VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY

UNIVERSITY OF INFORMATION TECHNOLOGY

COMPUTER SCIENCE DEPARTMENT

PROJECT REPORT

BUILDING STUDENT MANAGEMENT SOFTWARE

SE104.M22.PMCL

LECTURER: M.Sc. Mai Trong Khang

AUTHORS:

Pham Hoang Phuc	–	20520278
Do Thi Thu Trang	–	20520816
Le Tan Loc	–	20521546
Nguyen Thi Ngoc Nga	–	20521641

HO CHI MINH CITY, 06/2022

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1. Topic

- Unified Topic: Student List Management.
- Development Model: Three – Layer Model.

1.2. Actual survey

Currently, we are living in a developed society, not only the education and training of students more and more stressful, but the parents' demand for their children to go to school is also increasing after each year. Along with the development of education and the increase in the number of students, the classical way of managing and storing student information is causing a lot of disadvantages and a lot of effort. And when the explosion of services on the Internet in our country is getting bigger and bigger, the emergence of a student management software to help reduce the labor of teachers in managing and helping students' parents and students can easily track, quickly and accurately access information, look up learning status at any time, which is very necessary and possible at this time.

1.3. Defining requirements

Based on the actual survey, it can be found that there are jobs that take a lot of time but are still repeated every year such as: receiving student applications for admission to school every year, dividing classes for students so that balanced, the requirements for the management of the subjects for the students in the school, the arrangement of the timetable, the management and storage of the scores of the subjects for the students, etc. Usually, these jobs just stop at using Excel or some other tool to execute, manage and store the necessary data. Therefore, building a software “Student Management” is an extremely necessary issue.

CHAPTER 2: SOFTWARE REQUIREMENTS ANALYSIS AND MODELING

2.1. Introduction of the software and description of the main work process

2.1.1. Software introduction

To make a flawless and fully functional student management software, it will take a lot of work and build a lot of features for the software. Therefore, within the framework of the software that the team builds, there will be a brief overview of highly complex jobs and will focus on performing some of the main tasks that are:

1. Managing the admission of students - the condition is that the student must be in between the ages of 15 and 20 years old.

2. Implement class list builder function.

3. Building information management functions and learning outcomes.

4. Function to input and store the scoreboard of subjects.

5. Function to make summary report.

Besides, the software also has other functions, allowing users to change some default data such as: size of age that can be admitted to the school, maximum quantity of one class, number of classes or name of each class, number of subjects or name of each subject, and finally the passing score.

2.1.2. Implementation process

2.1.2.1. Accepting students

The admission record for each student will include information such as the student's name, gender, date of birth, address and email.

2.1.2.2. Make a class list

A class will include name, quantity and information for each student in that class.

2.1.2.3. Look up student information

The information to look up for any student will include the student's full name, affiliated class and the student's semester I and II average scores.

2.1.2.4. Manage course transcripts

Store and update 15-minute scores, 1-period scores and average scores of subjects by name of each student in each class and each semester.

2.1.2.5. Make a summary report

The preparation of the final report will have 2 parts, namely, the final report of the course and the final report of the semester. Each report section will contain class information, the quantity of that class along with the number of students passing and the percentage of students passing that class.

2.2. Classification of requirements

The student management software that our team builds will fully meet the following requirements:

2.2.1. Major requirements

No.	Request name	Form	Rule	Note
1	Accepting students	BM1	QĐ1	
2	Make a class list	BM2	QĐ2	
3	Look up students	BM3		
4	Get the subject score sheet	BM4	QĐ4	
5	Make a summary report	BM5	QĐ5	
6	Change the rules		QĐ6	

⇨ Professional requirements need to ensure correctness.

2.2.2. Evolution requirements

No.	Major	Parameters need to be changed	The value domain needs to be changed
1	Accepting students	Age of students admitted to the school	The oldest age, the youngest age
2	Make a class list	The structure of each class and the number of classes in the school	Maximum quantity of classes, number and names of classes in the school
3	Changing regulations on subjects	Subjects	Number of subjects, names of subjects
4	Changes to the final report	Subject scores	Passing score

2.2.3. Efficiency requirements

No.	Major	Work	Processing speed	Storage capacity	Note
1	Accepting students	Stable, efficient	Right now		
2	Make a class list	Stable, efficient	Right now		
3	Look up students	Stable, reliable	Right now		
4	Get the subject score sheet	Stable, efficient	Right now	Great	
5	Make a summary report	Stable, efficient, accurate	Right now		
6	Change the rules	Stable, efficient	Instantly		

2.2.4. Handy requirements

No.	Major	Ease of learning	Ease of use	Note
1	Accepting students	There are clear instructions specification	Easy	
2	Make a class list	There are clear instructions specification	Normal	
3	Look up students	There are clear instructions specification	Very easy	
4	Get the subject score sheet	There is a guide specification	Easy	
5	Make a summary report	There are clear instructions specification	Easy	
6	Change the rules	There are clear instructions specification	Normal	

⇨ Handy requirements also require the intuitiveness of the interface and most importantly, ease of use for the user.

2.2.5. Compatibility Requirements

No.	Major	Related objects	Note
1	Input data	Excel, SQL Server	
2	Data saving	SQL Server	
3	UI Design	Winform	
4	Output information	Winform	

2.2.6. Security requirements

No.	Major	System management	Administrators	Ministry	Other
1	Decentralization	X			
2	Receive			X	
3	Arrange class			X	
4	Search		X	X	X
5	Change of reception rules		X		
6	Change of class arrangement rules		X		
7	Changing the subject structure		X		

2.2.7. Safety requirements

No.	Major	Subject	Note
1	Rehabilitate	Student profile deleted	
2	Update	Student profile in school	
3	Real cancellation	Student profile deleted	
4	Deletion is not allowed	Class when there are students	

2.2.8. Technological requirements

No.	Request	Detailed description	Note
1	Easy to fix	Determine the average error in 10 minutes	Does not affect other functions
2	Easy to maintain	Add new functions quickly	Does not affect other functions
3	Reuse	Build management software in the best possible time	Along with the requirements
4	Easy to carry	Change to a new database management system in up to 2 days	Along with the requirements

2.3. Responsibility table for each type of requirement

2.3.1. Responsibility table for major requirements

No.	Major	User	Software	Note
1	Accepting students	Provide student profile information	Check the rules and save the student record information	Can edit, delete added profile information
2	Make a class list	Provide information class list	Check the rules and save the information in the class list	Allows deleting class, adding class and transferring class for pupil
3	Look up students	Provide information students need to find	Search and display student information that meets the requirements	
4	Enter subject transcripts	Enter the subject, semester, student ID, test score	Check the rules and save the course transcripts	Allows editing, adding, deleting points entered

2.3.2. Responsibility table for evolution requirements

No.	Major	User	Software	Note
1	Changes in student admission rules	Indicate minimum and maximum age	Update new value	
2	Change of class arrangement rules	Indicate the maximum quantity of classes, the number of classes and the names of the classes in the school	Update new value	
3	Change of course regulations	Indicate the number of subjects, the names of the subjects	Update new value	
4	Change the rules for making scoreboards	Indicate the passing score of the subject	Update new value	

2.3.3. Responsibility table for efficiency requirements

No.	Major	User	Software	Note
1	Accepting students		Follow the request properly	
2	Make a class list	Prepare class list in advance	Follow the request properly	
3	Look up students		Follow the request properly	

2.3.4. Responsibility table for handy requirements

No.	Major	User	Software	Note
1	Accepting students	Read the user manual	Follow the request properly	
2	Make a class list	Read the user manual	Follow the request properly	
3	Look up students		Follow the request properly	

2.3.5. Responsibility table for compatibility requirements

No.	Major	User	Software	Note
1	Get a list of students	Prepare Excel file with form structure	Follow the request properly	
2	Export class list	Install Winfax software and tell the class to export the list	Follow the request properly	
3	Get the subject score sheet	Prepare Excel file with form structure	Follow the request properly	
4	Output summary report	Install Winfax software	Follow the request properly	

2.3.6. Responsibility table for security requirements

No.	Major	User	Software	Note
1	Administration	Indicate new users and permissions	Record and do it right	
2	Administrators	Provide login account	Record and do it right	
3	Ministry	Provide login account	Record and do it right	

2.3.7. Responsibility table for safety requirements

No.	Major	User	Software	Note
1	Rehabilitate	Indicate student records to be restored	Rehabilitate	
2	Real cancellation	Indicate student records to be restored	Real cancellation	
3	Deletion is not allowed		Follow the request properly	

2.3.8. Responsibility table for technological requirements

No.	Major	Detailed description	Note
1	Easy to fix	Determine the average error in 15 minutes	When debugging one function does not affect the other functions
2	Easy to maintain	Add new functions fast	Does not affect existing functions
3	Reuse	Build software to manage students at grade 1 and 2 in 3 days	Along with the requirements
4	Easy to carry	Change to a new database management system up to 3 days	Along with the requirements

2.4. Data flow diagram for each type of requirement

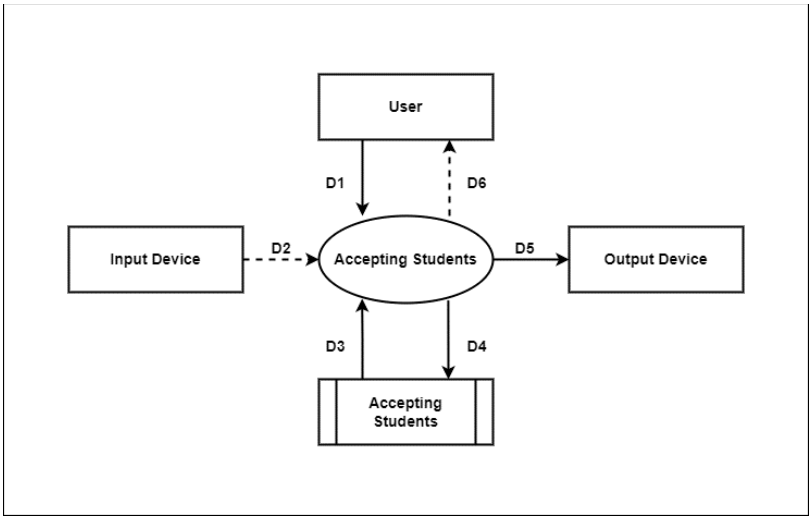
2.4.1. Data flow diagram for accepting students request

– Form 1 and Rule 1:

BM1:	**Student Records**
Full name:……	Gender:……
Date of Birth:……	Address:……
Email:…….	

QĐ1: Student age from 15 to 20.

– Data flow diagram:



- Data streams:
  + **D1:** Full name, Date of birth, Gender, Email, Address
  + **D2:** None
  + **D3:** Maximum age, Minimum age
  + **D4:** D1
  + **D5:** D4
  + **D6:** None

- Algorithm:
  + **Step 1.** Get D1 from the user.
  + **Step 2.** Connect to the database.
  + **Step 3.** Read D3 from auxiliary memory.
  + **Step 4.** Calculate student age.
  + **Step 5.** Check the minimum age requirement.
  + **Step 6.** Check the maximum age requirement.
  + **Step 7.** If all of the above conditions are not satisfied, go to step 10.
  + **Step 8.** Save D4 to secondary memory.
  + **Step 9.** Export D5 to printer (if required).
  + **Step 10.** Close the database connection.
  + **Step 11.** Finish.

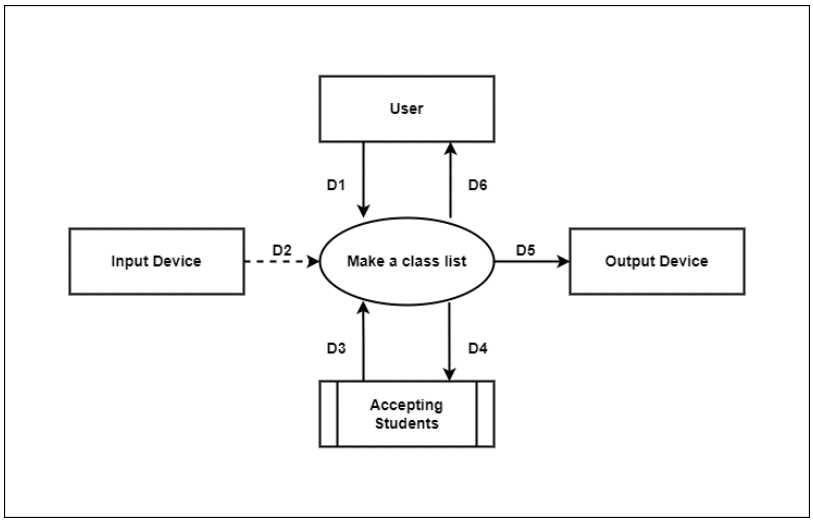
2.4.2. Data flow diagram for make a class list request

- Form 2 and Rule 2:

BM2:	**Class List**			
Class:……..		Quantity:……		
No.	**Full Name**	**Sex**	**Year of Birth**	**Address**
1				
2				

QĐ2: There are 3 grades (10, 11, 12). Grade 10 has 4 classes (10A1, 10A2, 10A3, 10A4). Grade 11 has 3 classes (11A1, 11A2, 11A3). Grade 12 has 2 classes (12A1, 12A2). Each class has no more than 40 students.

– Data flow diagram:



– Data streams:
  + **D1:** Class, Quantity
  + **D2:** None
  + **D3:** List of students, List of classes, Maximum number of students
  + **D4:** D1
  + **D5:** D1 + Student List
  + **D6:** D5

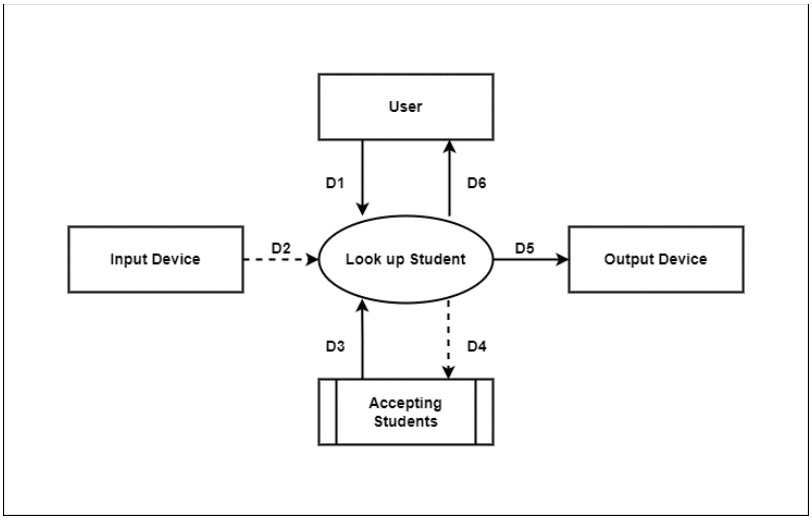
– Algorithm:
  + **Step 1.** Get D1 from the user.
  + **Step 2.** Connect to the database.
  + **Step 3.** Read D3 from auxiliary memory.
  + **Step 4.** Check that “Grade” (D1) is in “Class List” (D3).
  + **Step 5.** Calculate class quantity.
  + **Step 6.** Check the maximum class quantity.
  + **Step 7.** If all of the above conditions are not satisfied, go to step 11.
  + **Step 8.** Save D4 to secondary memory.
  + **Step 9.** Export D5 to the printer (if required).
  + **Step 10.** Return D6 to the user.
  + **Step 11.** Close the database connection.
  + **Step 12.** Finish.

2.4.3. Data flow diagram for look up student request

– Form 3:

BM3:	Student List			
No.	**Full Name**	**Class**	**1st Semester GPA**	**2nd Semester GPA**
1				
2				

– Data flow diagram:



– Data streams:
  + **D1:** Full Name, Class
  + **D2:** None
  + **D3:** Full name, Class, GPA of the first semester, GPA of the second semester of the student meets the requirements
  + **D4:** None
  + **D5:** D3
  + **D6:** D5

– Algorithm:
  + **Step 1.** Get D1 from the user.
  + **Step 2.** Connect to the database.
  + **Step 3.** Read D3 from auxiliary memory.
  + **Step 4.** Export D5.
  + **Step 5.** Return D6 to the user.

+ **Step 6.** Close the database connection.
+ **Step 7.** Finish.

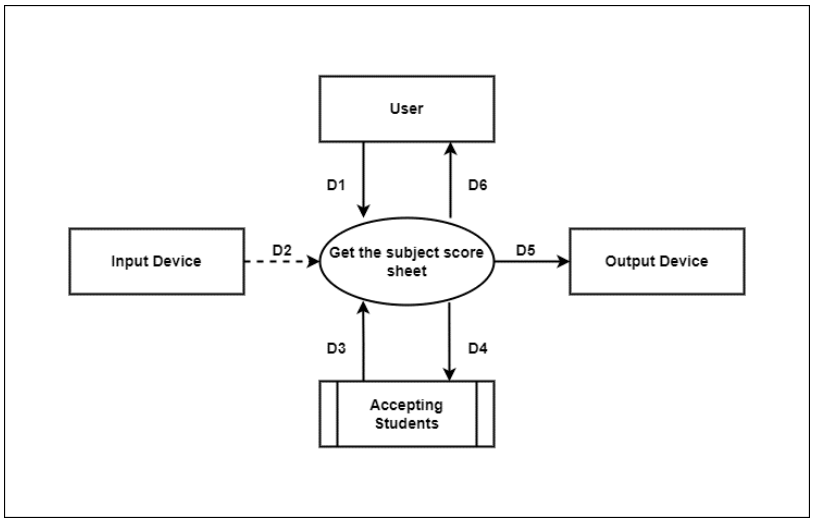
2.4.4. Data flow diagram for get the subject score sheet request

– Form 4 and Rule 4:

BM4:	**Subject Score Sheet**			
Class:……		Subject:……		
Semester:……				
No.	**Full Name**	**15-minute Score**	**1-period Score**	**Average Score**
1				
2				

QĐ4: There are 2 semesters (I, II). There are 9 subjects (Math, Physics, Chemistry, Biology, History, Geography, Literature, Civic Education, Physical Education). $0 \leq Score \leq 10$

– Data flow diagram:



– Data streams:
+ **D1:** Class, Subject, Semester, Student ID, 15-minute Score, 1-period Score
+ **D2:** None
+ **D3:** List of students
+ **D4:** D1 + GPA

+ **D5:** D4
+ **D6:** D5

- Algorithm:
  + **Step 1.** Get D1 from the user.
  + **Step 2.** Connect to the database.
  + **Step 3.** Read D3 from auxiliary memory.
  + **Step 4.** Check that "Subject" (D1) belongs to "Subjects List" (D3).
  + **Step 5.** Check if "Semester" (D1) is in "Semesters List" (D3).
  + **Step 7.** Check "15-minute Score" (D1) is within "0 – 10" range.
  + **Step 8.** Check if "1-period Score" (D1) is in the range "0 – 10".
  + **Step 9.** If all of the above conditions are not met, go to step 14.
  + **Step 10.** Calculate "Grade Point Average" from secondary memory.
  + **Step 11.** Save D4 to secondary memory.
  + **Step 12.** Export D5 to the printer (if required).
  + **Step 13.** Return D6 to the user.
  + **Step 14.** Close the database connection.
  + **Step 15.** Finish.

2.4.5. Data flow diagram for make a final report request

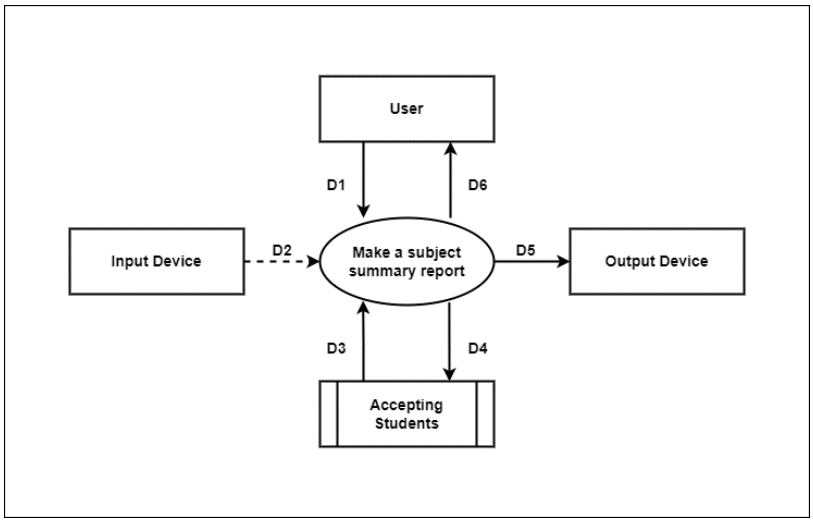
2.4.5.1. Data flow diagram for make a subject summary report request

- Form 5.1 and Rule 5:

BM5.1:	**Subject Summary Report**			
Subject:……		Semester:……		
No.	**Class**	**Quantity**	**Number of Passes**	**Ratio**
1				
2				

QĐ5: Students pass the subject/pass if the average score is ≥ 5.

– Data flow diagram:



– Data streams:
  + **D1:** Subject , Semester
  + **D2:** None
  + **D3:** Class List, Subject Transcript, Passing Score.
  + **D4:** D3
  + **D5:** D4
  + **D6:** D5

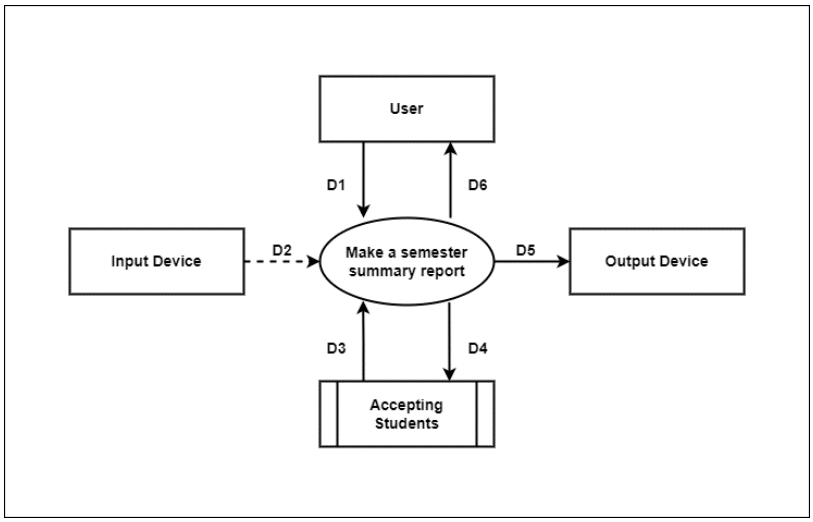
– Algorithm:
  + **Step 1.** Get D1 from the user.
  + **Step 2.** Connect to the database.
  + **Step 3.** Read D3 from auxiliary memory.
  + **Step 4.** Calculate the number of passes for each class of the subject.
  + **Step 5.** Calculate the pass rate for each class of the subject.
  + **Step 6.** Save D4 to secondary memory.
  + **Step 7.** Export D5 to the printer (if required).
  + **Step 8.** Return D6 to the user.
  + **Step 9.** Close the database connection.
  + **Step 10.** Finish.

2.4.5.2. Data flow diagram for make a semester summary report request

- Form 5.2:

BM5.2:	**Semester End Report**			
Semester:……				
No.	**Class**	**Quantity**	**Number of Passes**	**Ratio**
1				
2				

- Data flow diagram:



- Data streams:
  + **D1:** Semester
  + **D2:** None
  + **D3:** Class List, Subject Transcript, Passing Score
  + **D4:** Class List, Number of Passes, Ratio
  + **D5:** D4
  + **D6:** D5

- Algorithm:
  + **Step 1.** Get D1 from the user.
  + **Step 2.** Connect to the database.
  + **Step 3.** Read D3 from auxiliary memory.
  + **Step 4.** Calculate the number of passes for each class in the semester.
  + **Step 5.** Calculate the pass rate for each class in the semester.

