PROJECT REPORT

# **School Management**

Semester: Programming Fundamentals

Class: K22P.SE2.A.13

Group: Group 5

Instructor Võ Công Đình

Team member: Nguyễn Minh Hoàng Quốc

Vương Hưng Thành

Trần Thanh Dũng

1. Project introduction

This system will provide a few basic functional characteristics as follows:

The system has some main functions such as adding subjects, adding classes, adding teachers, adding students, adding teaching assignment schedules and some sub-functions as follows: creating tests, student attendance, notification of test or exam results, automatic grading,...

* 1. Proposed System
     + Create a system to display rosters so teachers can see which classes they teach.
     + Create a system that creates tests for teachers and creates exams for schools.
     + Create a test-taking or test-taking system for students.
     + Automatic scoring system.
     + Student grade management system.
     + The system summarizes the results.
     + Attendance system.
     + Student grade lookup system.
  2. The scope of the project to be applied

- A website for students, teachers, and school administrators.

* 1. System Name

- Teacher management system

- Student management system

- Class management system

- Division management system

- Attendance system.

- Examination system.

- Searching for information systems.

* 1. Deployment Environment

- Windows, Linux, Mac

* 1. Development Tools
* Visual Studio Code
* Draw.io Diagram
* MySQL Workbench 8.0 CE
  1. Customer Requirements
* Seller login
* Show laptops
* Create order
* Accountant login
* Show orders
* View order details
* Conﬁrm payment or Cancel order

1. Analyze System Requirements

Student information management: The system needs to manage student information including full name, date of birth, address, phone number, email, academic performance, academic results, class list, list of subjects studying, extracurricular activities, certificates earned, etc.

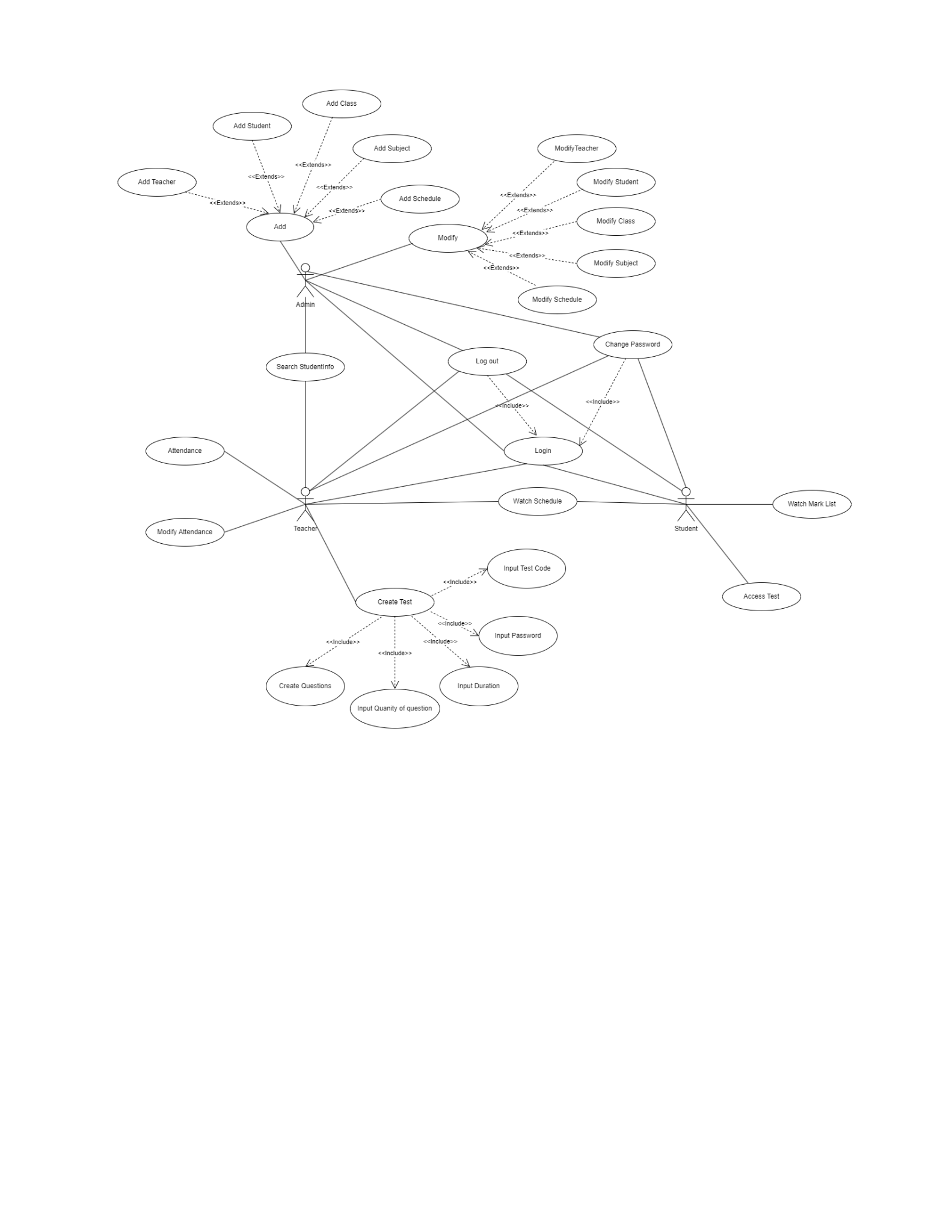
Management of lecturer information: The system needs to allow the management of lecturer information including full name, date of birth, address, phone number, email, major, education level, experience, etc.

