|  |
| --- |
| **Software design** |
| Student Management System |
| Version 1.0 |
| Prepared by class SE1316, Group 5  Hoàng Hồng Quang  Nguyễn Minh Hiếu  Đỗ Huy Hoàng  Đỗ Thành Đạt  April 2, 2020 |

**Contents**

[**Revision history** 3](#_Toc36733505)

[**1.** **Document description** 3](#_Toc36733506)

[**2.** **SRS overview** 3](#_Toc36733507)

[***2.1. List of system users*** 3](#_Toc36733508)

[***2.2. List of use cases*** 3](#_Toc36733509)

[***2.3. Use case diagram*** 3](#_Toc36733510)

[**2.3.1. Preliminary use case diagram** 3](#_Toc36733511)

[**2.3.2. Use case detail diagrams** 3](#_Toc36733512)

[***2.4. Activity diagrams of use case scenarios*** 3](#_Toc36733513)

[**2.4.1. Activity diagram for use case “”** 3](#_Toc36733514)

[**3.1.** **Table details** 3](#_Toc36733515)

[**3.1.1.** **Table name: Student** 3](#_Toc36733516)

[**3.1.2.** **Table name:** 4](#_Toc36733517)

[**3.1.3.** **Table name:** 4](#_Toc36733518)

[**3.1.4.** **Table name:** 4](#_Toc36733519)

[**3.1.5.** **Table name:** 5](#_Toc36733520)

[**4.** **Class diagram (p.61 – p70)** 5](#_Toc36733521)

[**4.1.** **List of classes in the Layers** 5](#_Toc36733522)

[**4.2.** **Modeling the Class Behaviors - sequence diagrams (p. 66, 77)** 5](#_Toc36733523)

[**4.2.1.** **Sequence diagram for use case “”** 6](#_Toc36733524)

[**5.** **User Interface Model Design** 6](#_Toc36733525)

[**5.1.** **User interface model design** 6](#_Toc36733526)

[**51.1. Screen “”** 6](#_Toc36733527)

[**51.2. Screen “”** 6](#_Toc36733528)

[**51.3. Screen “”** 6](#_Toc36733529)

[**51.4. Screen “”** 6](#_Toc36733530)

[**51.5. Screen “”** 6](#_Toc36733531)

[**5.2.** **Interface Flow Diagrams (p. 51)** 6](#_Toc36733532)

**Revision history**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Authors** | **Modifications** | **Version** |
| 02-April | Group 5 |  | 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. **Document description**

The number of students is increasing day by day, and the Managing Students required correctly and as fast as possible. We don't want to use paper or mind to save student information like StudentID, Student Name, Class, Gender, ... and the educator needs a program that could help them to store student pieces of information.

Student Management by C# - (SMS) helps the Educate owner to:

* Store Students, Teachers, Subjects, Classes, Marks information: Easy to find Student, Teachers, Subject, Classes, Marks information in need managing, comparing and the pieces of information are statistically. help to compare, select, Classify and help the Educator find the best Education solution to humans resourcing, Ground renting to runs their Schools, University.
* Adding more Student, Teacher, Subject, Classes, Marks Information: Yearly, new Students, Student, Teacher, Subject, Classes, Marks will be added. SMS helps these stuffs comfortable and minimizing the mistake taking ability. Marking and rating Student, Subject, Teacher, Syllabus will be much easier with SMS.
* Remove Student, Teacher, Subject, Classes, Marks Information: Student, Teacher, Subject, Classes, Marks Information will be old and these participators will be removed easily by SMS. This action helps the School's Database always in a good state and valuable.
* Edit Student, Teacher, Subject, Classes, Marks Information: Whenever the participants want to change their information, they can come easily and without any pen or eraser, saving our resources and Cost.
* Help Students move out of class easily by using a very Smart Algorithm of removal. Saved time, Cost, and valuation.
* Searching and filtering help the Educator skimming the information in 1 second, without any notebooks, pencil, minimized the ability of lost, forgot and save time.
* Our smart algorithm helps the owner calculate the rate of a student, who chooses a unique subject. Calculate an ability that Student study this subject.
* At the end of every semester, the system will automatically calculate the percent of a student who passed individual subject walls, helping the owner a lot while calculating the passing rate and failure rate.

