_ID,LastName,FirstName,MiddleName,Yearlevel_ID from Student

View Class Details

	Class_ID	descriptions	DFrom	DTo	semester	SYear
1	GCH0712	Top1dsadsad	1900-01-01	1	2	3
2	GCH0713	Top1dsadsa?	1900-01-01	2	4	5
3	GCH0714	Top1dsadsa?	1900-01-01	2	4	5
4	GCH0715	Top1dsadsa?	1900-01-01	2	4	5

Figure 29

Query: select Class_ID,descriptions,DFrom,DTo,semester,SYear
from Class

View Slot Details

	Slot_No	Dfrom	Dto
1	slot5	2	4
2	slot6	2	4
3	slot7	2	3

Figure 30

Figure 31

Query: select Slot_No,Dfrom,Dto
from Slot

View Subject Details

	Subject_ID	SubjectInitial	SubjectName	Attendance
1	1	entertainment	Professional Practice (1620)	1
2	2	education	Professional Practice (1620)	1
3	3	gender	Professional Practice (1620)	1

Figure 32

Query:

```
select Subject_ID,SubjectInitial,SubjectName,Attendance from Subject
```

IV. Test.

a. Test plan.

Table 1

Test	What is being tested	How	Test data used	Expected result
1	Year level insert, update, delete.	Enter typical values	Data set 1	Good data accepted; bad data rejected
2	Student insert, update, delete.	Enter typical values	Data set 1	Good data accepted; bad data rejected
3	Attendance insert, update,	Enter typical values	Data set 1	Good data accepted; bad
	delete.			data rejected

4	Class insert, update, delete.	Enter typical values	Data set 1	Good data accepted; bad data rejected
5	Subject insert, update, delete.	Enter typical values	Data set 1	Good data accepted; bad data rejected
6	Slot insert, update, delete.	Enter typical values	Data set 1	Good data accepted; bad data rejected

b. Test logs.

Table 2

Test case	Test Title	Test Summary	Test Steps	Test Data	Expected result	Post-condition	Actual Result	Status	Note
1	Test Year level insert	Insert normal	insert into Yearlevel values(‘Yearlevel_ID’, ‘Yearlevel’)	(1,'Fresher')	Data is added into table	Successful	New Student added into table	Passed	
		data validation	insert into Yearlevel values(‘Yearlevel_ID’, ‘Yearlevel’)	Yearlevel_ID =‘abc’	error, Yearlevel_ID must be integer	Error	Invalid column name 'abc'	Passed	
		Data	insert into Yearlevel values(‘Yearlevel_ID’, ‘Yearlevel’)	(Yearlevel_ID = -1)	Error Yearlevel ID must not be negative	Successful	New Yearlevel added into table	Failed	Set conditions for YearlevelID
	Test Yearlevel Update	Test Update	Update Yearlevel set ‘Yearlevel_ID=1’ where	('beginer' where Yearlevel_ID = 4)	Update Successful	Successful	Data changed as expected where Yearlevel_ID = 4	Passed	

			Yearlevel_ ID = 4						
	Test Yearlev el delete	Test delete	Delete Yearlevel where Yearlevel_ ID=1	Yearlevel_I D=1	Delete Success ful	Succe ssful	Yearlevel removed from table	Passed	
2	Test Student insert	Insert normal	insert into Student values(‘St udent_ID’, ‘LastName ’,‘MiddleN ame’,‘First Name’, ‘Yearlevel _ID’	('GCH1723 0','VINH','T HE','DO',1)	Data is added into table	Succe ssful	New Student added into table	Passed	
		data validati on	insert into Student values(‘St udent_ID’, ‘LastName ’,‘MiddleN ame’,‘First Name’,‘Ye arlevel_ID’	Student_ID = ‘abc’	error, Student _ID must be integer	Error	Invalid column name 'abc'	Passed	
		Data	insert into Student values(‘St udent_ID’, ‘LastName ’, ‘MiddleNa me’,‘First Name’,‘Ye arlevel_ID’	StudentID = -1	Error Student ID must not be negativ e	Succe ssful	New Student added into table	Failed	Set conditions for Student_ID
	Test Student Update	Test Update	update Student set LastName = 'TUAN', FirstName = 'ANH',	('TUAN', 'ANH', 'LƯU','GCH 17230')	Update Success ful	Succe ssful	Data changed as expected where StudentID = ‘GCH17230'	Passed	