Classroom information management: The system should allow to manage information about classes including class name, homeroom teacher, list of students in class, subjects taught in class, etc.

Score management: The system should allow the management of student grades including grades for each subject and the student's overall academic performance.

Teaching schedule management: The system needs to manage the teacher's teaching schedule and information about the lessons of each class.

* 1. Use Case
  2. Use Case



* 1. Login description

| Name | Login |
| --- | --- |
| ID | UC\_001 |
| Description | Được sử dụng để truy cập chương trình và xác định tính năng có thể truy cập tùy thuộc vào loại tài khoản. |
| Actor | Quản lý,giáo viên và học sinh. |
| Organizational Beneﬁts | Tính năng này được tạo ra để xác định các tính năng và dữ liệu mà Quản lý,Giáo viên và học sinh có thể truy cập trong chương trình. |
| Trigger | Đăng nhập sẽ được kích hoạt ngay sau khi chúng tôi kích hoạt chương trình. |
| Preconditions | Phải nhập đúng tên người dùng và mật khẩu,cơ sở dữ liệu phải được kết nối,tài khoản phải được cung cấp. |
| Postconditions | Success: Quản lý,giáo viên và học sinh đã đăng nhập vào hệ thống chính,Menu của chương trình phải được hiển thị.  Fail: Quản lý,giáo viên hoặc học sinh chưa đăng nhập vào hệ thống. |
| Main Course | 1. Nhập tên người dùng và mật khẩu. 2. Xác minh đăng nhập. 3. Hiển thị menu chính. |
| Alternate Courses | AC 2.1: Người dùng nhập sai tên đăng nhập,mật khẩu.   1. Lỗi hiển thị thông báo. 2. Hiện lại màn hình đăng nhập. 3. Nhập lại yêu cầu.   AC2.2: Tên đăng nhập và mật khẩu tồn tại.   1. Hiện thông báo đăng nhập thành công. 2. Hiện menu chính. |
| Exceptions | EX 1.1: Nhập tên người dùng/Mật khẩu.   1. Hiển thị: “Tên người dùng hoặc mật khẩu không hợp lệ, vui lòng nhập lại!” và yêu cầu nhập lại cho đến khi đúng. 2. Yêu cầu nhập lại tên người dùng và mật khẩu. 3. . EX 2.1: Ngắt kết nối với cơ sở dữ liệu. 4. Bước 1: Hiển thị “Lỗi!!! Không thể kết nối với cơ sở dữ liệu” 5. Yêu cầu kiểm tra lại đường truyền internet. |