Student Management makes things easier!

1. **SRS overview**

***2.1. List of system users***

1. Admin

***2.2. List of use cases***

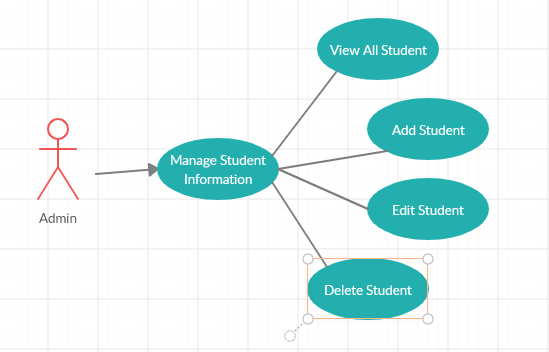
1. Manage Student Information
2. Change Class for Students
3. Manage Teacher Information
4. Manage Teaching Information
5. View Class
6. Adjust Marks

***2.3. Use case diagram***

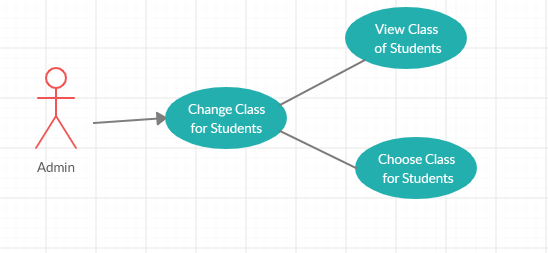
**2.3.1. Preliminary use case diagram**

**2.3.2. Use case detail diagrams**

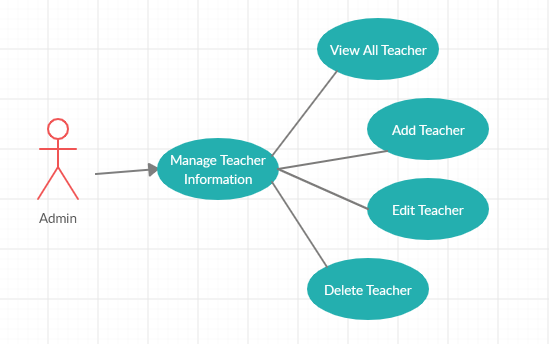
***2.3.2.1. Use case “Manage Student Information”***



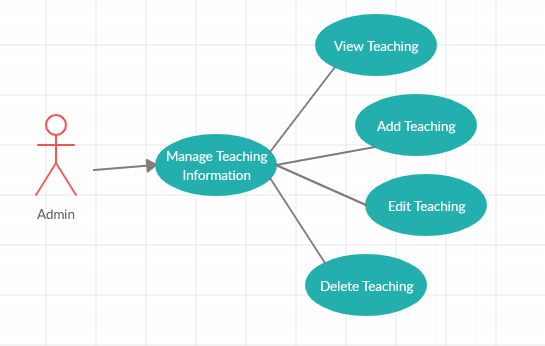
***2.3.2.2. Use case “Change Class for Students”***

******

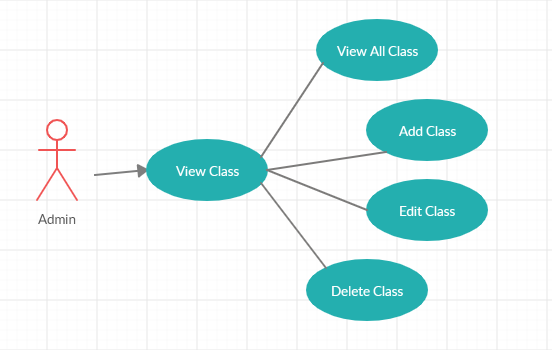
***2.3.2.3. Use case “Manage Teacher Information”***

