[I. OVERVIEW 3](#_Toc533630201)

[II. INTRODUCE 3](#_Toc533630202)

[2.1. Purpose of document 3](#_Toc533630203)

[2.2. Scope 4](#_Toc533630204)

[2.3. References 4](#_Toc533630205)

[2.4. Document description 4](#_Toc533630206)

[III. USE CASE MODEL 5](#_Toc533630207)

[3.1 Use case diagram 5](#_Toc533630208)

[3.2 Use case 5](#_Toc533630209)

[3.2.1 Login 5](#_Toc533630210)

[3.2.1.1 Scenarios 5](#_Toc533630211)

[3.2.1.2 Use case Description 5](#_Toc533630212)

[3.2.2. Register Subject 6](#_Toc533630213)

[3.2.2.1 Scenarios 6](#_Toc533630214)

[3.2.2.2 Use case Description 6](#_Toc533630215)

[3.2.3. View Student Record 6](#_Toc533630216)

[3.2.3.1 Scenarios 6](#_Toc533630217)

[3.2.3.2 Use case Description 7](#_Toc533630218)

[3.2.4. Manage entity information 7](#_Toc533630219)

[3.2.4.1. Scenarios 7](#_Toc533630220)

[3.2.4.2. Use Case Description 7](#_Toc533630221)

[3.2.5. Statistic tuition 7](#_Toc533630222)

[3.2.5.1. Scenarios 7](#_Toc533630223)

[3.2.5.2. Use Case Description 8](#_Toc533630224)

[3.2.6. Statistic tuition 8](#_Toc533630225)

[3.2.6.1. Scenarios 8](#_Toc533630226)

[3.2.6.2. Use Case Description 8](#_Toc533630227)

[3.2.7. View tuition fee information 8](#_Toc533630228)

[3.2.7.1 Use case scenarios 8](#_Toc533630229)

[3.2.7.2 Use case description 8](#_Toc533630230)

[3.2.8. Calculate tuition fee 9](#_Toc533630231)

[3.2.8.1 Scenarios 9](#_Toc533630232)

[3.2.8.2 Use case description 9](#_Toc533630233)

[3.2.9. Collect tuition fee 9](#_Toc533630234)

[3.2.9.1 Scenarios 9](#_Toc533630235)

[3.2.9.2 Use case description 10](#_Toc533630236)

[3.3. Actor description 10](#_Toc533630237)

[3.3.1. Student: 10](#_Toc533630238)

[3.3.2. Staff 10](#_Toc533630239)

[3.3.3. Database 11](#_Toc533630240)

[3.3.4. Tuition Fee Management System 11](#_Toc533630241)

[IV. SYSTEM ARCHITECTURE 11](#_Toc533630242)

[V. DATABASE DESIGN 11](#_Toc533630243)

[5.1 Entity relationship diagram 11](#_Toc533630244)

[5.1.1 Entities 11](#_Toc533630245)

[5.1.2 Entity relationship diagram 12](#_Toc533630246)

[5.2 Specification Tables 13](#_Toc533630247)

[5.3 Logical design 15](#_Toc533630248)

[VI. USER INTERFACE DESIGN 16](#_Toc533630249)

[6.1 Architecture MENU 16](#_Toc533630250)

[6.2. Interface of functions 18](#_Toc533630251)

[6.2.1 Form Login 18](#_Toc533630252)

[6.2.2 Form Collect Tuition Fees 19](#_Toc533630253)

[6.2.3 Form Manage Credit Class 22](#_Toc533630254)

[6.2.4 Form Manage Student 24](#_Toc533630255)

[6.2.5 Form Manage Class 26](#_Toc533630256)

[6.2.6 Form Manage Diminished Object 27](#_Toc533630257)

[6.2.7 Form Statistic Fee Payment List 28](#_Toc533630258)

[6.2.8 Form Statistic Fee Debt List 30](#_Toc533630259)

[6.2.9 Form Statistic Payment List Per Day 31](#_Toc533630260)

[6.2.10 Form View Student Record 32](#_Toc533630261)

[6.2.11 Form Register Credit 32](#_Toc533630262)

[6.2.12 Form View Payment Information 34](#_Toc533630263)

[VII. IMPLEMENTATION 36](#_Toc533630264)

[VIII. TESTING DOCUMENTS 36](#_Toc533630265)

[8.1. Test case 36](#_Toc533630266)

[8.2. Test Scenario 38](#_Toc533630267)

[8.3. Requirement Traceability Matrix 42](#_Toc533630268)

I. OVERVIEW

Today, in parallel with the development of science and technology, computer science holds a central place in most areas of society. It has achieved brilliant scientific achievements with its leaps and bounds. The application of science and technology to human life is increasing and constantly interfering in most of the work in life. Information technology is one of those sciences. Coupled with the development of computer-aided manufacturing technology, the adoption of software applications is critical to the application of this science.

In the world as well as in Vietnam, information technology has become a spearhead industry, it is an indispensable science and technology in applying to social activities such as management, business information, ...

In recent years, in our country, informatics has been widely applied in the field of education and training. The application of management software in schools has greatly assisted managers from the management, management of the learning process, training and management of tuition fees - students. Ensure fast and accurate science.

In the face of the increasing number of universities and faculties, it is a waste of time and effort to keep the old management method. Therefore, the application of information technology in the management of universities is very reasonable.

Many students, faculties, departments, and training types that lead to student tuition fees also differ, the management of tuition fees becomes increasingly difficult. To help minimize that difficulty and to actively support the university's fee collection team, our team works on the "Student fees collection management" program.

II. INTRODUCE

2.1. Purpose of document

This is a detailed description of student fees collection management software, which is the basis for programming functions as well as scripting testing and program testing.

2.2. Scope

Documentation is the result of the software design process, which is the communication basis of the developer, which is the basis for testing, commissioning,

maintenance, and provision of book management functions.

2.3. References

|  |  |  |
| --- | --- | --- |
| **Number** | **Name of document** | **File Name** |
| 1 | Requirements specification document | Requirements specification document.doc |

2.4. Document description

The structure of the document consists of 5 parts:

• **Part 1 - Overview:** General introduction of the document, help the reader visualize the content, purpose and general layout of the document.

• **Part 2 - Introduce:** Pose the most basic requirements that software must have.

• **Part 3 - Use case model.:** Building architectural models and describing system architecture.

• **Part 4 - System architecture**: Describe data dictionary and build data dictionary.

• **Part 5 - Database design:** Design subsystems of the system.

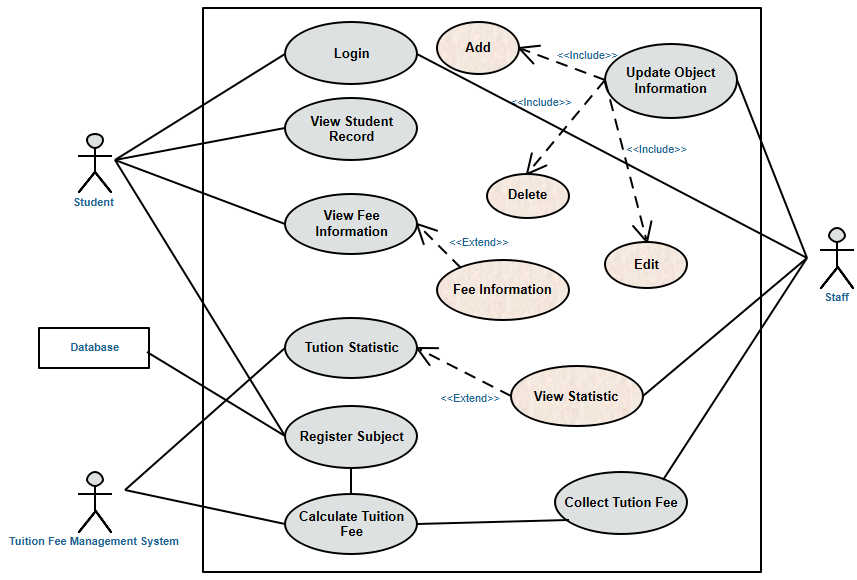
• **Part 6 - User Interface Design:** An overview of the interfaces, interface images, and accompanying operations.

• **Part 7 - Implementation:** An overview of the interfaces, interface images, and accompanying operations.

• **Part 8 – Testing documents:** An overview of the interfaces, interface images, and accompanying operations.

III. USE CASE MODEL

3.1 Use case diagram



*Figure 3.1: User case diagram for system “Student fees collection management”*

3.2 Use case

3.2.1 Login

3.2.1.1 Scenarios

***a) Successful scenario***

- Enter a valid account name

- Enter the password that matches the account name

- Staff and students on their account to perform the necessary functions

***b) Fail scenario***

- Login failed because of wrong account name

- Login failed because the password does not match the account

3.2.1.2 Use case Description

- Use case: Login

- Actors: Student, Staff

- Goal: Login to student and staff accounts so that the subjects perform the necessary functions.