* 1. Class description

| Name | Class |
| --- | --- |
| ID | UC\_002 |
| Description | Quản lý thông tin lớp học, danh sách học sinh, giáo viên chủ nhiệm và thời khóa biểu môn học của lớp học. |
| Actor | Quản lý,giáo viên và học sinh. |
| Organizational Beneﬁts | Quản lý lớp học: Cho phép giáo viên và quản lý trường học quản lý thông tin lớp học, danh sách học sinh, giáo viên chủ nhiệm và thời khóa biểu môn học.  Tối ưu hoá quy trình: Giúp giáo viên và quản lý trường học nhanh chóng tìm kiếm, cập nhật, xóa thông tin lớp học và sinh viên liên quan. |
| Trigger | Quản lý,giáo viên hoặc học sinh truy cập vào tính năng quản lý lớp học. |
| Preconditions | Hệ thống đã được triển khai và hoạt động,cơ sở dữ liệu phải được kết nối.  Quản lý,giáo viên hoặc học sinh đã đăng nhập vào hệ thống và có quyền truy cập vào tính năng quản lý lớp học. |
| Postconditions | Success:Thông tin lớp học, danh sách sinh viên, giáo viên chủ nhiệm và thời khóa biểu môn học được cập nhật trong hệ thống. |
| Main Course | 1. Giáo viên hoặc quản lý trường học chọn chức năng quản lý lớp học từ giao diện chính của hệ thống. 2. Hệ thống hiển thị danh sách các lớp học có sẵn trong hệ thống, bao gồm thông tin về ClassID, ClassName, MainTeacherID, SubjectScheduleID và số lượng sinh viên trong lớp học. 3. Giáo viên hoặc quản lý trường học có thể tìm kiếm lớp học bằng cách sử dụng tính năng Search, nhập các thông tin tìm kiếm như ClassID, ClassName, MainTeacherID, SubjectScheduleID. 4. Giáo viên hoặc quản lý trường học có thể cập nhật thông tin lớp học, danh sách sinh viên, giảng viên chủ nhiệm và thời khóa biểu môn học của lớp học bằng tính năng Update. 5. Giáo viên hoặc quản lý trường học có thể xóa lớp học hoặc sinh viên trong lớp học khỏi hệ thống bằng tính năng Remove. 6. Giáo viên hoặc quản lý trường học có thể xem thông tin chi tiết của lớp học, bao gồm danh sách sinh viên, giảng viên |
| Alternate Courses | AC3.1: 1a. Nếu giáo viên không tìm thấy kết quả tìm kiếm, hệ thống hiển thị thông báo không tìm thấy lớp học và yêu cầu giáo viên thực hiện lại tìm kiếm.  2a. Giáo viên tiếp tục nhập thông tin để tìm kiếm hoặc quay lại bước 1. |
| Exceptions | EX1:.1e. Nếu giáo viên không có quyền truy cập và quản lý thông tin lớp học, hệ thống hiển thị thông báo lỗi truy cập không được phép và yêu cầu giáo viên liên hệ với quản trị viên.  2e. Use case kết thúc.  1f. Nếu giáo viên nhập thông tin không hợp lệ hoặc thiếu thông tin cần thiết trong quá trình cập nhật hoặc thêm học sinh vào lớp học, hệ thống hiển thị thông báo lỗi và yêu cầu giáo viên nhập lại thông tin đúng định dạng hoặc đủ thông tin cần thiết.  2f. Giáo viên thực hiện lại nhập thông tin hoặc hoàn tất thông tin cần thiết.  3f. Quay lại bước 5 hoặc 10 tùy vào chức năng được chọn.  1g. Nếu giáo viên không xác nhận xoá lớp học khi yêu cầu xoá, hoặc hủy bỏ quá trình thêm học sinh vào lớp học khi yêu cầu thêm, hệ thống không thực hiện hành động tương ứng và quay lại bước trước đó.  2g. Quay lại bước 1 hoặc 10 tùy vào chức năng được chọn.  1h. Nếu xảy ra lỗi hệ thống hoặc lỗi kết nối trong quá trình cập nhật, xoá hoặc thêm học sinh vào lớp học, hệ thống hiển thị thông báo lỗi và yêu cầu giáo viên thử lại sau.  2h. Giáo viên thử lại hoặc liên hệ với quản trị viên để giải quyết vấn đề.  3h. Quay lại bước trước đó hoặc kết thúc use case. |