******

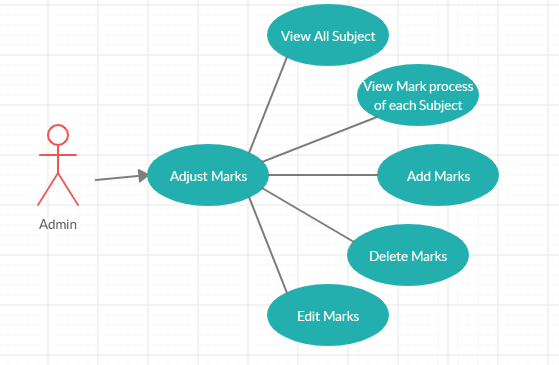
***2.3.2.4. Use case “Manage Teaching Information”***

******

***2.3.2.5. Use case “View Class”***

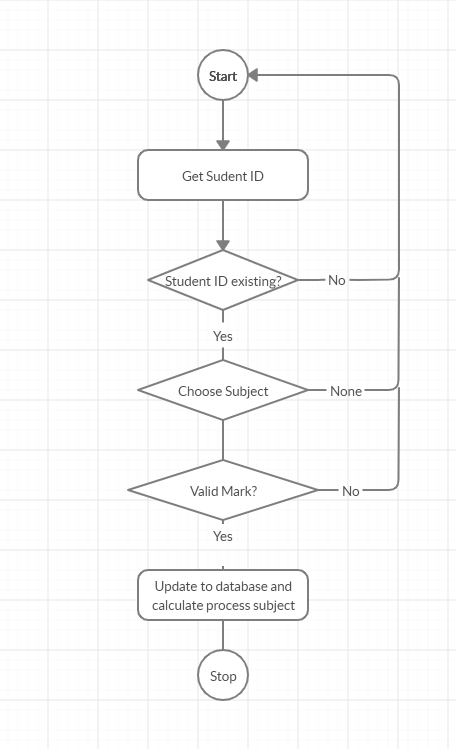
******

***2.3.2.6. Use case “Adjust Marks”***

******

***2.4. Activity diagrams of use case scenarios***

**2.4.1. Activity diagram for “Adjust Marks”**

****

Check reservations for this book?

Borrow (Update copy type & reservation status)

Reserved by any other user or Borrower is not the first in reservation

Not reserved by any other user or Borrower is the first in reservation

Available

1. **Table details**

Note: Attribute with underline is Primary key

* + 1. **Table name: Student**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Note** |
| StudentID | Integer | Unlimited | Primary key |
| Name | Nvarchar | 500 | Not null |
| Gender | Boolean | Unlimited | Not null |
| Date Of Birth | Date | Unlimited | Allows null |
| Address | Nvarchar | 500 | Allows null |
| Phone | Nvarchar | 500 | Allows null |
| Email | Nvarchar | 500 | Allows null |

* + 1. **Table name: Class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Note** |
| ClassID | Integer | Unlimited | Not null |
| ClassName | Nvarchar | 500 | Not null |

* + 1. **Table name: Mark**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Note** |
| MarkID | Integer | Unlimited | Not null |
| StudentID | Integer | Unlimited | Not null |
| SubjectID | Integer | Unlimited | Not null |
| Test1 | Integer | Unlimited | Allows null |
| Test2 | Integer | Unlimited | Allows null |
| Test3 | Integer | Unlimited | Allows null |
| FinalTest | Integer | Unlimited | Allows null |

* + 1. **Table name: Study**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Note** |
| StudyID | Integer | Unlimited | Not null |
| StudentID | Integer | Unlimited | Not null |
| ClassID | Integer | Unlimited | Not null |

* + 1. **Table name: Teach**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Note** |
| ID | Integer | Unlimited | Not null |
| TeacherID | Integer | Unlimited | Not null |
| SubjectID | Integer | Unlimited | Not null |

* + 1. **Table name: Teacher**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Note** |
| TeacherID | Integer | Unlimited | Not null |
| Name | Nvarchar | 500 | Allows null |
| Gender | Nchar | 10 | Allows null |
| Date Of Birth | Date | Unlimited | Allows null |
| Address | Nvarchar | 500 | Allows null |
| Phone | Nvarchar | 500 | Allows null |
| Email | Nvarchar | 500 | Allows null |

* + 1. **Table name: Subject**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Note** |
| SubjectID | Integer | Unlimited | Not null |
| SubjectName | Nvarchar | 500 | Allows null |
| SubjectCode | Nchar | 10 | Allows null |