- Description: When students and staff enter the system, they will log in to their account by entering their username and password. Students will register for the course of the semester; View their profile and view subscription information periodically. Staff will update the objects such as: class section, class management, students, subjects exemptions; List the tuition fees paid each day and collect the tuition fees of each student to submit the invoice.

3.2.2. Register Subject

3.2.2.1 Scenarios

***a) Successful scenario***

Students register for each subject for each semester

- Select the year

- Select a semester

- Select the subject

- Choose a teacher who will teach the subject

***b) Fail scenario***

- There is no subject to register in that semester

- There is no teacher who student wants to apply for the course

3.2.2.2 Use case Description

- Use case: Register Subject

- Actors: Student

- Goal: Students register periodically

- Description: Students register for each semester by choosing the academic year, select the semester, select the subject, select the teacher then proceed to register. When registering for the course, the course registration information will be automatically updated in the registration database.

3.2.3. View Student Record

3.2.3.1 Scenarios

***a) Successful scenario***

Check for personal information such as: Student ID, Full Name, Address, Gender, Date of Birth, Subject, Email, ... and completely match.

***b) Fail scenario***

Check your personal information such as: Student ID, Full Name, Address, Gender, Date of Birth, Subject, Email, ... and inaccurate information.

3.2.3.2 Use case Description

- Use case: View Student Record

- Actors: Student

- Goal: Allows students to check their personal information

- Description: When the student successfully login to their account, there will enter the "Student Profile" section to check their personal information such as Student ID, Full Name, Address, Gender, Audience, Email, ...

3.2.4. Manage entity information

3.2.4.1. Scenarios

***a) Successful scenario***

- The staff informed by the education authorities: The next semester will have more student A

- The staff enters the personal information of student A

- The system records the existence and storage of A’s information

***b) Fail scenario***

<Empty>

3.2.4.2. Use Case Description

Staff can perform additional operations, edit, delete information about objects such as: Class, Student, Management class, Exempt object to update for the System Database.

3.2.5. Statistic tuition

3.2.5.1. Scenarios

***a) Successful scenario***

- Student A wants to review the list of his tuition receipts submitted in the previous semester and asks the staff to look for them.

- The staff enters the time range of the previous semester.

- The system processes and displays a list of invoices collected in the previous semester.

***b) Fail scenario***

<Empty>

3.2.5.2. Use Case Description

Staff can perform a search to list the students who have paid tuition for a certain period of time, including information such as: student ID, student name, total amount, collection staff.

3.2.6. Statistic tuition

3.2.6.1. Scenarios

***a) Successful scenario***

- The staff performs login to the system using his account.

- The staff clicks on the tab view to perform the view function.

- The system displays a list of all the students along with the bills they have paid.

***b) Fail scenario***

<Empty>

3.2.6.2. Use Case Description

The staff uses his account to log in to the system and sees a list of all students who have paid tuition and related information.

3.2.7. View tuition fee information

3.2.7.1 Use case scenarios

***a) Successful scenario***

- Hoai logs into the system.

- She selects a View tuition fee payment information tab

- She selects Semester and Year

- She clicks the Load button

- She sees tuition fee payment information of that semester and year.

3.2.7.2 Use case description

- Use case: View tuition fee payment information

- Actors: Student

- Goal: To view tuition fee payment information

- Description:

When a student logs into the system, he/she clicks the View tuition fee payment information tab. He/she selects Semester and Year to see the tuition fee payment information of that semester of the year. The system will load information from bill entity. The student will see the tuition fee payment information of that semester of the year.

3.2.8. Calculate tuition fee

3.2.8.1 Scenarios

***a) Successful scenario***

- Hoai logs into the system

- She registers 9 subjects for semester 1 of the year 2018-2019

- The Tuition fee management system calculates and shows the tuition fee of that semester is 6050000VND.

3.2.8.2 Use case description

- Use case: Calculate tuition fee

- Actors: Tuition fee management system

- Goal: To calculate the tuition fee

- Description:

When a student logs into the system to register subjects for one semester. The tuition fee management system will calculate the tuition fee and show the tuition fee for each subject also the total tuition fee of that semester.

3.2.9. Collect tuition fee

3.2.9.1 Scenarios

***a) Successful scenario***

- Hoai arrives at Finance Department to pay tuition fee for semester 1 of year 2018-2019.

- Phuong enters Hoai’s ID, selects Bill Name, Semester and Year. The system shows total bill information.

- Phuong tells Hoai the total tuition fee she has to pay.

- Hoai pays for the tuition fee.

- Phuong exports the bill and give it to Hoai.

***a) Fail scenario***

- Hoai arrives at Finance Department to pay the tuition fee for semester 1 of the year 2018-2019.

- Phuong enters Hoai’s ID, selects Bill Name, Semester and Year. The system shows total bill information.

- Phuong tells Hoai the total tuition fee she has to pay.

- Hoai realizes she does not have enough money to pay the tuition fee. She tells Phuong that she will come back on another day to pay.

3.2.9.2 Use case description

- Use case: Collect tuition fee

- Actors: Staff

- Goal: To collect tuition fee

- Description:

When a student arrives to pay a tuition fee, the staff enter his/her ID, bill name, semester and year. The system will show details of tuition fee bill. The student pays for tuition fee, the staff exports bill and give it to the student.

3.3. Actor description

3.3.1. Student:

Students use the system to login to their account; Check your personal information such as: Student ID, Full Name, Address, Gender, Date of Birth, Subject, Email,... as well as change password in "Student Profile" section; Register the course for each semester, select the Year, Semester, Subjects and Subject Teachers in the "Credit Registration" section. In addition, students will use the system to view their payment information, in the "Payment Information" section will list the subjects of students in a term of a certain year and the status of payment of fees of the subjects in this semester.

3.3.2. Staff

The Staffs use the system to manage entity information, view tuition fee statistic and collect tuition fee.

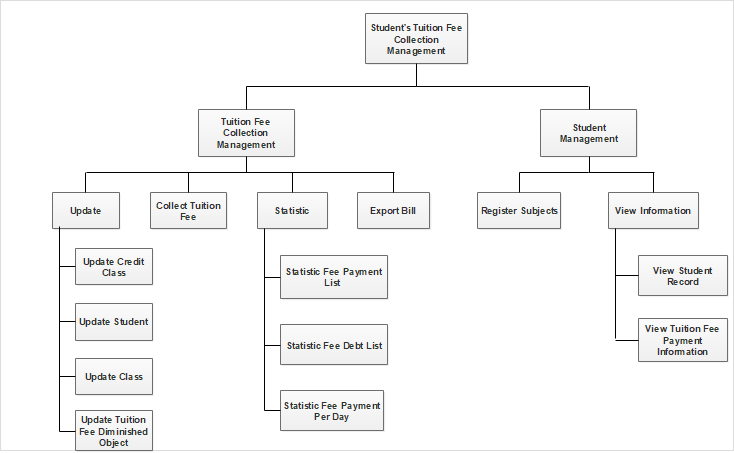
3.3.3. Database

System input is a list of modules and lists the names of the teachers teaching that module.

3.3.4. Tuition Fee Management System

Calculates the total tuition of a given period of a student and statistics each period of tuition of each student as well as the statistics of tuition fees that the school collects on a daily basis.

IV. SYSTEM ARCHITECTURE



*Figure 4.1. Business Fuction Diagram of Student’s Tuition Fee Collection Management System*

V. DATABASE DESIGN

5.1 Entity relationship diagram

5.1.1 Entities

Based on the specific activities of the Student’s Tuition Fee Collection Management System and the above functional decomposition model, we identify the entities:

1. Student

2. Class

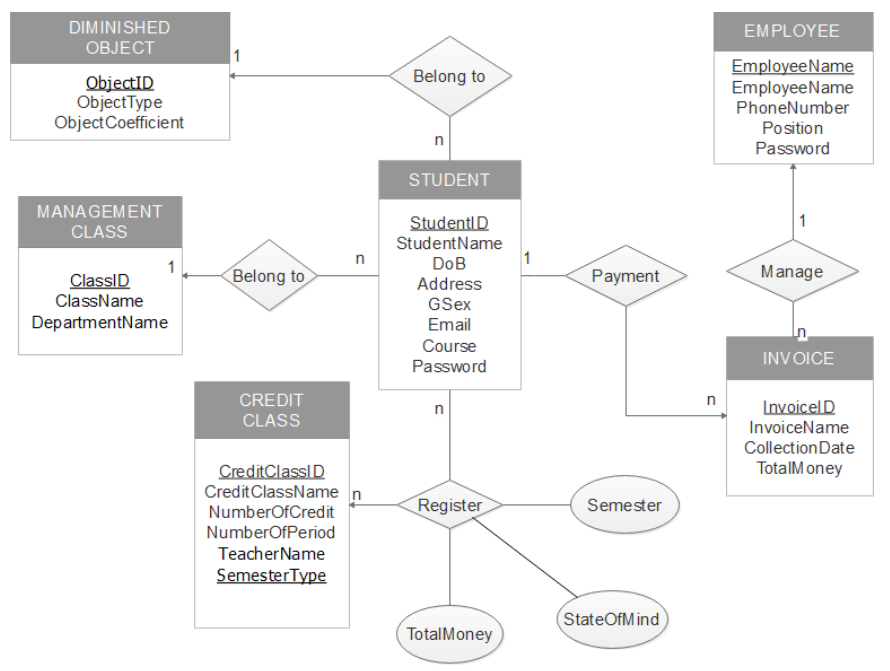
3. CreditClass

4. Employee

5. Bill

6. DiminishedObject

5.1.2 Entity relationship diagram



*Figure 5.1. Entity Relationship Diagram*

5.2 Specification Tables

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CREDITCLASS** | | | | | |
| *Field* | *Type* | *Index* | *NULL* | *Default* | *Note* |
| **CreditClassID** | **CHAR(10)** | **PRIMARY** | **NO** |  | ID |
| CreditClassName | NVARCHAR(50) |  | YES |  | Name |
| NumOfCredit | INT |  | YES |  | Number of credit |
| NumOfPeriod | INT |  | YES |  | Number Of Period |
| TeacherName | NVARCHAR(50) |  | YES |  | Name Of Teacher |
| **SemesterType** | **CHAR(10)** | **PRIMARY** | **NO** |  | Type Of Semester |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STUDENT** | | | | | |
| *Field* | *Type* | *Index* | *NULL* | *Default* | *Note* |
| **StudentID** | **CHAR (10)** | **PRIMARY** | **NO** |  | ID |
| StudentName | NVARCHAR(100) |  | YES |  | Name |
| DoB | DATE |  | YES |  | Date Of Birth |
| StudentAdd | NVARCHAR(50) |  | YES |  | Address |
| Sex | NVARCHAR(10) |  | YES |  | Sexuality |
| Email | VARCHAR(50) |  | YES |  | Email |
| Term | VARCHAR(50) |  | YES |  |  |
| Password | CHAR(10) |  | YES |  | Password |
| **ClassID** | **CHAR(10)** | **REFERENCES** | **NO** |  | **Class ID** |
| **ObjectID** | **CHAR(10)** | **REFERENCES** | **NO** |  | **Object ID** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DIMINISHEDOBJECT** | | | | | |
| *Field* | *Default* | *Index* | *NULL* | *Default* | *Note* |
| **ObjectID** | **CHAR (10)** | **PRIMARY** | **NO** |  | ObjectID |
| ObjectType | NVARCHAR(30) |  | YES |  | Type Object |
| Coeff | FLOAT |  | YES |  | Diminish Coefficient |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **BILL** | | | | | |
| *Field* | *Type* | *Index* | *NULL* | *Default* | *Note* |
| **BillID** | **CHAR(10)** | **PRIMARY** | **NO** |  | ID |
| BillName | NVARCHAR(50) |  | YES |  | Name |
| PaymentDate | DATE |  | YES |  | Date of Payment |
| Total | BIGINT |  | YES |  | Total money |
| **StudentID** | **CHAR(10)** | **REFERENCES** | **NO** |  | Student ID |
| **EmployeeID** | **CHAR(10)** | **REFERENCES** | **NO** |  | Employee ID |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EMPLOYEE** | | | | | |
| *Field* | *Type* | *Index* | *NULL* | *Default* | *Note* |
| **EmployeeID** | **CHAR (10)** | **PRIMARY** | **NO** |  | ID |
| EmployeeName | NVARCHAR(50) |  | YES |  | Name |
| PhoneNum | CHAR(15) |  | YES |  | Phone Number |
| Position | NVARCHAR(20) |  | YES |  | Position |
| Password | CHAR(10) |  | YES |  | Password |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CLASS** | | | | | |
| *Field* | *Type* | *Index* | *NULL* | *Default* | *Note* |
| **ClassID** | **CHAR(10)** | **PRIMARY** | **NO** |  | ID |
| ClassName | NVARCHAR(50) |  | YES |  | Name |
| Faculty | NVARCHAR(50) |  | YES |  | Faculty |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REGISTER** | | | | | |
| *NULL* | *Default* | *Index* | *NULL* | *Default* | *Note* |
| **StudentID** | **CHAR (10)** | **PRIMARY** | **NO** |  | ID |
| **CreditClassID** | **CHAR (10)** | **PRIMARY** | **NO** |  | ID |
| **SemesterType** | **CHAR (10)** | **PRIMARY** | **NO** |  | Type |
| Year | CHAR(50) |  | YES |  | School Year |
| Total | BIGINT |  | YES |  | Total Money |