			MiddleName = 'LU'U' where Student_ID = 'GCH17230';						
	Test Student Delete	Test Delete	delete from Student where Student_ID='GCH17232'	Student_ID = 'GCH17232'	Delete Successful	Successful	Student removed from table	Passed	
3	Test Attendance insert	Insert normal	Insert into Attendance values(Attendance_ID,Student_ID,Status,Class_ID, Slot_No)	(1,'GCH17230','Absent','GCH0712','SLOT5')	Data is added into table	Successful	New Statuss added into table	Passed	
	Test Attendance update	Test Update	update Attendance set Student_ID = 'GCH17231', Statuss = 'Absent' where Attendance_ID = 2	('GCH17231','Absent', 2)	Update Successful	Successful	Data changed as expected where Attendance ID = 2	Passed	
4	Test Class insert	Insert normal	Insert into Class values(Class_ID, descriptions, DFrom, DTo,	('GCH0712','Top1dsadsad','12:30',1,2,3)	Data is added into table	Successful	New Class added into table	Passed	

			semester, SYear						
	Test class Update	Test Update	update Class set descriptions = 'top1234', DFrom = '1900-0101' where Class_ID = 'GCH0712'	('top1234', '1900-01-01', 'GCH0712')	Update Successful	Successful	Data changed as expected where ClassID = GCH0712	Passed	
	Test Class Delete	Test Delete	delete from Class where Class_ID = 'GCH0712''	Class_ID = 'GCH0712''	Delete Successful	Successful	Class removed from table	Passed	
5	Test Subject insert	Insert normal	insert into Subject values('Subject_ID', 'SubjectInitial', 'SubjectName', 'Attendance')	(1,'entertainment','Professional Practice (1620)',1)	Data is added into table	Successful	New Subject added into table	Passed	
		data validation	insert into Subject values('Subject_ID', 'SubjectInitial', 'SubjectName', 'Attendance')	Subject_ID = 'abc'	error, Subject_ID must be integer	Error	Invalid column name 'abc'	Passed	
	Test	Test	update	('gender',	Update	Succe	Data changed as	Passed	

	Subject Update	Update	Subject set SubjectInitial = 'gender', SubjectName = 'database developer (1622)' where Subject_ID = 2	'database developer(1622)', 2)	Successful	ssful	expected where Subject_ID = 2		
	Test Subject Delete	Test Delete	delete from Subject where Subject_ID = 'SubjectID'	Subject_ID = '1620'	Delete Successful	Successful	Subject removed from table	Passed	
6	Test Slot insert	Insert normal	insert into Slot values(' Slot_No', 'Dfrom', 'Dto')	('slot6',2,4)	Data is added into table	Successful	New Slot added into table	Passed	
	Test Slot update	Test Update	update Slot set Dfrom = 5, Dto = 6 where Slot_No = 'slot6'	(5,6 'slot6')	Update Successful	Successful	Data changed as expected where Slot_No = 'slot6'	Passed	
	Test Slot delete	Test delete	delete from Slot where Slot_No = 'Slot5'	'Slot5'	Delete Successful	Successful	Slot removed from table	Passed	

V. Technical document.

- Introduction.

