****

**Ho Chi Minh, 01/7/2020**

**Lecturer: Huynh Xuan Phung**

**Members: Phan Vinh Loc 17110049**

**Nguyen Tan Loc 17110048**

**Truong Viet Hoang 17110033**

**TOPIC: STUDENT’S PROJECT MANAGEMENT**

**HO CHI MINH UNIVERSITY OF TECHNOLOGY AND EDUCATION**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC KINH BẮC**

**SUBJECT: WINDOW PROGRAMMING**

**FINAL PROJECT**

**Score**

|  |  |  |  |
| --- | --- | --- | --- |
| Criteria | Content | Presentation | Overall |
| Score |  |  |  |

Observations of Instructor:

Evaluation:

Nguyen Tan Loc:   
Truong Viet Hoang:   
Phan Vinh Loc:

Teacher Guide

**Phung**

**Huynh Xuan Phung**

**Contents:**

[**List of figures:** ***Error! Bookmark not defined.***](#_Toc44607318)

[**List of tables: 5**](#_Toc44607319)

[**Chapter 1: Program Overview 7**](#_Toc44607320)

[1.1 About the Project 7](#_Toc44607321)

[**1.1.1 Purpose 7**](#_Toc44607322)

[**1.1.2 Requirements 7**](#_Toc44607323)

[**1.1.3 Project Analysis 7**](#_Toc44607324)

[**1.1.4 Expected Interface 7**](#_Toc44607325)

[**1.1.5 Technical Requirements 7**](#_Toc44607326)

[**1.1.6 Technology and Tool 8**](#_Toc44607327)

[**Chapter 2: Work Assignment 8**](#_Toc44607328)

[2.1. Work Plan 8](#_Toc44607329)

[2.2 Work Assignment 8](#_Toc44607330)

[**Chapter 3: Design Software 9**](#_Toc44607331)

[3.1 Design Interface 9](#_Toc44607332)

[**3.1.1 Program Interface 9**](#_Toc44607333)

[Error! Bookmark not defined.](#_Toc44607334)

[**3.1.2 Interface Specification 12**](#_Toc44607335)

[3.2 Design Class 14](#_Toc44607336)

[**3.2.1 Sample list of classes used in the program 14**](#_Toc44607337)

[**3.2.2 The sample table describes the methods in a class 15**](#_Toc44607338)

[3.2 Design Database 16](#_Toc44607339)

[**3.3.1 Database Diagram 17**](#_Toc44607340)

[**3.3.2 Database Specification and Field Decsription 17**](#_Toc44607341)

[**Chapter 4: Install and Test 20**](#_Toc44607426)

[***Error! Bookmark not defined.***](#_Toc44607427)

[**Chapter 5: Conclusion and Development Direction 39**](#_Toc44607428)

[5.1 Conclusion 39](#_Toc44607429)

[5.2 Development Direction 40](#_Toc44607430)

[**Reference 40**](#_Toc44607431)

List of figures:

[Figure 1: Login Interface 9](#_Toc44630165)

[Figure 2: Interface of user when log in successful (Home Page) 9](#_Toc44630166)

[Figure 3: Form Student’s Information 10](#_Toc44630167)

[Figure 4: Form Lecture’s Information 10](#_Toc44630168)

[Figure 5: Form Project 11](#_Toc44630169)

[Figure 6: Form Project Detail 11](#_Toc44630170)

[Figure 7: Form Progress of Student 12](#_Toc44630171)

[Figure 8: Test 1 20](#_Toc44630172)

[Figure 9: Test 2 21](#_Toc44630173)

[Figure 10: Test 3 22](#_Toc44630174)

[Figure 11: Test 4 23](#_Toc44630175)

[Figure 12: Test 5 24](#_Toc44630176)

[Figure 13: Test 6 24](#_Toc44630177)

[Figure 14: Test 7 25](#_Toc44630178)

[Figure 15: Test 8 26](#_Toc44630179)

[Figure 16: Test 9 26](#_Toc44630180)

[Figure 17: Test 10 27](#_Toc44630181)

[Figure 18: Test 11 27](#_Toc44630182)

[Figure 19: Test 12 28](#_Toc44630183)

[Figure 20: Test 13 29](#_Toc44630184)

[Figure 21: Test 14 29](#_Toc44630185)

[Figure 22: Test 15 30](#_Toc44630186)

[Figure 23: Test 16 30](#_Toc44630187)

[Figure 24: Test 17 31](#_Toc44630188)

[Figure 25: Test 18 32](#_Toc44630189)

[Figure 26: Test 19 33](#_Toc44630190)

[Figure 27: Test 20 34](#_Toc44630191)

[Figure 28: Test 21 35](#_Toc44630192)

[Figure 29: Test 22 36](#_Toc44630193)

[Figure 30: Test 23 37](#_Toc44630194)

[Figure 31: Test 24 38](#_Toc44630195)

[Figure 32: Test 25 39](#_Toc44630196)

# List of tables:

[Table 1: Interface Specification 12](#_Toc44630400)

[Table 2: Design Class 14](#_Toc44630401)

[Table 3: The sample table describes the method in a class 15](#_Toc44630402)

[Table 4: UserLogin 17](#_Toc44630403)

[Table 5: Student 18](#_Toc44630404)

[Table 6: Lecture 18](#_Toc44630405)

[Table 7: Project 19](#_Toc44630406)

[Table 8: Progress 20](#_Toc44630407)

[Table 9: ProjectManagemnet 20](#_Toc44630408)

# 

# 

# Chapter 1: Program Overview

## About the Project

* Design and build object-oriented software to create t software which can manage the student’s project

### Purpose

* Help instructor can manage the student’s project
* We can know the owner of the project
* We can know the instructor of the project
* We can know the detail of the project

### Requirements

* Manage student’s information
* Instructors (Lecture)
* Project’s information and report

### Project Analysis

* Building object – oriented software
* The output of data is the application

### Expected Interface

This project includes the following forms:

* Student
* Lecture
* Project

### Technical Requirements

- Fulfilling the requirements set out by the project

- Application of object-oriented programming

- Software capacity is light, stable running

### 1.1.6 Technology and Tool

- Design Graphical User Interface (GUI) by Visual Studio 2019 (Tool)

- Design View Model and Data Model by Visual Studio 2019 (Code C#)

- Building software on EntityFramework.6.2.0

- SQL Server

# Chapter 2: Work Assignment

## 2.1. Work Plan

|  |  |  |
| --- | --- | --- |
| **Week** | **Date** | **Work** |
| Week 10 | 02/06 | Design Data Base But Wrong |
| Week 11 | 09/06 | Design Data Base |
| Week 12 | 16/06 | Design Class + GUI |
| Week 13 | 23/06 | Design View Model and Data Model |
| Week 14 | 30/06 | Finish |

## 2.2 Work Assignment

|  |  |  |  |
| --- | --- | --- | --- |
| **Number Order** | **Name of student** | **Description** | **Contribute (%)** |
| 1 | Nguyen Tan Loc | Design Class | 40% |
| 2 | Truong Viet Hoang | Design GUI, Function Login | 30% |
| 3 | Phan Vinh Loc | Design Data Base Install and Test Write Report | 30% |

# Chapter 3: Design Software

## 3.1 Design Interface

### Program Interface

Student name: Truong Viet Hoang

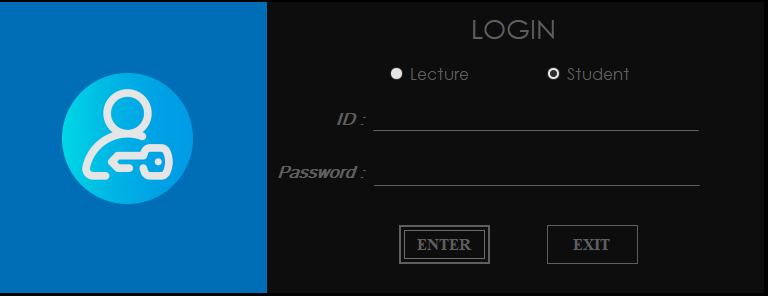


Figure 1: Login Interface



Figure 2: Interface of user when log in successful (Home Page)