* + 1. **Table name: Subject**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Size** | **Note** |
| ID | Integer | Unlimited | Not null |
| ClassID | Integer | Unlimited | Not null |
| TeacherID | Integer | Unlimited | Not null |

1. **Class diagram (p.61 – p70)**
   1. **List of classes in the Layers**

Layer GUI

Layer DA

Student Database

TeacherDAO

TeacherGUI

ClassGUI

SubjectGUI

StudentGUI

Note: Allow to use

BL: Business logic DT: Data transfer

DA: Data access

Layer DT

Teacher

MainFormGUI

About

SubjectDetailGUI

TeacherSubjectManagement

Student

Class

Subject

Teacher Class

Class

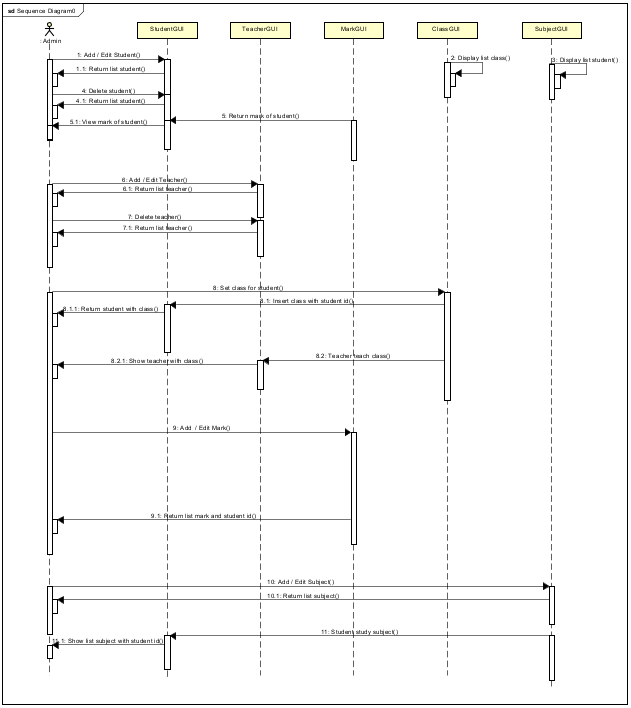
DAO

ClassDAO

StudentDAO

SubjectDAO

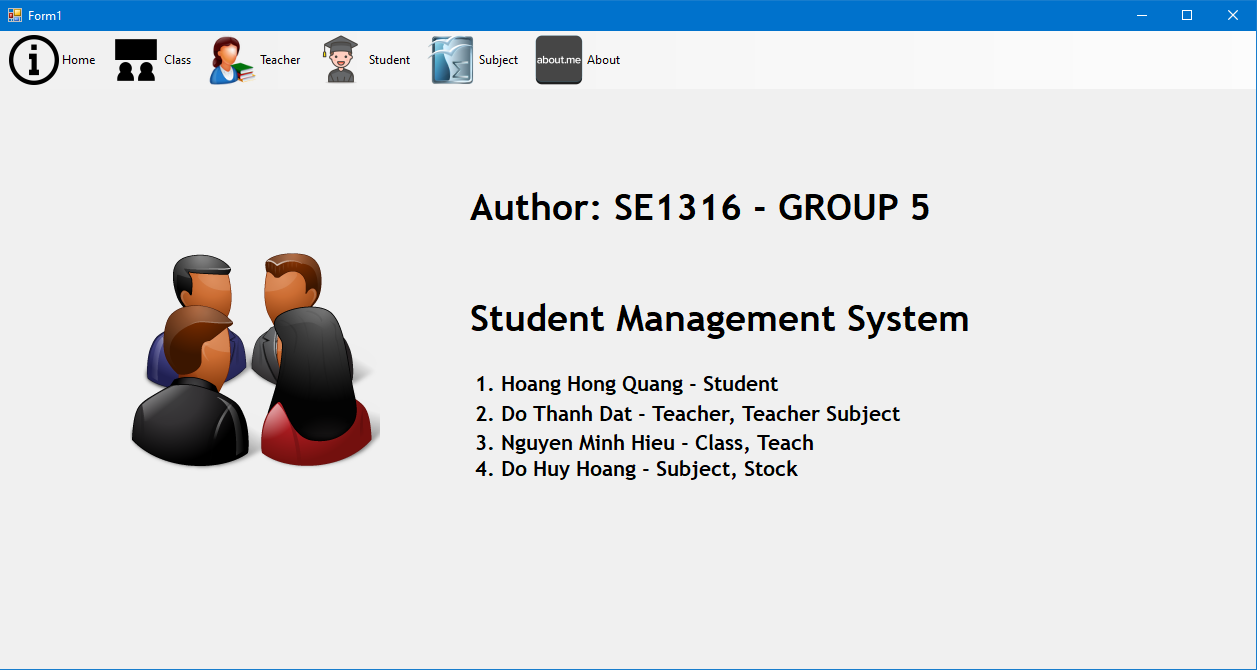
* 1. **Modeling the Class Behaviors - sequence diagrams (p. 66, 77)**