5.3 Logical design

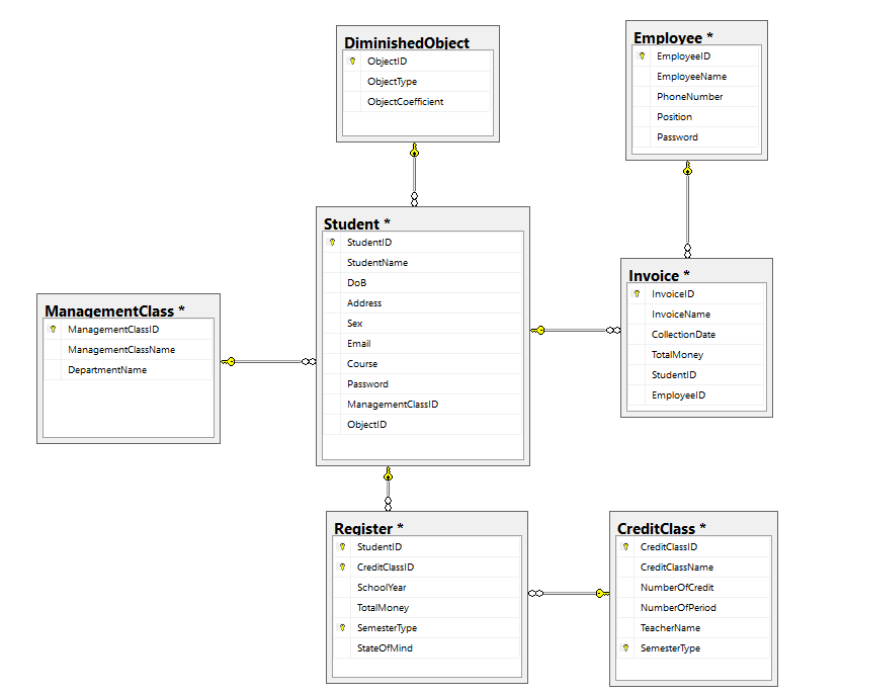


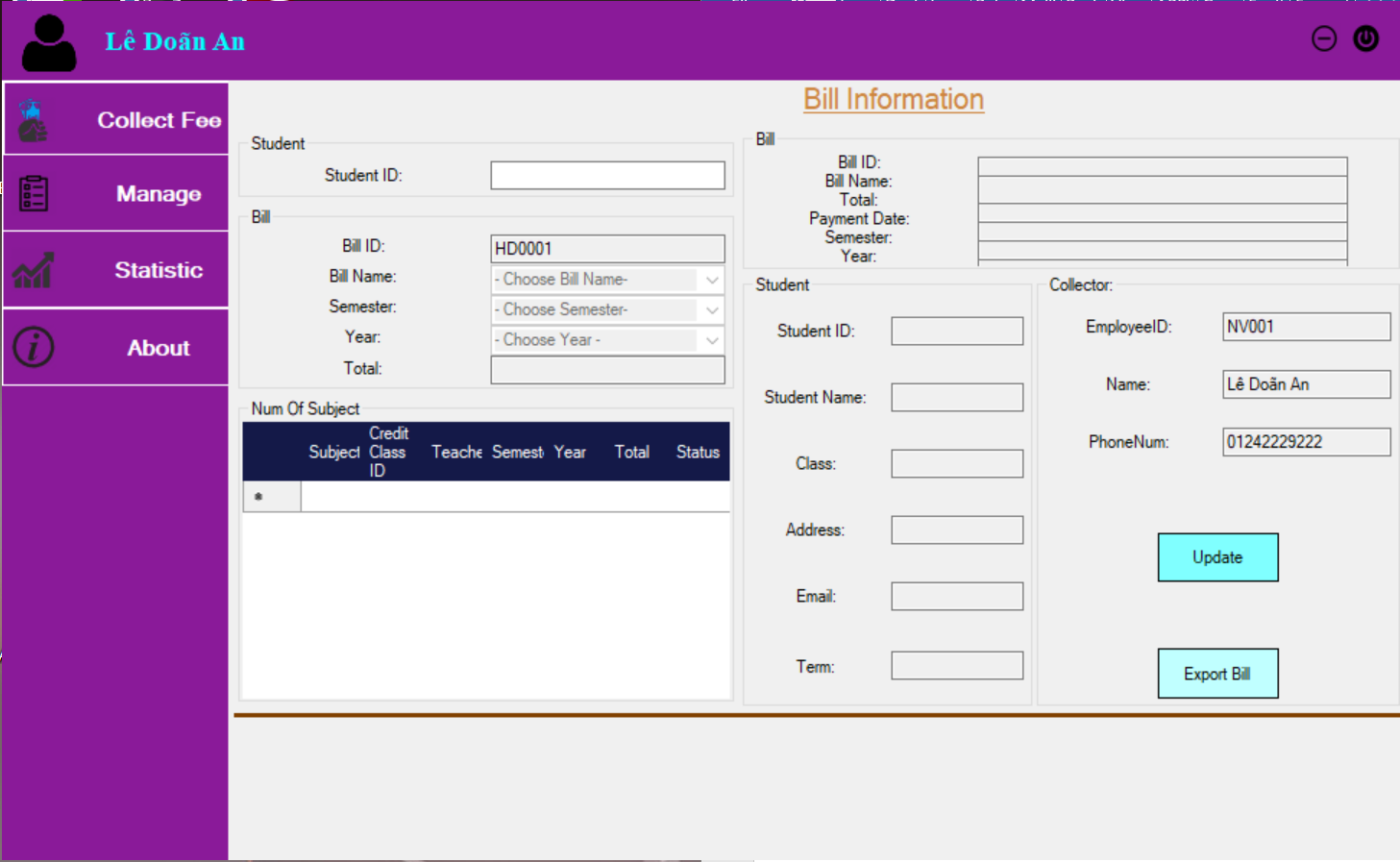
Figure 5.2. Logical design

VI. USER INTERFACE DESIGN

6.1 Architecture MENU

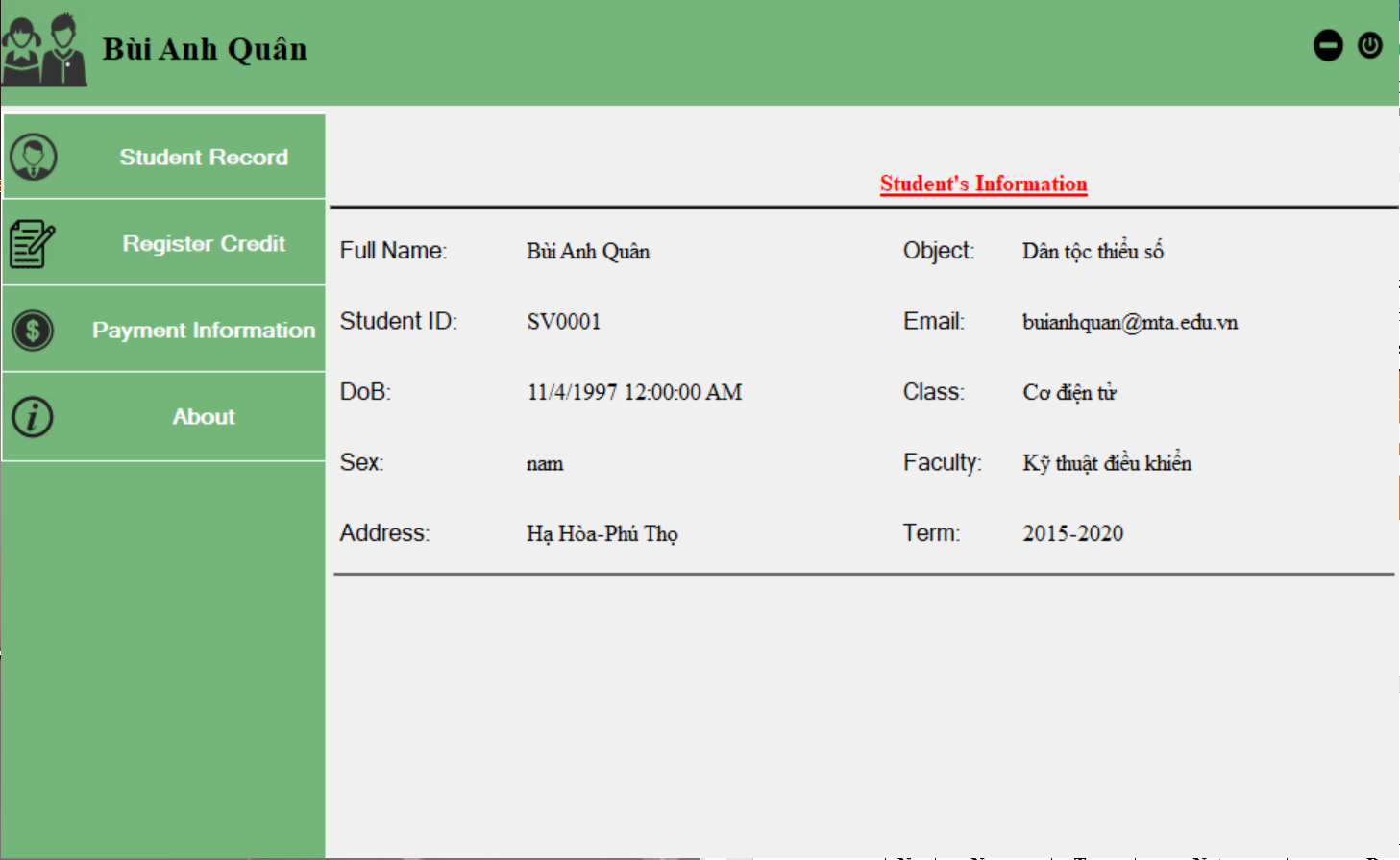
The main menu of the Menu is built for two different users: Employee and Student.

- For Employee access:



*Figure 6.1: Interface for login under Employee access*

- For Student Access:



*Figure 6.2: Interface for login under Student access*

6.2. Interface of functions

6.2.1 Form Login

When the program runs, a login window appears, allowing the user to enter a User name (this use Employee Code or Student Code) and Password.

There are two subjects that are built as Employees and Students.

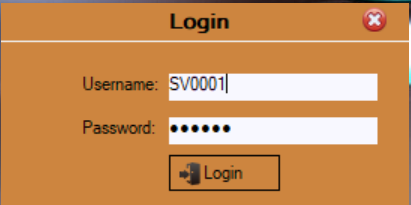


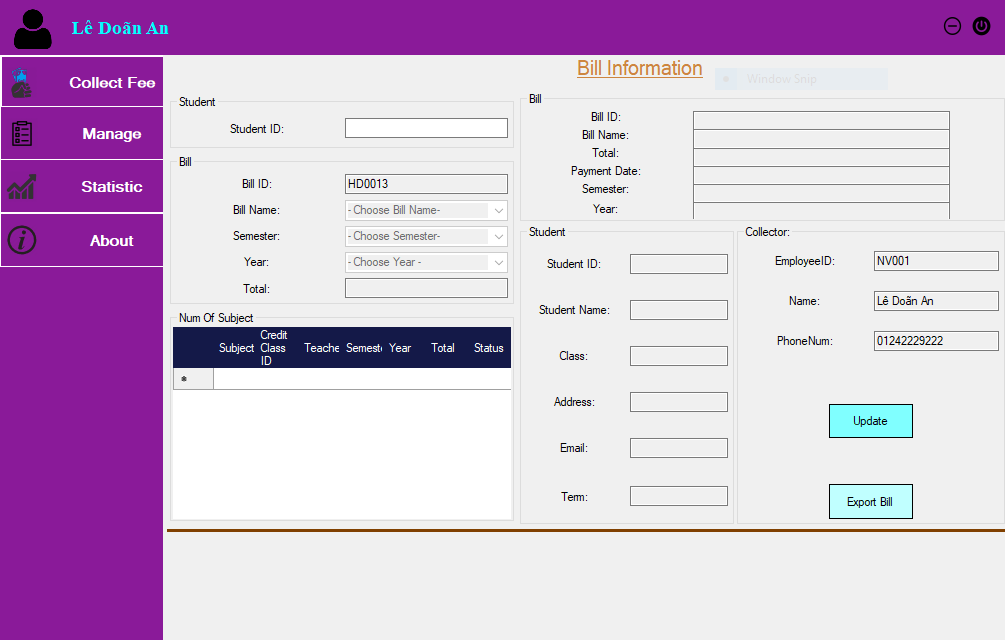
Figure 6.3: The login window

Description:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Note** | **Describe** |
| 1 | txtUserName | Textbox | None | Allow the user to enter username |
| 2 | txtPassword | Textbox | PasswordChar = \* | Allow the user to enter Password |
| 3 | btnLogIn | Button | None | When the user click the button, system check the values of texboxs, if information is correct, announces to user log in successfully, the system will hide this form and load FormMain |

6.2.2 Form Collect Tuition Fees

This section allows the Employee to collect tuition fees and view the payment information of each Student Periodically for each academic year when the Student pays fees; update student’s pay tuition information and billing information in the database; Print invoice information for students.