+ **Step 6.** Save D4 to secondary memory.
+ **Step 7.** Export D5 to the printer (if required).
+ **Step 8.** Return D6 to the user.
+ **Step 9.** Close the database connection.
+ **Step 10.** Finish.

2.4.6. Data flow diagram for change the rules request

Rule 6:

QĐ6: Users can change the rules as follows:

+ QĐ1: Change the minimum age, maximum age.

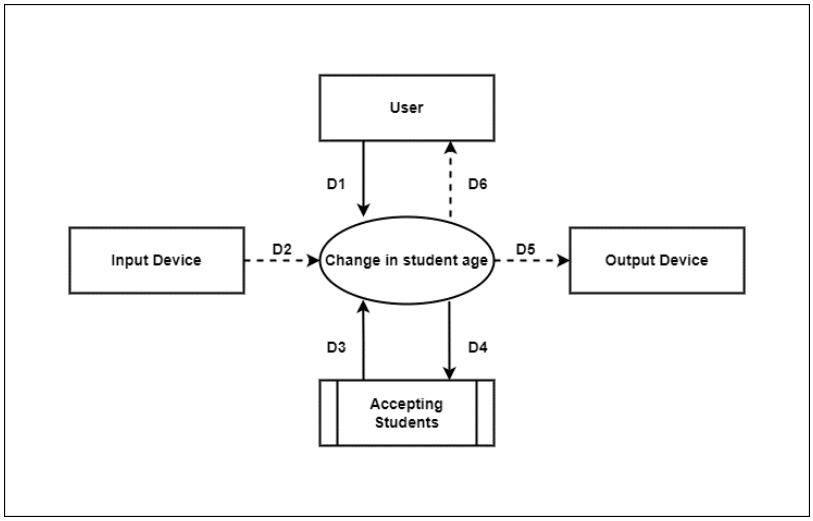
+ QĐ2: Change the maximum quantity of classes, change the number and names of classes in the school.

+ QĐ4: Change number and name of subjects.

+ QĐ5: Change of passing points.

2.4.6.1. Data flow diagram for change in student age request

- Data flow diagram:



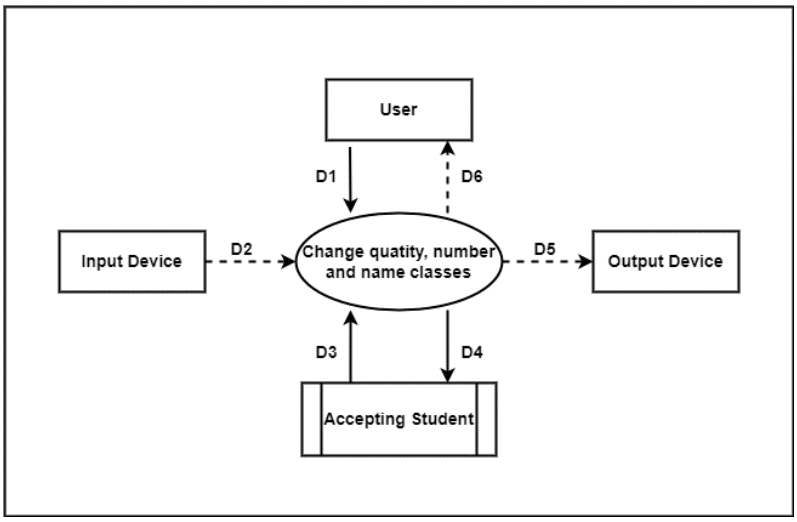
- Data streams:
  + **D1:** Maximum age, Minimum age
  + **D2:** None
  + **D3:** Student's year of birth

+ **D4:** D1
+ **D5:** None
+ **D6:** None

- Algorithm:
  + **Step 1.** Get D1 from the user.
  + **Step 2.** Connect to the database.
  + **Step 3.** Calculate the age of the student.
  + **Step 4.** Check if the new value of the maximum age (D1) is greater than or equal to the student's age (step 3) .
  + **Step 5.** Check if the new value of the minimum age (D1) is less than or equal to the student's age (step 3) .
  + **Step 6.** If the above conditions are not satisfied, go to step 8.
  + **Step 7.** Save D4 to secondary memory.
  + **Step 8.** Close the database connection.
  + **Step 9.** Finish.

2.4.6.2. Data flow diagram for change the quantity, number and name of classes

- Data flow diagram:

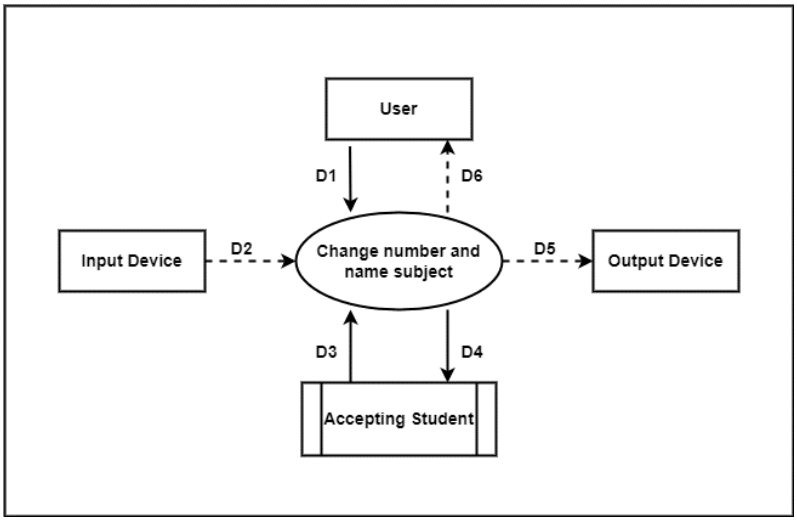


- Data streams:
  + **D1:** Applicable class (Class name, class code), Maximum quantity
  + **D2:** None
  + **D3:** List of classes (Class name, class code)
  + **D4:** D1
  + **D5:** None
  + **D6:** None

- Algorithm:
  + **Step 1.** Get D1 from the user.
  + **Step 2.** Connect to the database.
  + **Step 3.** If you change the number and names of classes, go to step 6
  + **Step 4.** Calculate class quantity
  + **Step 5.** Check if the new value of maximum class quantity (D1) is greater than or equal to class quantity (step 3). If yes, go to step 11. If not, go to step 12.
  + **Step 6.** If you delete the class, go to step 10.
  + **Step 7.** If you update the class, go to step 9.
  + **Step 8.** Check if the class name (D1) is in the class list (D3). If not, add the class to the class list and then go to step 11. If not, go to step 12.
  + **Step 9.** Check if the class name (D1) is in the class list (D3). If not, update the name in the class list and then go to step 11. If not, go to step 12.
  + **Step 10.** Check if the class name (D1) is in the class list (D3). If yes, delete the corresponding class from the class list and then go to step 11. If not, go to step 12.
  + **Step 11.** Save D4 to secondary memory.
  + **Step 12.** Close the database connection.
  + **Step 13.** Finish.

2.4.6.3. Data flow diagram for change the number and name of subjects

- Data flow diagram:



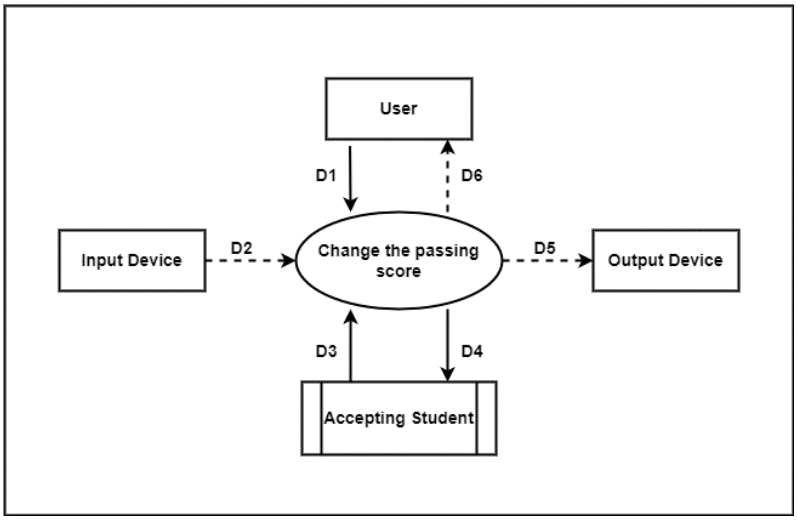
- Data streams:
  + **D1:** Subject to be applied (Subject name, subject code)
  + **D2:** None
  + **D3:** List of subjects (Subject name, subject code)
  + **D4:** D1

+ **D5:** None
+ **D6:** None

- Algorithm:
  + **Step 1.** Get D1 from the user.
  + **Step 2.** Connect to the database.
  + **Step 3.** If you delete a subject, go to step 7.
  + **Step 4.** If you edit the subject name, go to step 6.
  + **Step 5.** Check if the subject (D1) is in the list of subjects (D3). If not, add the subject to the list of subjects and then go to step 8. Otherwise go to step 9.
  + **Step 6.** Check if the subject name (D1) is in the list of subjects (D3). If not, update the name in the subject list, then go to step 8. Otherwise go to step 9.
  + **Step 7.** Check if the subject name (D1) is in the list of subjects (D3). If not, delete the corresponding subject from the subject list and then go to step 8. Otherwise go to step 9.
  + **Step 8.** Save D4 to secondary memory.
  + **Step 9.** Close the database connection.
  + **Step 10.** Finish.

2.4.6.4. Data flow diagram for change the passing score

- Data flow diagram:

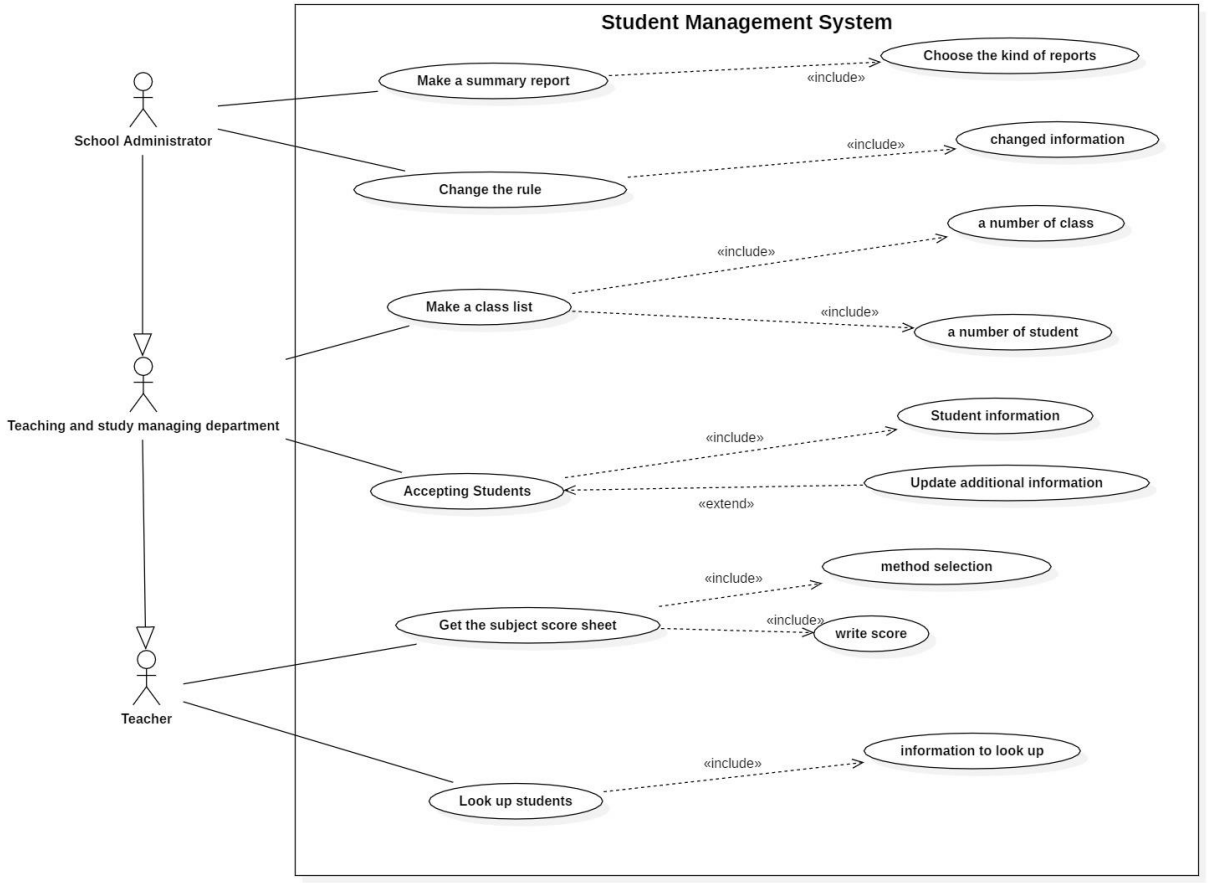


- Data streams:
  + **D1:** Passing score
  + **D2:** None
  + **D3:** None
  + **D4:** D1

    - **D5:** None
    - **D6:** None

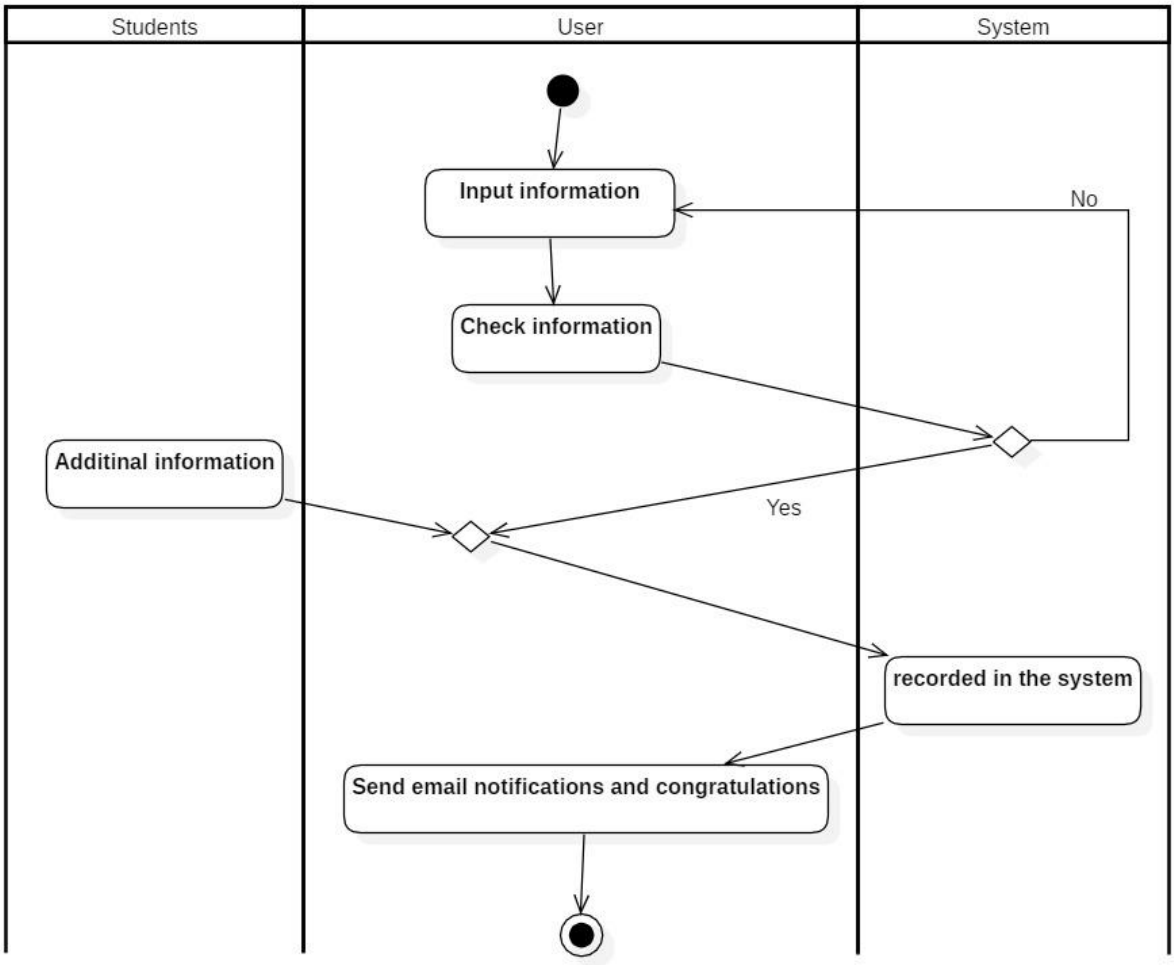
- Algorithm:
    - **Step 1.** Get D1 from the user.
    - **Step 2.** Connect to the database.
    - **Step 3.** Save D4 to secondary memory.
    - **Step 4.** Close the database connection.
    - **Step 5.** Finish.

CHAPTER 3: DRAW USE CASE DIAGRAM

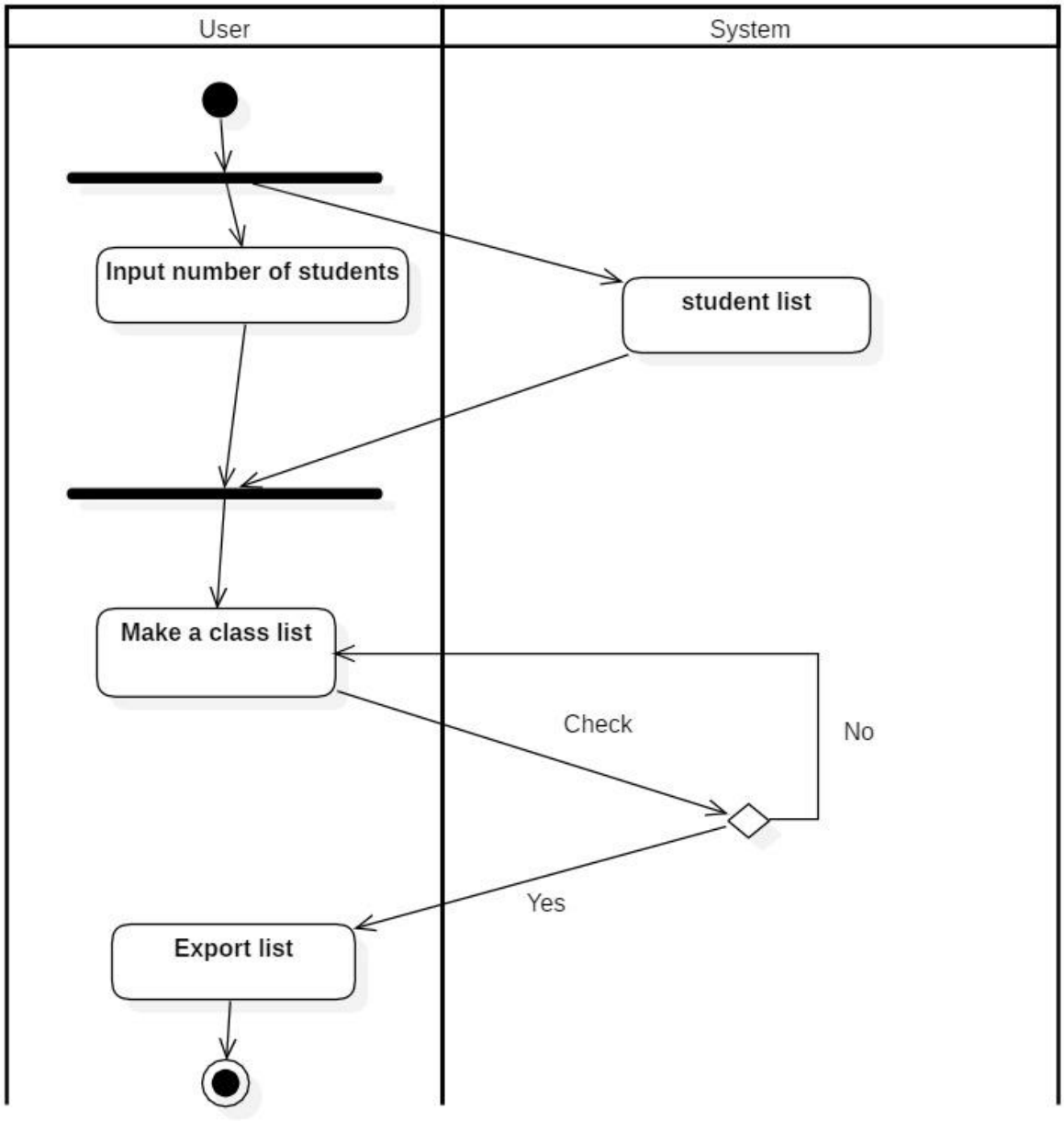


CHAPTER 4: DRAW ACTIVITY DIAGRAM

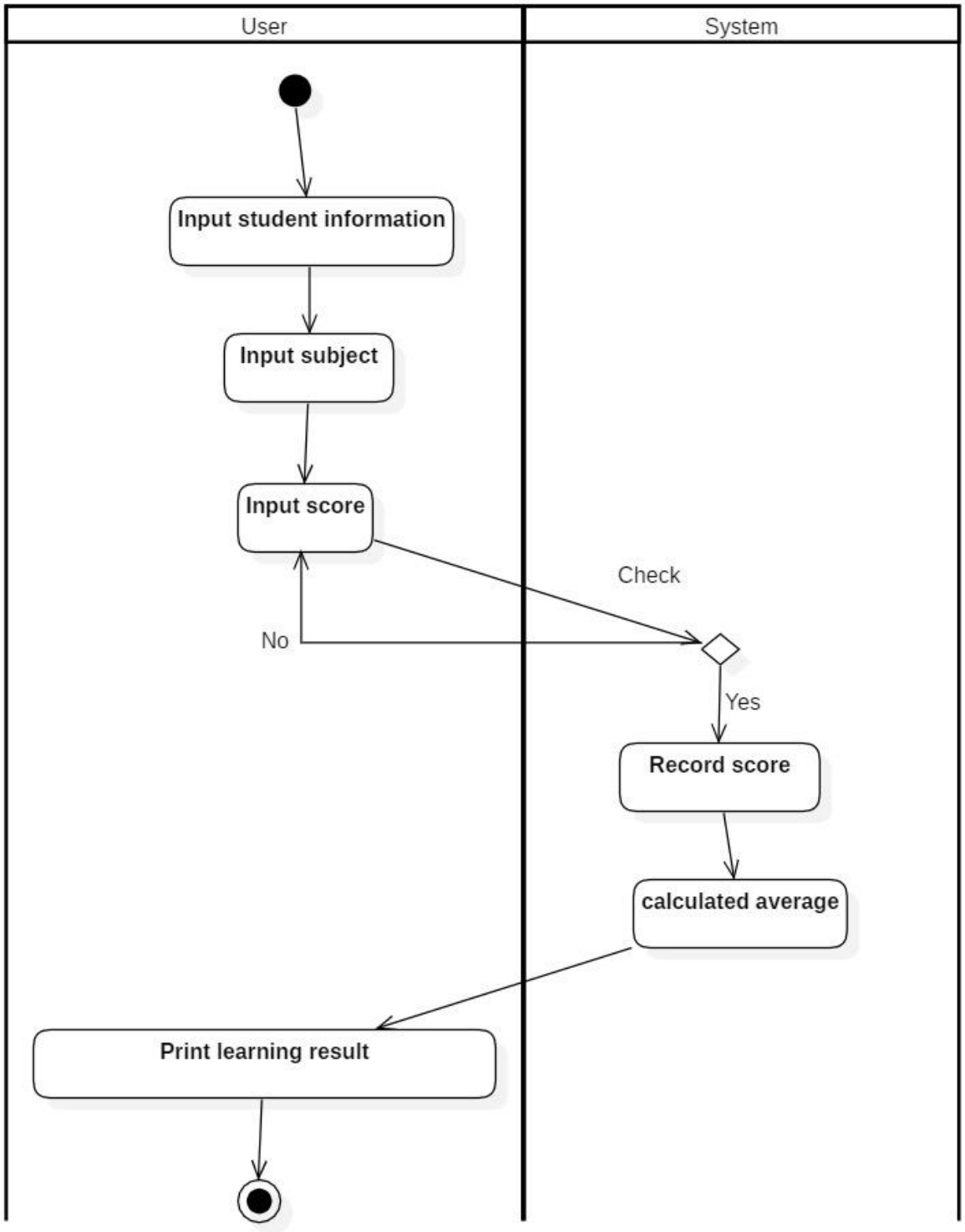
4.1. Activity diagram for accepting students request