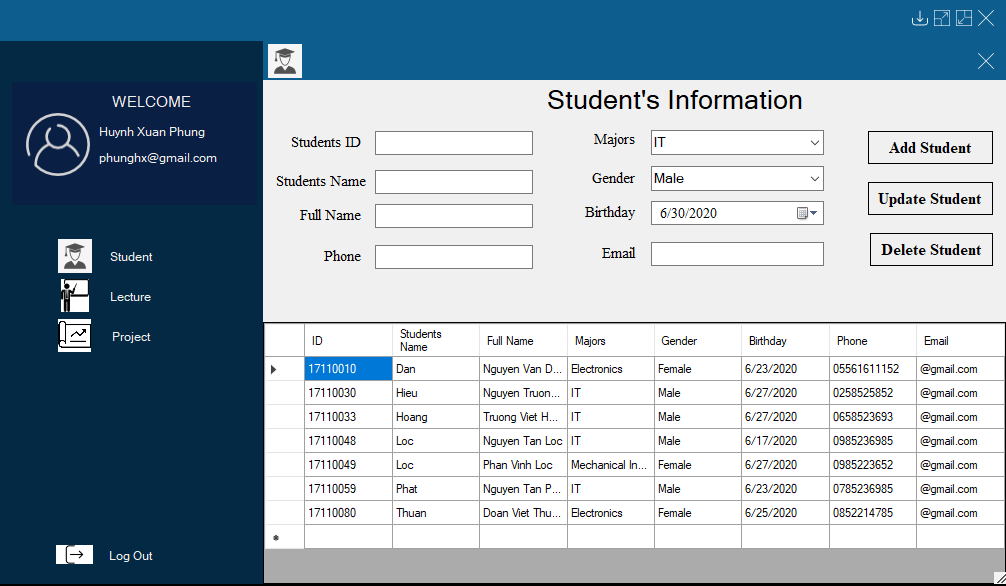


Figure 3: Form Student’s Information

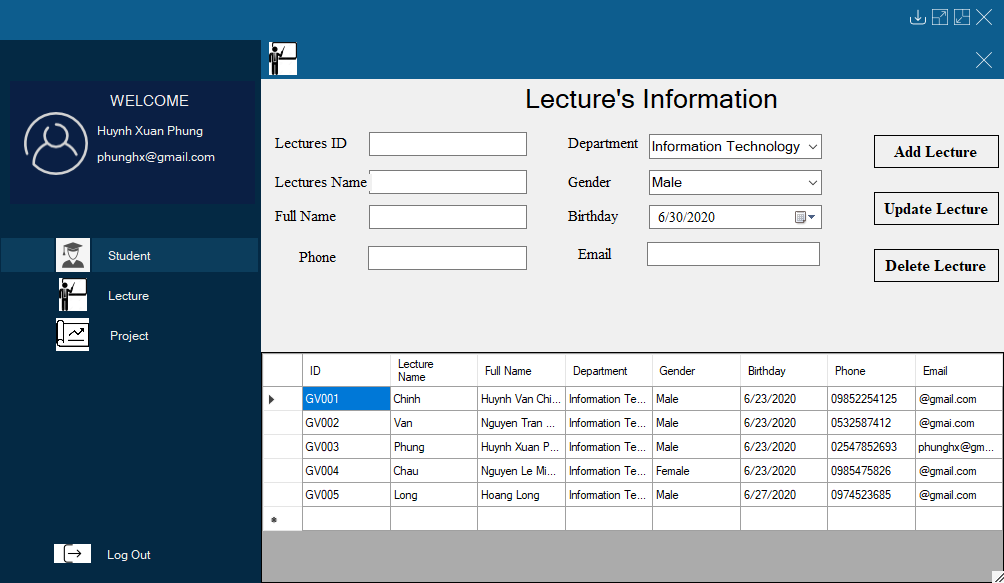


Figure 4: Form Lecture’s Information

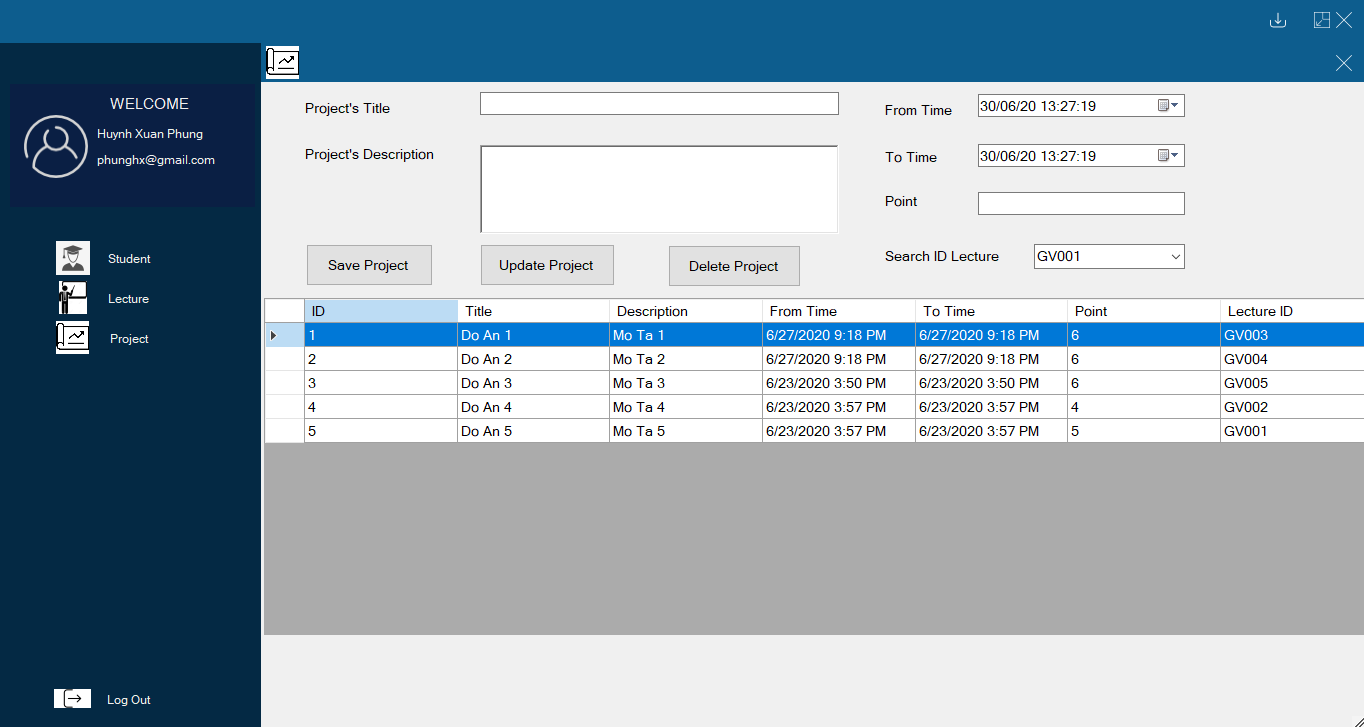


Figure 5: Form Project

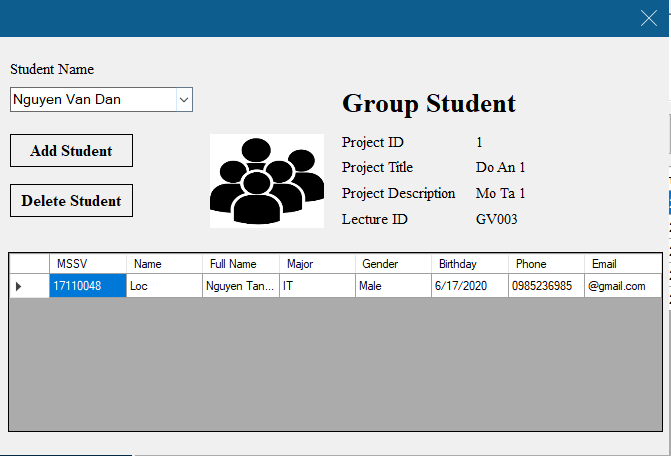


Figure 6: Form Project Detail

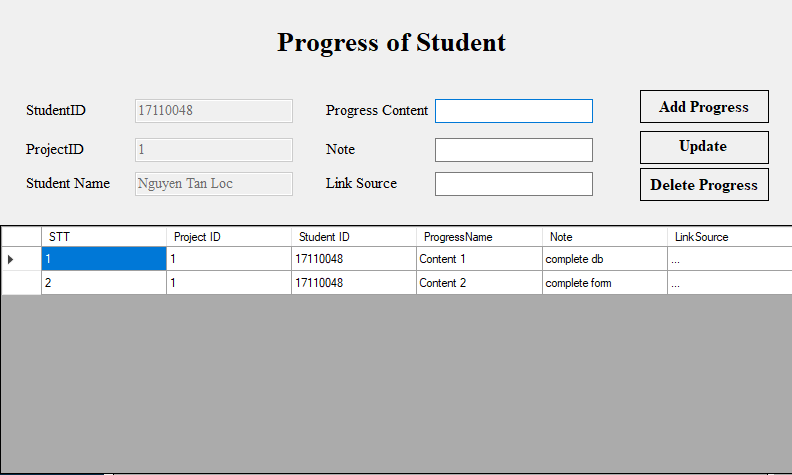


Figure 7: Form Progress of Student

About Interface of Student such as Lecture but can’t Add, Update And Delete, etc. The student just can view

### Interface Specification

Table 1: Interface Specification

|  |  |  |
| --- | --- | --- |
| **Form** | **Type of button** | **Purpose** |
| Login Form | Radio Button (Select Role) | Allow you select type of user to login |
| Button (Login) | Check your ID and password after that let you log in |
| Button (Exit) | You can exit the program when you don’t use |
| Home page Form | Button (Student) | Let you enter in student form and edit on it |
| Button (Lecture) | Let you enter in lecture form and edit on it |
| Button (Project) | Let you enter in project form and edit on it |
| Button (Log out) | Allow log your account out and you can log in with another account |
| Student’s Information Form | Button (Add Student) | Allow you add information of new student |
| Button (Update Student) | Allow you edit information of the student who is added |
| Button (Delete Student) | Allow you delete information of the student |
| Lecture’s Information Form | Button (Add lecture) | Allow you add information of new lecture |
| Button (Update lecture) | Allow you edit information of the lecture who is added |
| Button (Delete lecture) | Allow you delete information of the lecture |
| Project Form | Button (Add project) | Allow you add information of new project |
| Button (Update project) | Allow you edit information of the project which is added |
| Button (Delete project) | Allow you delete information of the project |
| Project Detail Form | Combo box (List of students who are had information in Student Form) | Allow you select the name of student |
| Button (Add Student) | Add the student you choose in the combo box into the project which you chosen before |
| Button (Delete Student) | Delete the student you choose in the data grid view of the project which you chosen before |
| Progress of Student Form | Button (Add progress) | Allow you access into the project’s progress and add progress for student |
| Button (Update progress) | Allow you access into the project’s progress and update progress for student |
| Button (Delete progress) | Allow you access into the project’s progress and delete progress for student |

## 3.2 Design Class

### 3.2.1 Sample list of classes used in the program

Student name: Nguyen Tan Loc

Table 2: Design Class

|  |  |  |
| --- | --- | --- |
| **TT** | **Class name**  **Inheritance or lead**  **export from (specify name**  **base class) – if have** | **Purpose** |
| 1 | * Class UserLogin | Before using the program, the user must create a login account |
| 2 | * Class Lecture * Foreign Key IDLoginL references UserLogin(ID) | Store information of lecturer and decentralization for teachers when creating accounts (Lecturers have the right to use the add, edit, delete functions) |
| 3 | * Class Student * Foreign Key IDLogin references UserLogin(ID) | Store information of students and decentralization for students when creating accounts (Students only have permission to view) |
| 4 | * Class Project * Foreign Key L\_ID references Lecture(L\_ID) | Store information of project and save the name of that project lecturer |
| 5 | * Class ProjectManagement * Foreign Key P\_ID references Project(P\_ID) * Foreign Key S\_ID references Student(S\_ID) * Set P\_ID and S\_ID is primary key that get all the information of Project and Student | From this table we can save the project information, know the name of the lecturer and the teacher can create groups for students. |
| 6 | * Class Progress * Foreign Key P\_ID references ProjectManagement (P\_ID) * Foreign Key S\_ID references ProjectManagement (S\_ID) * Set P\_ID and S\_ID is primary key that get the ID of Project and Student | Every student needs to record the process of completing the project. Get the ID of Project and Student to know what projects students do, lecturer, and what each student can do |