* 1. Person description

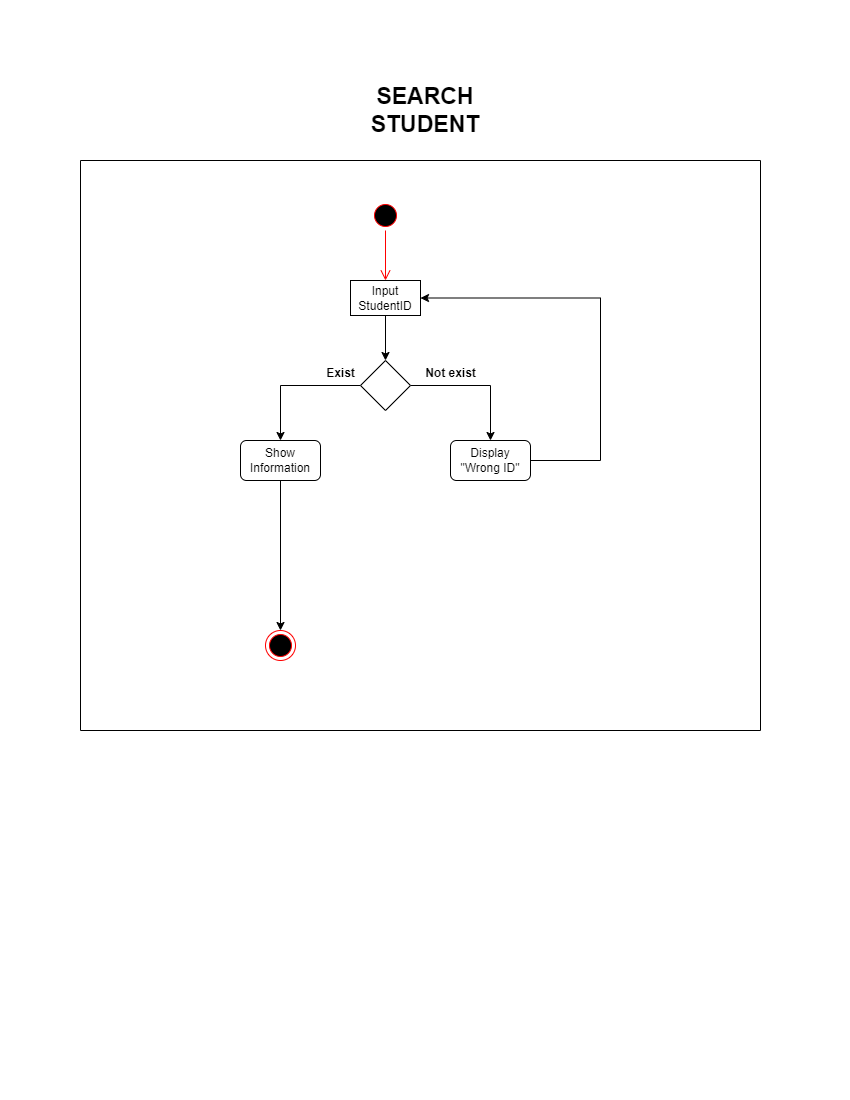
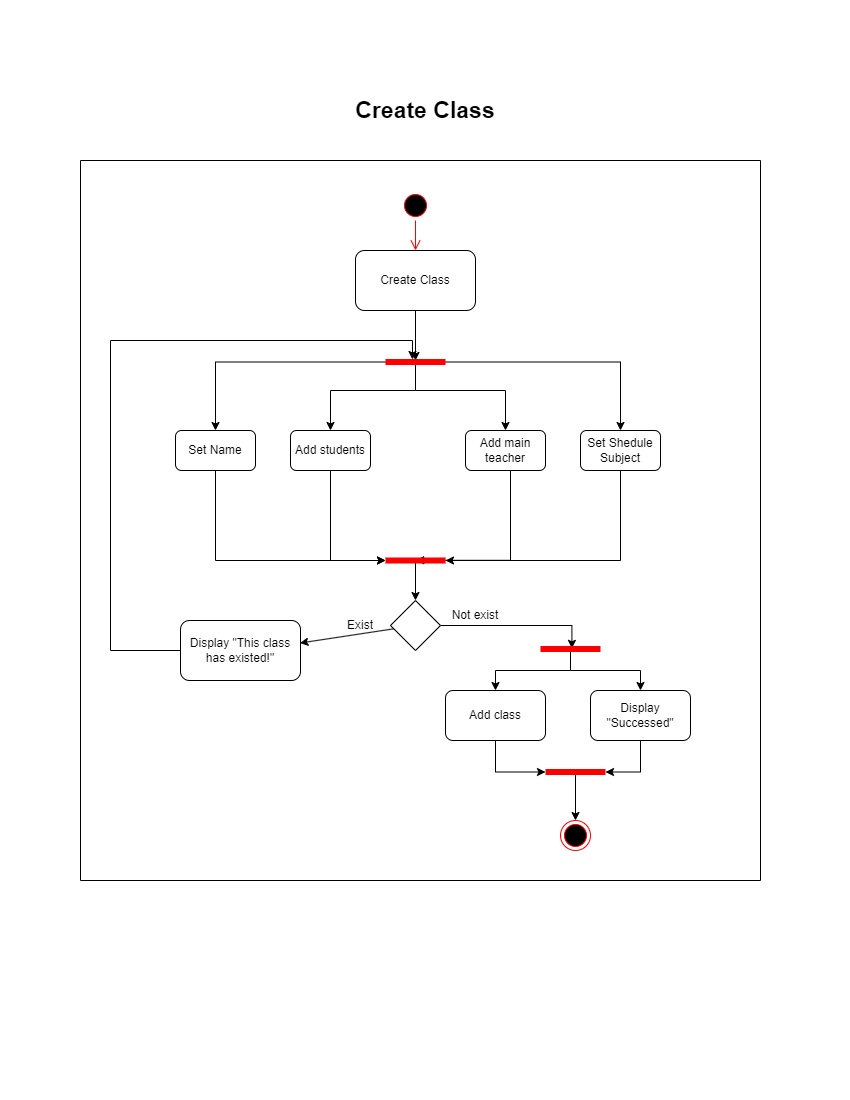
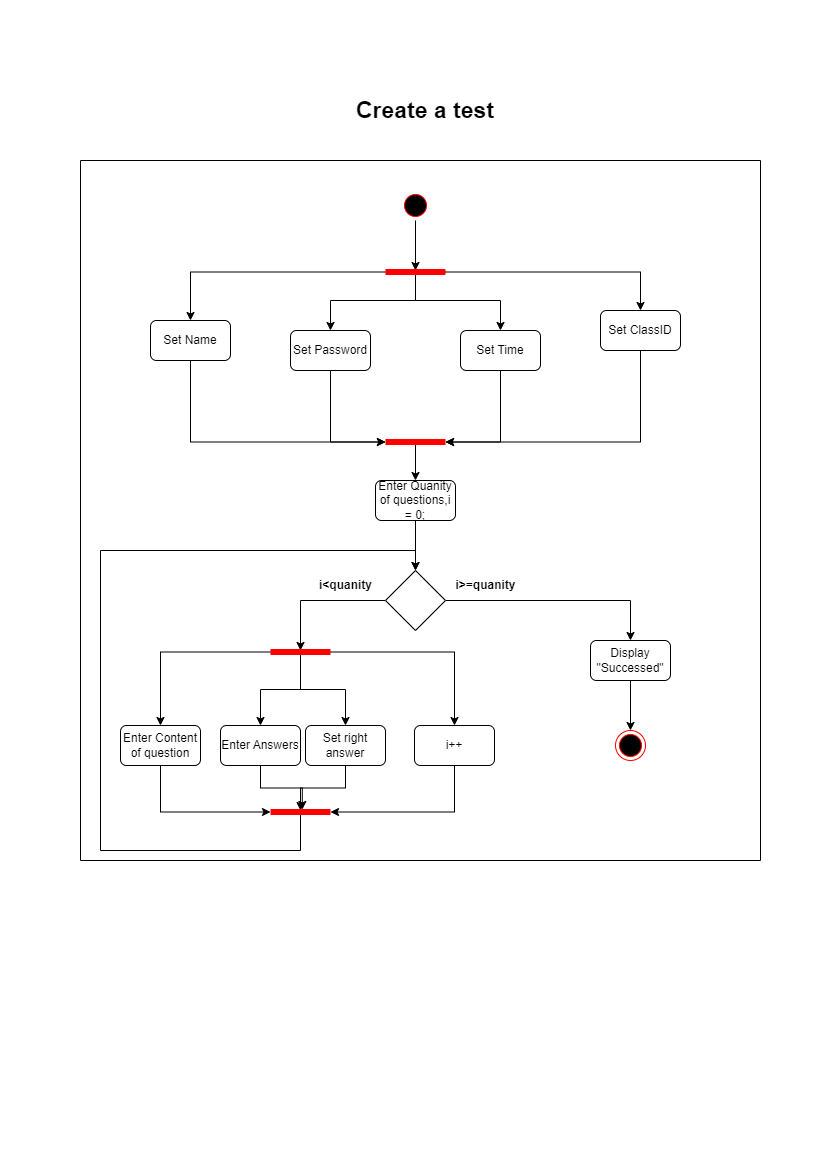
| Name | Person. |
| --- | --- |
| ID | UC\_003 |
| Description | Quản lý thông tin cá nhân của người dùng sau khi đăng nhập vào hệ thống quản lý trường học. |
| Actor | Quản lý,giáo viên,học sinh. |
| Organizational Beneﬁts | Cung cấp cho người dùng khả năng quản lý thông tin cá nhân, bao gồm cập nhật thông tin, thay đổi mật khẩu, và quản lý thông tin liên hệ.  Đảm bảo tính bảo mật và chính xác của thông tin cá nhân của người dùng.  Nâng cao trải nghiệm người dùng và tăng cường tương tác giữa người dùng và hệ thống quản lý trường học. |
| Trigger | Người dùng muốn quản lý thông tin cá nhân sau khi đã đăng nhập vào hệ thống quản lý trường học. |
| Preconditions | Success:Người dùng đã đăng nhập vào hệ thống quản lý trường học. |
| Postconditions | Người dùng đã cập nhật, thay đổi thông tin cá nhân thành công trong hệ thống quản lý trường học. |