The report will provide an overview of the system and its intended use. Reports include ERD, Use case diagram, logical design ..., followed by system analysis and

evaluation. Clearly analyze the attendance system to help users better understand it, highlight strengths and weaknesses and things to improve

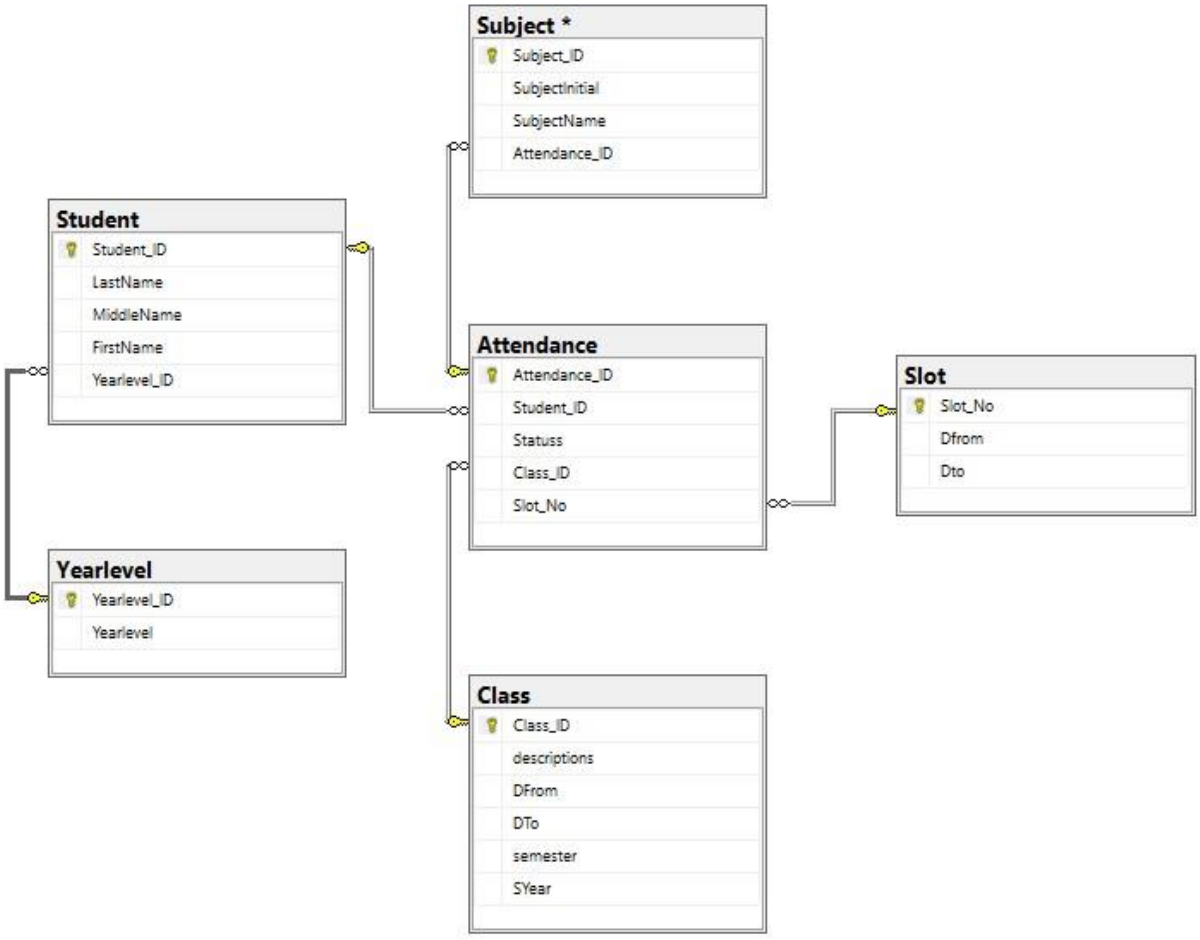

Figure 33

Evaluating user criteria, the database is divided into 6 tables, each complying with standardized rules to minimize redundancy. First for students can only view their attendance information, for teachers can attend, update attendance, watch attendance, view slots, view subject. - **Physical design:**

Table 3

Table Name	Attribute	Content	Data Type	PK	Referenced
	Name			or FK	
Yearlevel	Yearlevel_ID Yearlevel	Year level ID Year level	Int Varchar(50)	PK	