### 3.2.2 The sample table describes the methods in a class

Student name: Nguyen Tan Loc

Table 3: The sample table describes the method in a class

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Method** | **Purpose** | **File name, line number containing the declaration** |
| 1 | public static Student getStudentInfomationAfterLogin(string idLogin) Input: idLogin Output: If class Students have IDLogin == idLogin => return Student | Check if the student class has an account that matches the ID of the UserLogin class.  If identical then return student  Else error note | DBProjectStudent/Controller/StudentController.cs (159) |
| 2 | public static List<Progress> listProgressOfStudent(int P\_ID,string S\_ID) Input: P\_ID, S\_ID Output: list Progress of Student | Find the ID student done ID project is to record the project completion process | DBProjectStudent/Controller/StudentController.cs (150) |
| 3 | public static List<Student> getStudentDetail() Input: nothing Output: list Student | Get all full name student that lecturer create group | DBProjectStudent/Controller/StudentController.cs (134) |
| 4 | public static List<Student> getAllStudentbyIDproject(int idproject) Input: idproject Output: list student | Show the group student working on that project | DBProjectStudent/Controller/ ProjectManagementController.cs (42) |
| 5 | public static Lecture getLectureInfomationAfterLogin(string idLogin) Input: idLogin Output: If class Lecturer have IDLogin == idLogin => return Lecturer | Check if the lecturer class has an account that matches the ID of the UserLogin class.  If identical then return lecturer  Else error note | DBProjectStudent/Controller/LectureController.cs (128) |

## Design Database

Student Name: Phan Vinh Loc

### 3.3.1 Database Diagram

A screenshot of a social media post

Description automatically generated

### 3.3.2 Database Specification and Field Description

Table 4: UserLogin

|  |  |  |
| --- | --- | --- |
| Column name | Data Type | Note |
| ID | Varchar(20) |  |
| Pass | Varchar(20) |  |
| Roleuser | Varchar(10) |  |

Table 5: Student

|  |  |  |
| --- | --- | --- |
| Column name | Data Type | Note |
| S\_ID | Varchar(20) | Student ID |
| S\_Name | Nvarchar(100) | Student Name |
| S\_Fullname | Varchar(50) | Student Fullname |
| S\_Major | Varchar(50) | Student Major |
| S\_Birthday | Varchar(50) | Student Birthday |
| S\_Phone | Varchar(50) | Student Phone |
| S\_Email | Varchar(50) | Student Email |
| S\_Gender | Varchar(50) | Student Gender |
| ID | int | ID Login |

Table 6: Lecture

|  |  |  |
| --- | --- | --- |
| Column name | Data Type | Note |
| L\_ID | Varchar(20) | Lecture ID |
| L\_Name | Nvarchar(100) | Lecture Name |
| L\_Fullname | Varchar(50) | Lecture Fullname |
| L\_Department | Varchar(50) | Lecture Department |
| L\_Birthday | Datetime | Lecture Birthday |
| L\_Phone | Varchar(50) | Lecture Phone |
| L\_Email | Varchar(50) | Lecture Email |
| L\_Gender | Varchar(50) | Lecture Gender |
| ID | int | ID Login |

Table 7: Project

|  |  |  |
| --- | --- | --- |
| Column name | Data Type | Note |
| P\_ID | int | Project ID |
| P\_Title | Nvarchar(100) | Project Tile |
| P\_Description | Nvarchar(200) | Project Description |
| P\_Fromtime | datetime | Project Fromtime |
| P\_Totime | datetime | Project Totime |
| P\_Point | Varchar(10) | Project Point |
| L\_ID | Varchar(20) | Lecture ID |

Table 8: Progress

|  |  |  |
| --- | --- | --- |
| Column name | Data Type | Note |
| STT | Nvarchar(50) | Number order |
| P\_ID | int |  |
| Progress Name | Nvarchar(100) |  |
| Note | Nvarchar(100) |  |
| Link Source | Nvarchar(100) |  |

Table 9: ProjectManagemnet

|  |  |  |
| --- | --- | --- |
| Column name | Data Type | Note |
| P\_ID | int |  |
| S\_ID | Varchar(20) |  |

# Chapter 4: Install and Test

Figure 8: Test 1

Figure 8: Test 1 shows the login interface. Account you can add and allow role user in the data base (SQL server). In this case I log in by account lecture which have ID: 123123 and pass:123123

Test form Student’s Information  
Add 1 Student who have ID: 17110000, Student’s Name: ABC, Full Name: ABCXYZ, Phone: 0102030405, Major: IT, Gender: Male, Birthday: 1/2/1999, Email: abcxyz@gmail.com  
Before Add