****

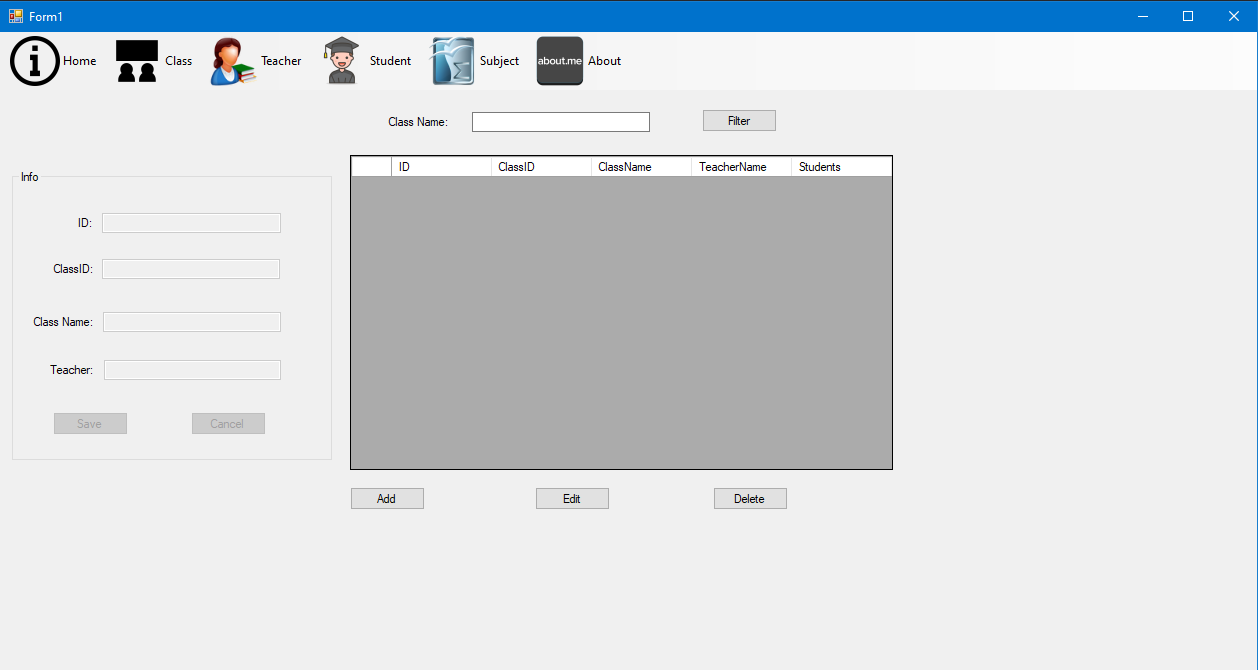
**5.User Interface Model Design**

* 1. **User interface model design**

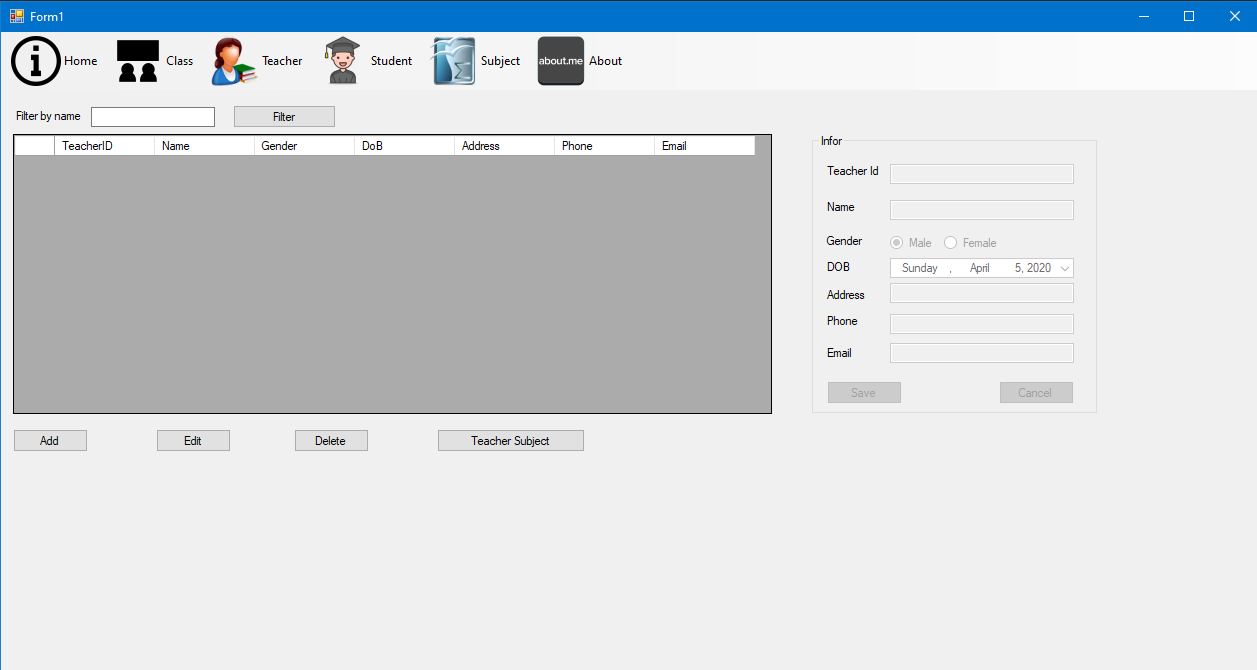
**51.1. Screen “Home”**



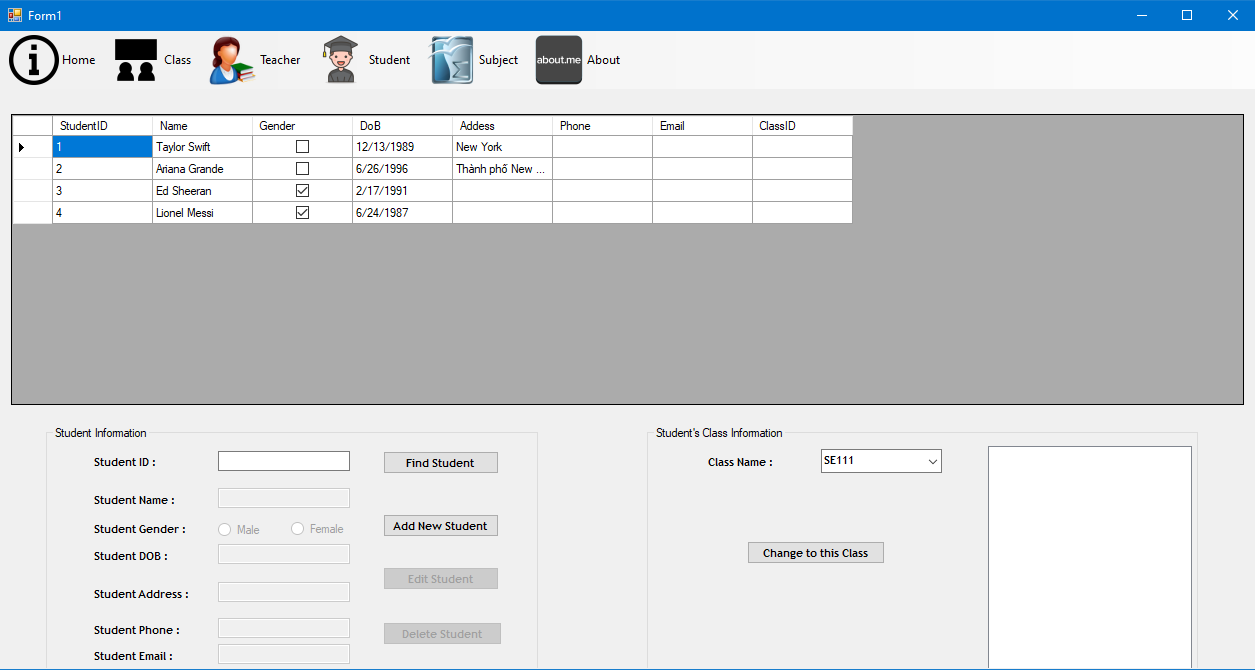