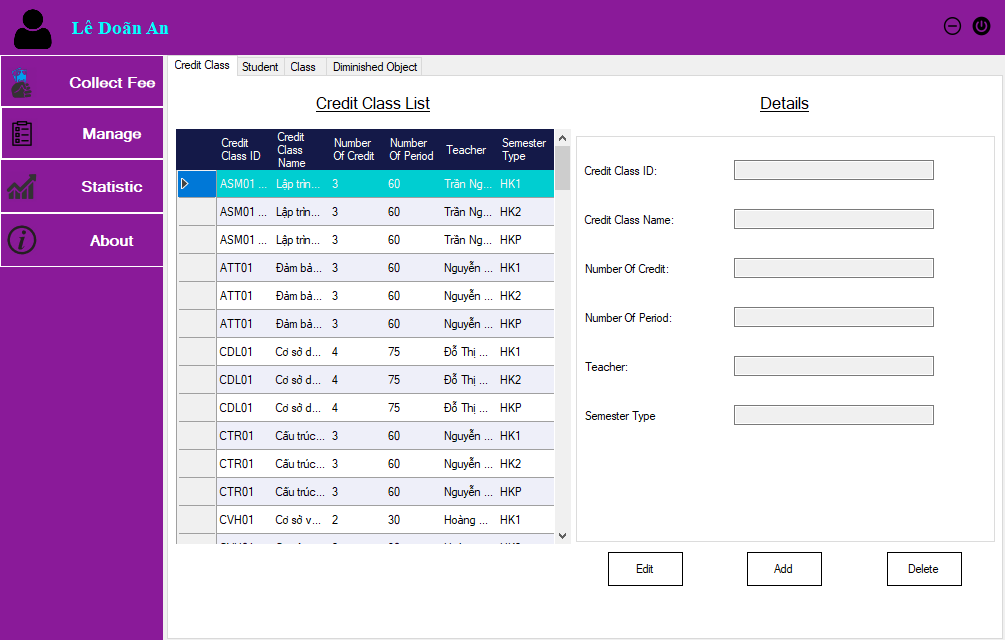


*Figure 6.4: Collect tuition fee interface and billing information*

Description:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Note** | **Describe** |
| 1 | txtStudentID | Textbox | None | Allow the employee to enter Student ID |
| 2 | txtBillID | Textbox | None | Bill ID is created automatically |
| 3 | cbbBillName | ComboBox | None | Allow employee choose bill name |
| 4 | cbbSemester | ComboBox | None | Allow employee choose semester |
| 5 | cbbYear | ComboBox | None | Allow employee choose year |
| 6 | txtTotal | Textbox | None | Display total money the student have to pay for credits registered in selected semester |
| 7 | txtBillID | Textbox | None | Display bill ID |
| 8 | txtBillName | Textbox | None | Display bill name |
| 9 | txtTotal | Textbox | None | Display total money |
| 10 | txtPaymentDate | Textbox | None | Display payment date |
| 11 | txtSemester | Textbox | None | Display semester |
| 12 | txtYear | Textbox | None | Display year |
| 13 | txtStudentID | Textbox | None | Display student ID |
| 14 | txtStudentName | Textbox | None | Display student name |
| 15 | txtClass | Textbox | None | Display student class |
| 16 | txtAddress | Textbox | None | Display student address |
| 17 | txtEmail | Textbox | None | Display student email |
| 18 | txtTerm | Textbox | None | Display student term |
| 19 | txtEmpID | Textbox | None | Display employee ID |
| 20 | txtEmpName | Textbox | None | Display employee name |
| 21 | txtEmpPhone | Textbox | None | Display employee phone number |
| 22 | btnUpdate | Button | None | When employee clicks on button Update, system will update bill information in database. |
| 23 | btnExport | Button | None | When employee clicks on button Export Bill, system will export bill information into pdf |

6.2.3 Form Manage Credit Class

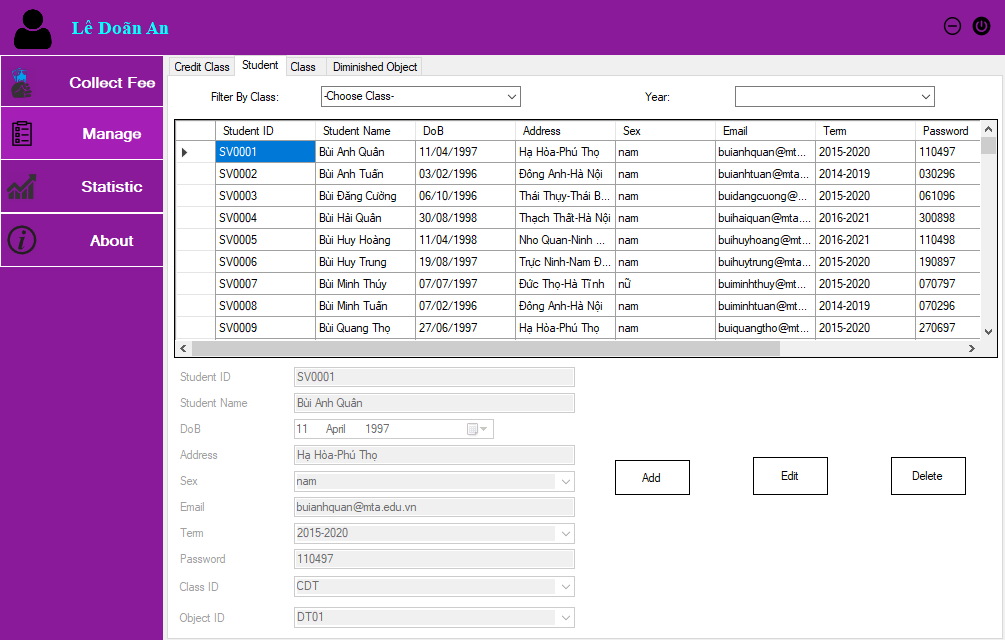


*Figure 6.5: Credit Class management interface*

Description:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Note** | **Describe** |
| 1 | dgvCreClassList | Data Grid View | None | Display credit class list |
| 2 | txtCreClassID | Textbox | None | Allow employee enter credit class ID |
| 3 | txtCreClassName | Textbox | None | Allow employee enter credit class name |
| 4 | txtNumOfCredit | Textbox | None | Allow employee enter number of credit |
| 5 | txtNumOfPeriod | Textbox | None | Allow employee enter number of period |
| 6 | txtTeacher | Textbox | None | Allow employee enter teacher name |
| 7 | txtSemType | Textbox | None | Allow employee enter semester type |
| 8 | btnEdit | Textbox | None | When employee clicks on button Edit, she can edit credit class information |
| 9 | txtAdd | Textbox | None | When employee clicks on button Add, system will add credit class information into database |
| 10 | txtDelete | Textbox | None | When employee clicks on button Delete, system will delete credit class information from database |

6.2.4 Form Manage Student

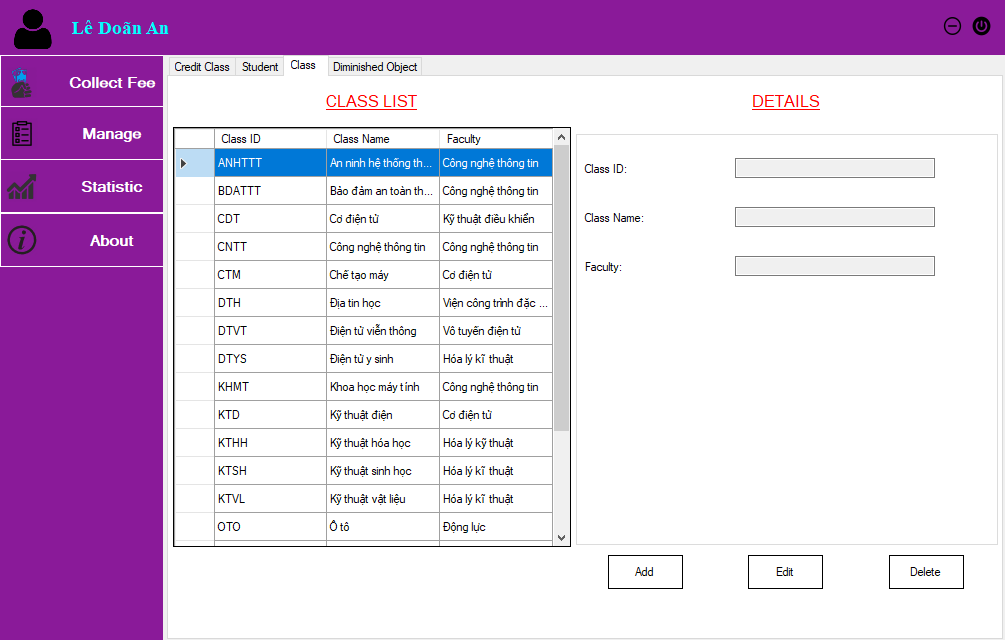


*Figure 6.6: Student Management Interface*

Description:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Note** | **Describe** |
| 1 | dgvCreClassList | Data Grid View | None | Display credit class list |
| 2 | txtCreClassID | Textbox | None | Allow employee enter credit class ID |
| 3 | txtCreClassName | Textbox | None | Allow employee enter credit class name |
| 4 | txtNumOfCredit | Textbox | None | Allow employee enter number of credit |
| 5 | txtNumOfPeriod | Textbox | None | Allow employee enter number of period |
| 6 | txtTeacher | Textbox | None | Allow employee enter teacher name |
| 7 | txtSemType | Textbox | None | Allow employee enter semester type |
| 8 | btnEdit | Textbox | None | When employee clicks on button Edit, she can edit credit class information |
| 9 | txtAdd | Textbox | None | When employee clicks on button Add, system will add credit class information into database |
| 10 | txtDelete | Textbox | None | When employee clicks on button Delete, system will delete credit class information from database |