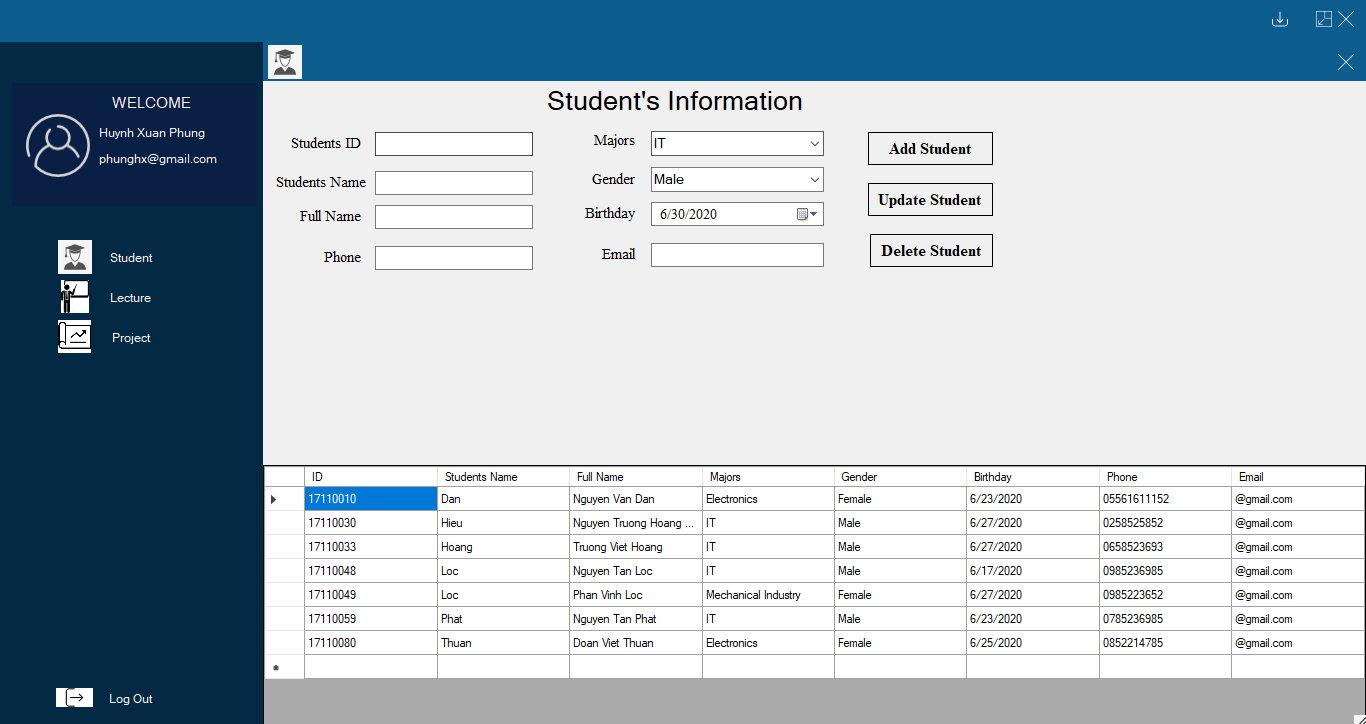


Figure 9: Test 2

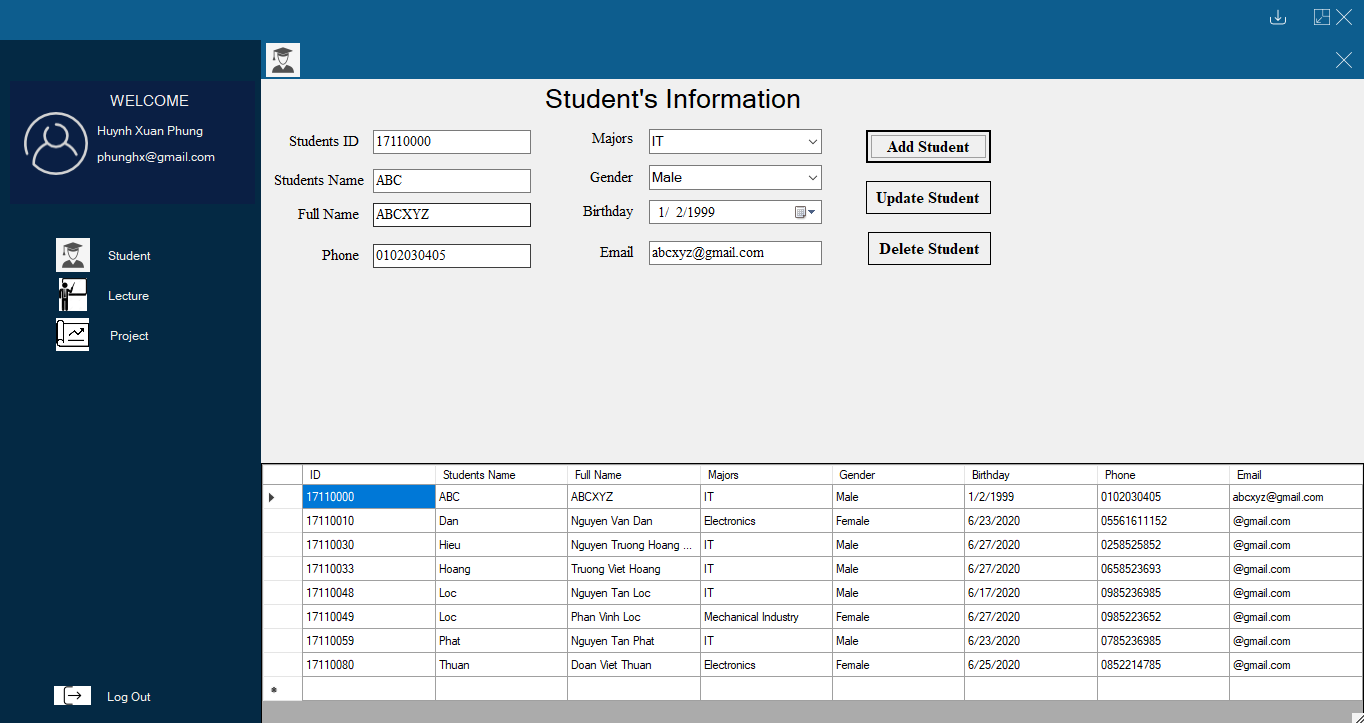
After Add  


Figure 10: Test 3

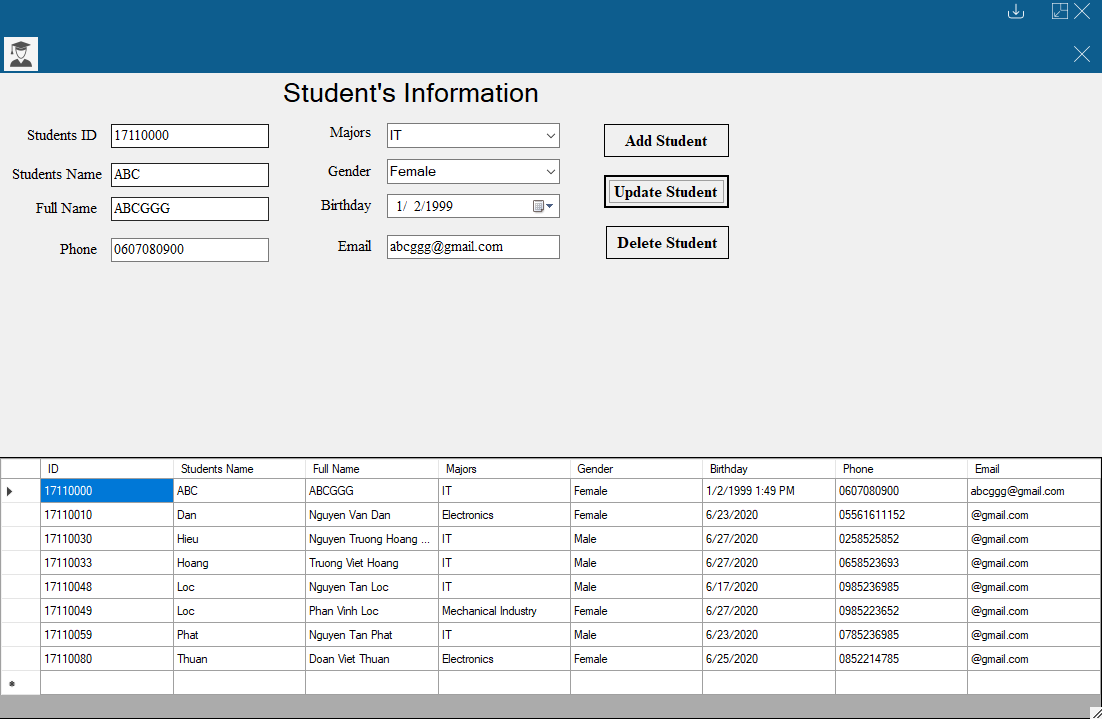
Update 1 Student who have ID: 17110000, Student’s Name: ABC, Full Name: ABCGGG, Phone: 0607080900, Major: IT, Gender: Female, Birthday: 1/2/1999, Email: abcggg@gmail.com 

Figure 11: Test 4

Delete 1 Student  
I will delete student who have ID:17110000  
Before Delete

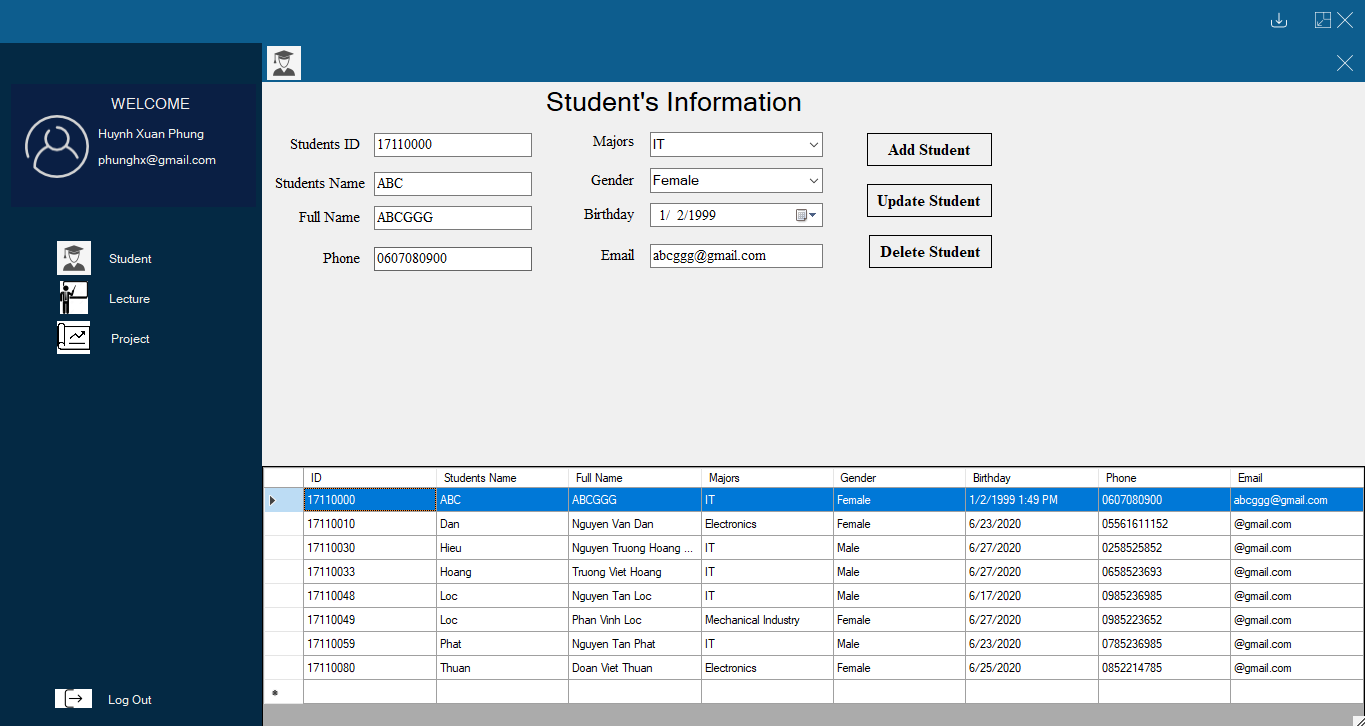


Figure 12: Test 5

After Delete

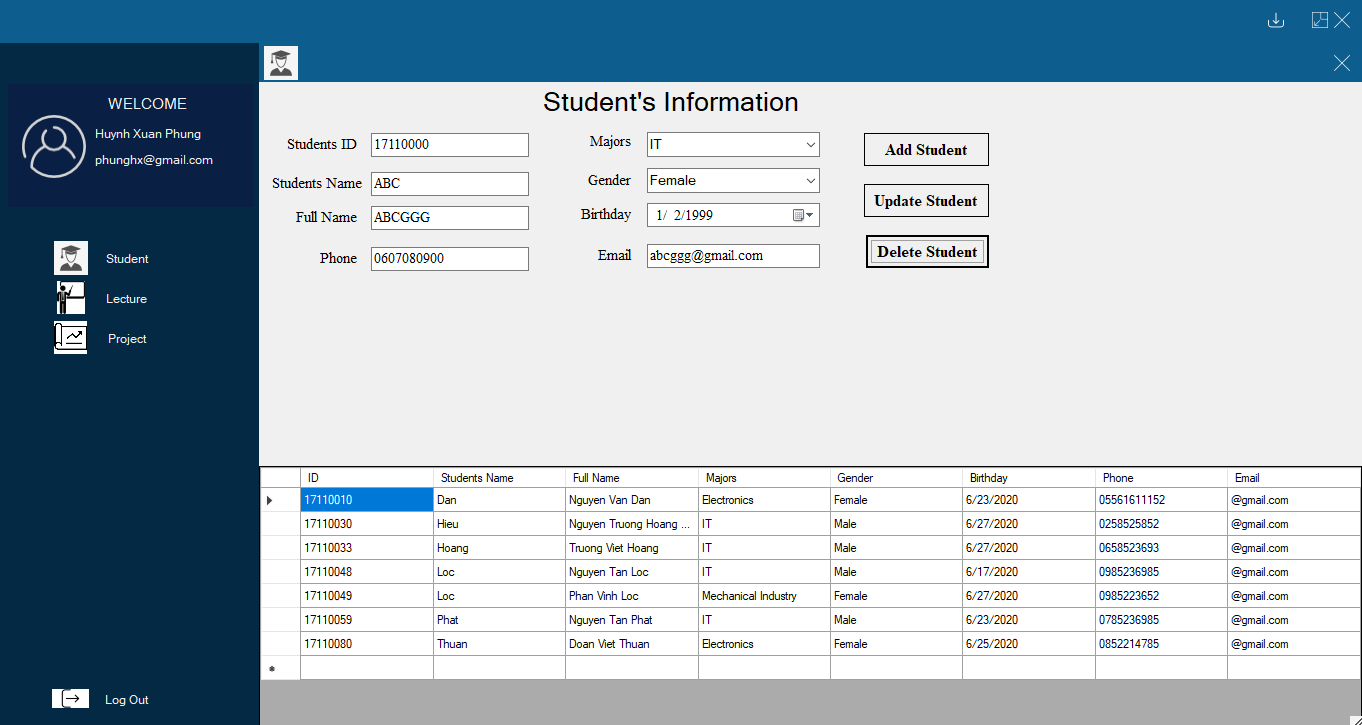


Figure 13: Test 6

The student have ID: 17110000 is deleted   
  
Test form Lecture’s Information  
Add 1 Lecture  
I will add 1 lecture have Lectures ID: GV000 , Lectures Name: Quang, Fullname: Nguyen Dang Quang , Phone:0700010203 , Department: Computer Engineer, Gender: Male, Birthday: 6/30/2020, Email: quangdn@gmail.com  
Before Add