| Main Course | 1. Người dùng truy cập vào tính năng quản lý thông tin cá nhân trong hệ thống quản lý trường học. 2. Hệ thống hiển thị thông tin cá nhân hiện tại của người dùng, bao gồm tên, địa chỉ, số điện thoại, email, v.v. 3. Người dùng có thể cập nhật, thay đổi thông tin cá nhân bằng cách điền vào các trường thông tin cần thay đổi. 4. Người dùng xác nhận và lưu lại các thay đổi thông tin cá nhân. 5. Hệ thống kiểm tra tính hợp lệ của thông tin cập nhật, nếu hợp lệ, cập nhật thông tin cá nhân của người dùng trong hệ thống. 6. Hệ thống hiển thị thông tin cá nhân đã được cập nhật và thông báo cho người dùng. |
| Alternate Courses | AC2.1: Nếu người dùng muốn thay đổi mật khẩu, họ có thể sử dụng chức năng thay đổi mật khẩu để nhập mật khẩu cũ và mật khẩu mới, sau đó hệ thống sẽ kiểm tra tính hợp lệ của mật khẩu cũ và thực hiện thay đổi mật khẩu mới.  Nếu người dùng muốn cập nhật thông tin liên hệ khác như số điện thoại, địa chỉ email, v.v., họ có thể sử dụng chức năng cập nhật thông tin liên hệ để nhập thông tin mới và hệ thống sẽ cập nhật lại thông tin liên hệ của người dùng. |
| Exceptions | EX1:Người dùng không có quyền truy cập hoặc cập nhật thông tin. |