6.2.5 Form Manage Class

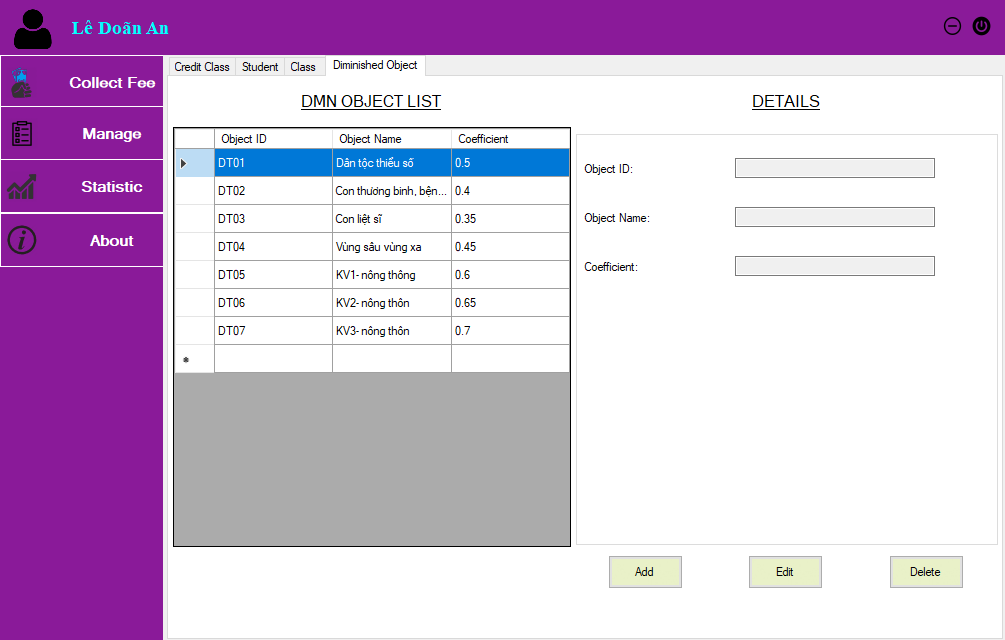


*Figure 6.7: Management Class Management Interface*

Description:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Note** | **Describe** |
| 1 | dgvClass | Data Grid View | None | Display class list |
| 2 | txtClassID | Textbox | None | Allow employee enter class ID |
| 3 | txtClassName | Textbox | None | Allow employee enter class name |
| 4 | txtFaculty | Textbox | None | Allow employee enter faculty |
| 5 | btnAdd | Button | None | If employee clicks on button Add, system will add class information into database |
| 6 | btnEdit | Button | None | If employee clicks on button Edit, she can edit class information |
| 7 | btnDelete | Button | None | If employee clicks on button Delete, system will delete class information from database |

6.2.6 Form Manage Diminished Object



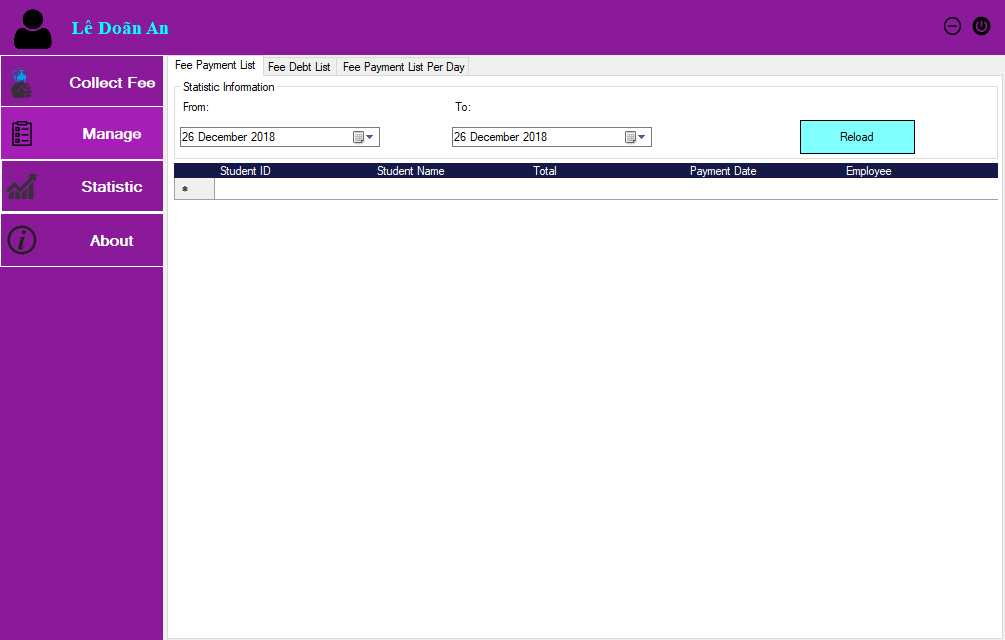
*Figure 6.8: Diminished Object Management interface*

Description:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Note** | **Describe** |
| 1 | dgvObject | Data Grid View | None | Display diminished object list |
| 2 | txtObjectID | Textbox | None | Allow employee enter object ID |
| 3 | txtObjectName | Textbox | None | Allow employee enter object name |
| 4 | txtCoefficient | Textbox | None | Allow employee enter coefficient |
| 5 | btnAdd | Button | None | If employee clicks on button Add, system will add diminished object information into database |
| 6 | btnEdit | Button | None | If employee clicks on button Edit, she can edit diminished object information |
| 7 | btnDelete | Button | None | If employee clicks on button Delete, system will delete diminished object information from database |

6.2.7 Form Statistic Fee Payment List

This section allows the Employee to list the payment schedule for a given day or payment list for a certain period of time.



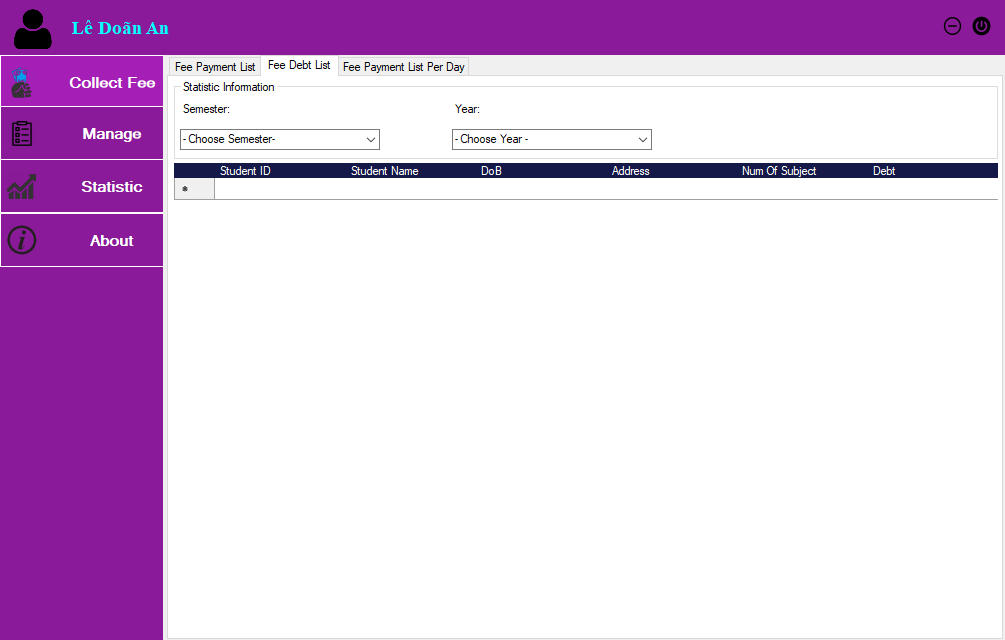
*Figure 6.9: Statistics of Charge List over a period of time*