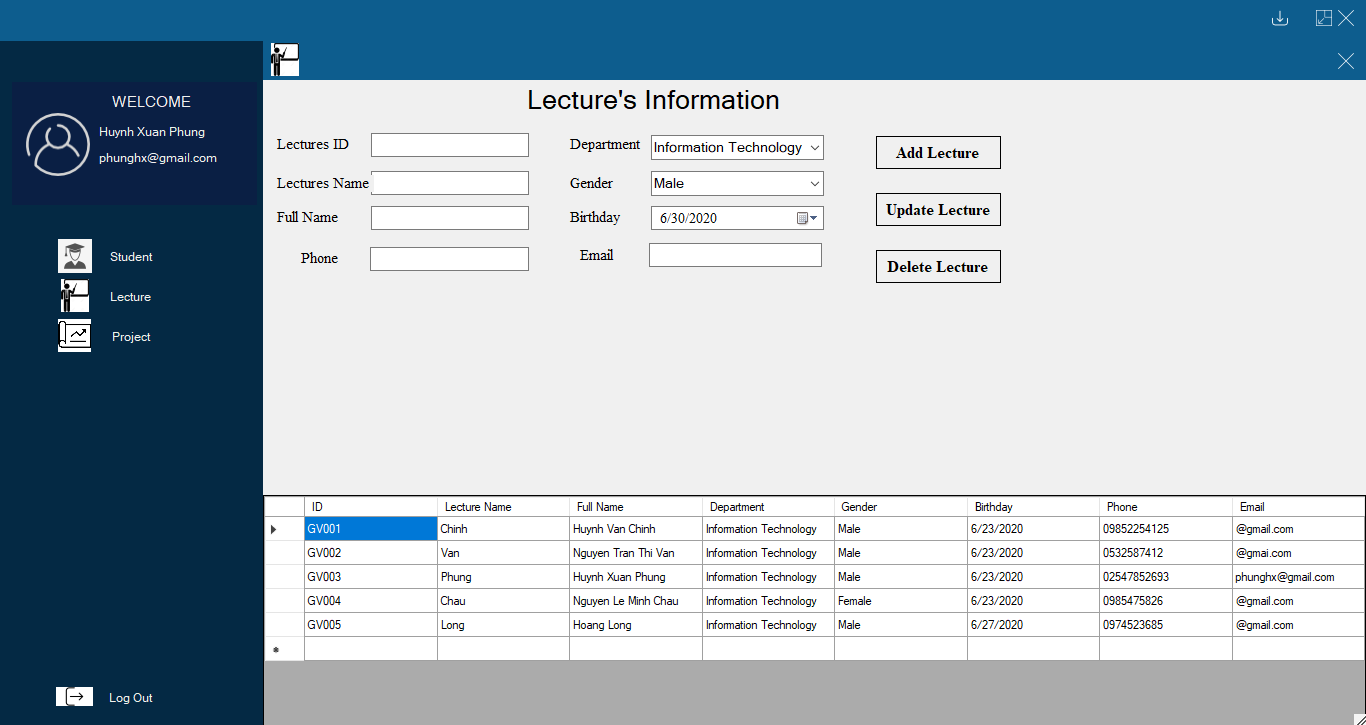


Figure 14: Test 7

After Add

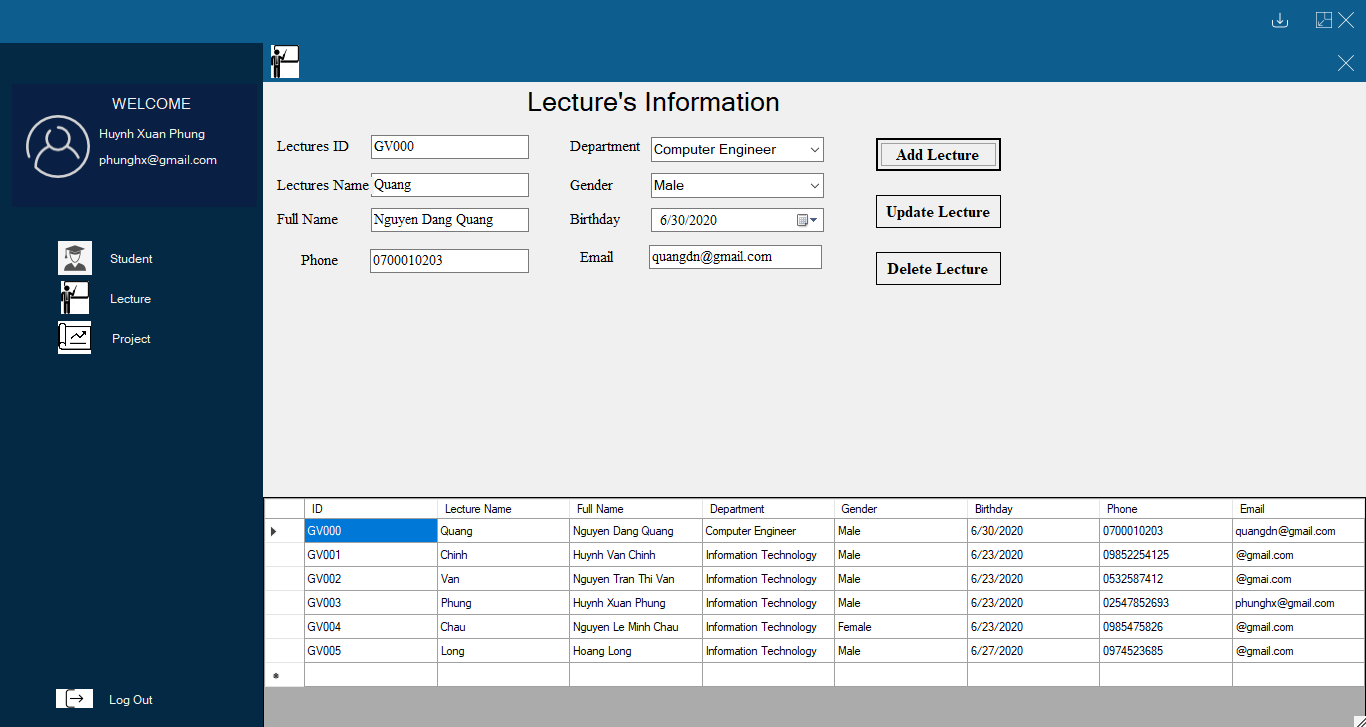


Figure 15: Test 8

Update 1 Lecture  
I will update the lecture who have ID: GV000, Department from Computer Engineer to Data Science