* 1. Create Test description.

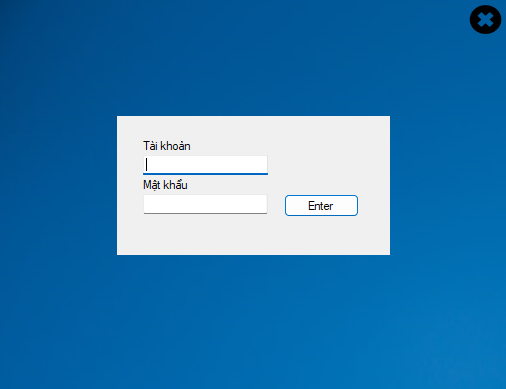
| Name | Create Test. |
| --- | --- |
| ID | UC\_004 |
| Description | Tạo bài kiểm tra trong hệ thống quản lý trường học. |
| Actor | Giáo viên và học sinh. |
| Organizational Beneﬁts | Giáo viên có khả năng tạo bài kiểm tra đơn giản và hiệu quả trong hệ thống quản lý trường học.  Giáo viên có thể tự do thiết kế, định dạng, và quản lý nội dung bài kiểm tra dựa trên nội dung giảng dạy và mục tiêu đào tạo.  Tạo bài kiểm tra trong hệ thống quản lý trường học giúp giảm thời gian và công sức cần thiết cho quá trình tạo bài kiểm tra truyền thống trên giấy. |
| Trigger | Giáo viên muốn tạo bài kiểm tra cho học sinh trong hệ thống quản lý trường học. |
| Preconditions | Giáo viên đã đăng nhập vào hệ thống quản lý trường học.  Giáo viên đã có quyền truy cập và tạo bài kiểm tra trong hệ thống. |
| Postconditions | Success: Bài kiểm tra đã được tạo thành công trong hệ thống quản lý trường học. |
| Main Course | 1. Giáo viên truy cập vào tính năng tạo bài kiểm tra trong hệ thống quản lý trường học. 2. Hệ thống hiển thị giao diện tạo bài kiểm tra, cho phép giáo viên nhập thông tin cơ bản của bài kiểm tra như tên bài kiểm tra, thời gian làm bài. 3. Giáo viên nhập nội dung câu hỏi, đáp án và thông tin liên quan cho mỗi câu hỏi của bài kiểm tra. 4. Giáo viên kiểm tra lại thông tin đã nhập và xác nhận tạo bài kiểm tra. 5. Hệ thống lưu lại thông tin bài kiểm tra và hiển thị thông báo tạo bài kiểm tra thành công. |
| Alternate Courses | 1. Nếu giáo viên muốn chỉnh sửa bài kiểm tra đã tạo, họ có thể sử dụng chức năng chỉnh sửa bài kiểm tra để thay đổi nội dung câu hỏi, đáp án, hoặc thông tin khác của bài kiểm tra. 2. Nếu giáo viên muốn xóa bài kiểm tra đã tạo, họ có thể sử dụng chức năng xóa bài kiểm tra để xóa bài kiểm tra không còn cần thiết trong hệ thống. |
| Exceptions | EX1:Giáo viên không có quyền truy cập hoặc tạo bài kiểm tra trong hệ thống.   * Thông tin nhập không chính xác.   . |