Description:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Note** | **Describe** |
| 1 | dgvPaymentList | Data Grid View | None | Display fee payment list |
| 2 | dtpStart | Date Time Picker | None | Allow employee choose starting date to view statistic fee payment list |
| 3 | dtpEnd | Date Time Picker | None | Allow employee choose ending date to view statistic fee payment list |
| 4 | btnReload | Button | None | Reload fee payment list |

6.2.8 Form Statistic Fee Debt List

Allows Employee to list students' debt arrears for a semester of any one year



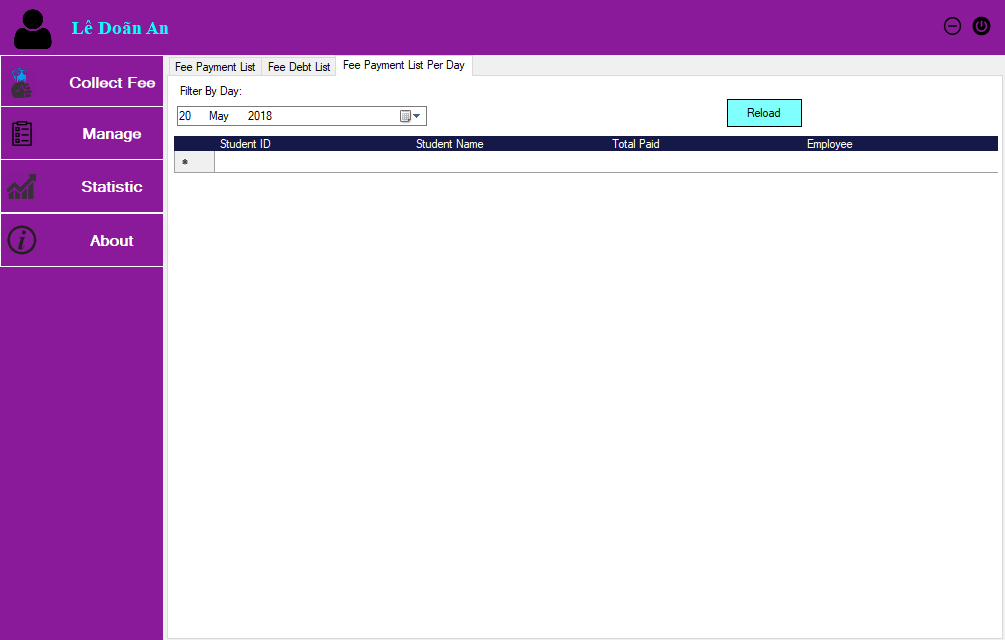
*Figure 6.10: Statistic list of fee debt about semester*

Description:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Note** | **Describe** |
| 1 | dgvDebtList | Data Grid View | None | Display fee debt list |
| 2 | cbbSemester | ComboBox | None | Allow employee choose semester |
| 3 | cbbYear | ComboBox | None | Allow employee choose year |

6.2.9 Form Statistic Payment List Per Day

Allows Employees to view students' payment of the day



*Figure 11: Statistics list of daily fee payment*

Description:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Note** | **Describe** |
| 1 | dgvListPerDay | Data Grid View | None | Display fee payment list per day |
| 2 | dtpDate | Date Time Picker | None | Allow employee choose date to view statistic |
| 3 | btnReload | Button | None | Reload fee payment list per day |

6.2.10 Form View Student Record

This section will allow students to check their personal information such as StudentID, Full Name, Address, Sex, Date of Birth, Course, Email, ...



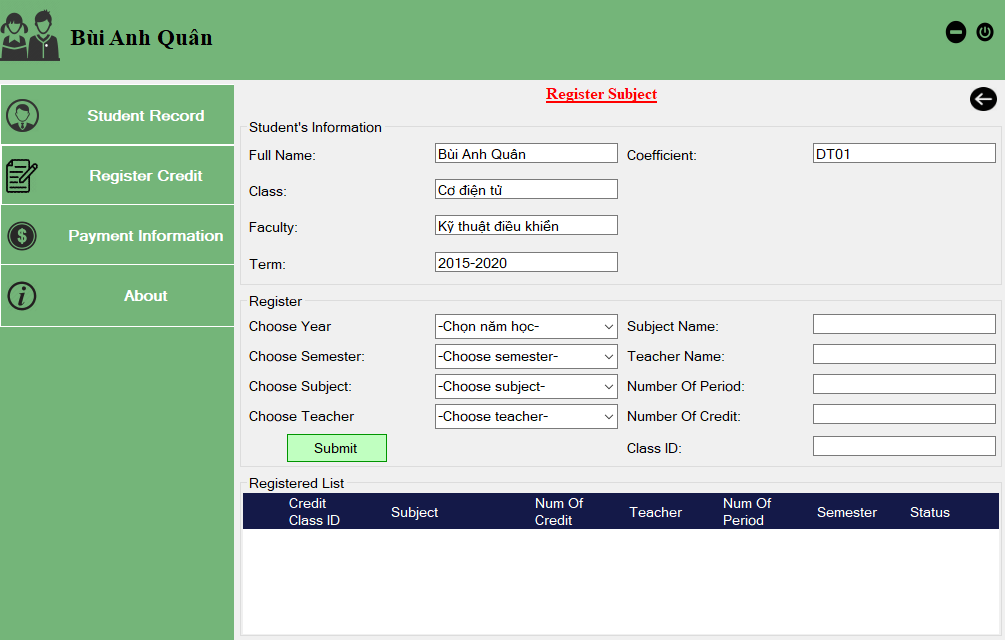
*Figure 6.12: Student Profile*

Description:

Display student’s information

6.2.11 Form Register Credit

Provides an interface for students to register subjects for each semester of them. In the register section, allows Student select the Year, Semester, Subjects, and Subject Teachers.



*Figure 6.13: Course registration interface*

When complete registering for the course, the course registration information will be updated automatically in the register table in the database.

Description:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Note** | **Describe** |
| 1 | dgvRegList | Data Grid View | None | Display registered credit class list |
| 2 | txtName | Textbox | None | Display student’s name |
| 3 | txtClass | Textbox | None | Display student’s class |
| 4 | txtFaculty | Textbox | None | Display student’s faculty |
| 5 | txtTerm | Textbox | None | Display student’s term |
| 6 | txtCoeff | Textbox | None | Display student’s diminished coefficient |
| 7 | cbbYear | ComboBox | None | Allow student choose year |
| 8 | cbbSem | ComboBox | None | Allow student choose semester |
| 9 | cbbSubject | ComboBox | None | Allow student choose subject |
| 10 | cbbTeacher | ComboBox | None | Allow student choose teacher |
| 11 | txtSubject | Textbox | None | Display subject name |
| 12 | txtTeacher | Textbox | None | Display teacher name |
| 13 | txtPeriod | Textbox | None | Display number of period |
| 14 | txtCredit | Textbox | None | Display number of credit |
| 15 | txtClass | Textbox | None | Display class ID |
| 16 | btnSubmit | Textbox | None | When student clicks on Submit button, she submits the selection |

6.2.12 Form View Payment Information

List the student's courses for one semester in a year and the status of the payment for subjects during that semester to provide information to the student about his or her fee payment status.



*Figure 6.14: Payment Information*

Descrition:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Note** | **Describe** |
| 1 | dgvSubInf | Data Grid View | None | Display payment information |
| 2 | cbbSem | ComboBox | None | Allow student to choose semester to view payment information |
| 3 | cbbYear | ComboBox | None | Allow student to choose year to view payment information |
| 4 | btnReload | Button | None | Reload payment information |
| 5 | btnExport | Button | None | Export payment information |