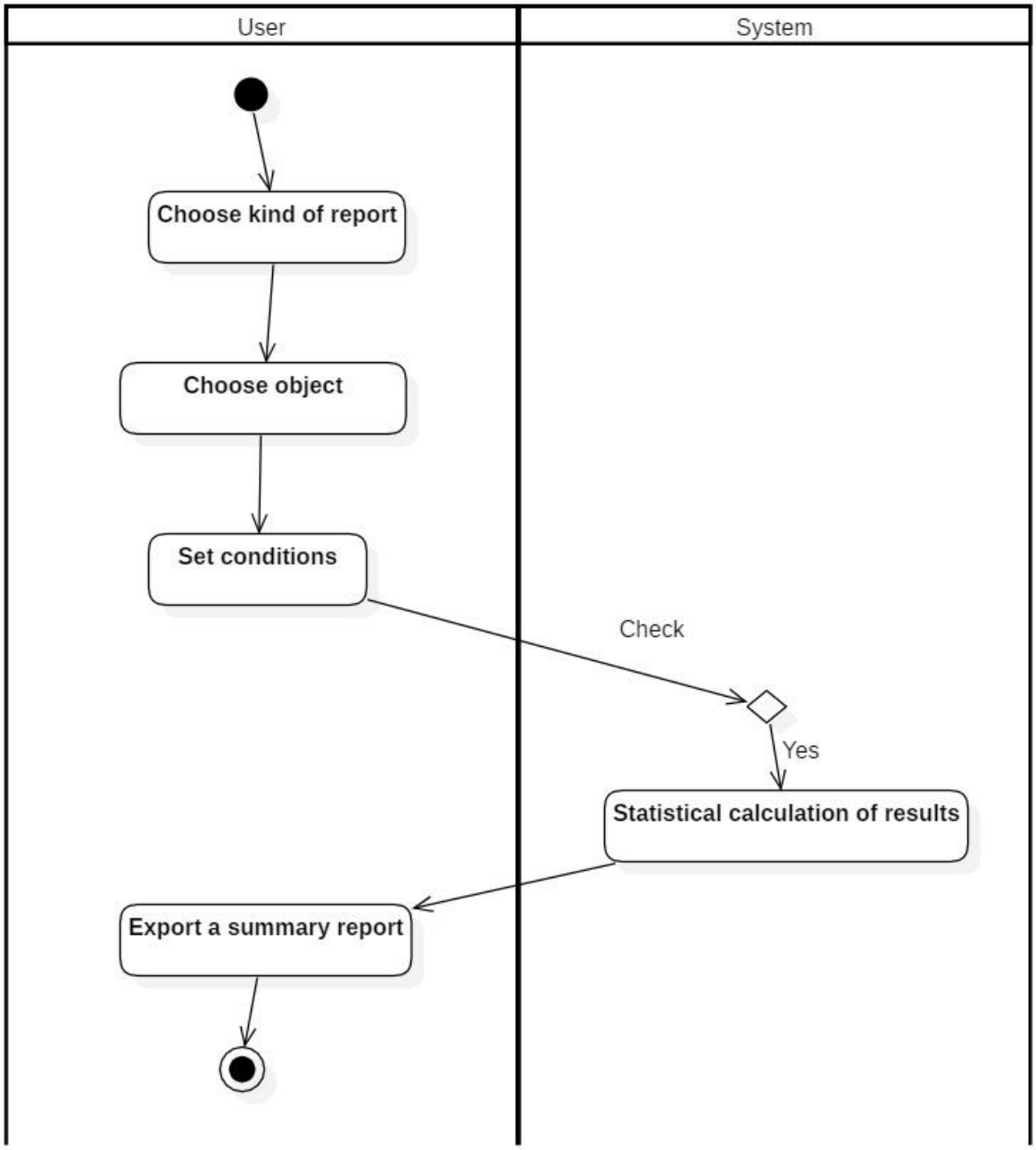
4.2. Activity diagram for make a class list request



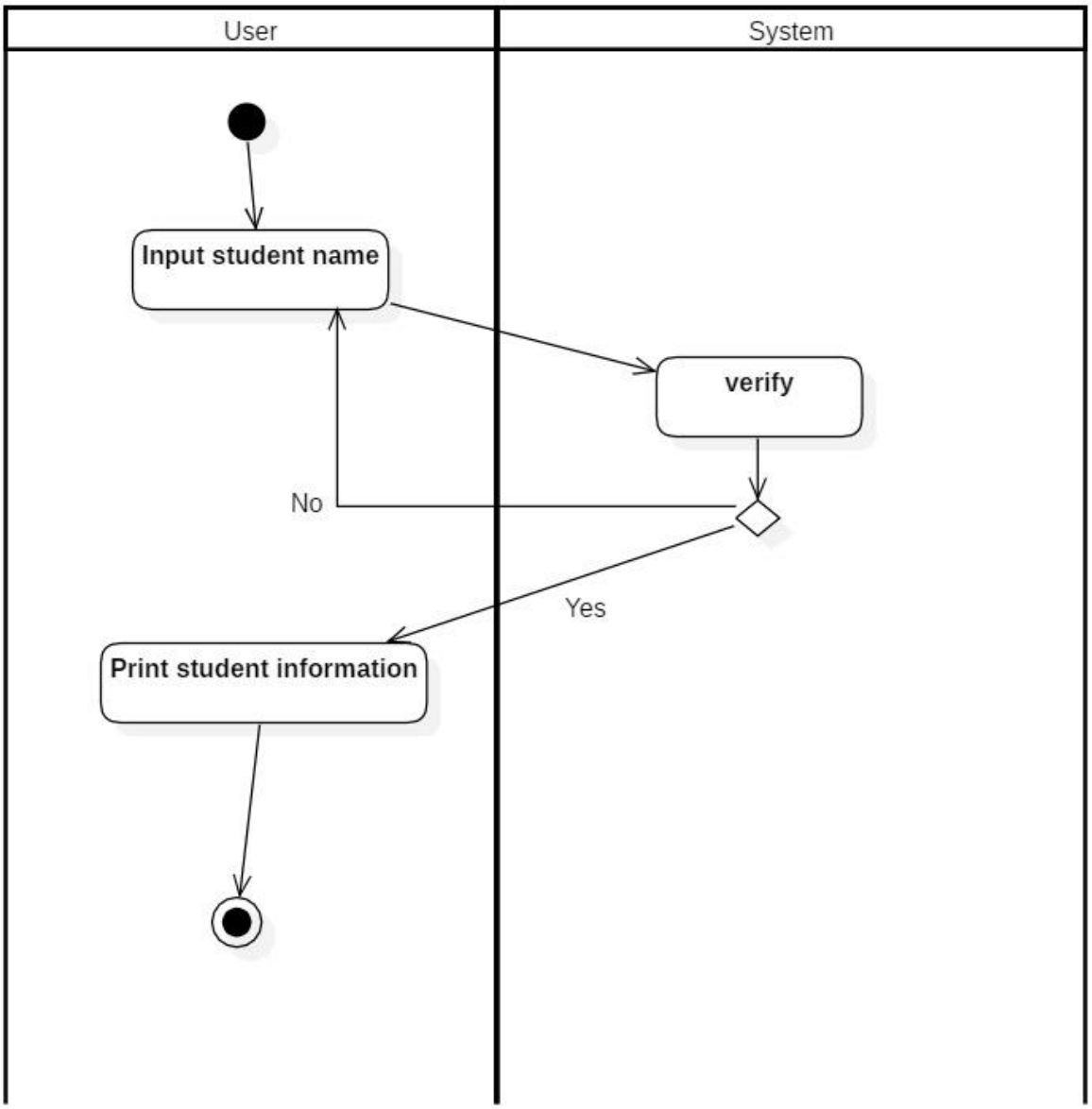
4.3. Activity diagram for scoring request



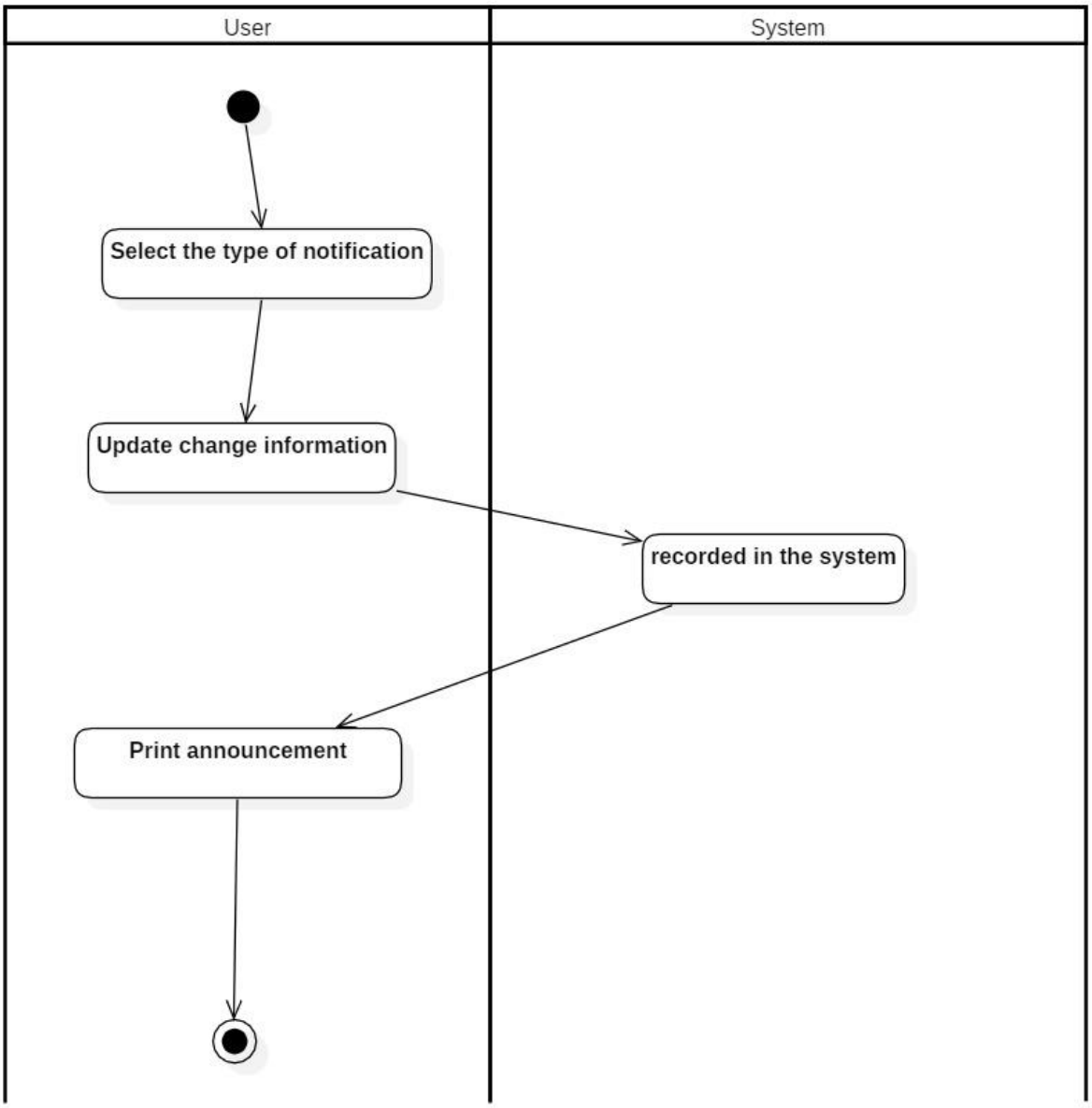
4.4. Activity diagram for make a summary report request



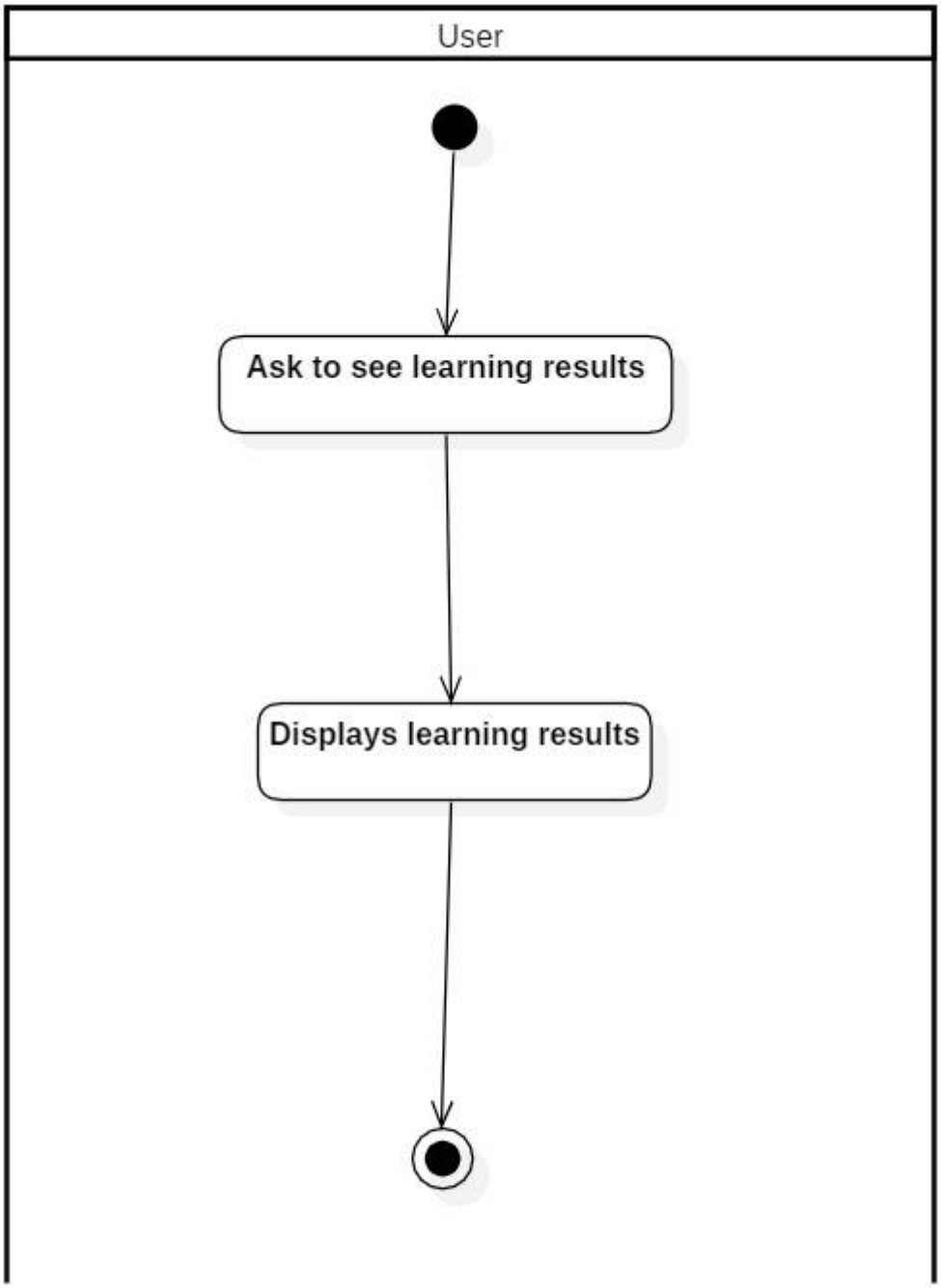
4.5. Activity diagram for look up students request



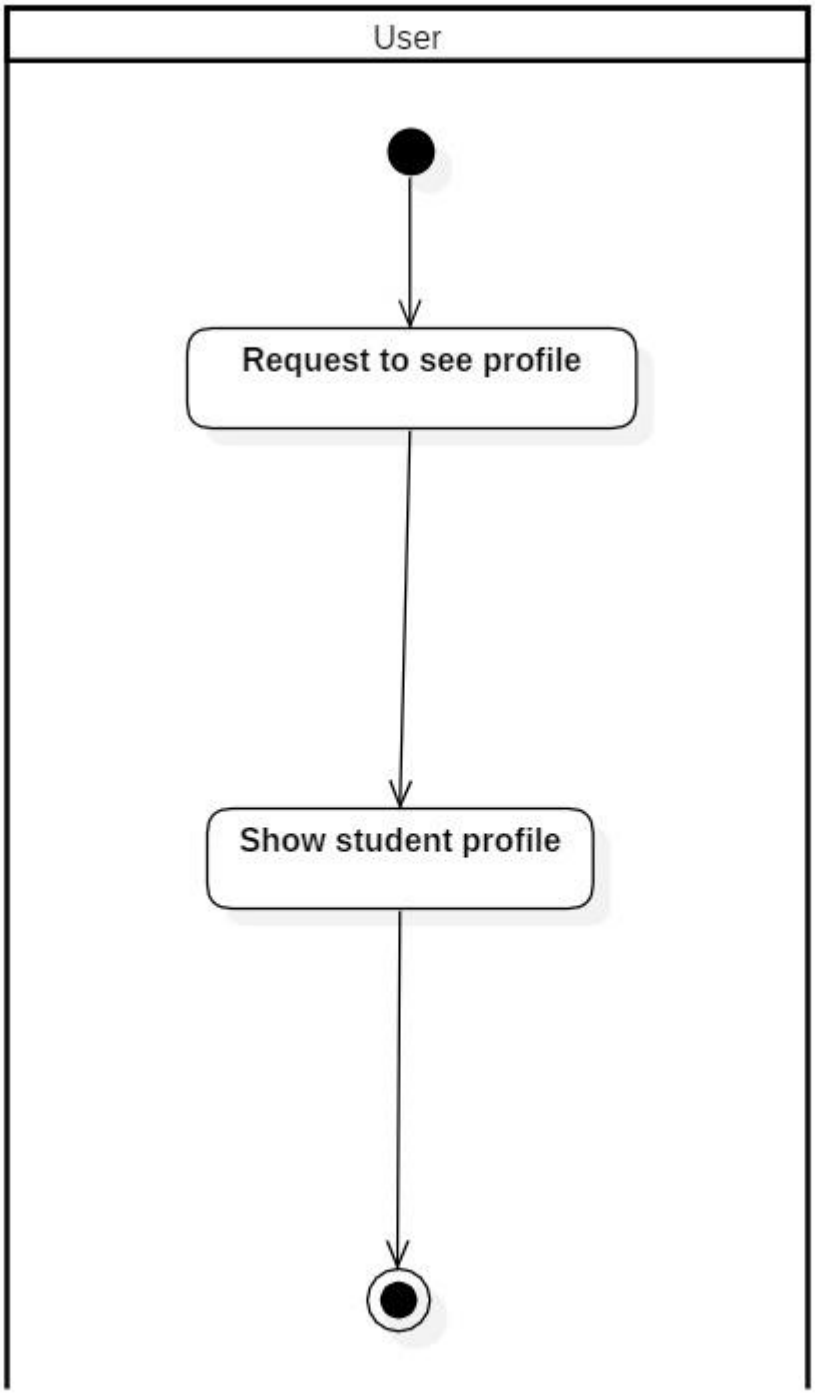
4.6. Activity diagram for update changed information request



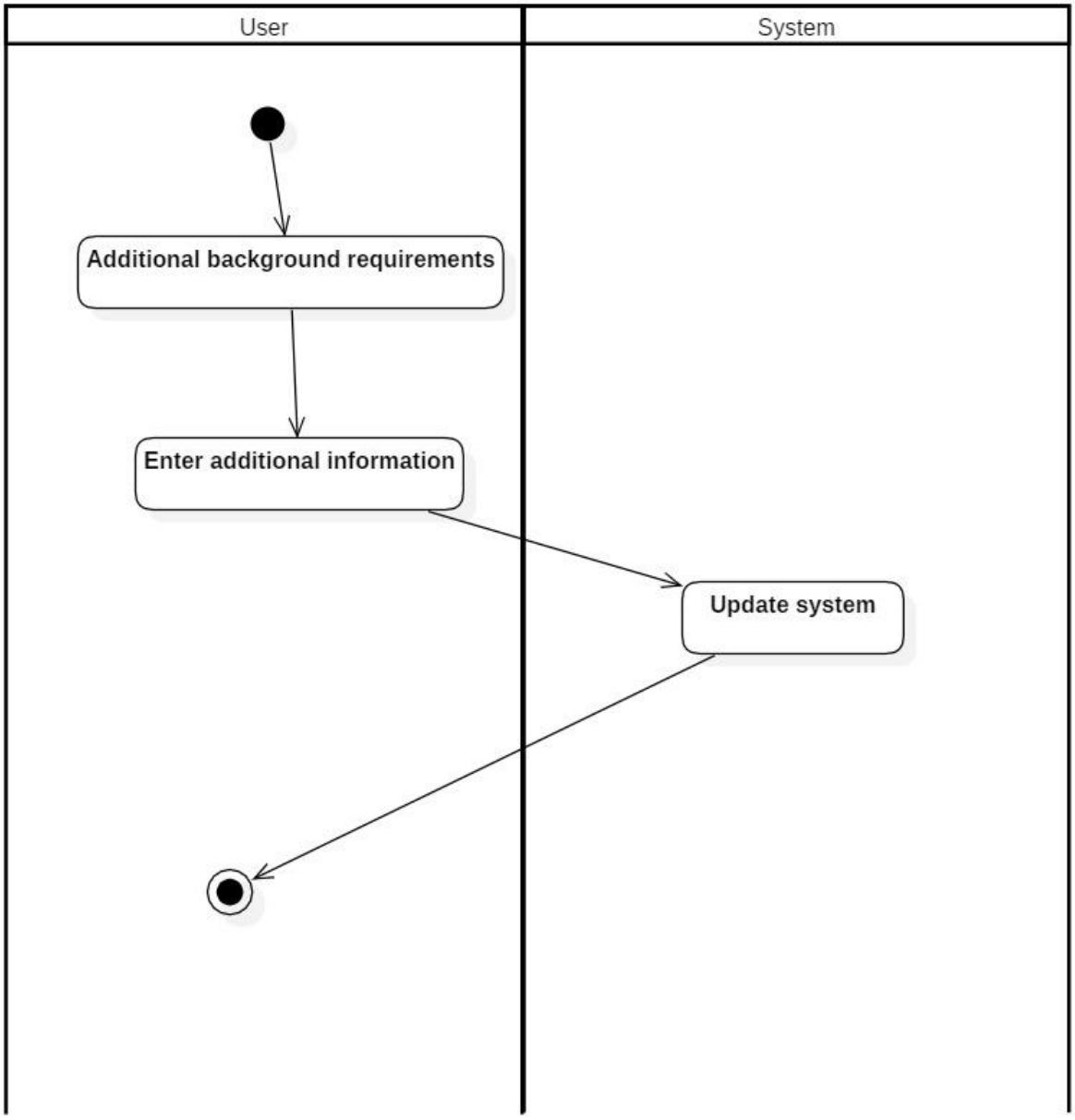
4.7. Activity diagram for view learning result request