Student	Student_ID LastName MiddleName FristName Yearlevel_ID	Student_ID LastName MiddleName FristName Yearlevel_ID	Varchar(8) Varchar(50) Varchar(50) Varchar(50) int	PK FK	 Yearlevel table
Class	Class_ID Descriptions Dfrom Dto Semester SYear	Class_ID Descriptions Dfrom Dto Semester SYear	Varchar(7) Varchar(50) Date Int Int Int	PK	
Slot	Slot_No Dfrom Dto	Slot_No Dfrom Dto	Varchar(8) Int Int	PK	
Attendance	Attendance_ID Student_ID Statuss Class_ID Slot_No	Attendance_ID Student_ID Statuss Class_ID Slot_No	Int Varchar(8) Varchar(8) Varchar(7) Varchar(8)	PK FK FK FK	 Student table Class table Slot table
Subject	Subject_ID SubjectInitial SubjectName Attendance	Subject_ID SubjectInitial SubjectName Attendance	Int Varchar(30) Varchar(30) Int	PK FK	 Attendance table

All physical designs follow 3 Normalization Forms because the value domains of the columns in the table contain only the value of the element values, the nonmain attribute of the relationship depends on the entire key tablets and no dependency function bridging

- Use-case diagram:

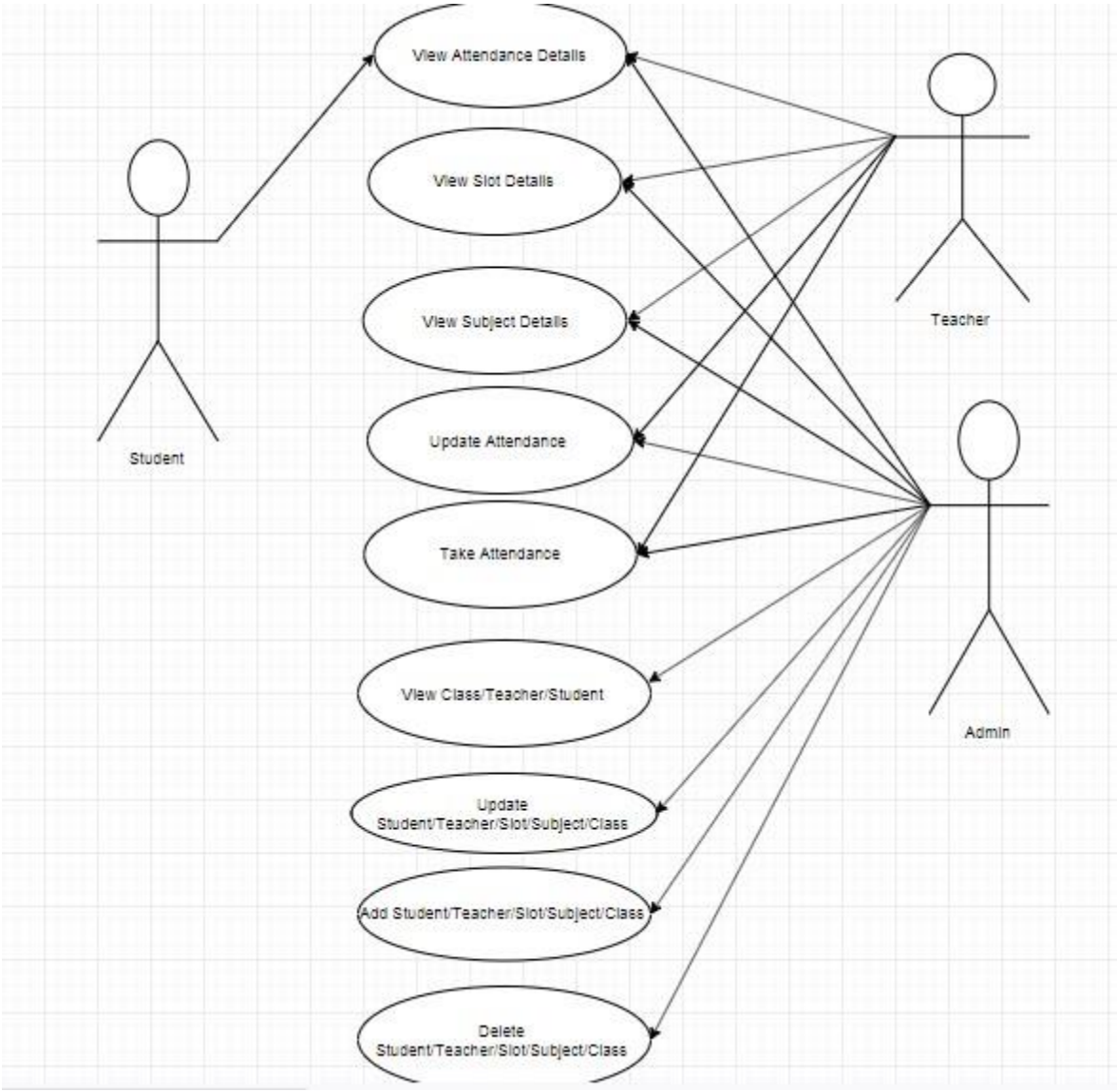

Figure 34

VI. Evaluate.

In this report, it can be seen that the database I designed is extremely simple and optimal and limits data redundancy and minimizes unnecessary data and helps the system work better. And the strength of this database is not using much memory. The database is simple and easy to use

Figure 35

and this is the interface of the system when logging in with an account including the username and password provided by the school. After logging in each account will have different interfaces - Student:

Students have only one function to view their attendance information. This is the student's login interface and to view each participant's information, just click the view attendance details button

Student Login

View Attendance Details

Figure 36

Figure 37

This is the teacher's login interface
Function: view attendance information, subject information, slot information, teachers can attend and edit attendance. In this interface teachers can:

- Click the view attendance button details to see attendance information
- Click the view session details button to view session information
- Click the view subject details button to view the subject information
- Click the take attendance button to take attendance
- Click the update attendance button to edit the attendance records

If the teacher selects the feature to see the system will return the output and if you choose to attend and update the system will display another interface to help you edit the data.

Take Attendance

AttendanceID	
StudentID	Select ◊
ClassID	Select ◊
SlotNo	
Status	◯ Absent
	◯ Present

Add

Figure 38

Figure 39

This is the interface when teachers choose to attend or update attendance. To attend, you need to add information to the schools then click on more. To update user information, fill in the fields and then click the update button to complete the data

Figure 40

This is the interface after Admin login:

If you want to add data, clicking the add button the system will display the additional interface. After completing the data provided by the system, then press the add button to complete the data.