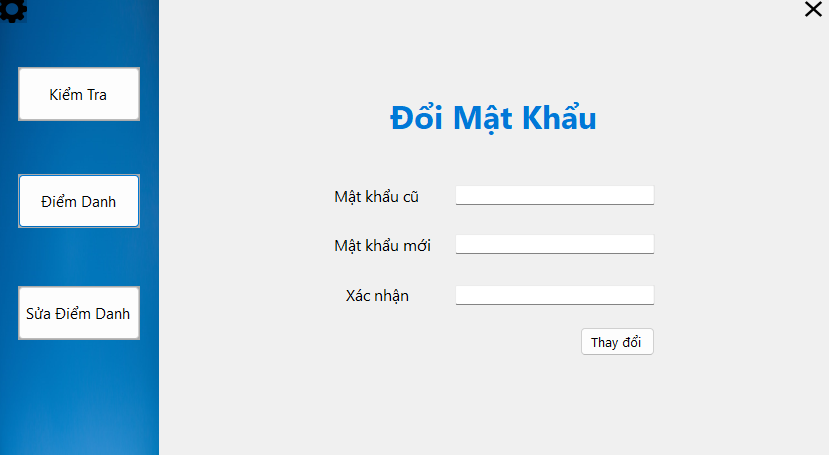
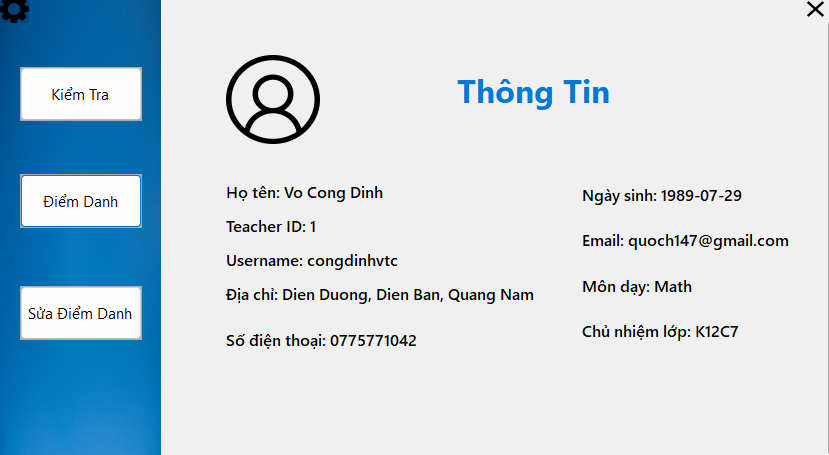
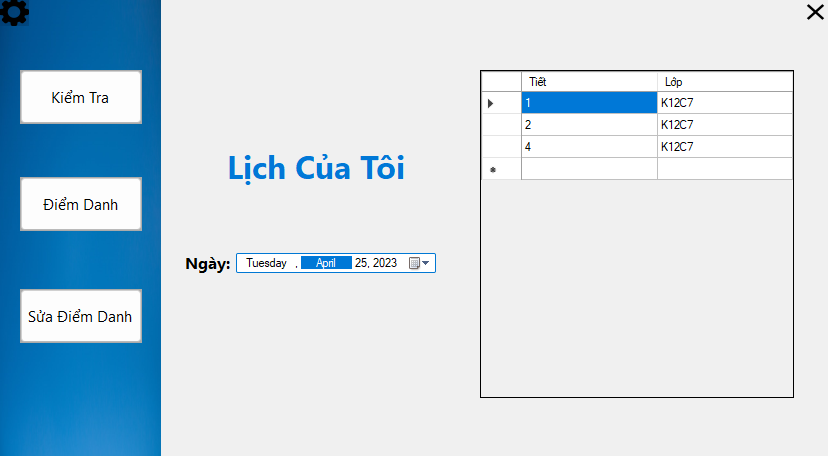
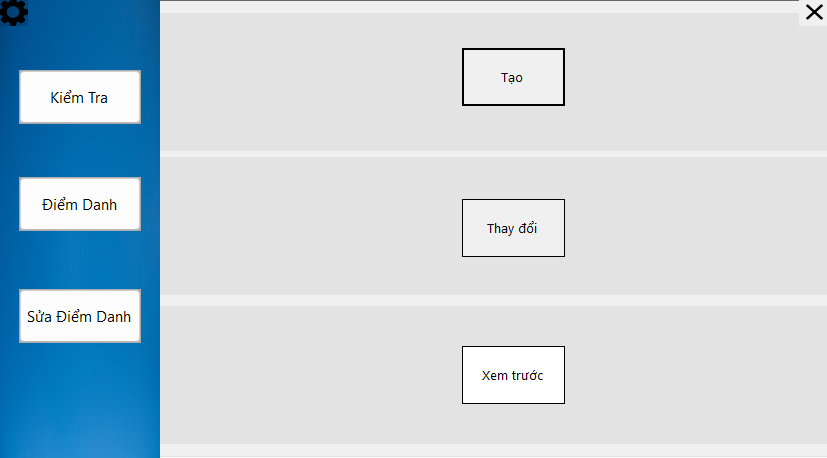
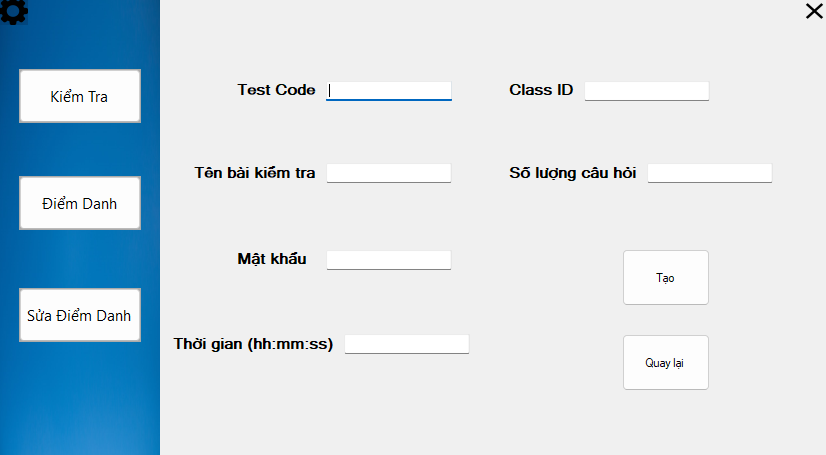