4.8. Activity diagram for view student resumes request

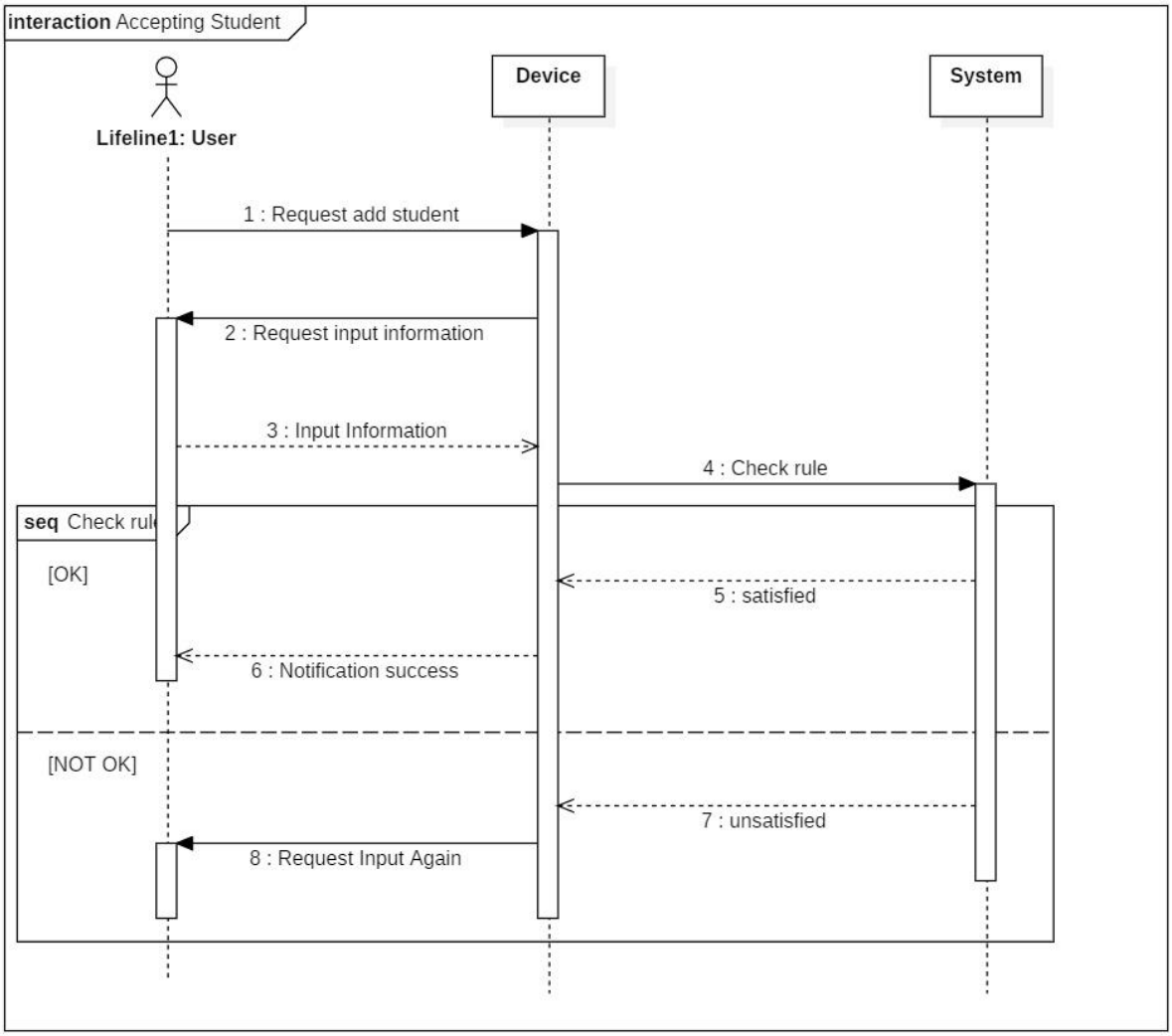


4.9. Activity diagram for additional resumes update request

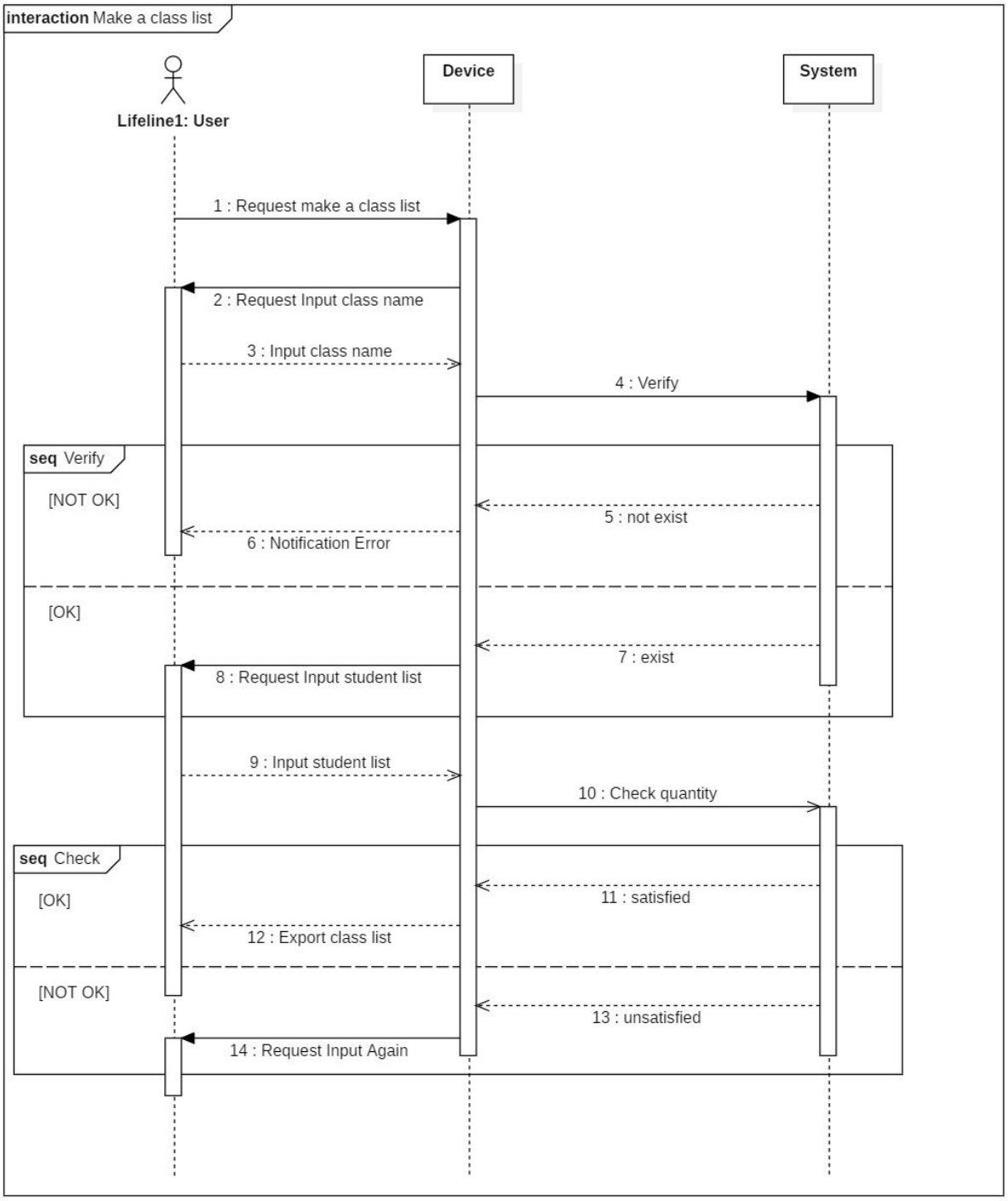


CHAPTER 5: DRAW SEQUENCE DIAGRAM

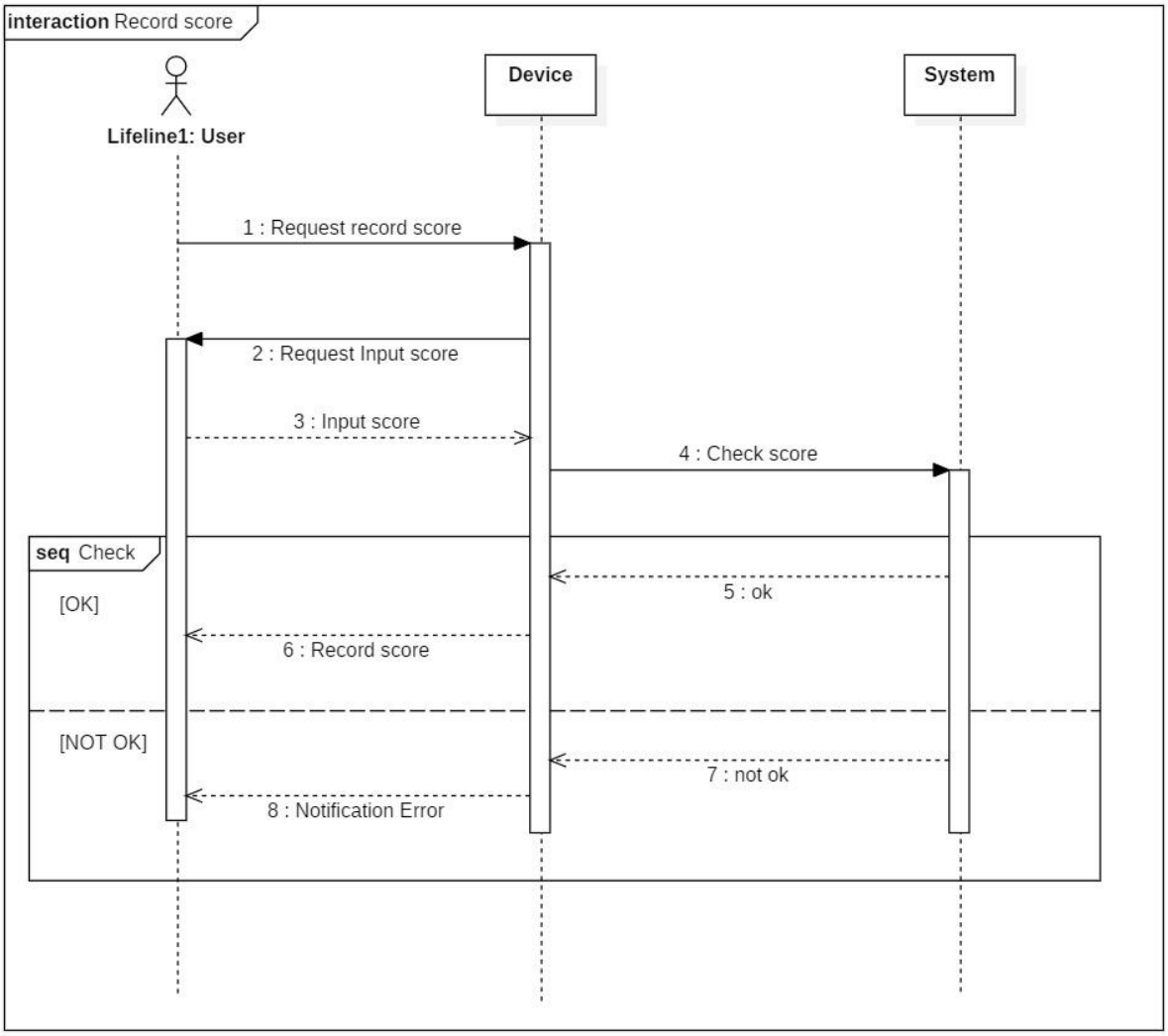
5.1. Sequence diagram for accepting student request



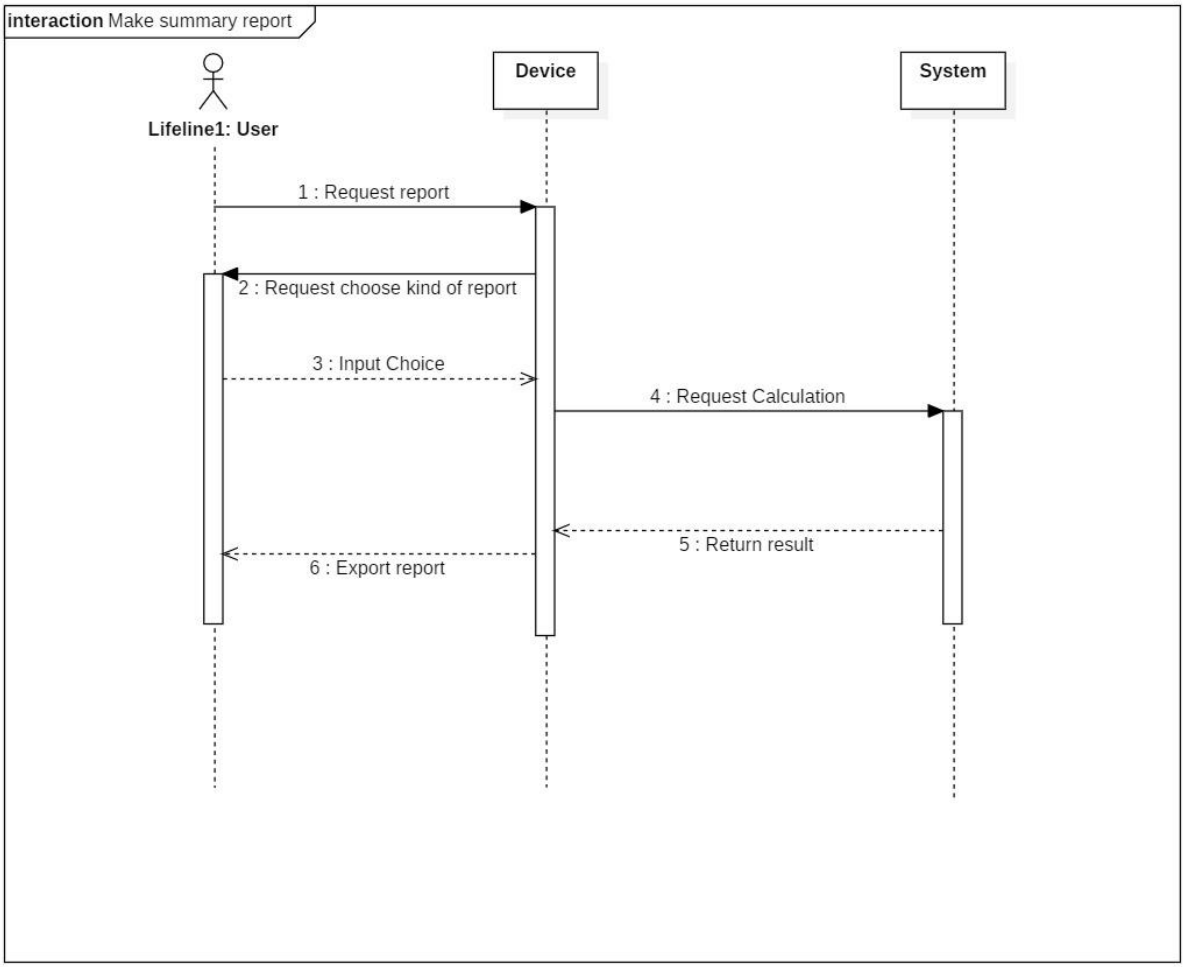
5.2. Sequence diagram for make a class list request



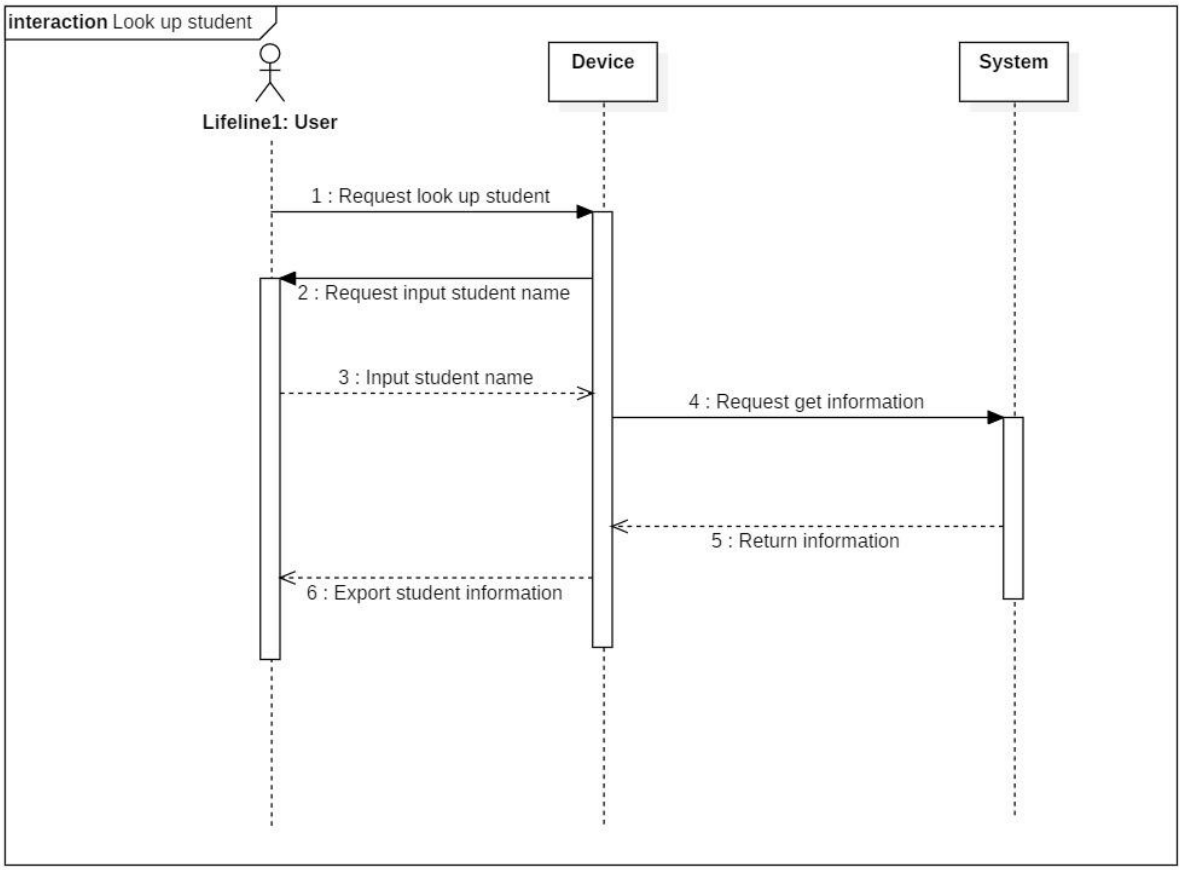
5.3. Sequence diagram for scoring request



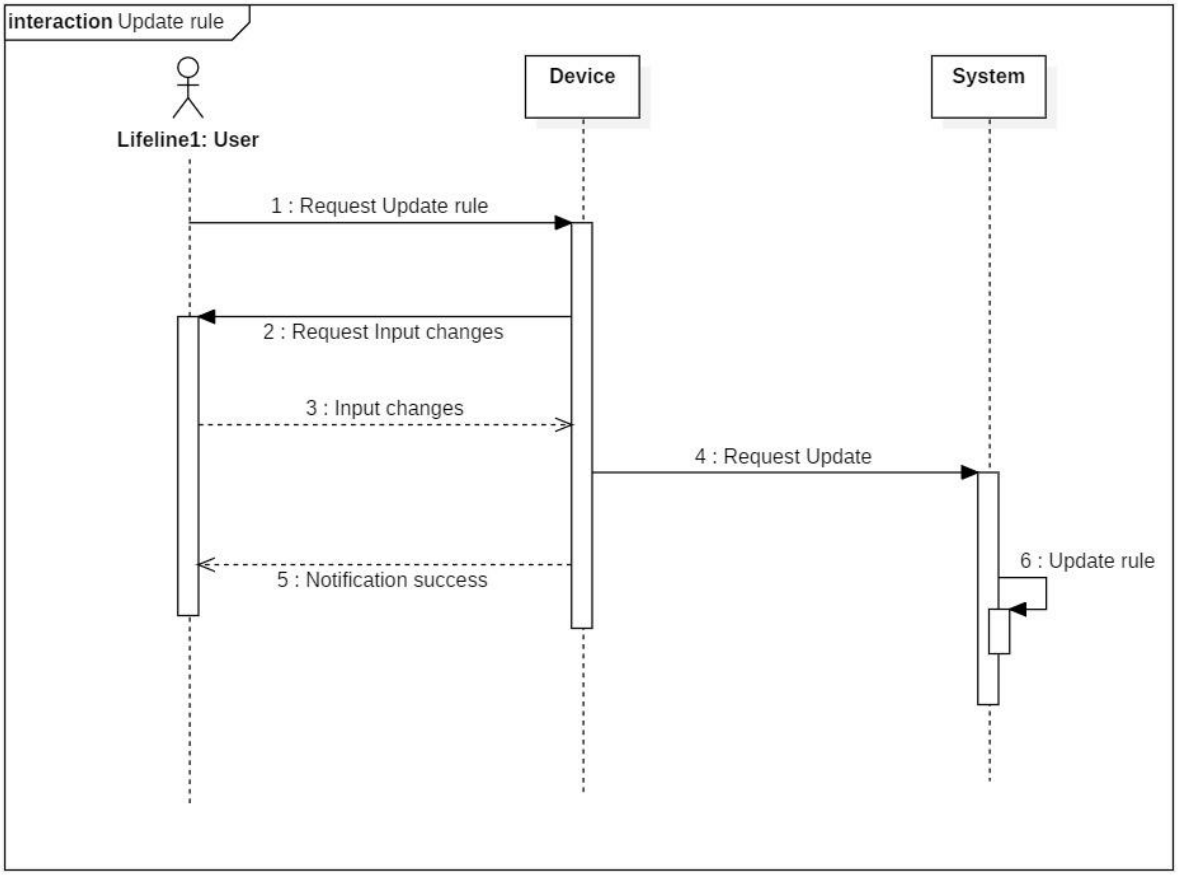
5.4. Sequence diagram for make summary report request



5.5. Sequence diagram for look up student request



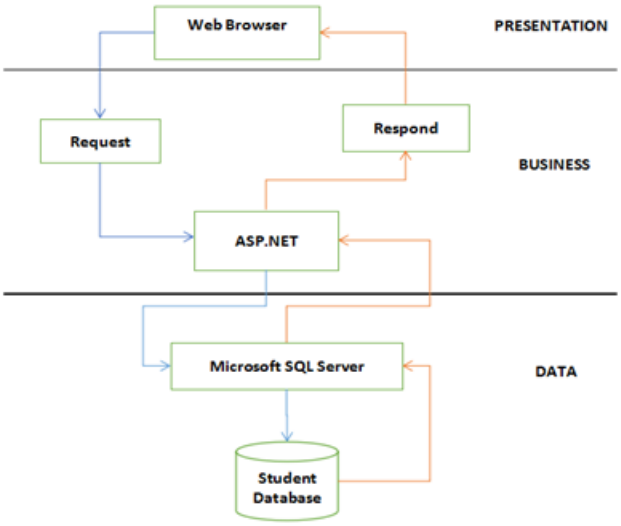
5.6. Sequence diagram for update rule request



CHAPTER 6: DATA DESIGN

6.1. System design

6.1.1. System architecture



In this software project "Student Management", my team will use a 3-layer model for the system design. This model includes:

6.1.1.1. Presentation layer

Used to communicate with users. This class includes interface components such as win form, web form, etc. and performs tasks such as inputting data, displaying retrieval results, checking the correctness of data to be ready for the next step.