VII. IMPLEMENTATION

VIII. TESTING DOCUMENTS

8.1. Test case

|  |  |  |  |
| --- | --- | --- | --- |
| **Seq.No** | **Condition to be tested** | **Test data** | **Expected result** |
| **1** | Login | Enter valid username and valid password | User should be able to log in |
| **2** | Login | Enter valid username and invalid password | User should not be able to log in |
| **3** | Login | Enter invalid username and valid password | User should not be able to log in |
| **4** | Login | Enter invalid username and invalid password | User should not be able to log in |
| **5** | View student record | Login with right type of user | User should be able to view student record |
| **6** | View student record | Not login | User should not be able to view student record |
| **7** | View student record | Login fail | User should not be able to view student record |
| **8** | View fee information | Login with right type of user | User should be able to view fee information |
| **9** | View fee information | Not login | User should not be able to view fee information |
| **10** | View fee information | Login fail | User should not be able to view fee information |
| **11** | Register subject | Login with right type of user and choose all fields rightly | User should be able to register subject |
| **12** | Register subject | Not login | User should not be able to register subject |
| **13** | Register subject | Login fail | User should not be able to register subject |
| **14** | Register subject | There is no subject in the semester user chose | User should not be able to register subject |
| **15** | Register subject | There is no teacher who student wants to apply for the course | User should not be able to register subject |
| **16** | Update object information | Login with right type of user | User should be able to update object information |
| **17** | Update object information | Not login | User should not be able to update object information |
| **18** | Update object information | Login fail | User should not be able to update object information |
| **19** | View statistic | Login with right type of user | User should be able to view statistic |
| **20** | View statistic | Not login | User should not be able to view statistic |
| **21** | View statistic | Login fail | User should not be able to view statistic |
| **22** | Statistic tuition | Login with right type of user and choose all fields rightly | System should be able to statistic tuition |
| **23** | Statistic tuition | Not login | System should not be able to statistic tuition |
| **24** | Statistic tuition | Login fail | System should not be able to statistic tuition |
| **25** | Statistic tuition | Choose fields incorrectly | System should not be able to statistic tuition |
| **26** | Calculate tuition fee | Login with right type of user and register subjects correctly | System should be able to calculate tuition fee |
| **27** | Calculate tuition fee | Not login | System should not be able to calculate tuition fee |
| **28** | Calculate tuition fee | Login fail | System should not be able to calculate tuition fee |
| **29** | Calculate tuition fee | Register subjects fail | System should not be able to calculate tuition fee |
| **30** | Collect tuition fee | Login with right type of user and choose all fields rightly | User should be able to collect tuition fee |
| **31** | Collect tuition fee | Not login | User should not be able to collect tuition fee |
| **32** | Collect tuition fee | Login fail | User should not be able to collect tuition fee |
| **33** | Collect tuition fee | Choose fields incorrectly | User should not be able to collect tuition fee |

8.2. Test Scenario

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Seq.No** | **Requirement Name** | **Test Scenario(TS)** | | **Test Case(TC)** | |
| **TS Name** | **TS ID** | **TS Name** | **TC ID** |
| **1** | **Login**  **(UC01)** | ok | TS\_UC01.01 | 1. Enter valid UserName and valid Password | TS\_UC01.01\_TC01 |
| Cancel | TS\_UC01.02 | 1. Enter valid UserName and invalid Password | TS\_UC01.02\_TC01 |
| 2. Enter invalid UserName and valid Password | TS\_UC01.02\_TC02 |
| 3. Enter invalid UserName and invalid Password | TS\_UC01.02\_TC03 |
| **2** | **Register Subject**  **(UC02)** | Successful | TS\_UC02.01 | 1. Students register for each subject for each semester(Select a year, a subject, Choose a teacher who will teach the subject) | TS\_UC02.01\_TC01 |
| Fail | TS\_UC02.02 | 1. There is no subject to register in that semester | TS\_UC02.02\_TC01 |
| 2. There is no teacher who student wants to apply for the course | TS\_UC02.02\_TC02 |
| **3** | **View Student Record**  **(UC03)** | Successful | TS\_UC03.01 | 1. Check for personal information such as: Student ID, Full Name, Address, Gender, Date of Birth, Subject, Email, ... and completely match | TS\_UC03.01\_TC01 |
| Fail | TS\_UC03.02 | 1. StudentID information is incorrect | TS\_UC03.02\_TC01 |
| 2. Full Name information is incorrect | TS\_UC03.02\_TC02 |
| 3. Adress information is incorrect | TS\_UC03.02\_TC03 |
| 4. Date of birth information is incorrect | TS\_UC03.02\_TC04 |
| .................... |  |
| **4** | **Manage entity information**  **(UC04)** | Successful | TS\_UC04.01 | 1. Employees manage information about objects such as Credit Class, Student, Management Class, Diminished Object with basic functions such as Add, Edit, Delete for each object | TS\_UC04.01\_TC01 |
| Fail | TS\_UC04.02 | 1. Add invalid objects information | TS\_UC04.02\_TC01 |
| 2. Could not add objects information | TS\_UC04.02\_TC02 |
| **5** | **View Fee Information (UC05)** | Successful | TS\_UC05.01 | 1. List success the student's courses for one semester in a year and the status of the payment for subjects during that semester | TS\_UC05.01\_TC01 |
| Fail | TS\_UC05.02 | 1. Statistic wrong | TS\_UC05.02\_TC01 |
| 2. Can't login to view the fee information | TS\_UC05.02\_TC02 |
| **6** | **Statistic tuition**  **(UC06)** | Successful | TS\_UC06.01 | 1. Login with right type of user and choose all fields rightly | TS\_UC06.01\_TC01 |
| Fail | TS\_UC06.02 | 1. Statistic wrong | TS\_UC06.02\_TC01 |
| 2. Can't login to view the statistic | TS\_UC06.02\_TC02 |
| **7** | **Collect tuition fee**  **(UC07)** | Successful | TS\_UC07.01 | 1. The exact amount of the student's tuition is paid and the total amount of tuition paid by the student in that semester | TS\_UC07.01\_TC01 |
| Fail | TS\_UC07.02 | 1. Register subjects fail | TS\_UC07.02\_TC01 |
| 2. Login fail | TS\_UC07.02\_TC02 |
| 3. Not login | TS\_UC07.02\_TC03 |
| **8** | **Calculate Tuition fee**  **(UC08)** | Successful | TS\_UC08.01 | 1. Preferred tuition fee | TS\_UC08.01\_TC01 |
| Fail | TS\_UC08.02 | Preferred invalid money | TS\_UC08.02\_TC01 |
| **9** | **View Statistic**  **(UC09)** | Successful | TS\_UC09.01 | list success the payment schedule for a given day or payment list for a certain period of time. | TS\_UC09.01\_TC01 |
| Fail | TS\_UC09.02 | 1. Login fail | TS\_UC09.02\_TC01 |
| 2. Not login | TS\_UC09.02\_TC02 |

8.3. Requirement Traceability Matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Seq.No** | **UCID** | **TSID** | **TCID** | **Status** | **Defects** |
| **1** | UC01 | TS\_UC01.01 | TS\_UC01.01\_TC01 | Passed | No |
| **2** | UC01 | TS\_UC01.02 | TS\_UC01.02\_TC01 | Passed | No |
| **3** | UC01 | TS\_UC01.02 | TS\_UC01.02\_TC02 | Passed | No |
| **4** | UC01 | TS\_UC01.02 | TS\_UC01.02\_TC03 | Passed | No |
| **5** | UC02 | TS\_UC02.01 | TS\_UC02.01\_TC01 | Passed | No |
| **6** | UC02 | TS\_UC02.02 | TS\_UC02.02\_TC01 | Passed | No |
| **7** | UC02 | TS\_UC02.02 | TS\_UC02.02\_TC02 | Passed | No |
| **8** | UC03 | TS\_UC03.01 | TS\_UC03.01\_TC01 | Passed | No |
| **9** | UC03 | TS\_UC03.02 | TS\_UC03.02\_TC01 | Passed | No |
| **10** | UC03 | TS\_UC03.02 | TS\_UC03.02\_TC02 | Passed | No |
| **11** | UC03 | TS\_UC03.02 | TS\_UC03.02\_TC03 | Passed | No |
| **12** | UC03 | TS\_UC03.02 | TS\_UC03.02\_TC04 | Passed | No |
| **13** | UC04 | TS\_UC04.01 | TS\_UC04.01\_TC01 | Passed | No |
| **14** | UC04 | TS\_UC04.02 | TS\_UC04.02\_TC01 | Passed | No |
| **15** | UC04 | TS\_UC04.02 | TS\_UC04.02\_TC02 | Passed | No |
| **16** | UC05 | TS\_UC05.01 | TS\_UC05.01\_TC01 | Passed | No |
| **17** | UC05 | TS\_UC05.02 | TS\_UC05.02\_TC01 | Passed | No |
| **18** | UC05 | TS\_UC05.02 | TS\_UC05.02\_TC02 | Passed | No |
| **19** | UC06 | TS\_UC06.01 | TS\_UC06.01\_TC01 | Passed | No |
| **20** | UC06 | TS\_UC06.02 | TS\_UC06.02\_TC01 | Passed | No |
| **21** | UC06 | TS\_UC06.02 | TS\_UC06.02\_TC02 | Passed | No |
| **22** | UC07 | TS\_UC07.01 | TS\_UC07.01\_TC01 | Passed | No |
| **23** | UC07 | TS\_UC07.02 | TS\_UC07.02\_TC01 | Passed | No |
| **24** | UC07 | TS\_UC07.02 | TS\_UC07.02\_TC02 | Passed | No |
| **25** | UC07 | TS\_UC07.02 | TS\_UC07.02\_TC03 | Passed | No |
| **26** | UC08 | TS\_UC08.01 | TS\_UC08.01\_TC01 | Passed | No |
| **27** | UC08 | TS\_UC08.02 | TS\_UC08.02\_TC01 | Passed | No |
| **28** | UC09 | TS\_UC09.01 | TS\_UC09.01\_TC01 | Passed | No |
| **29** | UC09 | TS\_UC09.02 | TS\_UC09.02\_TC01 | Passed | No |
| **30** | UC09 | TS\_UC09.02 | TS\_UC09.02\_TC02 | Passed | No |