Add

Add Yearlevel	Add Class	Add Slot
Add Student	Subject	

Figure 41

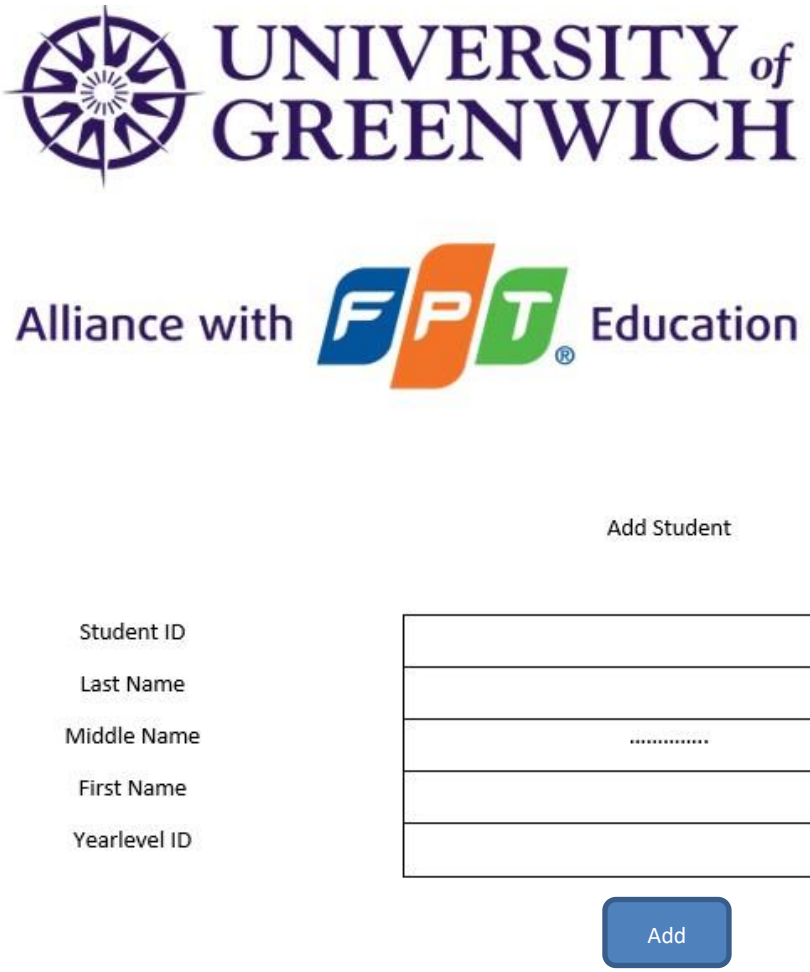

Figure 42

This is the extra student interface, just add data to the fields and click the add button to insert data. Note to edit the data, please enter the ID of the data to be edited

Update

Update Year level	Update Attendance	Update Subject
Update Student	Update Class	Update Slot

Figure 43

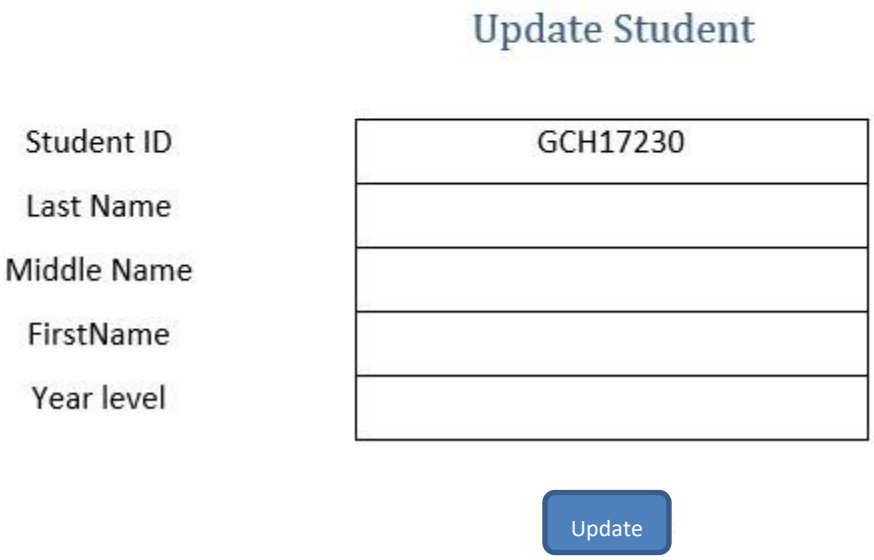

Figure 44

finally delete function, will need to click on the delete button to show the system delete interface and then select the data to delete. Examples are students or subjects