**51.2. Screen “Class”**



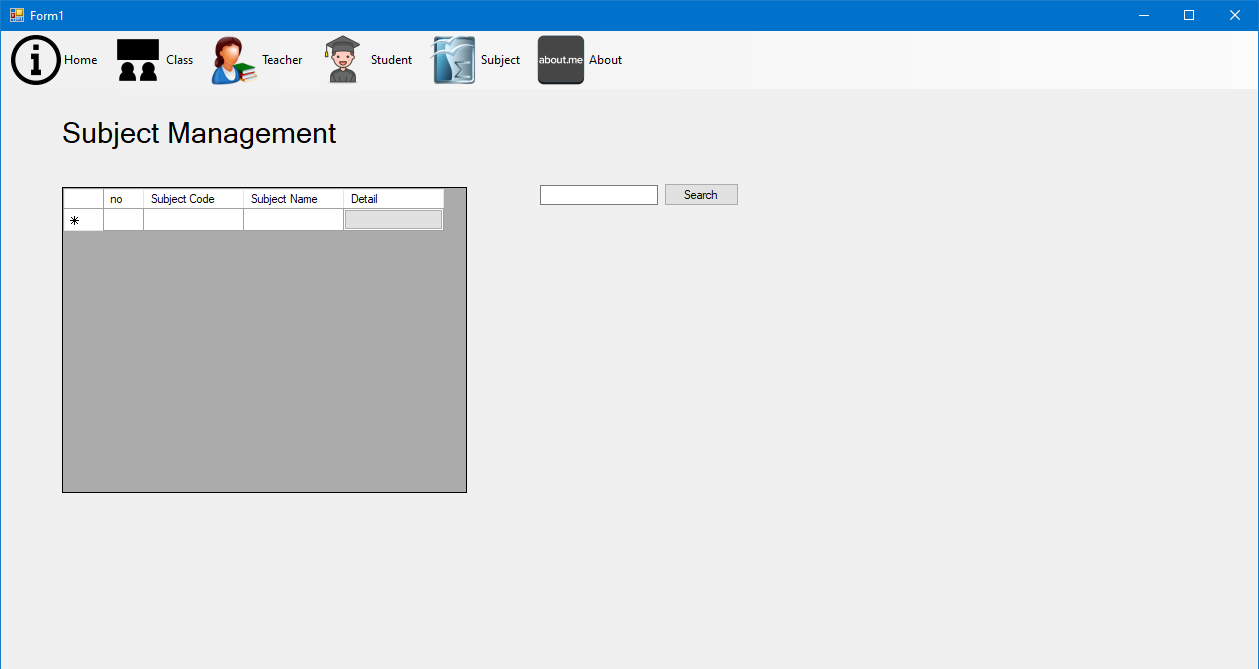
**51.3. Screen “Teacher”**



**51.4. Screen “Student”**



**51.5. Screen “Subject”**



* 1. **Interface Flow Diagrams (p. 51)**

Main form (main menu)

Subject

Class Teach

Teacher

Student

[Click]

[Click]

[Click]

[Click]

Home

About

Student Class Management

[Click]

[Click]

Teacher Subject Management

Mark management

[Click]

[Click]

[Click]

Class

[Click]