6.1.1.2. Business layer

Is the layer that meets the specific requirements of the application, here it is possible to display a list of students. This layer mainly processes the data source from the Presentation layer before transmitting to the Data layer and then saving it to the DBMS.

6.1.1.3. Data layer

It has the function to communicate with the DBMS and perform tasks related to data storage and query (search, add, delete, update,…)

6.1.2. Description of the components in the system

No.	Element	Explain
1	Web Browser	Runs with 2 protocols, WML and HTML, responsible for providing user interface and performing data input and output operations, error reporting.
2	Request	Requirements between layers in the system.
3	Respond	Respond to requests received.
4	ASP.NET	Make connections to SQL and execute queries, and fulfill business requirements such as constraint correctness, data integrity, and validity checks.
5	Microsoft SQL Server	Receive queries and return results to ASP.NET, and perform storage-related tasks (add, delete, update,…)
6	Student Database	Data storage for student management software.

6.2. Data design

6.2.1. Algorithm for logic diagramming

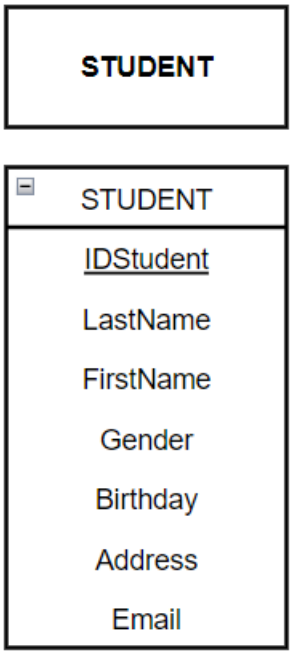
6.2.1.1. Accepting students request

- Data design with correctness:

BM1:	**Student Records**
Full name:……	Gender:……
Date of Birth:……	Address:……
Email:…….	

+ Related Form: Form 1
+ Data Flow Diagram: Diagram 2.4.1
+ New attributes: IDStudent, LastName, FirstName, Gender, Birthday, Address, Email
+ Data design: table STUDENT
+ Abstract properties: IDStudent

+ Logic Diagram:

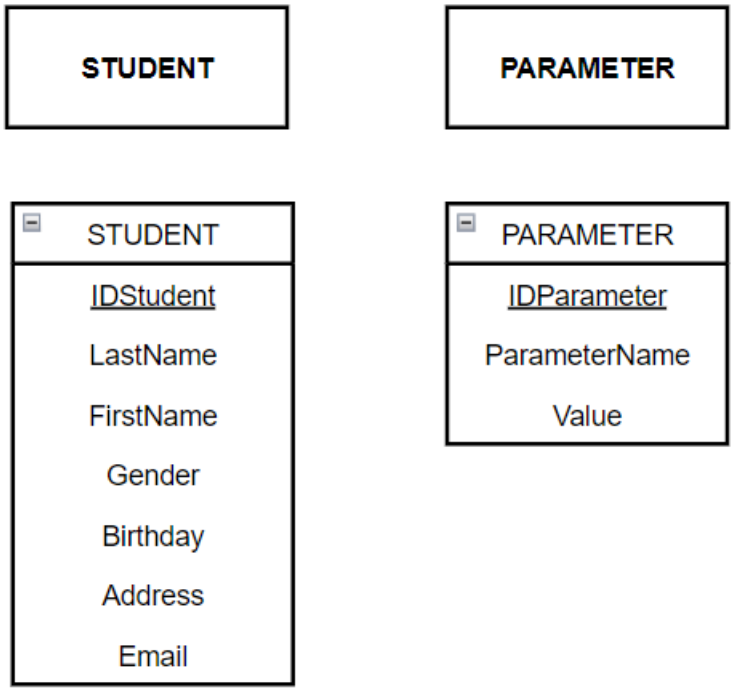


– Data design with evolution:

QĐ1: Student age from 15 to 20.

+ Related Rule: Rule 1
+ Rule change Data Flow Diagram: Diagram 2.4.6.1
+ New attributes: IDParameter, ParameterName, Value
+ New parameters: MinimumAge, MaximumAge
+ Data design: table STUDENT, table PARAMETER
+ Abstract properties: IDParameter

+ Logic Diagram:



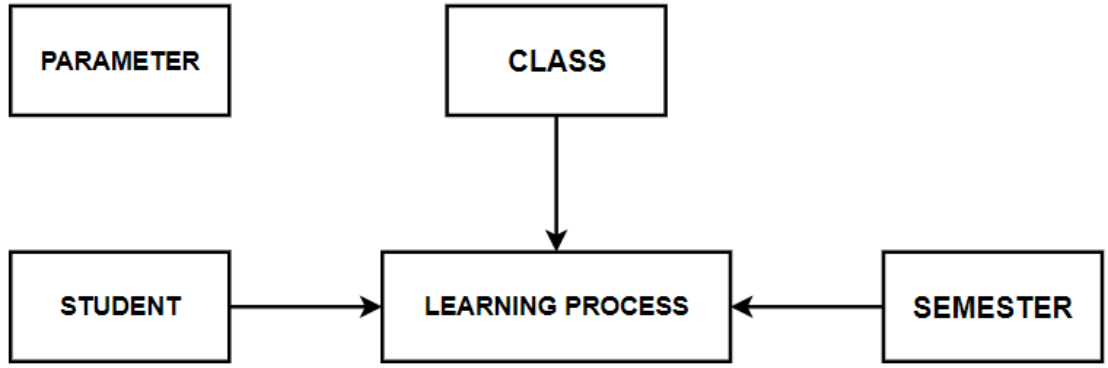
6.2.1.2. Make a class list request

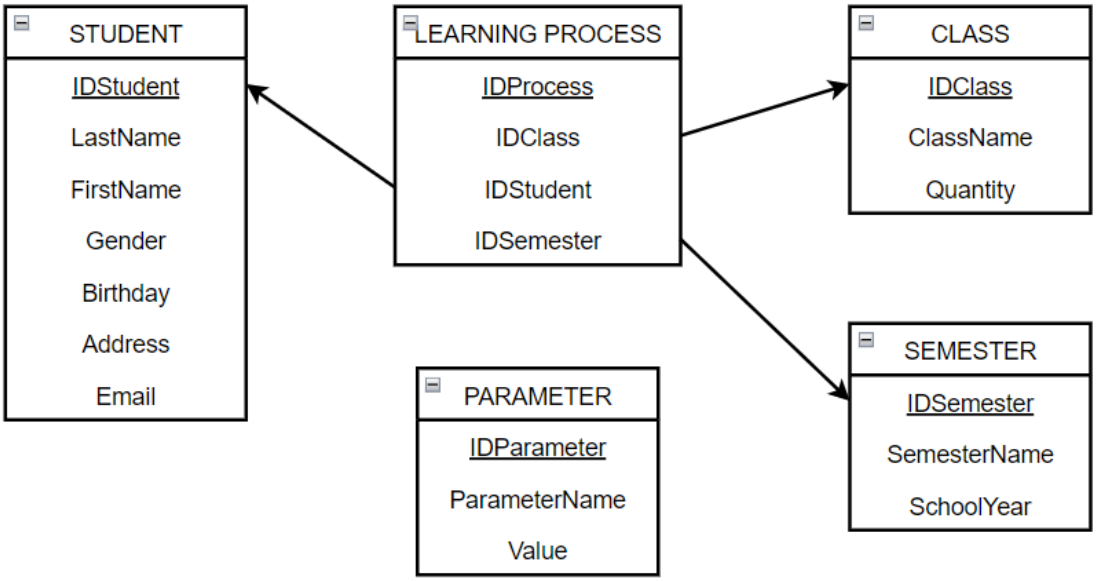
– Data design with correctness:

BM2:	**Class List**			
Class:……..		Quantity:……		
No.	**Full Name**	**Sex**	**Year of Birth**	**Address**
1				
2				

+ Related Form: Form 2
+ Data Flow Diagram: Diagram 2.4.2
+ New attributes: IDClass, IDProcess, IDSemester, Quantity, ClassName, SchoolYear, SemesterName
+ Data design: table STUDENT, table PARAMETER, table LEARNING PROCESS, table SEMESTER, table CLASS
+ Abstract properties: IDClass, IDProcess, IDSemester

+ Logic Diagram:

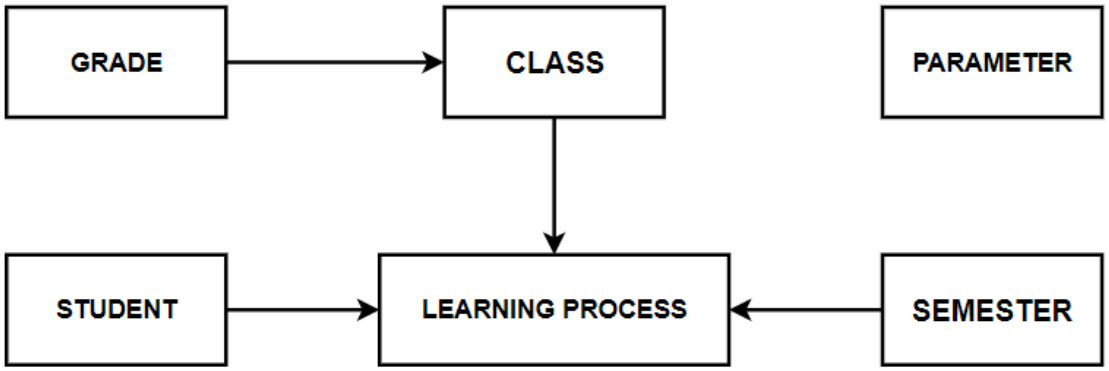


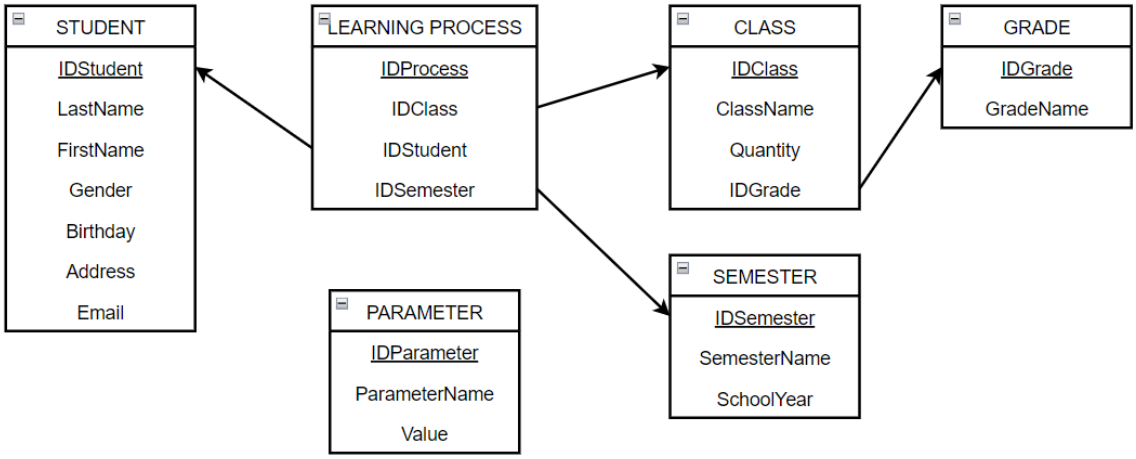


– Data design with evolution:

QĐ2: There are 3 grades (10, 11, 12). Grade 10 has 4 classes (10A1, 10A2, 10A3, 10A4). Grade 11 has 3 classes (11A1, 11A2, 11A3). Grade 12 has 2 classes (12A1, 12A2). Each class has no more than 40 students.

+ Related Rule: Rule 2
+ Rule change Data Flow Diagram: Diagram 2.4.6.2
+ New Attributes: IDGrade, GradeName
+ New parameters: MaximumQuantity
+ Data design: table STUDENT, table PARAMETER, table LEARNING PROCESS, table SEMESTER, table CLASS, table GRADE
+ Abstract Attributes: IDGrade
+ Logic Diagram:



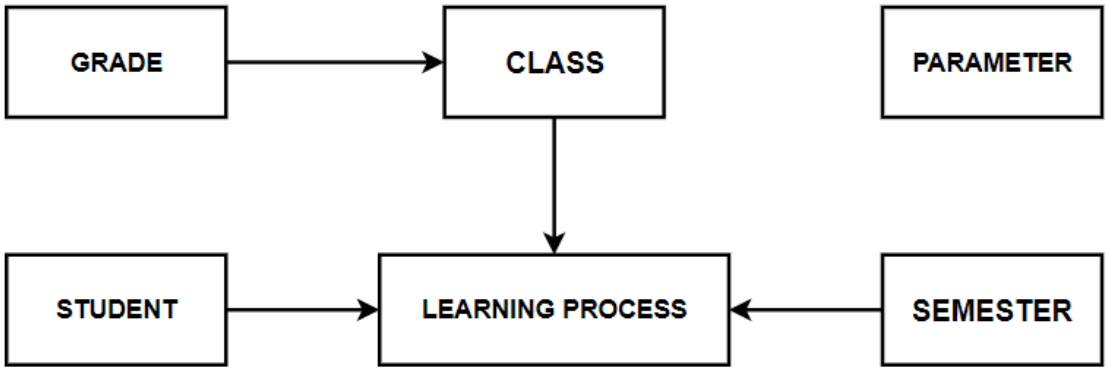


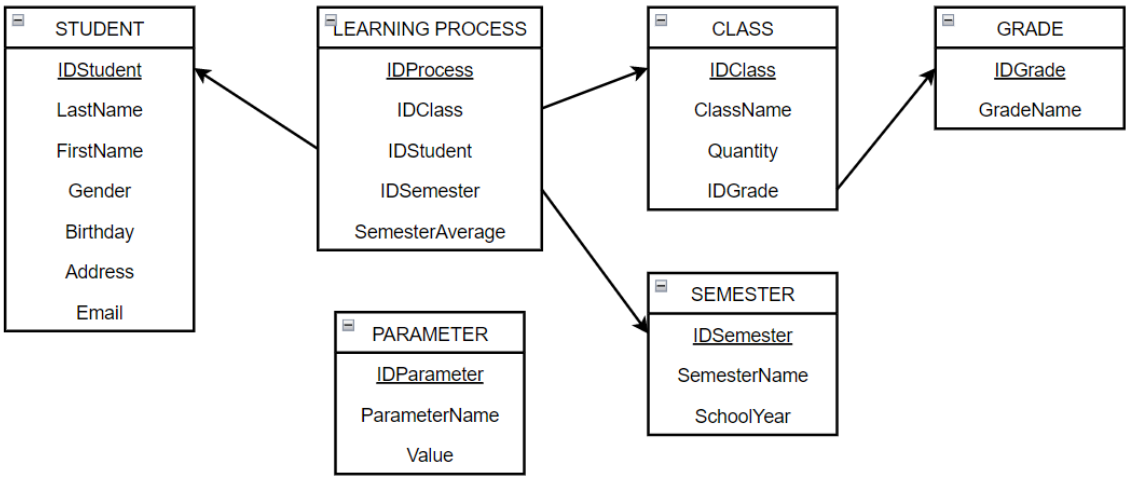
6.2.1.3. Look up students request

– Data sign with correctness:

BM3:	Student List			
No.	**Full Name**	**Class**	**1st Semester GPA**	**2nd Semester GPA**
1				
2				

+ Related Form: Form 3
+ Data Flow Diagram: Diagram 2.4.3
+ New properties: SemesterAverage
+ Data design: table STUDENT, table PARAMETER, table LEARNING PROCESS, table SEMESTER, table CLASS, table GRADE
+ Abstract properties: None
+ Logic Diagram:





– Data design with evolution:
  + Related Rule: None
  + Rule change Data Flow Diagram: None
  + New attributes: None
  + New parameters: None
  + Data design: table STUDENT, table PARAMETER, table LEARNING PROCESS, table SEMESTER, table CLASS, table GRADE
  + Abstract properties: None
  + Logic Diagram: None

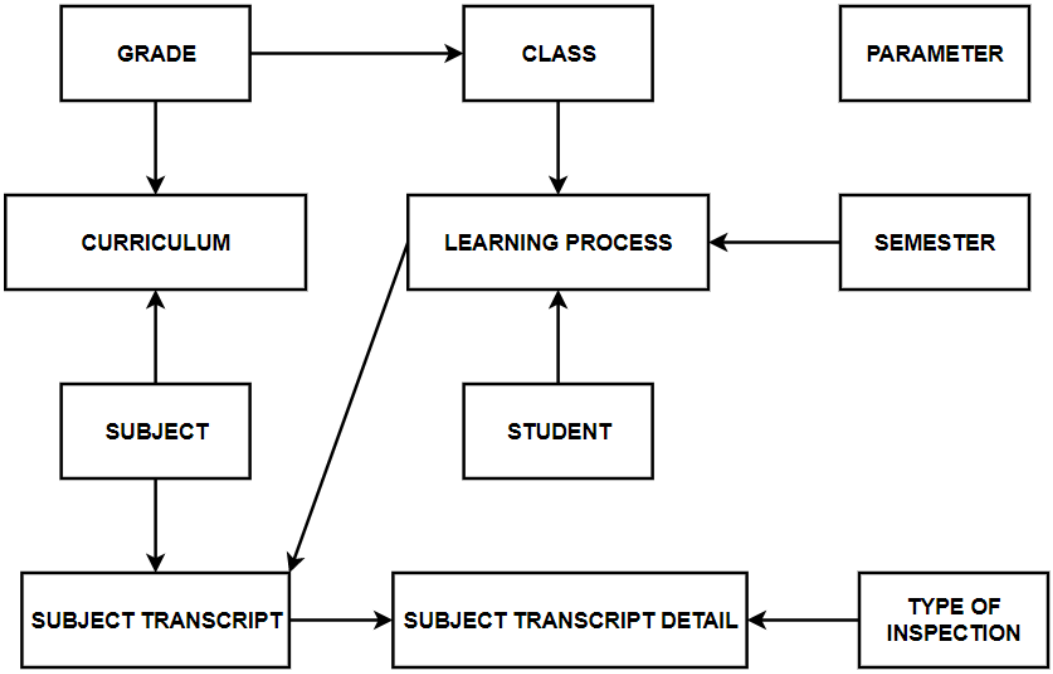
6.2.1.4. Get the subject score sheet request

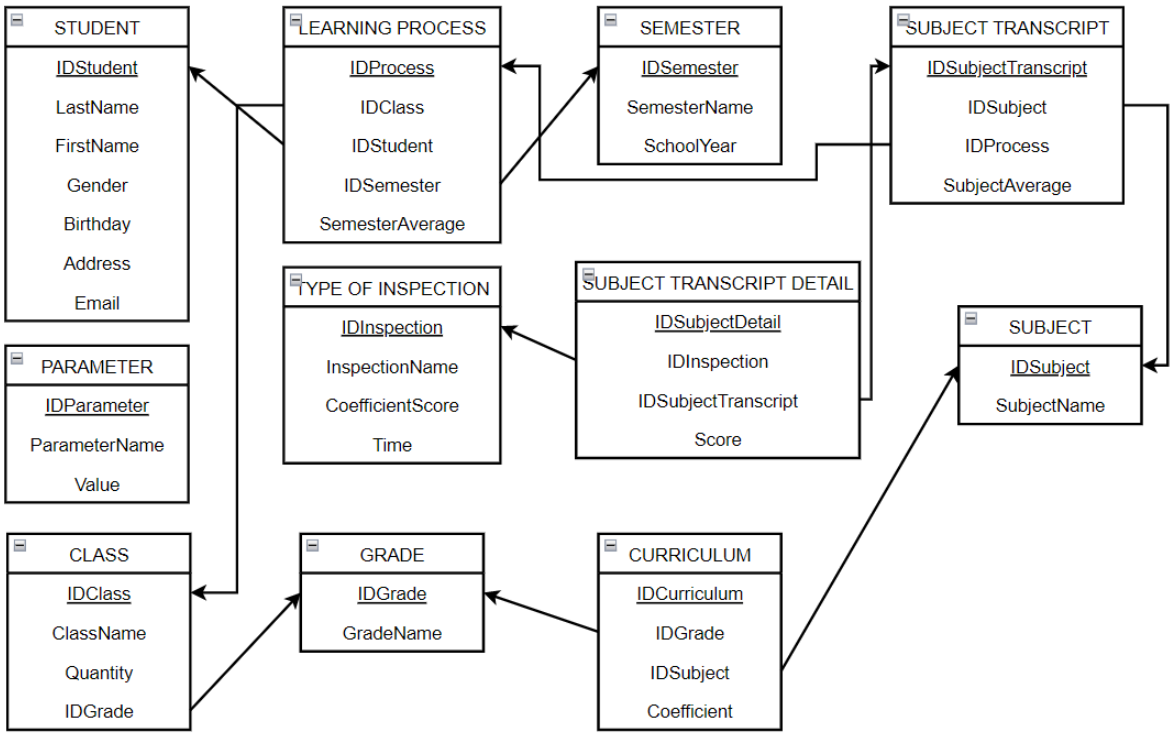
– Data design with correctness:

BM4:	**Subject Score Sheet**			
Class:……		Subject:……		
Semester:……				
No.	**Full Name**	**15-minute Score**	**1-period Score**	**Average Score**
1				
2				

  + Related Form: Form 4
  + Data Flow Diagram: Diagram 2.4.4

+ New attributes: IDSubject, SubjectName, IDSubjectTranscript, SubjectAverage, DIEM, IDInspection, InspectionName, CoefficientScore, Time, IDCurriculum, Coefficient, IDSubjectDetail, Score
+ Data design: table STUDENT, table PARAMETER, table LEARNING PROCESS, table SEMESTER, table CLASS, table GRADE, table SUBJECT, table SUBJECT TRANSCRIPT, table TYPE OF INSPECTION, table CURRICULUM, table SUBJECT TRANSCRIPT DETAIL
+ Abstract properties: IDSubject, IDSubjectTranscript, IDInspection, IDCurriculum, IDSubjectDetail
+ Logic Diagram:



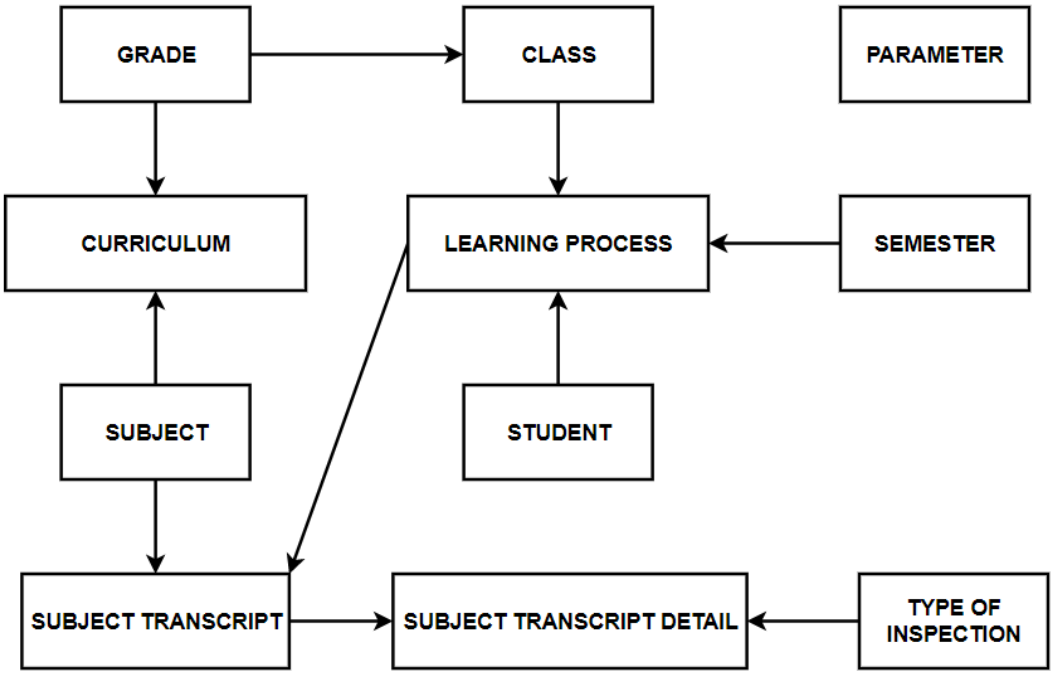


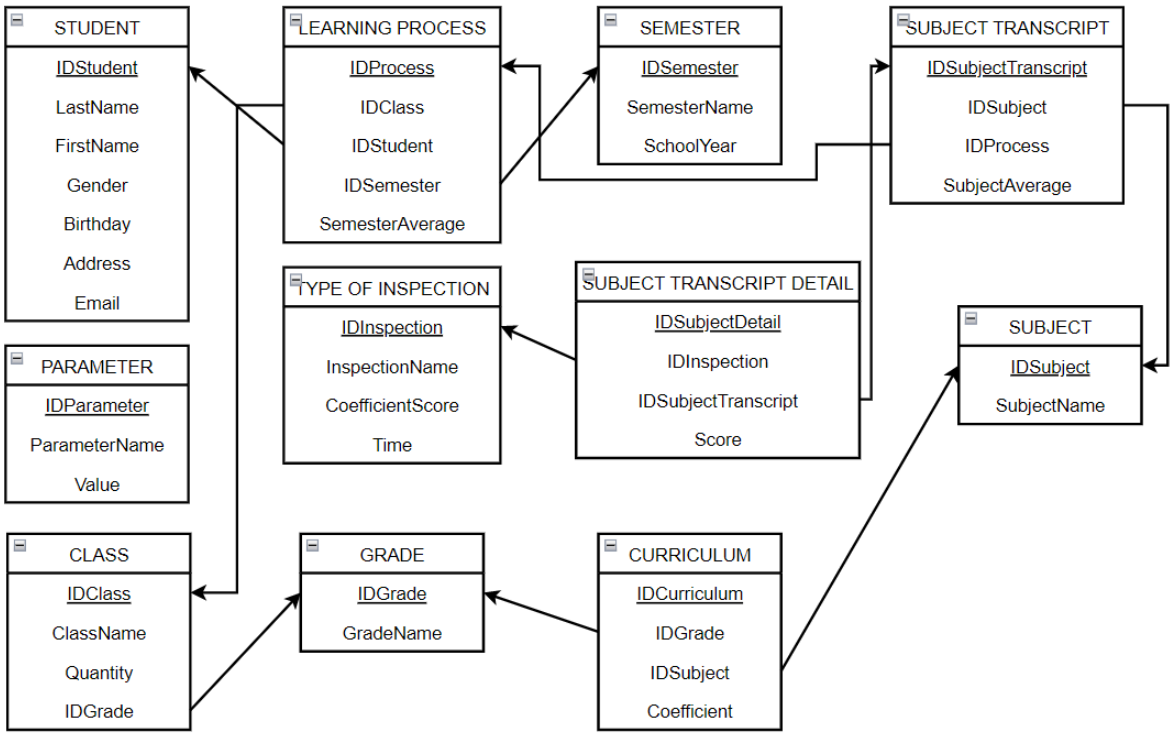
– Data design with evolution:

QĐ4: There are 2 semesters (I, II). There are 9 subjects (Math, Physics, Chemistry, Biology, History, Geography, Literature, Civic Education, Physical Education). $0 \leq Score \leq 10$

+ Related Rule: Rule 4
+ Rule change Data Flow Diagram: Diagram 2.4.6.3
+ New properties: None
+ New parameters: MinimumScore, MaximumScore
+ Data design: table STUDENT, table PARAMETER, table LEARNING PROCESS, table SEMESTER, table CLASS, table GRADE, table SUBJECT, table SUBJECT TRANSCRIPT, table TYPE OF INSPECTION, table CURRICULUM, table SUBJECT TRANSCRIPT DETAIL
+ Abstract properties: None

+ Logic Diagram:



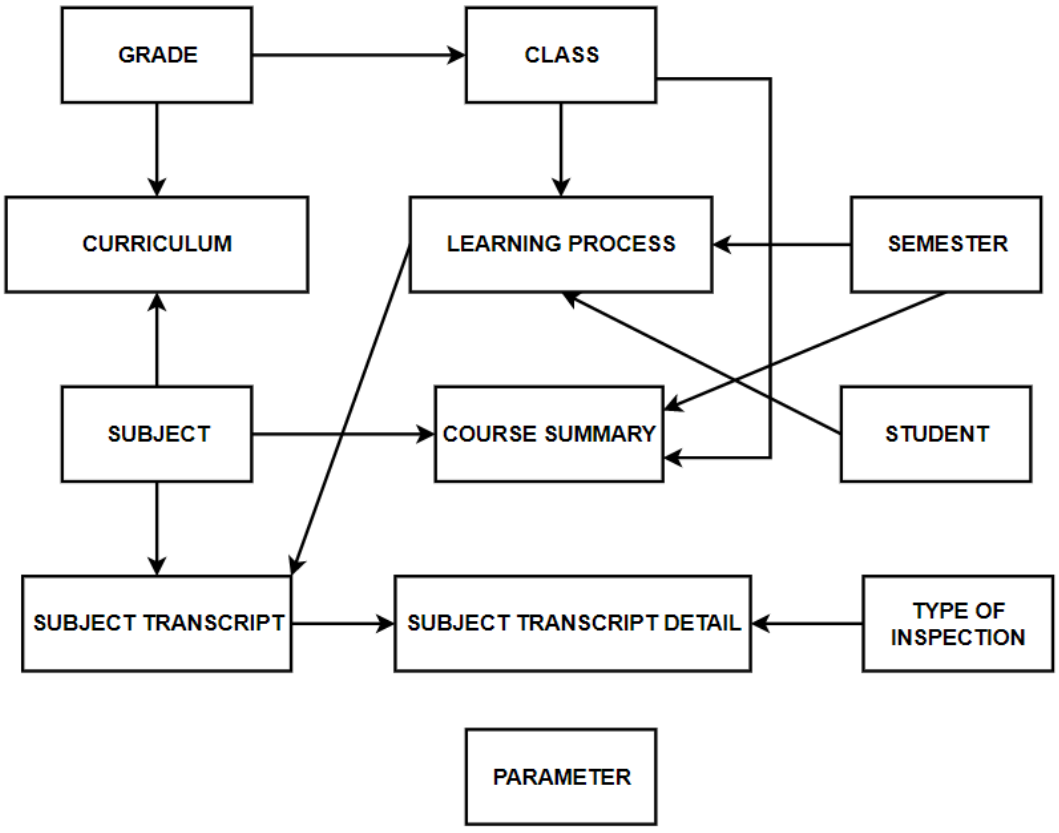


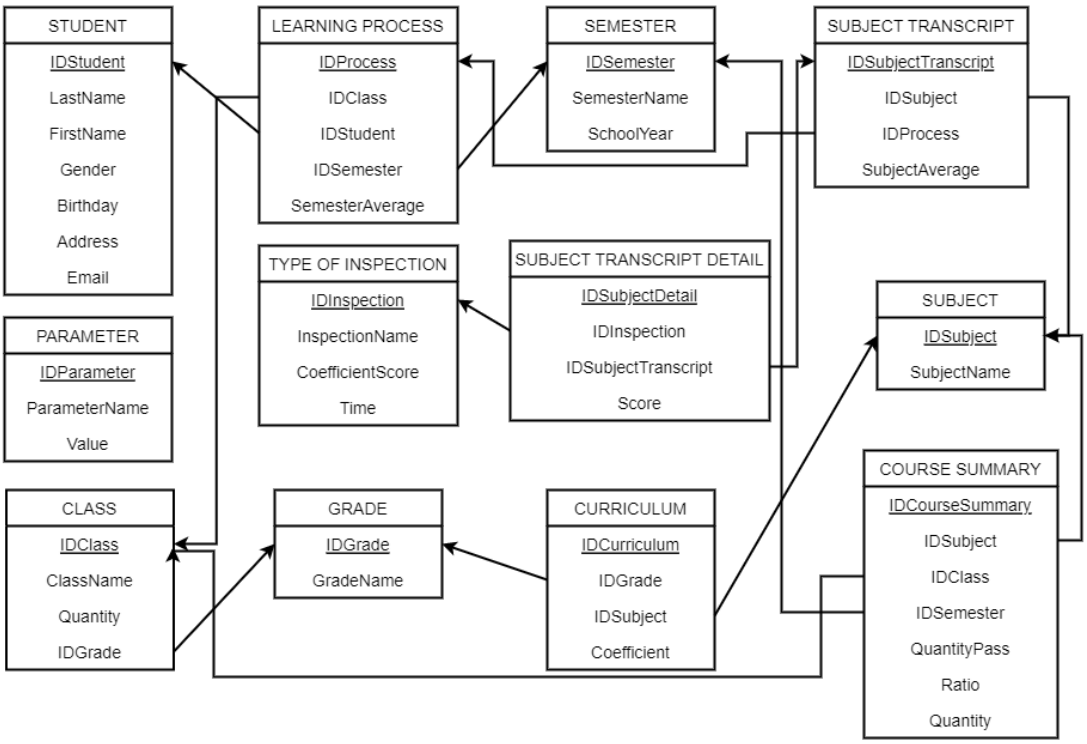
6.2.1.5. Make a subject summary report request

– Data design with correctness:

BM5.1:	Subject Summary Report			
Subject:……		Semester:……		
No.	**Class**	**Quantity**	**Number of Passes**	**Ratio**
1				
2				

+ Related Form: Form 5.1
+ Data Flow Diagram: Diagram 2.4.5.1
+ New attributes: IDCourseSummary, QuantityPass, Ratio
+ Data design: table STUDENT, table PARAMETER, table LEARNING PROCESS, table SEMESTER, table CLASS, table GRADE, table SUBJECT, table SUBJECT TRANSCRIPT, table TYPE OF INSPECTION, table CURRICULUM, table SUBJECT TRANSCRIPT DETAIL, table COURSE SUMMARY
+ Abstract properties: IDCourseSummary
+ Logic Diagram:



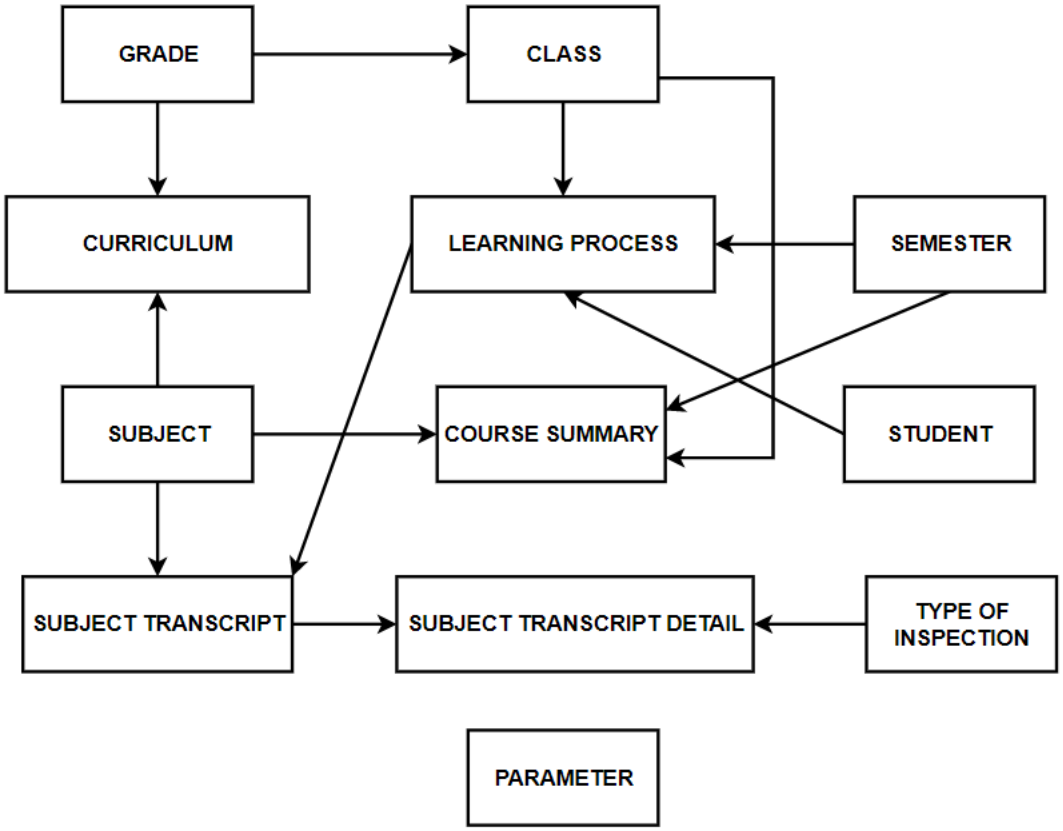


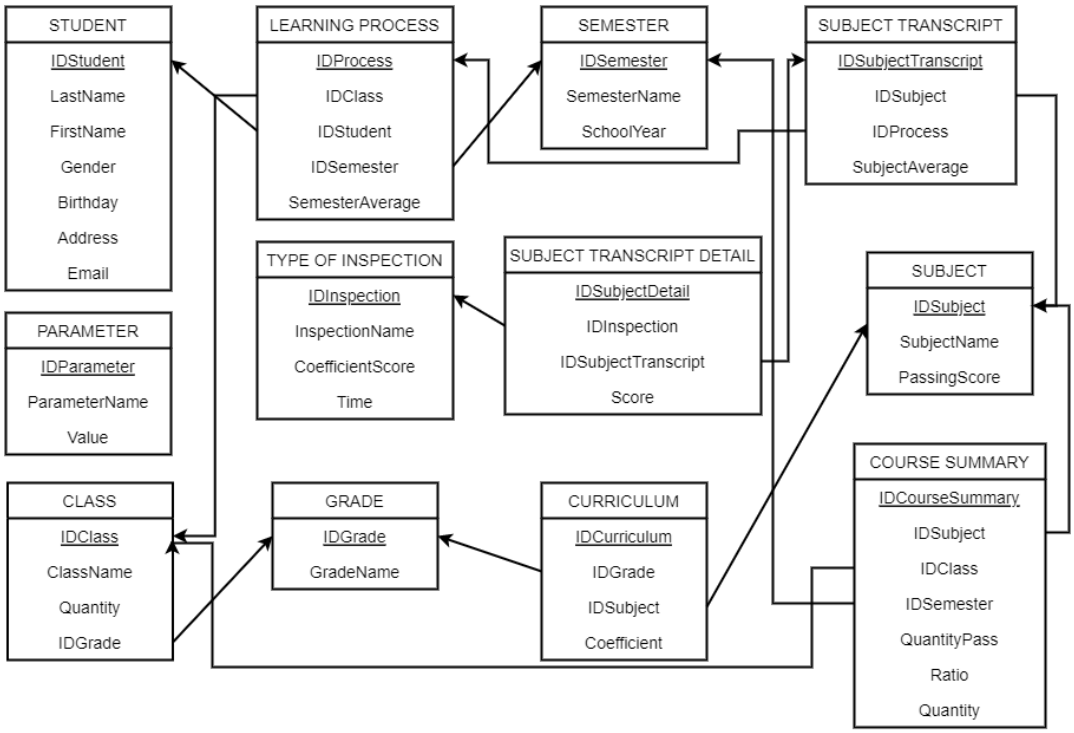
– Data design with evolution:

QĐ5: Students pass the subject/pass if the average score is ≥ 5.

+ Related Rule: Rule 5
+ Rule change Data Flow Diagram: Diagram 2.4.6.4
+ New properties: PassingScore
+ New parameters: None
+ Data design: table STUDENT, table PARAMETER, table LEARNING PROCESS, table SEMESTER, table CLASS, table GRADE, table SUBJECT, table SUBJECT TRANSCRIPT, table TYPE OF INSPECTION, table CURRICULUM, table SUBJECT TRANSCRIPT DETAIL, table COURSE SUMMARY
+ Abstract properties: None

+ Logic Diagram:



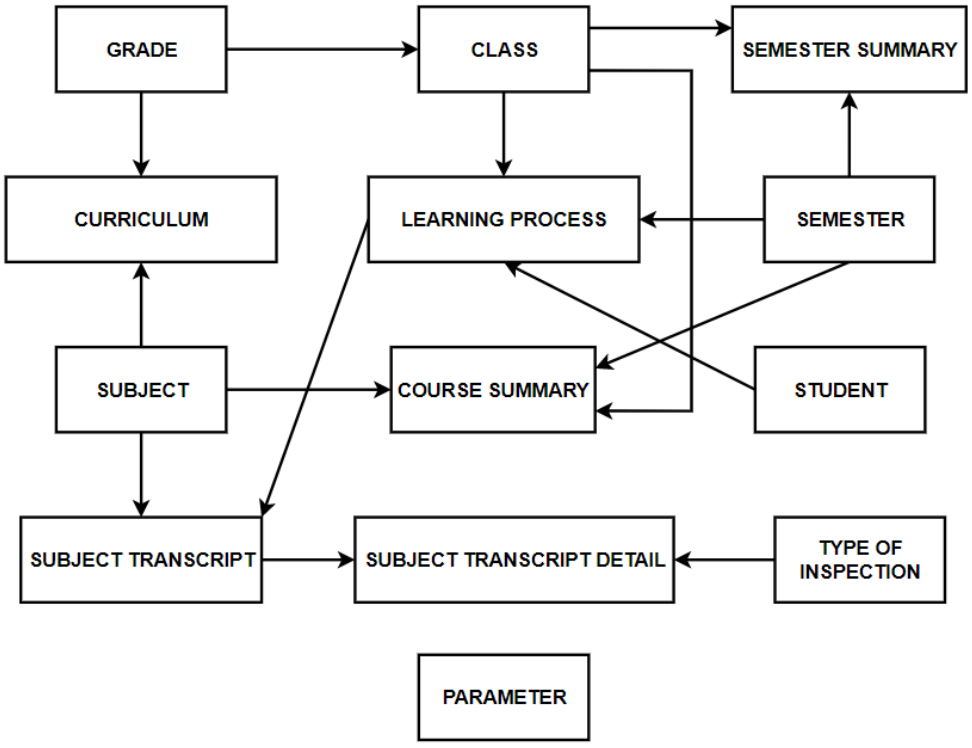


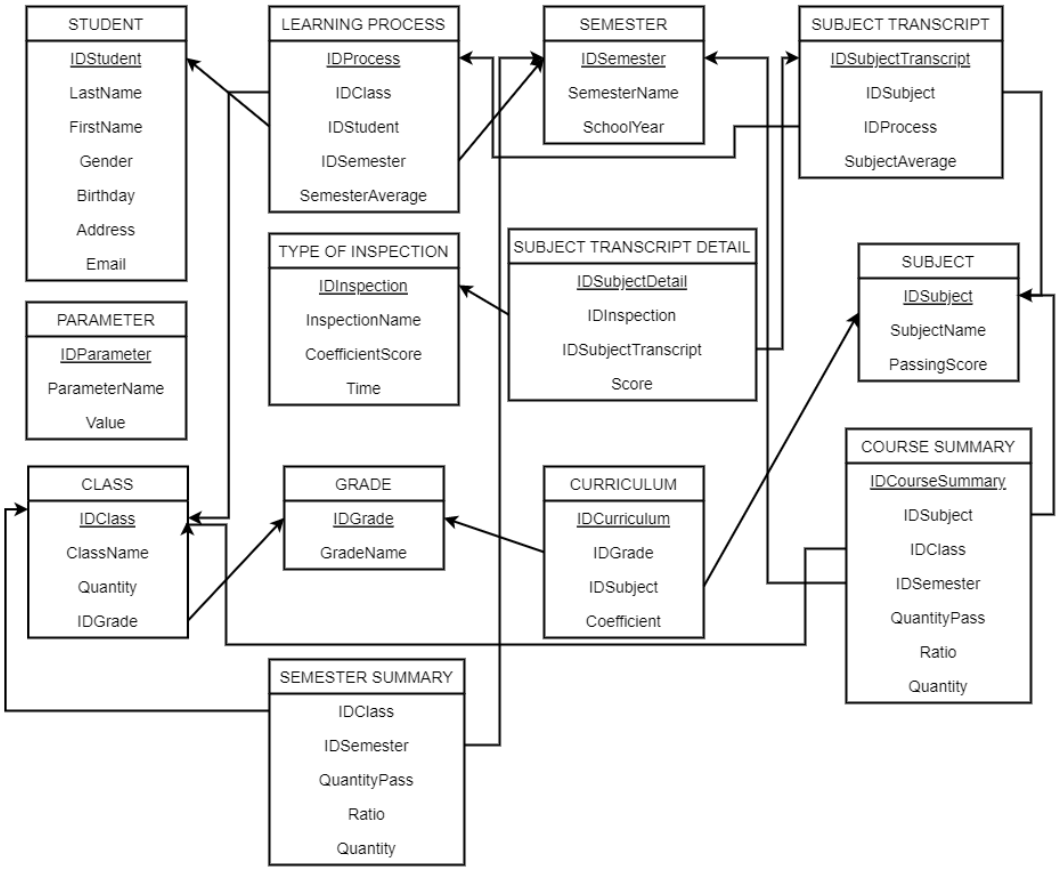
6.2.1.6. Make a semester summary report request

– Data design with correctness:

BM5.2:	Semester End Report			
Semester:……				
No.	**Class**	**Quantity**	**Number of Passes**	**Ratio**
1				
2				

+ Related Form: Form 5.2
+ Data Flow Diagram: Diagram 2.4.5.2
+ New properties: QuantityPass, Ratio
+ Data design: table STUDENT, table PARAMETER, table LEARNING PROCESS, table SEMESTER, table CLASS, table GRADE, table SUBJECT, table SUBJECT TRANSCRIPT, table TYPE OF INSPECTION, table CURRICULUM, table SUBJECT TRANSCRIPT DETAIL, table COURSE SUMMARY, table SEMESTER SUMMARY
+ Abstract properties: None
+ Logic Diagram:





- Data design with evolution:
  + Related Rule: None
  + Rule change Data Flow Diagram: None
  + New attributes: None
  + New parameters: None
  + Data design: table STUDENT, table PARAMETER, table LEARNING PROCESS, table SEMESTER, table CLASS, table GRADE, table SUBJECT, table SUBJECT TRANSCRIPT, table TYPE OF INSPECTION, table CURRICULUM, table SUBJECT TRANSCRIPT DETAIL, table COURSE SUMMARY, table SEMESTER SUMMARY
  + Abstract properties: None
  + Logic Diagram: None

6.2.2. List of tables in the diagram

No.	Table name	Explain
1	STUDENT	Student records
2	CLASS	Class information
3	GRADE	Information of grades
4	LEARNING PROCESS	Student learning process at school
5	SEMESTER	Semester
6	SUBJECT	List of subjects
7	SUBJECT TRANSCRIPT	Scoreboard of subjects
8	TYPE OF INSPECTION	Test form and score factor
9	SUBJECT TRANSCRIPT DETAIL	Details of the subject score sheet
10	CURRICULUM	Curriculum of all grades
11	COURSE SUMMARY	Course summary report
12	SEMESTER SUMMARY	Semester summary report
13	PARAMETER	Parameter table