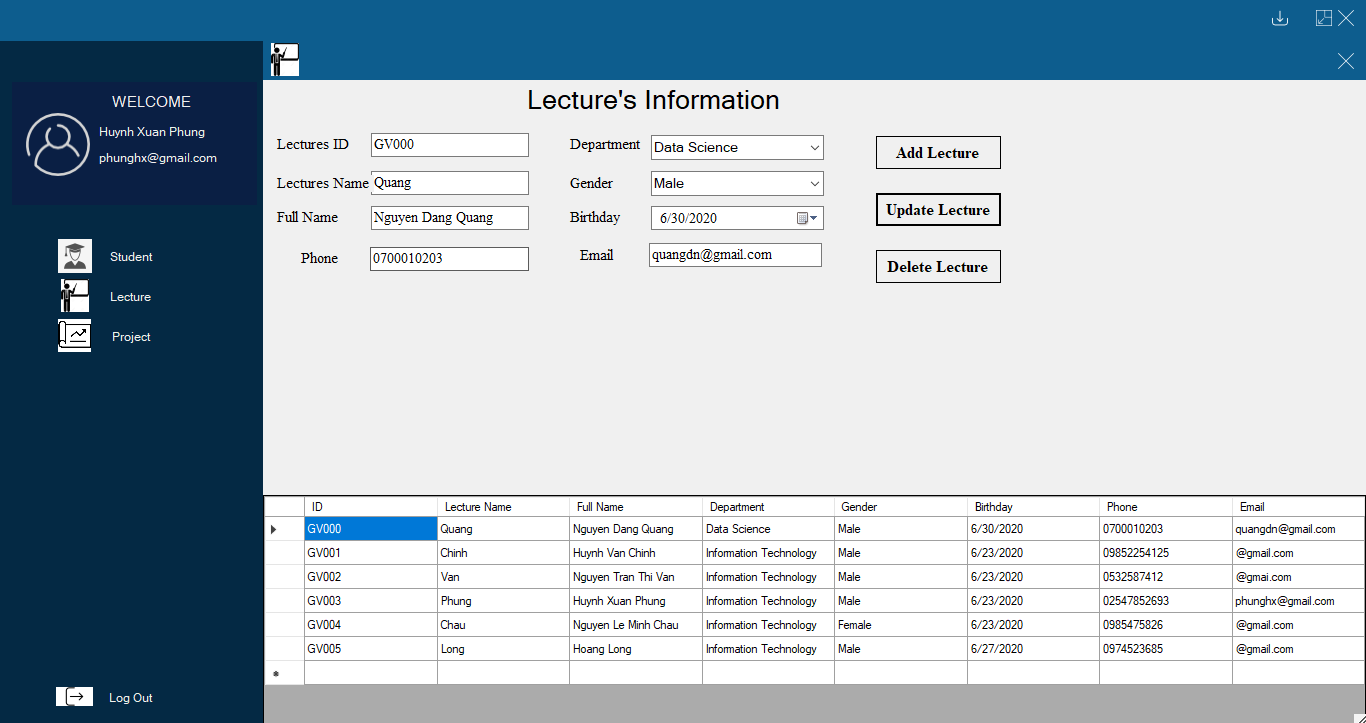


Figure 16: Test 9

Delete 1 Lecture  
I will delete the lecture who have ID: GV000  
Before Delete

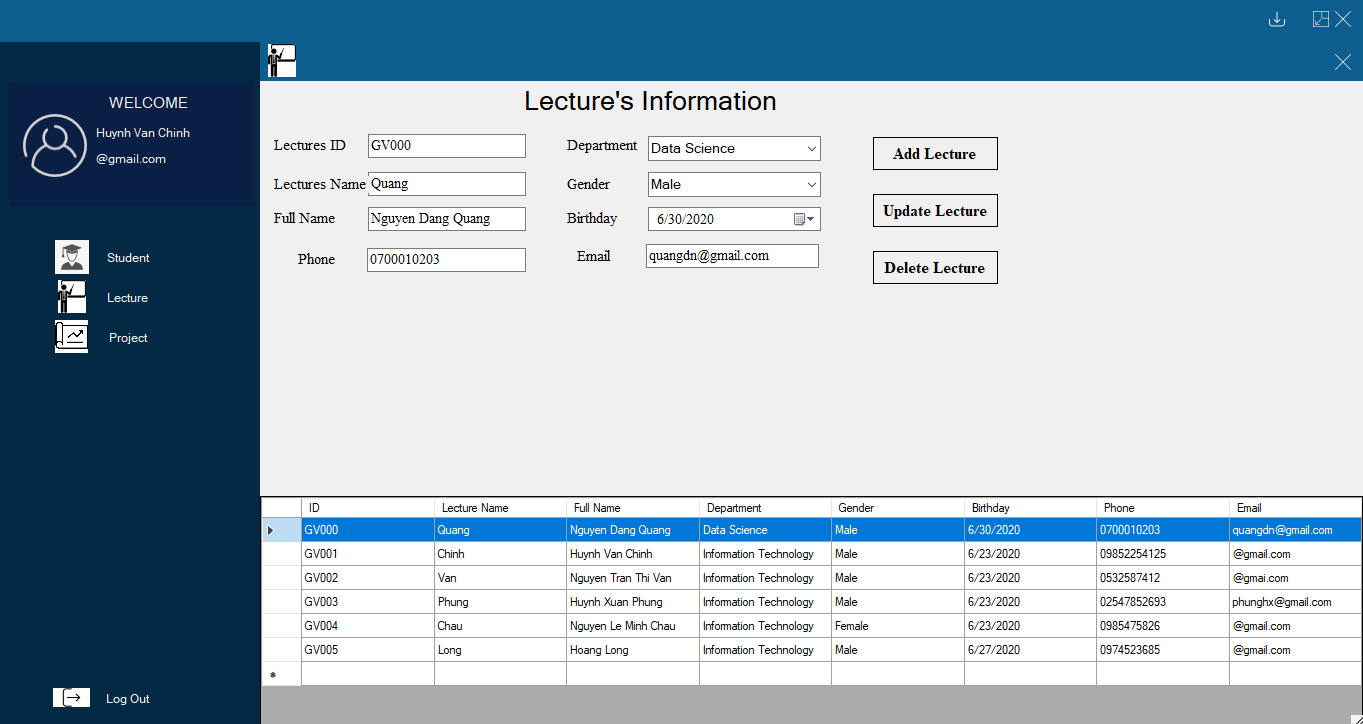


Figure 17: Test 10

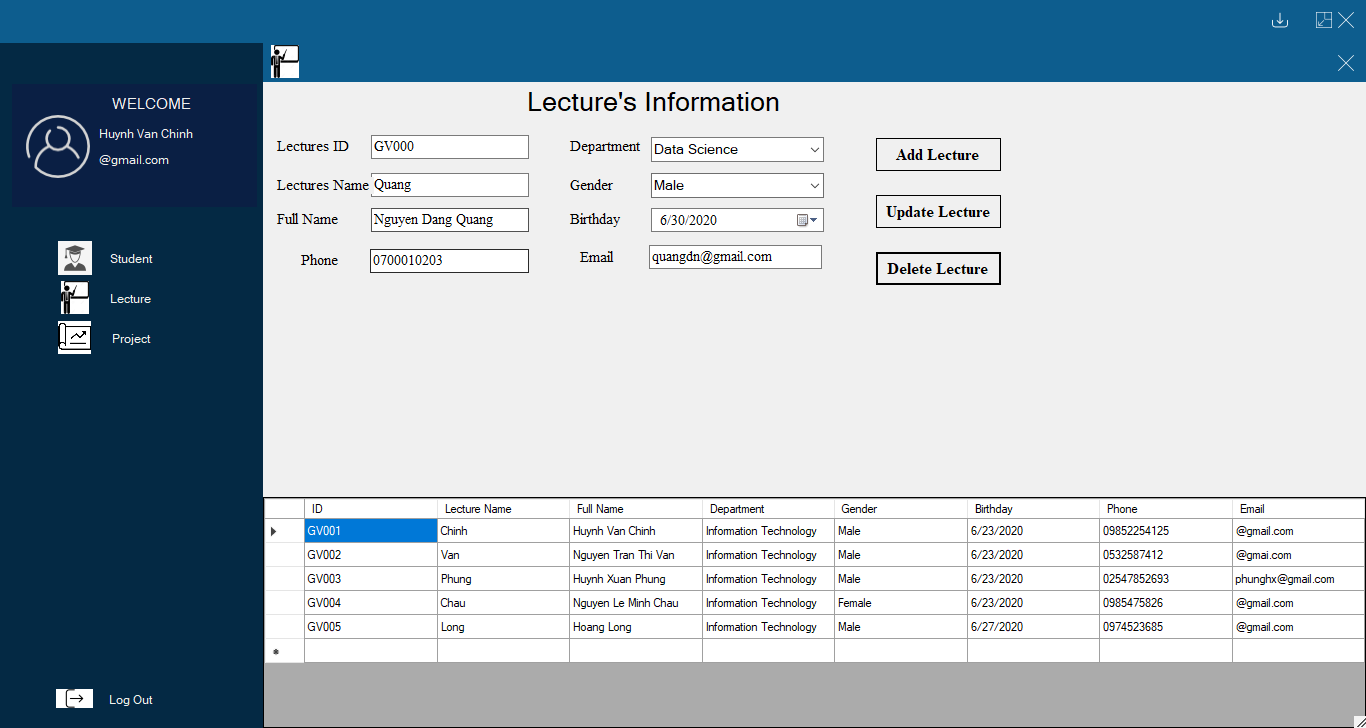
After Delete

Figure 18: Test 11

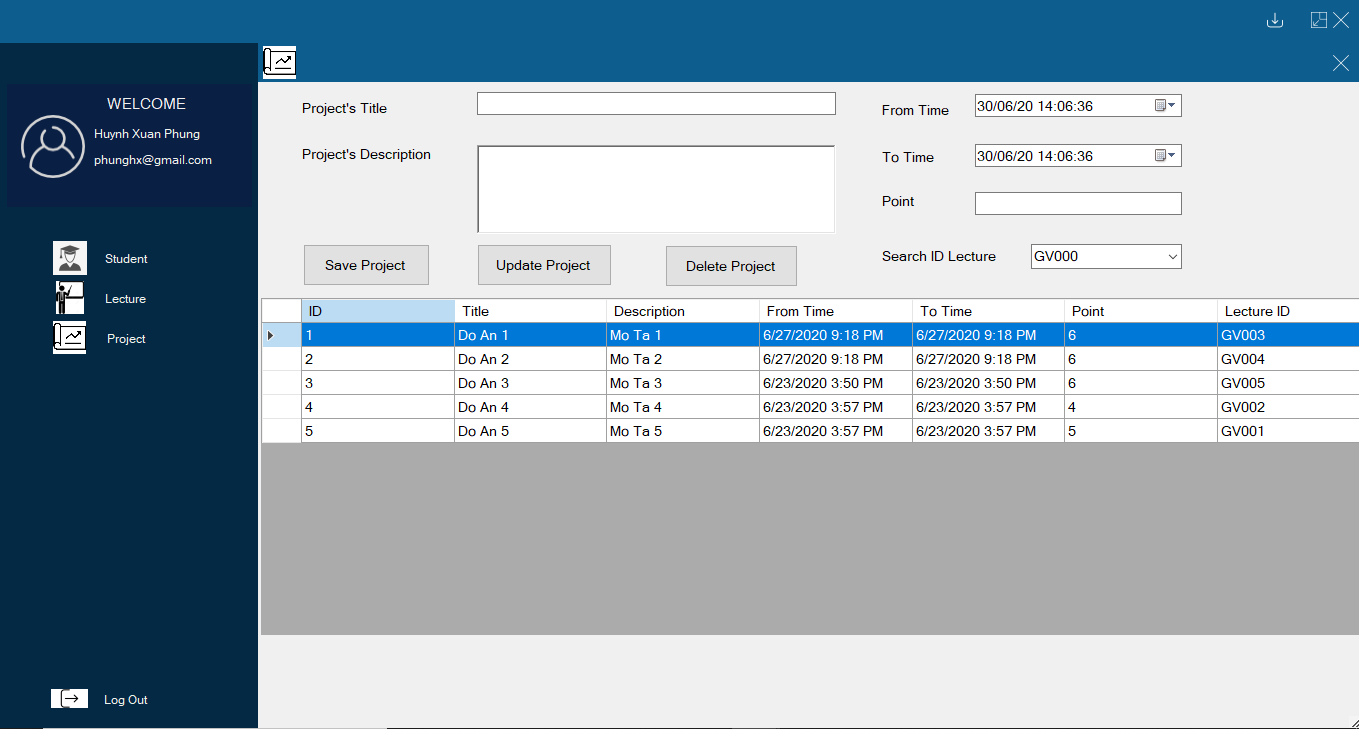
The lecture who have ID: GV000 is deleted  
Test form Project  
Add 1 project  
I will add 1 project which have Project’s Title: Do An 6, Project’s Description: Mo Ta 6, Form Time: 6/30/2020, To Time: 7/30/2020, Point: Lack, Lecture ID: GV000  
Before Add

Figure 19: Test 12

After Add

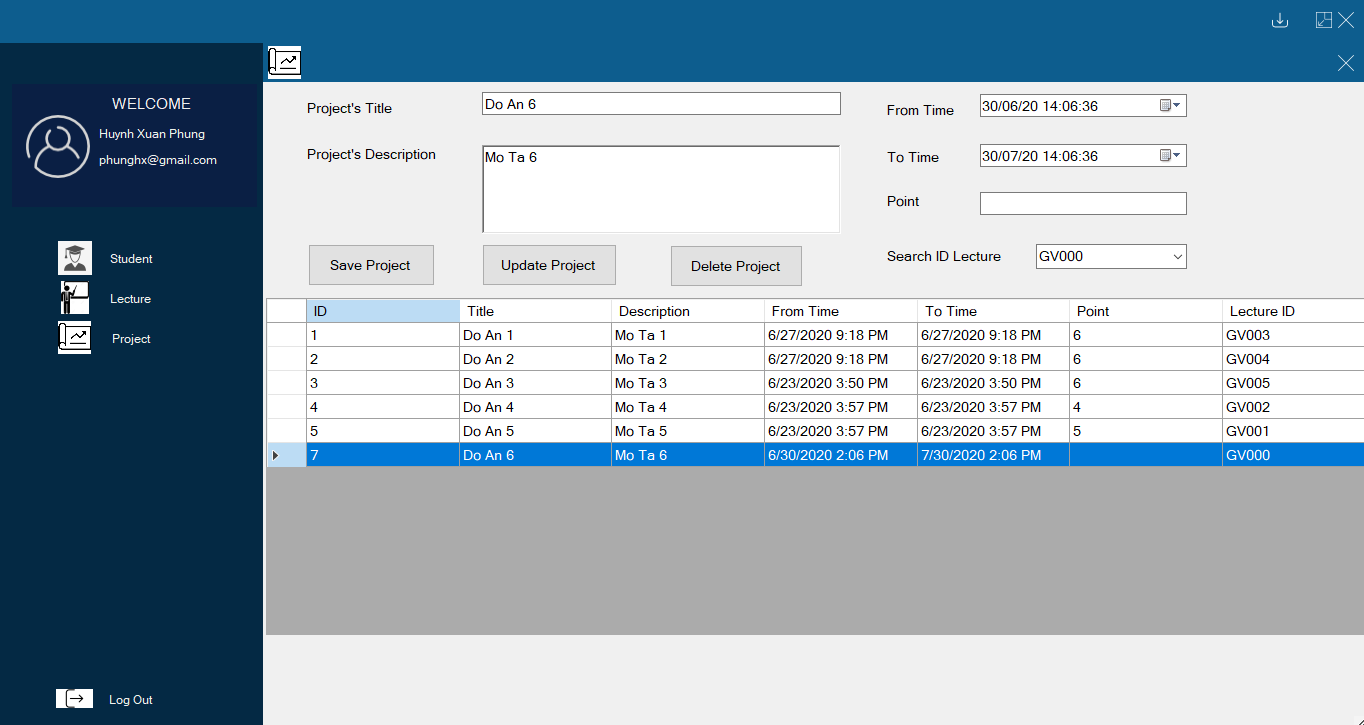


Figure 20: Test 13

Update 1 Project  
I will update project which Title: Do An 6 and change the Description to Update Mo ta 6, Change time 6/30/2020 to 8/30/2020