Delete

Delete Year level	Delete Subject	Delete Slot
Delete Student	Delete Class	

Figure 45

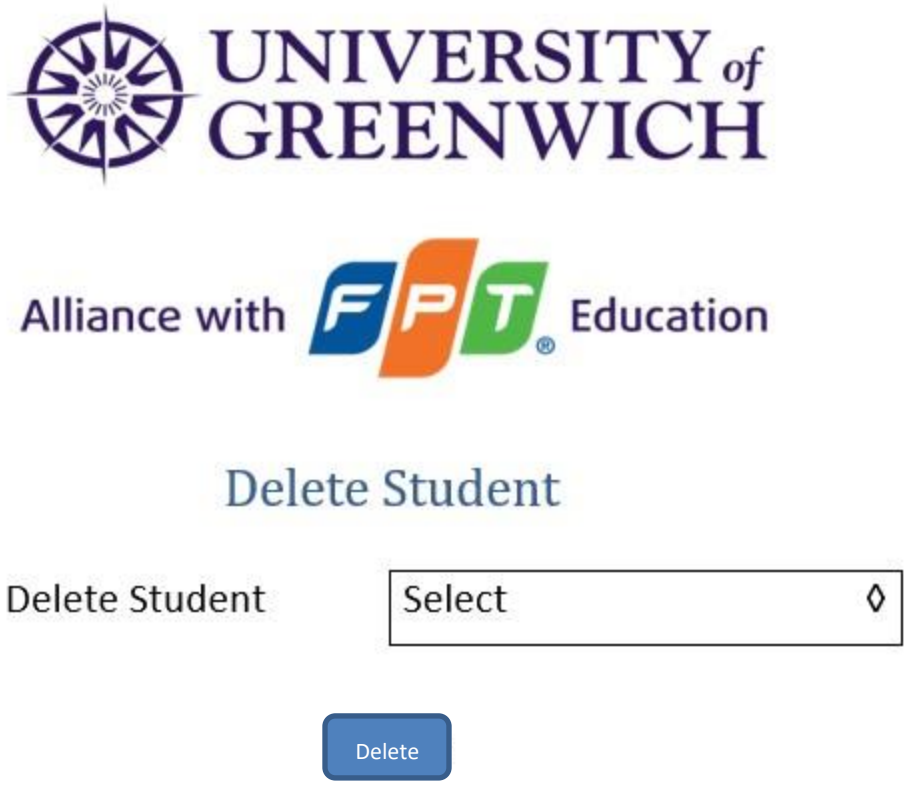

Figure 46

This is a successful notification after adding a correction to the system's data.

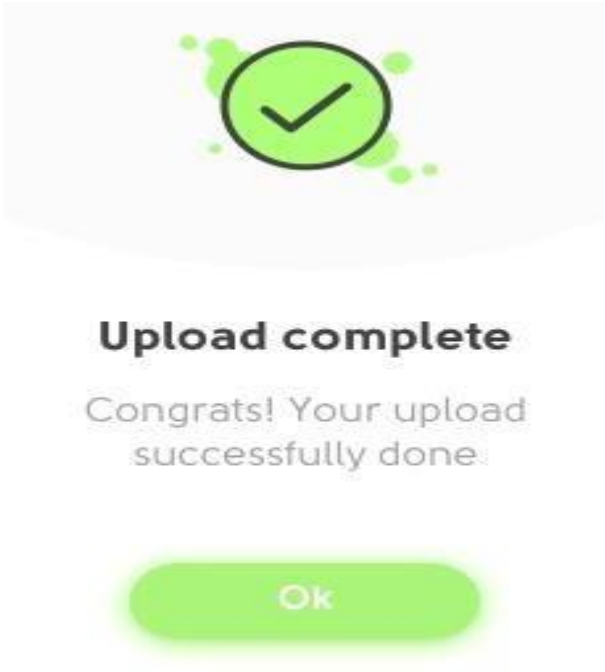

Figure 47

This is an error message that occurs from the system when the user inserts invalid data

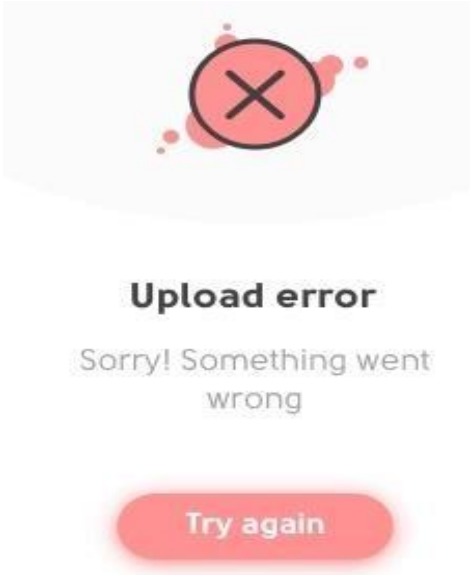

Figure 48

Note: users should check additional data for deletion.