6.2.3. Description of each data table

6.2.3.1. Table STUDENT

No.	Attribute name	Datatypes	Constraint	Explain
1	IDStudent	int identity	PRIMARY KEY	Student ID
2	LastName	nvarchar(40)	NOT NULL	Last name
3	FirstName	nvarchar(10)	NOT NULL	First name
4	Gender	nvarchar(20)	NOT NULL	Sexual
5	Birthday	date	NOT NULL	Date of birth
6	Address	nvarchar(100)	NOT NULL	Address
7	Email	varchar(40)	NOT NULL	Email

6.2.3.2. Table LOP

No.	Attribute name	Datatypes	Constraint	Explain
1	IDClass	int identity	PRIMARY KEY	Class ID
2	ClassName	nvarchar(100)	NOT NULL	Class name
3	Quantity	int		Quantity of class
4	IDGrade	int	FOREIGN KEY	Grade ID

6.2.3.3. Table GRADE

No.	Attribute name	Datatypes	Constraint	Explain
1	IDGrade	int identity	PRIMARY KEY	Grade ID
2	GradeName	nvarchar(100)	NOT NULL	Grade name

6.2.3.4. Table LEARNING PROCESS

No.	Attribute name	Datatypes	Constraint	Explain
1	IDProcess	int identity	PRIMARY KEY	Learning process ID
2	IDClass	int	FOREIGN KEY	Class ID
3	IDStudent	int	FOREIGN KEY	Student ID
4	IDSemester	int	FOREIGN KEY	Semester ID
5	SemesterAverage	float	NOT NULL	GPA of the semester

6.2.3.5. Table SEMESTER

No.	Attribute name	Datatypes	Constraint	Explain
1	IDSemester	int identity	PRIMARY KEY	Semester ID
2	SemesterName	nvarchar(100)	NOT NULL	Semester name
3	SchoolYear	int	NOT NULL	School year

6.2.3.6. Table SUBJECT

No.	Attribute name	Datatypes	Constraint	Explain
1	IDSubject	int identity	PRIMARY KEY	Subject ID
2	SubjectName	nvarchar(100)	NOT NULL	Subject name
3	PassingScore	float	Default=5	Subject passing score

6.2.3.7. Table SUBJECT TRANSCRIPT

No.	Attribute name	Datatypes	Constraint	Explain
1	IDSubjectTranscript	int identity	PRIMARY KEY	Subject transcript ID
2	IDSubject	int	FOREIGN KEY	Subject ID
3	IDProcess	int	FOREIGN KEY	Learning process ID
4	SubjectAverage	float	NOT NULL	GPA of the subject

6.2.3.8. Table TYPE OF INSPECTION

No.	Attribute name	Datatypes	Constraint	Explain
1	IDInspection	int identity	PRIMARY KEY	Test type ID
2	InspectionName	nvarchar(200)	NOT NULL	Name of test type
3	CoefficientScore	float	NOT NULL	Score factor
4	Time	int	NOT NULL	Test time

6.2.3.9. Table SUBJECT TRANSCRIPT DETAIL

No.	Attribute name	Datatypes	Constraint	Explain
1	IDSubjectDetail	int identity	PRIMARY KEY	Detail of subject score sheet ID
2	IDInspection	int	FOREIGN KEY	Test type ID
3	IDSubjectTranscript	int	FOREIGN KEY	Subject transcript ID
4	Score	float	NOT NULL	Corresponding score

6.2.3.10. Table CURRICULUM

No.	Attribute name	Datatypes	Constraint	Explain
1	IDCurriculum	int identity	PRIMARY KEY	Curriculum ID
2	GradeID	int	FOREIGN KEY	Grade ID
3	SubjectID	int	FOREIGN KEY	Subject ID
4	Coefficient	float	NOT NULL	Coefficient of the subject

6.2.3.11. Table COURSE SUMMARY

No.	Attribute name	Datatypes	Constraint	Explain
1	IDCourseSummary	int identity	PRIMARY KEY	Course Summary ID
2	IDSubject	int	FOREIGN KEY	Subject ID
3	IDClasss	int	FOREIGN KEY	Class ID
4	IDSemester	int	FOREIGN KEY	Semester ID
5	QuantityPass	int	NOT NULL	Number of passing subjects
6	Ratio	float	NOT NULL	Pass rate = QuantityPass / count(IDStudent)
7	Quantity	int	NOT NULL	Quantity of class

6.2.3.12. Table SEMESTER SUMMARY

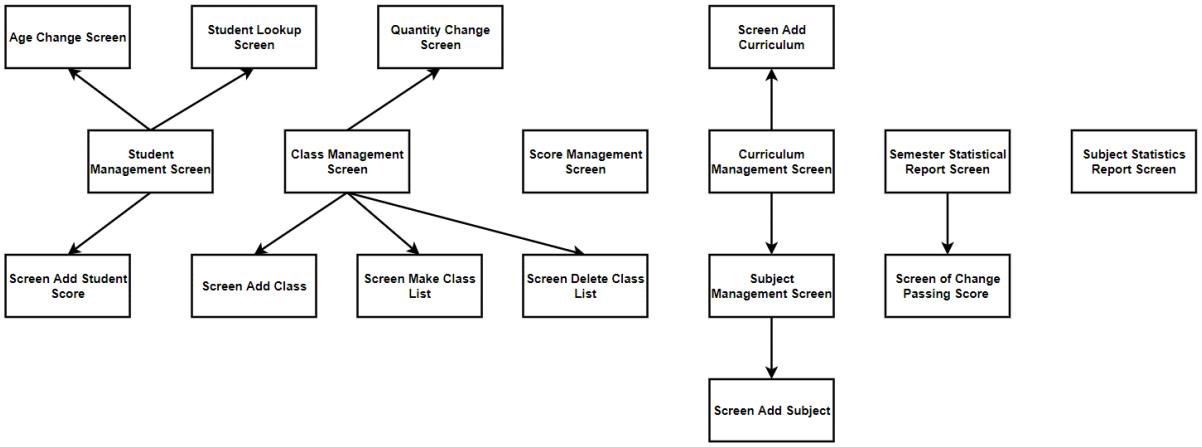
No.	Attribute name	Datatypes	Constraint	Explain
1	IDClass	int	FOREIGN KEY	Class ID
2	IDSemester	int	FOREIGN KEY	Semester ID
3	QuantityPass	int	NOT NULL	Number of passes
4	Ratio	float	NOT NULL	Pass rate
5	Quantity	int	NOT NULL	Quantity of class

6.2.3.13. Table PARAMETER

No.	Attribute name	Datatypes	Constraint	Explain
1	IDParameter	int identity	PRIMARY KEY	Parameter ID
2	ParameterName	nvarchar(100)	NOT NULL	Parameter name
3	Value	float	NOT NULL	The value of the parameter

CHAPTER 7: USER INTERFACE DESIGN

7.1. Screen link diagram



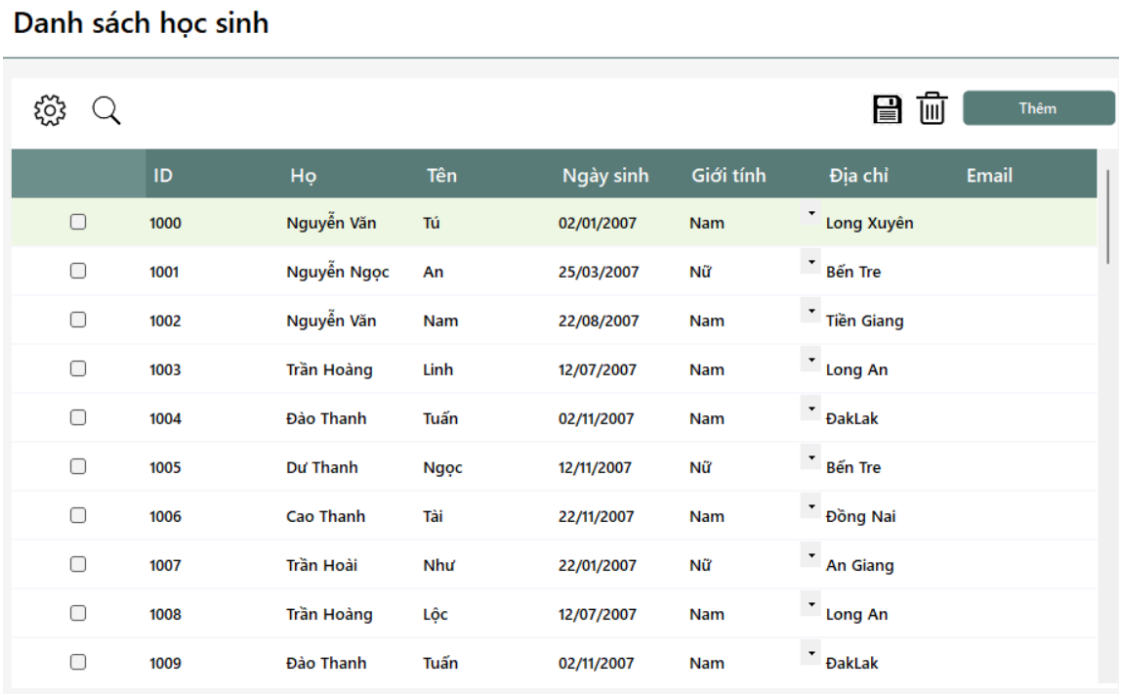
7.2. List of screens

No.	Screen	Screen Type	Function
1	Student management Screen	Input Screen	Export student list
2	Age change Screen	Input Screen	Change the minimum age, maximum age of a student
3	Student lookup Screen	Lookup Screen	Allows input of search criteria and display of search results
4	Students add Screen	Input Screen	Allows input and storage of student information
5	Class management Screen	Input Screen	Export the list of classes
6	Quantity change Screen	Input Screen	Change the rule on the maximum quantity of each class
7	Add class Screen	Input Screen	Allows inputing and storing information of classes
8	Make class list Screen	Input Screen	Allows adding students who do not have classes to existing classes
9	Updating class list Screen	Input Screen	Permission to remove a student from their current class
10	Curriculum management Screen	Input Screen	Export the list of subjects included in the grades and the coefficients corresponding to the subject

11	Subject management Screen	Input Screen	Export the list of subjects and corresponding passing scores
12	Subjects add Screen	Input Screen	Allows entry of subjects and corresponding passing scores, storing information of subjects
13	Curriculum add Screen	Input Screen	Allows adding subjects and corresponding coefficients of that subject to grades, storing information of curriculum
14	Score management Screen	Input Screen	Export the list of scores of each student's subjects in each class
15	Subject statistics report Screen	Report	Present the number of passes and the pass rate for each respective subject in the classes
16	Semester statistics report Screen	Report	Present the number of passes and the pass rate of the classes in the respective semester
17	Semester passing score change Screen	Input Screen	Changes in rule on passing score

7.3. Description of screens

7.3.1. Student management screen



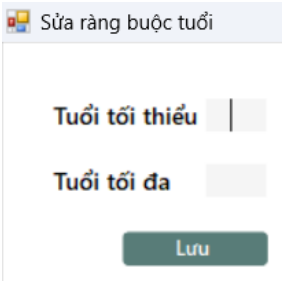
– Description of objects on the screen:

No.	Name	Type	Constraint	Function
1	btSetting	button		Allow opening screen to change age
2	btSearch	button		Allow to open the student lookup screen
3	btSave	button		Allows updating student information
4	btDelete	button		Allows deleting selected student information
5	btAdd	button		Allows opening the screen to add students
6	GridStudent	DataGridView		Contains student list information

– List of events and actions on the screen:

No.	Event	Action
1	Select the Settings button	Display the age change screen
2	Select the Search button	Display the student lookup screen
3	Select the Save button	Update student information
4	Select the Delete button	Delete information of selected students
5	Select the Add button	Display the student add screen
6	LTextbox_Click on GridStudent	Change the student's first name, last name, gender, address, email
7	LCheckbox_Click on GridStudent	Select the students you want to remove from the list
8	Initialize screen	Set the Controls and display information to the default state

7.3.2. Age change screen



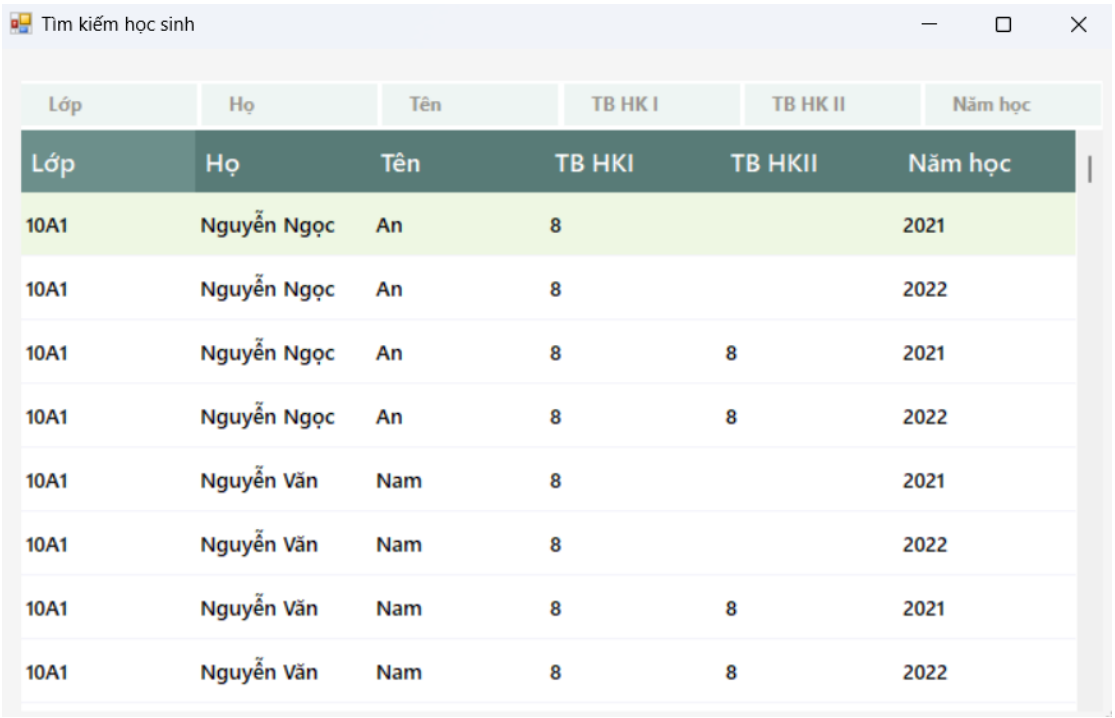
– Description of objects on the screen:

No.	Name	Type	Constraint	Function
1	lbMinAge	label		Notice where to enter the minimum age
2	lbMaxAge	label		Notice where to enter maximum age
3	txtMinAge	textbox	Enter numbers from 0-9	Enter new minimum age
4	txtMaxAge	textbox	Enter numbers from 0-9	Enter new maximum age
5	btSave	button		Newly updated minimum age and maximum age

– List of events and actions on the screen:

No.	Event	Action
1	Select the Save button	Update the new minimum and maximum age of the admitted students and save that value in the data
2	Initialize screen	Set the Controls and display information to the default state

7.3.3. Student lookup screen



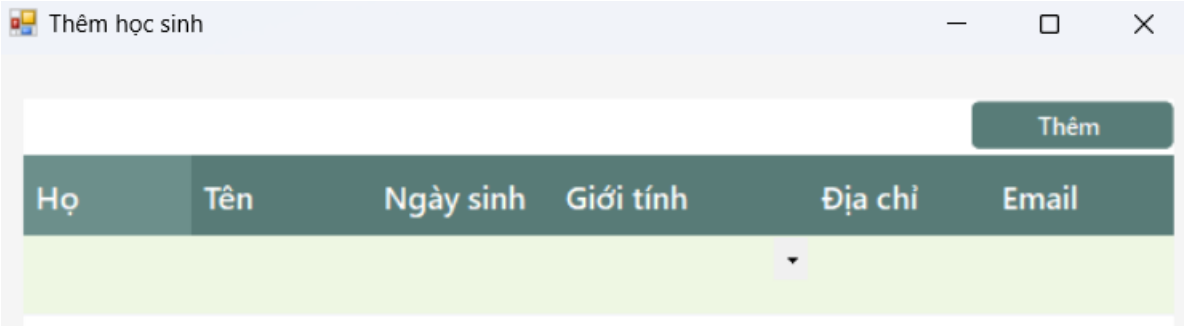
– Description of objects on the screen:

No.	Name	Type	Constraint	Function
1	txtLop	textbox		Enter the class name of the student to be searched
2	txtHo	textbox		Enter the student's last name to search
3	txtTen	textbox		Enter the name of the student to be searched
4	txtTBI	textbox		Enter the GPA of the first term of the student to be searched
5	txtTBII	textbox		Enter the GPA of the second term of the student to be searched
6	txtNamHoc	textbox		Enter the school year that there are students looking for
7	SearchMultiRow	DataGridView		Contains information of the student to be searched

– List of events and actions on the screen:

No.	Event	Action
1	Enter the value in txtLop	Display on the DataGridView the students whose class corresponds to the value entered
2	Enter the value in txtHo	Display on the DataGridView the students whose last name corresponds to the value entered
3	Enter the value in txtTen	Display on the DataGridView the students whose first name corresponds to the value entered
4	Enter the value in txtTBI	Display on the DataGridView the students whose GPA of first term corresponds to the value entered
5	Enter the value in txtTBII	Display on the DataGridView the students whose GPA of second term correspons to the value entered
6	Enter the value in txtNamHoc	Display on the DataGridView the students in the list by year corresponding to the entered value
7	Initialize screen	Set the Controls and display information to the default state

7.3.4. Students add screen



– Description of objects on the screen:

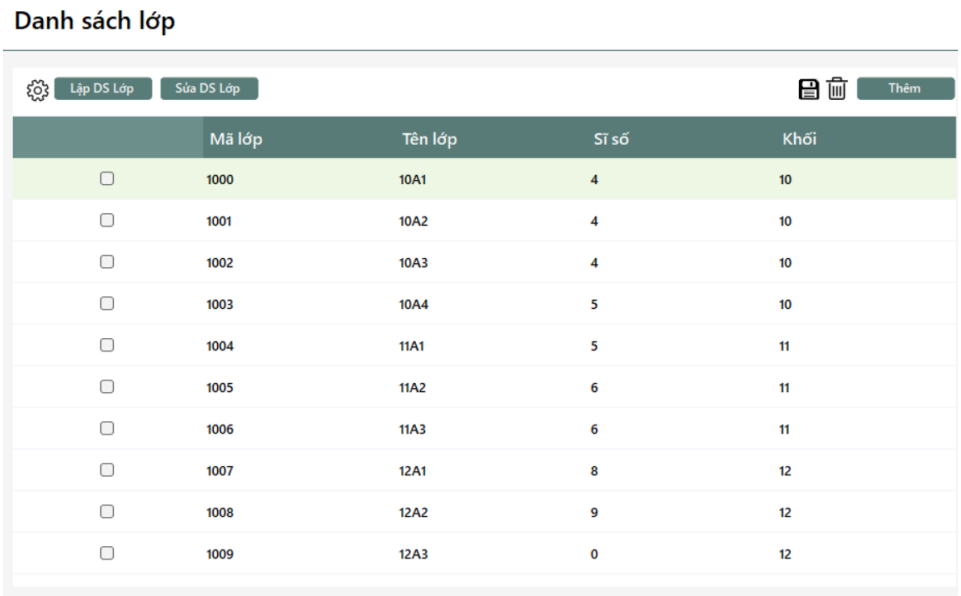
No.	Name	Type	Constraint	Function
1	GridAddStudent	DataGridView		Where to enter information students need to add
2	txtHo	DataGridViewTextboxColumn		Enter the student's last name to be added
3	txtTen	DataGridViewTextboxColumn		Enter the name of the student to be added
4	txtRightBirth	DataGridViewTextboxColumn	Enter the corresponding value of day, month, year	Enter the student's date of birth to add
5	cbGioiTinh	DataGridViewComboboxColumn		Select the gender of the student to add

6	txtDiaChi	DataGridViewTextboxColumn		Enter the address of the student to be added
7	txtEmail	DataGridViewTextboxColumn		Enter the student's email to be added
8	btAdd	Button		Add new student informati on to the student list

– List of events and actions on the screen:

No.	Event	Action
1	Select the Add button	Save value information of last name, first name, date of birth, gender, address, email just entered into the data
2	Initialize screen	Set the Controls and display information to the default state

7.3.5. Class management screen



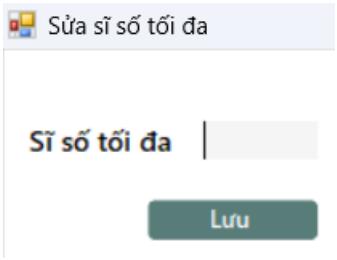
– Description of objects on the screen:

No.	Name	Type	Constraint	Function
1	btSetting	button		Allows opening the screen to change the maximum quantity of a class
2	btLapDanhSach	button		Allows opening the class list making screen
3	btSuaDanhSach	button		Allows opening the class list updating screen
4	btSave	button		Allows updating information about classes
5	btDelete	button		Allows deleting selected classes
6	btAdd	button		Allow to open the class add screen
7	GridClass	DataGridView		Contains class list information

– List of events and actions on the screen:

No.	Event	Action
1	Select the Settings button	Display on the screen change the maximum quantity of a class
2	Select the button LapDanhSach	Display on the list making screen
3	Select the button SuaDanhSach	Display on the list updating screen
4	Select the Save button	Update class information
5	Select the Delete button	Remove information of selected class
6	Select the Add button	Display the class add screen
7	LTextbox_Click on GridClass	Change class name, class grade information
8	LCheckbox_Click on GridStudent	Select the classes you want to remove from the list
9	Initialize screen	Set the Controls and display information to the default state

7.3.6. Quantity change screen



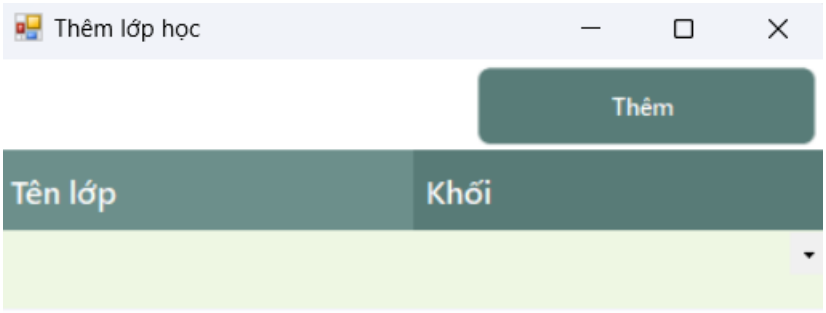
– Description of objects on the screen:

No.	Name	Type	Constraint	Function
1	lbMaSISO	label		Notice where to enter the new maximum quantity value
2	txtMaSISO	textbox	Enter numbers from 0-9	Enter the new maximum quantity value
3	BtSave	button		Update the maximum quantity of a class

– List of events and actions on the screen:

No.	Event	Action
1	Select the Save button	Update the new maximum quantity of a class and save that value in the data
2	Initialize screen	Set the Controls and display information to the default state