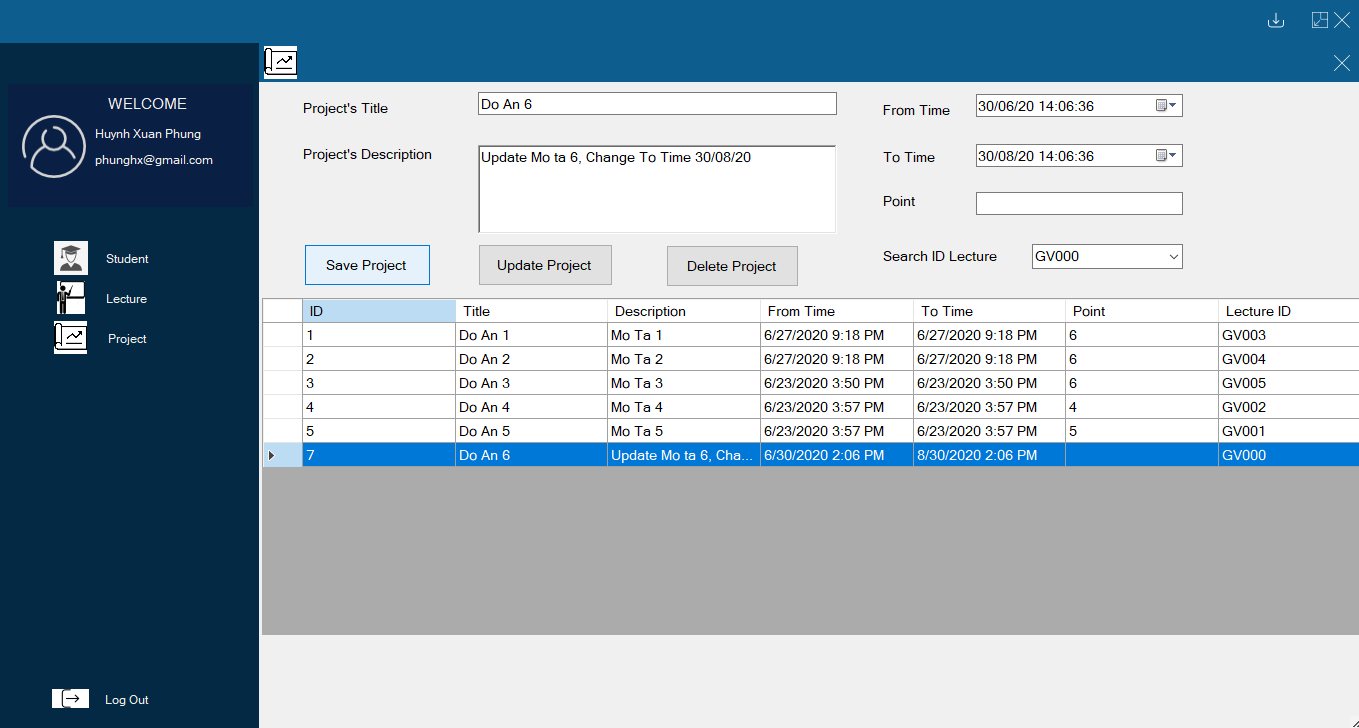


Figure 21: Test 14

Delete 1 Project  
I will delete project which have Title: Do An 6  
Before Delete

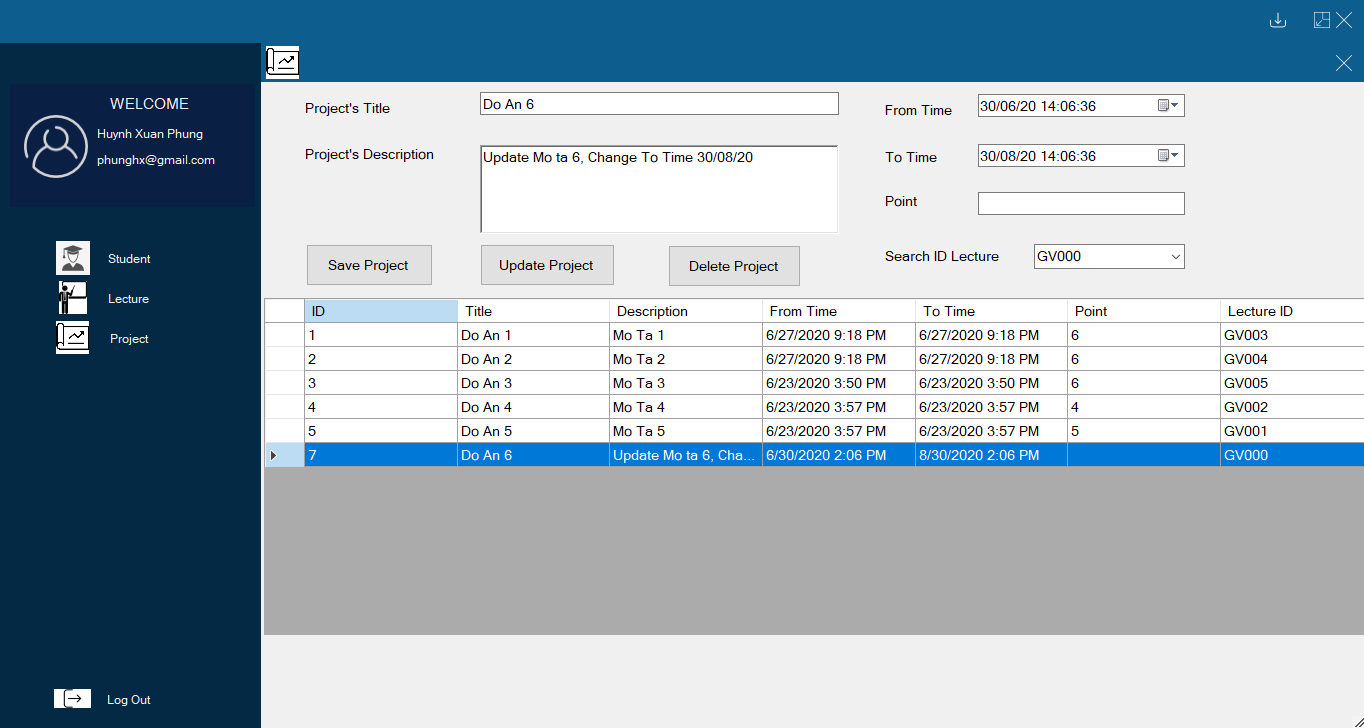


Figure 22: Test 15

After Delete

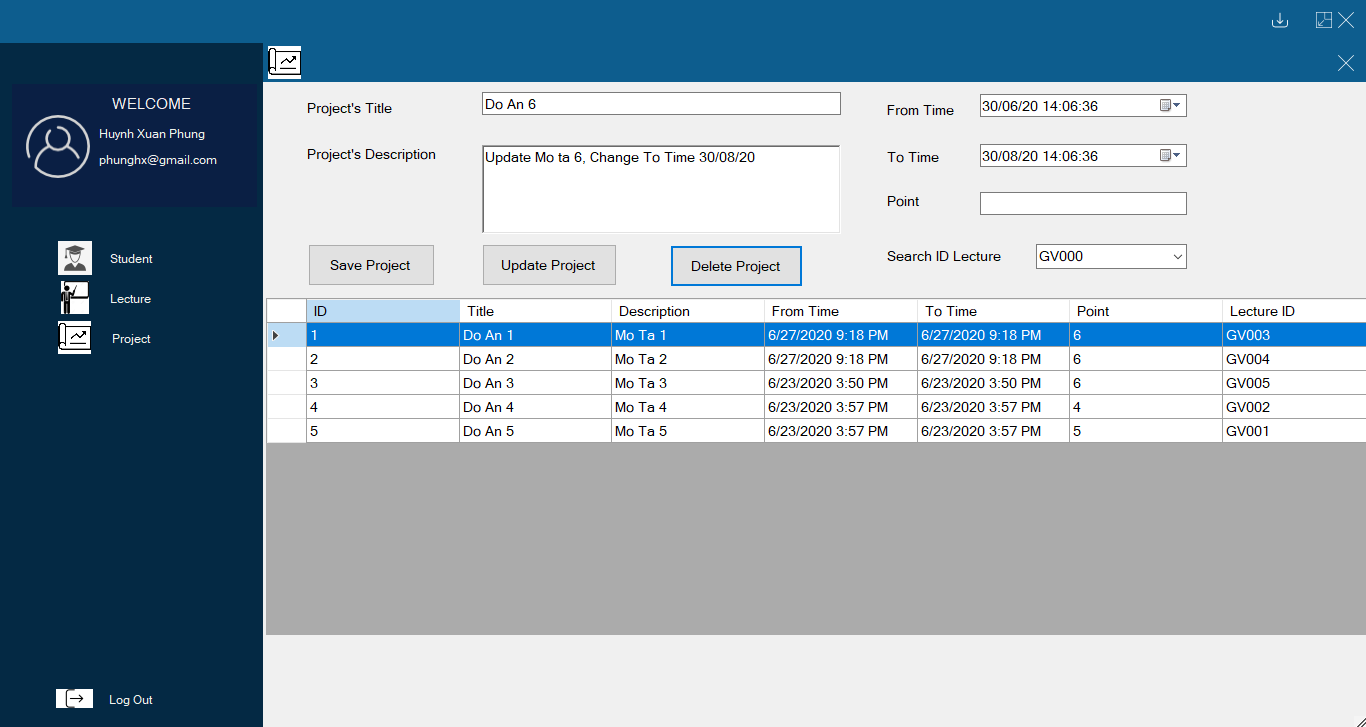


Figure 23: Test 16

The project which have title: Do An 6 is deleted  
Test form Project Detail  
Add 1 Student in Project Detail  
I will add 1 more student in project detail. You can select the student in list Student Name that I added before  
Before Add

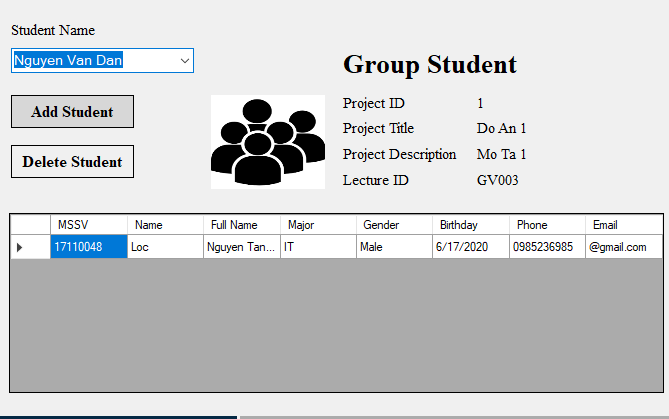


Figure 24: Test 17

After Add

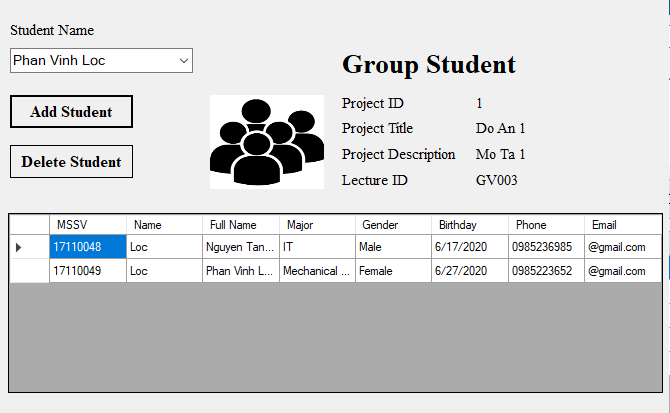


Figure 25: Test 18

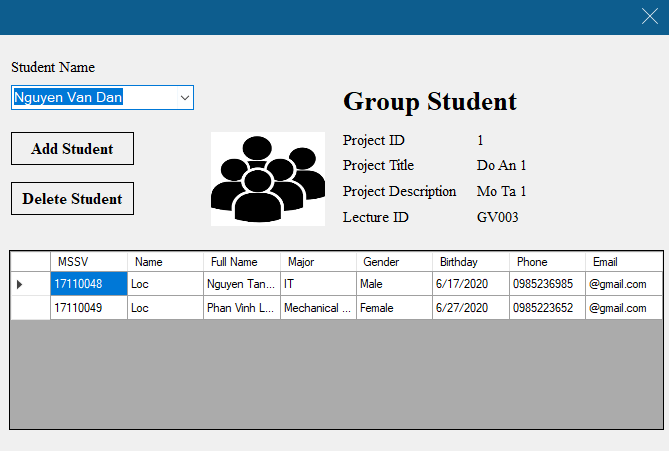
Delete 1 Student in Project Detail  
I will add the student Nguyen Tan Loc  
Before Delete

Figure 26: Test 19

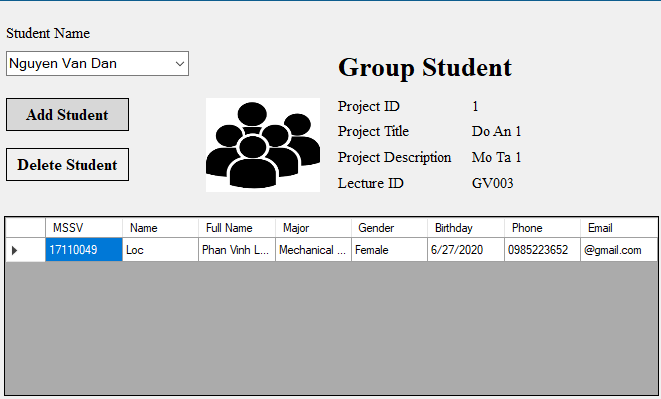
After Delete

Figure 27: Test 20

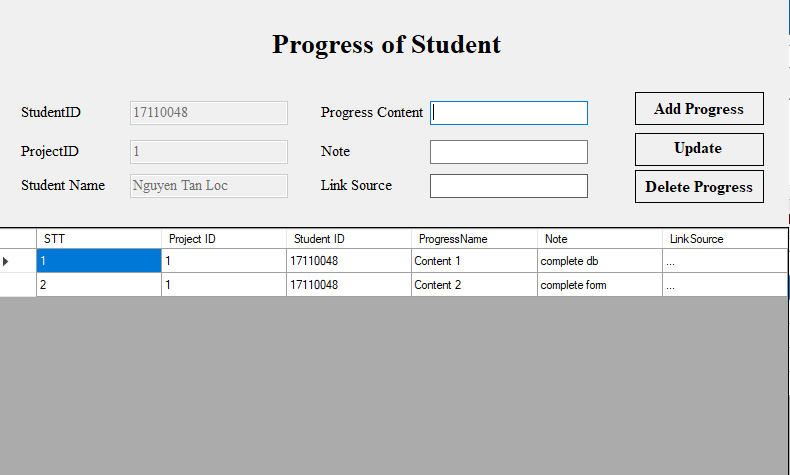
Test form Progress Detail  
Add 1 Progress  
I will add more progress at the project id 1 with Progress Content: Content 3 and Note: complete report  
Before Add

Figure 28: Test 21

After Add

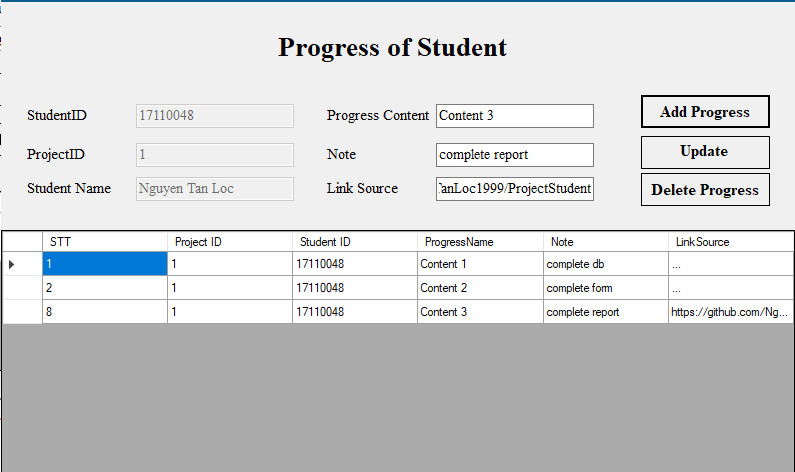


Figure 29: Test 22

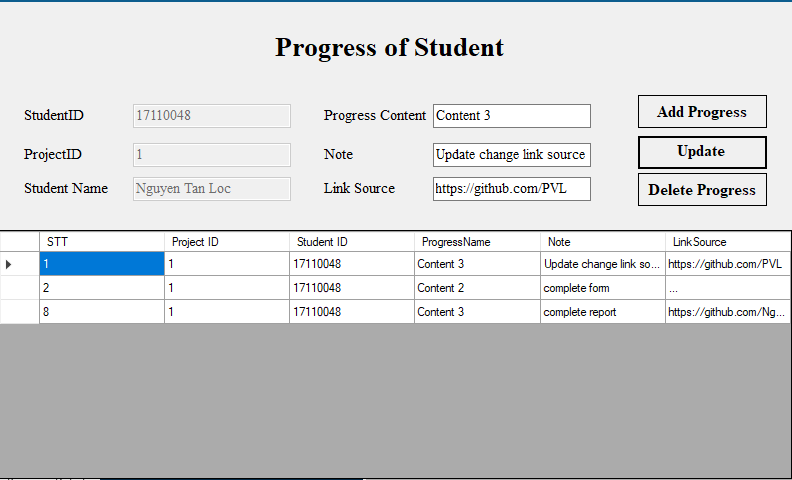
Update 1 Progress  
I will change the note of the project id 1 from complete db to update change link source   


Figure 30: Test 23

Delete 1 Progress  
I will delete the progress name Content3, STT: 1  
Before Delete

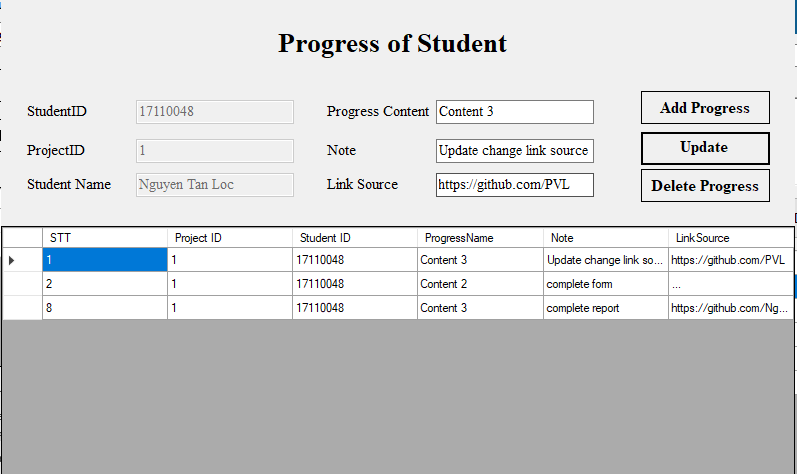


Figure 31: Test 24

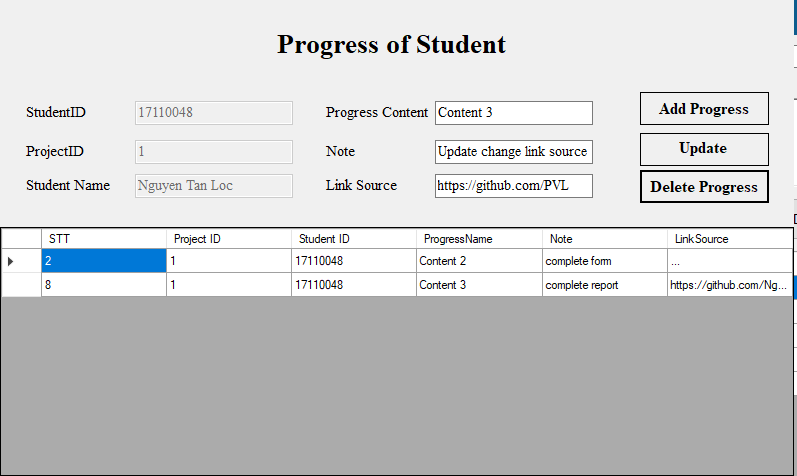
After Delete

Figure 32: Test 25

The progress have STT:1 is deleted

# Chapter 5: Conclusion and Development Direction

## 5.1 Conclusion

Basically, the team commented that its software solved 95% of the requirements set by the project. Following are the advantages and disadvantages of the software:

* Advantages:   
  The interface is neat, accessible, and easy to get acquainted.  
  Capacity quite light  
  The program consumes very little system resources when operating.  
  The program runs stably, gives accurate results, does not crash in the process of executing user requests.
* Disadvantages:

Hard to use LinQ  
Lack of Report in part Requirements

## 5.2 Development Direction

- Write more function Report in part Requirements  
- Write more function which can show all the project of any student/lecture in 1 course

# Reference

* We used the knowledge Mr. Phung taught to complete the project
* There are references sql query on google.com
* The specifically is on the website: stackoverflow.com