7.3.7. Class add screen



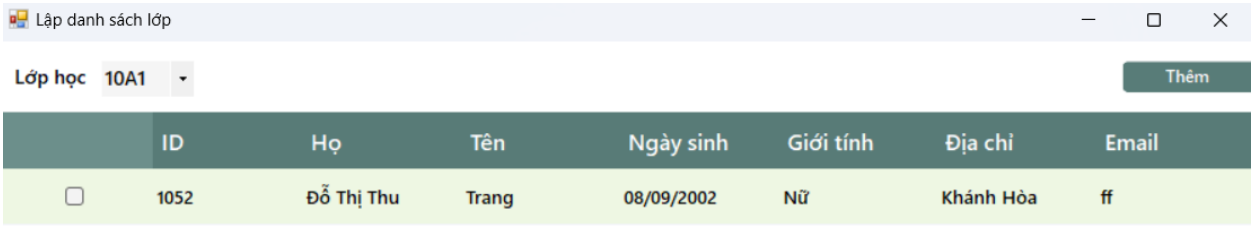
– Description of objects on the screen:

No.	Name	Type	Constraint	Function
1	GridAddClass	DataGridView		Where to enter class information to be added
2	txTenLop	DataGridViewTextboxColumn		Enter additional class name
3	cbTenKhoi	DataGridViewComboboxColumn		Select the grade of the added class
4	btAdd	Button		Add new class to the list

– List of events and actions on the screen:

No.	Event	Action
1	Select the Add button	Save information about the value of the class name, newly entered grade in the data
2	Initialize screen	Set the Controls and display information to the default state

7.3.8. Class list making screen



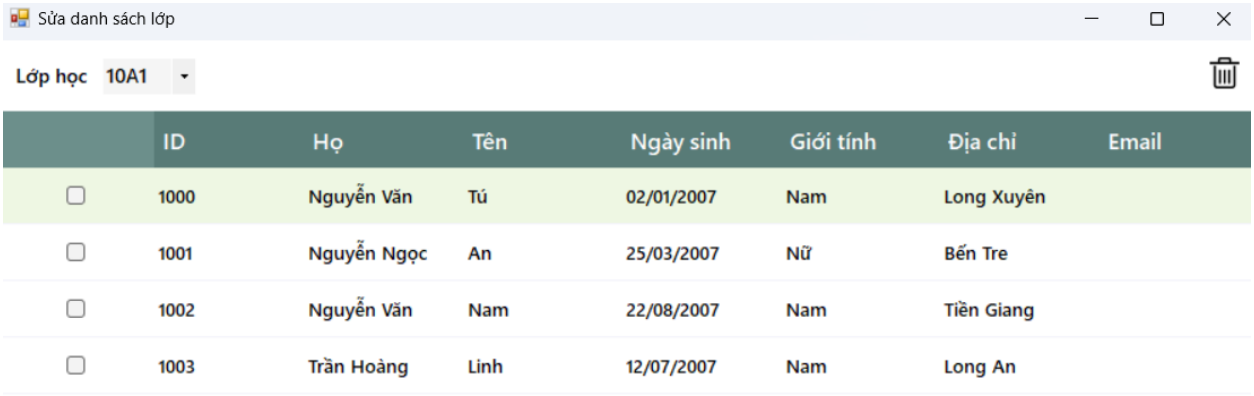
– Description of objects on the screen:

No.	Name	Type	Constraint	Function
1	lbLophoc	label		Notice where to choose the class to add
2	cbLopHoc	comboxbox		Let the user select the class that needs to add students
3	btAdd	button		Add students to the class
4	GridStudent	DataGridView		Contains information for students who do not have a class
5	cbStudent	DataGridViewCheckboxColumn		Select the student you want to add to the class

– List of events and actions on the screen:

No.	Event	Action
1	Select the Add button	Update the information of the selected students in the selected class list
2	Select LCheckbox_Click on GridStudent	Get the student information corresponding to the selected row
3	Initialize screen	Set the Controls and display information to the default state

7.3.9. Class list updating screen



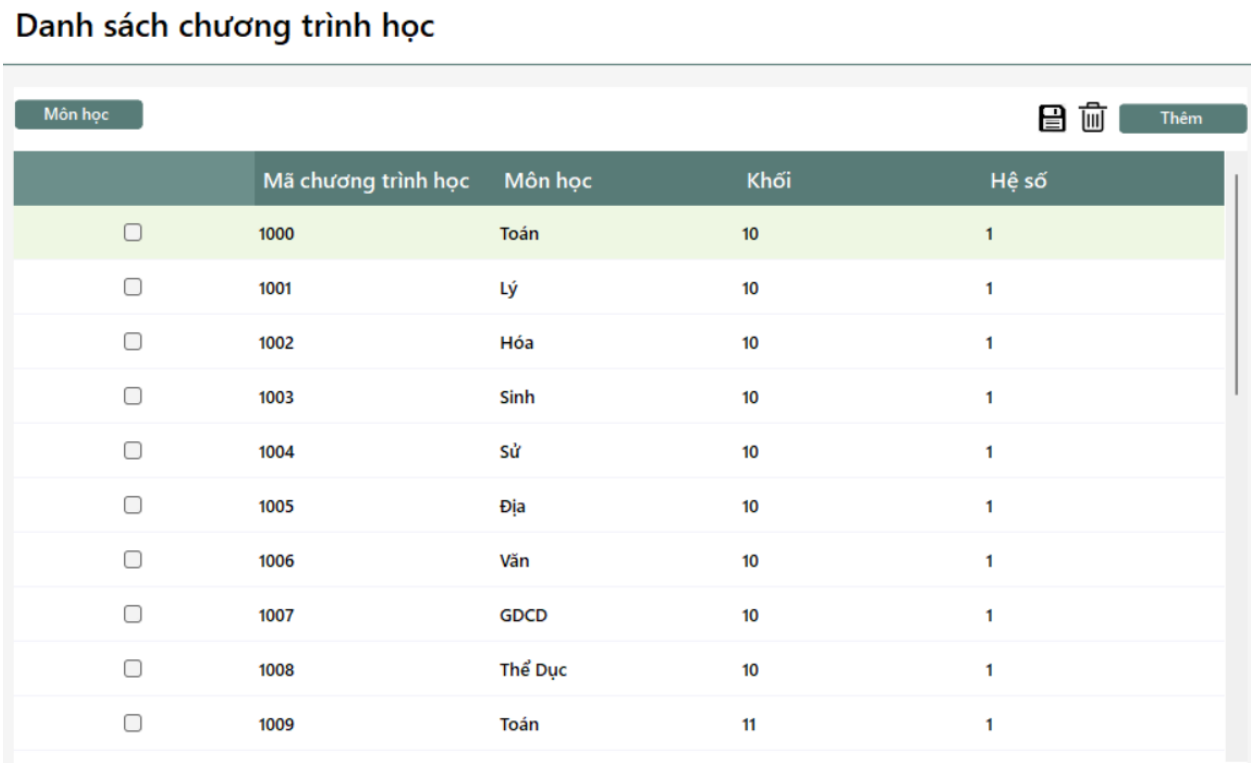
– Description of objects on the screen:

No.	Name	Type	Constraint	Function
1	lbLophoc	label		Notice where to select the class to be edited
2	cbLopHoc	comboxbox		Select the class to edit the list
3	btDelete	button		Remove selected student from class
4	GridStudent	DataGridView		Contains information about students currently in the selected class
5	cbStudent	DataGridViewCheckboxColumn		Select the student you want to remove from the class

– List of events and actions on the screen:

No.	Event	Action
1	Select the Delete button	Remove selected students from the respective class list
2	Select LCheckbox_Click on GridStudent	Get the student information corresponding to the selected row
3	Initialize screen	Set the Controls and display information to the default state

7.3.10. Curriculum management screen



– Description of objects on the screen:

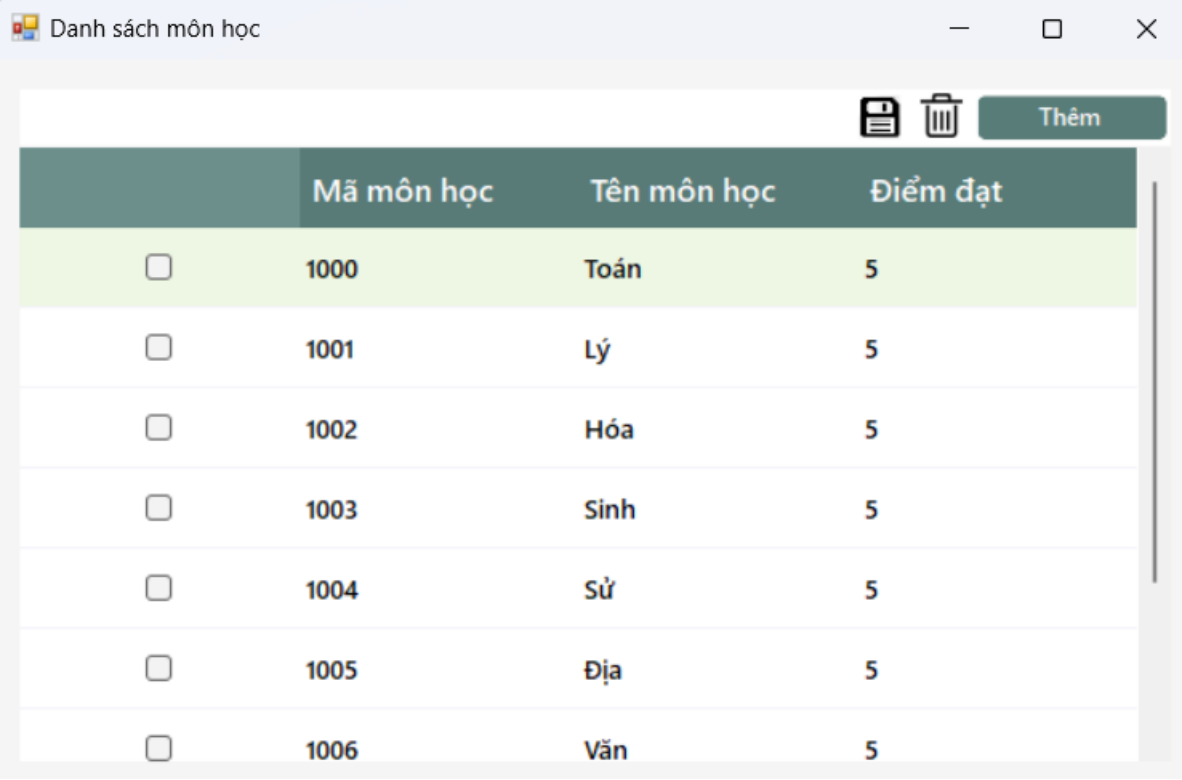
No.	Name	Type	Constraint	Function
1	btMonHoc	button		Allows opening the subject management screen
2	btSave	button		Allows updating of curriculum information
3	btDelete	button		Allows deletion of selected curriculums
4	btAdd	button		Allows opening the screen to add a curriculum

5	DataGridCTH	DataGridView		Contains information about the subjects included in the grades and the corresponding coefficients of each subject

– List of events and actions on the screen:

No.	Event	Action
1	Select button MonHoc	Display on the subject management screen
2	Select the Save button	Update the curriculum of the grades
3	Select the Delete button	Delete selected curriculum information
4	Select the Add button	Show curriculum adding screen
5	LCheckbox_Click on GridStudent	Select the curriculums you want to remove from the list
6	Initialize screen	Set the Controls and display information to the default state

7.3.11. Subject management screen



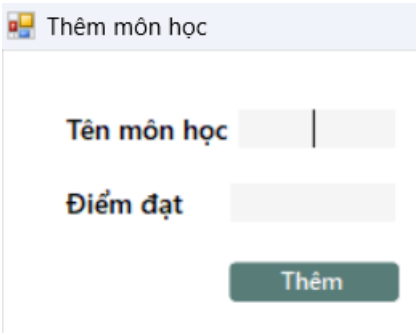
– Description of objects on the screen:

No.	Name	Type	Constraint	Function
1	btSave	button		Allows updating of course information
2	btDelete	button		Allows deleting subjects
3	btAdd	button		Allows opening the screen to add more subjects
4	DataGridMonHoc	DataGridView		Contains information about subjects

– List of events and actions on the screen:

No.	Event	Action
1	Select the Save button	Update course information
2	Select the Delete button	Delete selected subject information
3	Select the Add button	Display subjects adding screen
4	LTextbox_Click on GridClass	Change the subject name and passing score of the subject
5	LCheckbox_Click on GridStudent	Select the subjects you want to remove from the list
6	Initialize screen	Set the Controls and display information to the default state

7.3.12. Subject adding screen



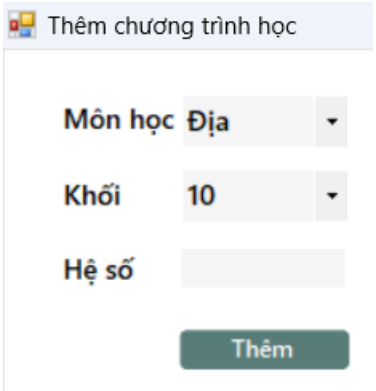
– Description of objects on the screen:

No.	Name	Type	Constraint	Function
1	lbTenMonHoc	label		Notice where to enter subject name
2	lbDiemdata	label		Announcement of the place to enter the passing score corresponding to the subject
3	txtTenMonHoc	textbox		Enter new subject name
4	txtDiemdata	textbox	Enter digits from 0-9 and value from 1-10	Enter the passing score of the added subject
5	btAdd	button		Add new subject information to the subject list

– List of events and actions on the screen:

No.	Event	Action
1	Select the Add button	Store information including the name of the new subject and the passing score corresponding to that subject in the subject list
2	Initialize screen	Set the Controls and display information to the default state

7.3.13. Curriculum adding screen



– Description of objects on the screen:

No.	Name	Type	Constraint	Function
1	lbMonhoc	label		Notice where to enter the subject name
2	lbKhoi	label		Notice where to enter grade name
3	lbHeso	label		Notice where to enter coefficients for subjects
4	cbMonhoc	combo box		Choose from the list of subjects
5	cbKhoi	combo box		Select from grades list
6	txtHeso	textbox	Enter the digits from 0-9, and the value from 1-10	Enter the coefficient of the corresponding selected subject

– List of events and actions on the screen:

No.	Event	Action
1	Select the Add button	Store new curriculum information in the data
2	Initialize screen	Set the Controls and display information to the default state

7.3.14. Score management screen

– Description of objects on the screen:

No.	Name	Type	Constraint	Function
1	lbLophoc	label		Announce where to select the class you want to display student scores
2	lbMonhoc	label		Announcement where to select the subject to display the score
3	lbHocky	button		Announcement of where to choose the semester and school year for the class
4	cbLophoc	combo box		Select a class from the class list
5	cbMonhoc	comboxbox		Choose a subject from the list of subjects
6	cbHocky	combo box		Select the semester and school year to display scores
7	GridScore	DataGridView		Contains information about the selected subject scores of students in the selected class and semester
8	cbScore	DataGridViewComboBoxColumn		Select the student who wants to add or change the score
9	tx15p	DataGridViewTextBoxColumn		Add, remove, or change the student's 15-minute score respectively
10	tx1tiet	DataGridViewTextBoxColumn		Add, remove or change the corresponding student's 1-period score
11	txhocki	DataGridViewTextBoxColumn		Add, remove or change the respective student's semester scores
12	btSave	button		Update student's score
13	btDelete	button		Delete the score of the selected student
14	btAdd	button		Add scores for selected students

– List of events and actions on the screen:

No.	Event	Action
1	LTextbox_Click on GridSocre	Add, delete, change the student's 15-minute score, 1-period score, semester score
2	LCheckBox_Click on GridScore	Select the student you want to add, remove or change grades scores
3	Select the Add button	Add scores for selected students
4	Select the Delete button	Delete the corresponding score of the selected student
5	Select the Save button	Update the students' scores and recalculate the average score, store it in the score list
6	Initialize screen	Set the Controls and display information to the default state

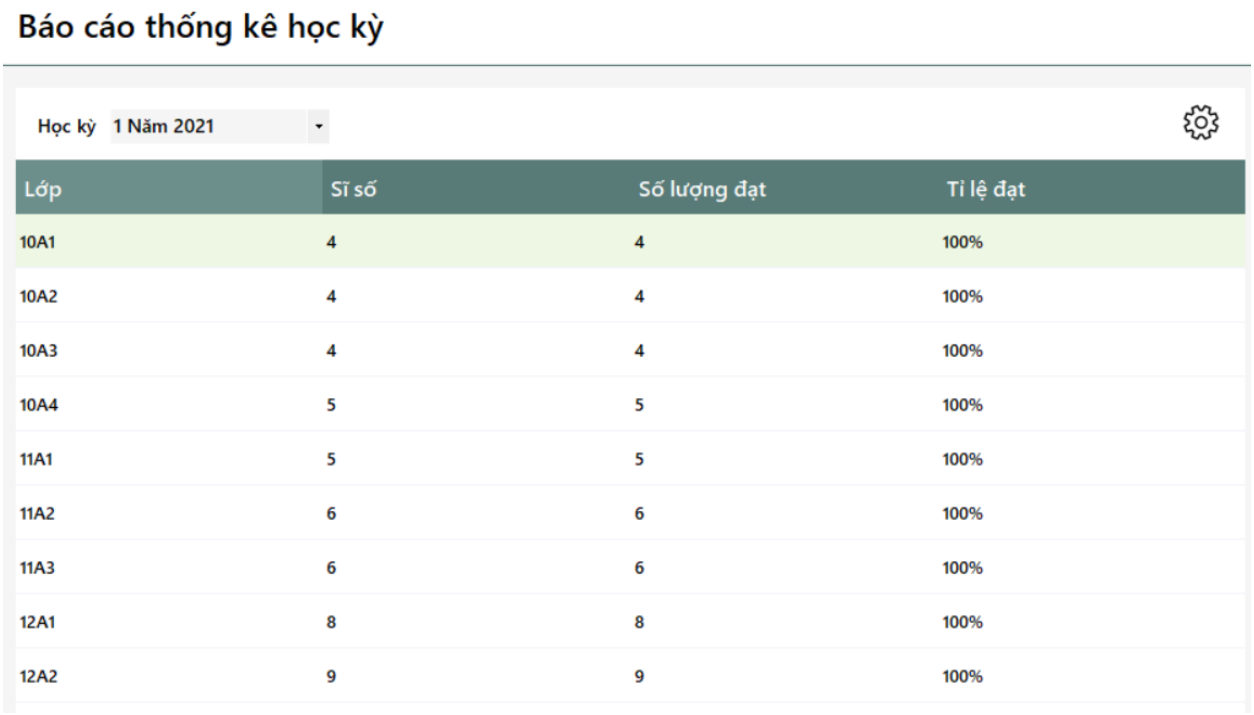
7.3.15. Subject statistics report screen

– Description of objects on the screen:

No.	Name	Type	Constraint	Function
1	lbMonhoc	label		Announcement of the place to choose the subject to see the report result
2	lbHocky	label		Announcement of the place to choose the semester
3	cbMonhoc	combo box		Select the subject to see the report result in the subject list
4	cbHocky	combo box		Select the corresponding semester and school year
5	DataGridReport	DataGrid View		Display statistics report for the subject
6	lbMonhoc	label		Announcement of the place to choose the subject to see the report result

– List of events and actions on the screen: None

7.3.16. Semester statistics report screen



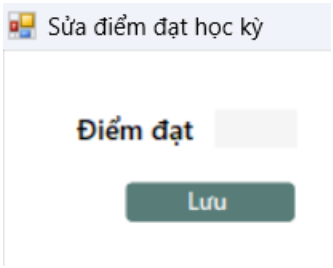
– Description of objects on the screen:

No.	Name	Type	Constraint	Function
1	lbHocky	label		Announcement where to select the semester to see the report results
2	cbHocky	combo box		Select the semester and school year to see the report results
3	DataGridReport	DataGrid View		Show statistics report for the semester
4	btSetting	button		Allow to open the screen to change the semester passing score

– List of events and actions on the screen:

No.	Event	Action
1	Select the Settings button	Display the semester passing score changing screen
2	Initialize screen	Set the Controls and display information to the default state

7.3.17. Semester passing score change screen



– Description of objects on the screen:

No.	Name	Type	Constraint	Function
1	lbDiemdata	label		Announcement where to enter the new score
2	txtMaSISO	textbox	Enter a digit from 0-9 and a value from 0-10	Enter new score value
3	BtSave	button		Update the passing score for the semester

– List of events and actions on the screen:

No.	Event	Action
1	Select the Save button	Update the new passing score value for the semester and save it in the data
2	Initialize screen	Set the Controls and display information to the default state

CHAPTER 8: INSTALLATION AND TESTING

No.	Function	Completion (%)	Note
1	Student list management	100	Output a list containing information about students
2	Changes in minimum and maximum age rule	100	
3	Look up students	95	Search by last name, first name; by class or by GPA of the semester
4	Adding students	100	
5	Class list management	100	
6	Change the rule on maximum quantity of a class	100	Maximum number of students a class can have
7	Add class	100	
8	Make a class list	90	Add students without classes to classes
9	Update class list	90	
10	Curriculum management	90	Output information about which subjects a grade has studied and the coefficient of each subject
11	Subject management	100	Output information about subjects and can update the rule on passing scores for that subject
12	Adding subjects	100	
13	Adding curriculum	95	Add subjects to the curriculum of each grade
14	Subject score management	95	You can update, add or delete a certain subject's score column for students and automatically recalculate the GPA
15	Course statistical report	95	Displays the number of passes and calculates the percentage of students who have met the subject's score
16	Semester statistical report	95	Display the number of passes and calculate the percentage of students passing in the semester
17	Change rule on passing scores/subjects	100	

CHAPTER 9: COMMENTS AND CONCLUSION

In this section, our team would like to present the advantages and limitations of the software after installing and testing:

❖ **About advantages:**

- The software program implemented by the team has basically completed in a relatively complete way the requirements and regulations of the given topic, meeting the correctness and evolution.
- Time to access and update is relatively stable, meeting the effectiveness of software quality.
- Besides, the software also has an interface arranged in a relatively clear, intuitive and consistent layout, and the software's features are easy to use for users.
- The software is designed in a 3-layer model and uses object-oriented methods, packaged according to the requirements and corresponding regulations, so it will be easy to maintain and upgrade if there are errors during use.
- Meet the progress of the project.

❖ **Regarding limitations:**

- In the process of building the project, the team encountered many difficulties in data design and software construction due to having to approach a new programming language and a new platform resulting in some features not yet fully completed, for example: If the size of age is changed, then the current students whose age is not in accordance with the regulations will still be unaffected,…
- The software has only been able to adapt to some of the basic changing requirements of users but has not yet fulfilled the more complex requirements, leading to the software's evolution not being really good.
- The interface is clear and intuitive to the user, but it is still simple, have not been optimized and impressed with the user.

Thus, the software has relatively completed the requirements set forth, meeting some basic and necessary features for users, but there will still be limitations and shortcomings due to using new programming language. In addition, in the process of developing and implementing the project, the team also coordinated to clearly divide the work, regularly held meetings to exchange and contribute ideas to perfect the product, however some team members have not kept up with the work of others, leading to shortcomings in the software testing process.

CHAPTER 10: DEVELOPMENT DIRECTION

- Try to overcome the limitations of the software.
- Re-edit the interface, try to build a more beautiful interface and impress users.
- Develop software to better platforms.
- Build some new functions so that the software becomes more and more practical and can be used in some schools.

REFERENCES

[1] Tài liệu lập trình Winform C# cơ bản để tự học hiện nay, https://codegym.vn/blog/2021/04/05/tai-lieu-lap-trinh-winform-c/

[2] Winform C# programming tutorials courses

[3] Lecture slides and practice materials included in the Introduction to Software Engineering course (SE104)