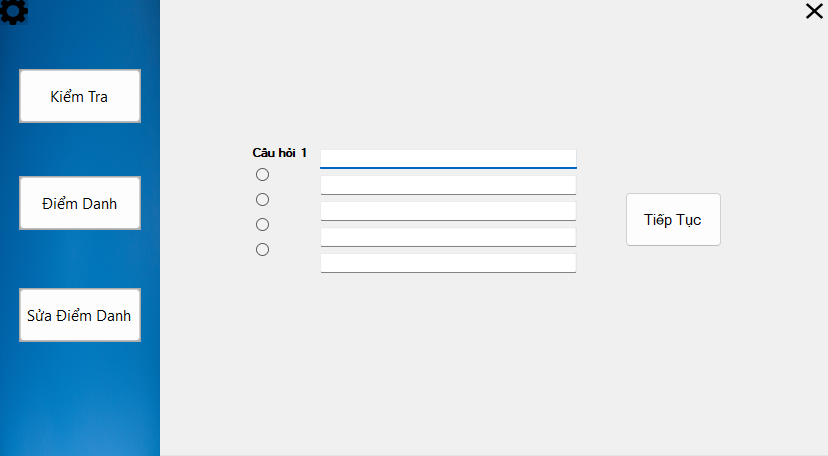
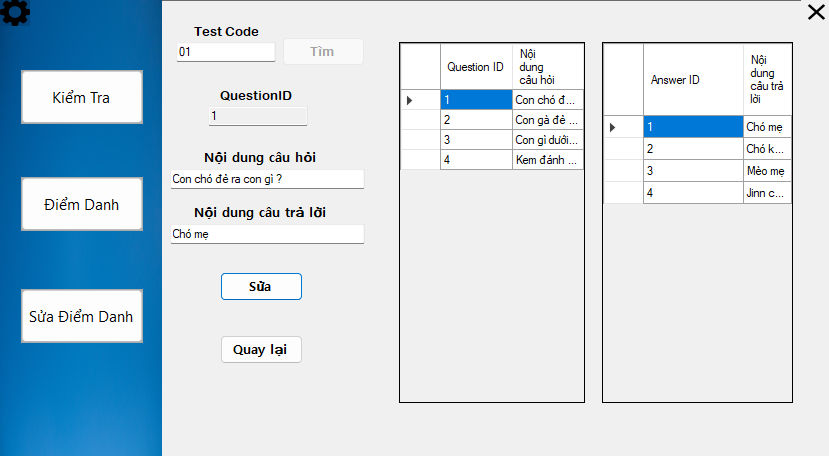
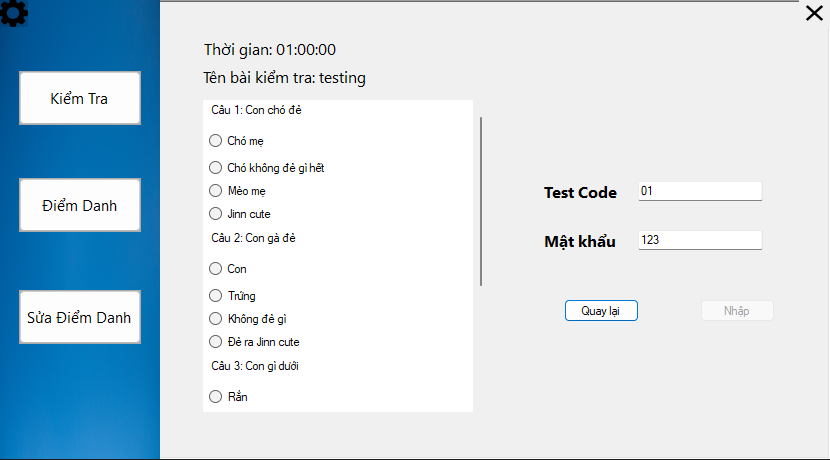
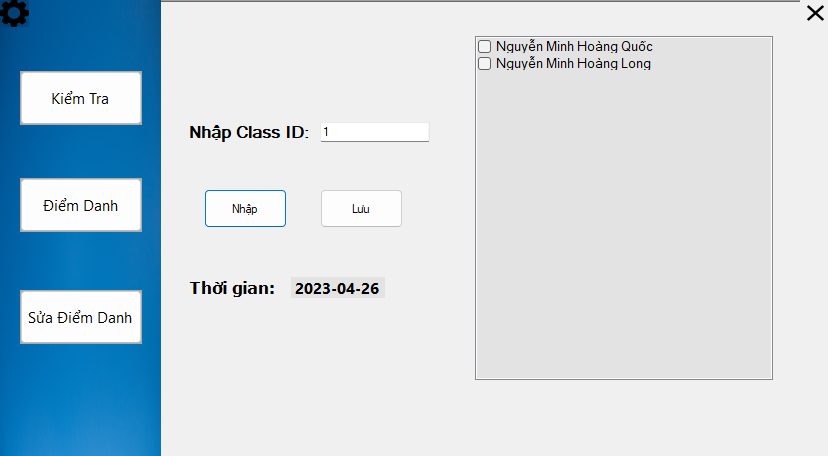
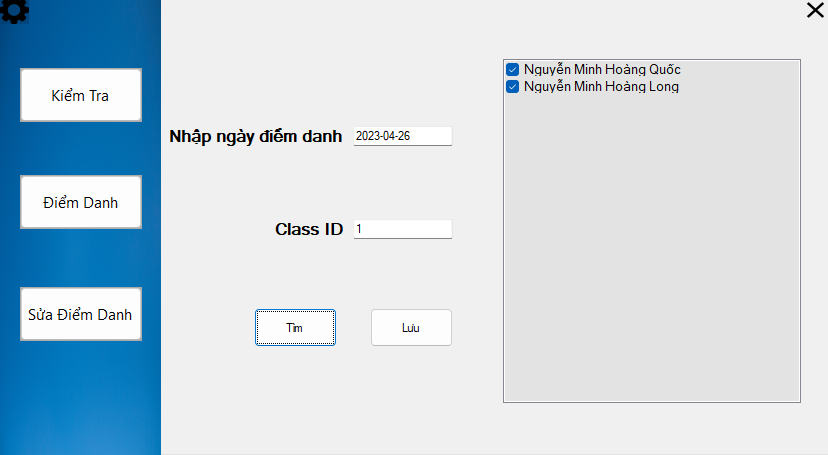
* 1. Checkpoint description

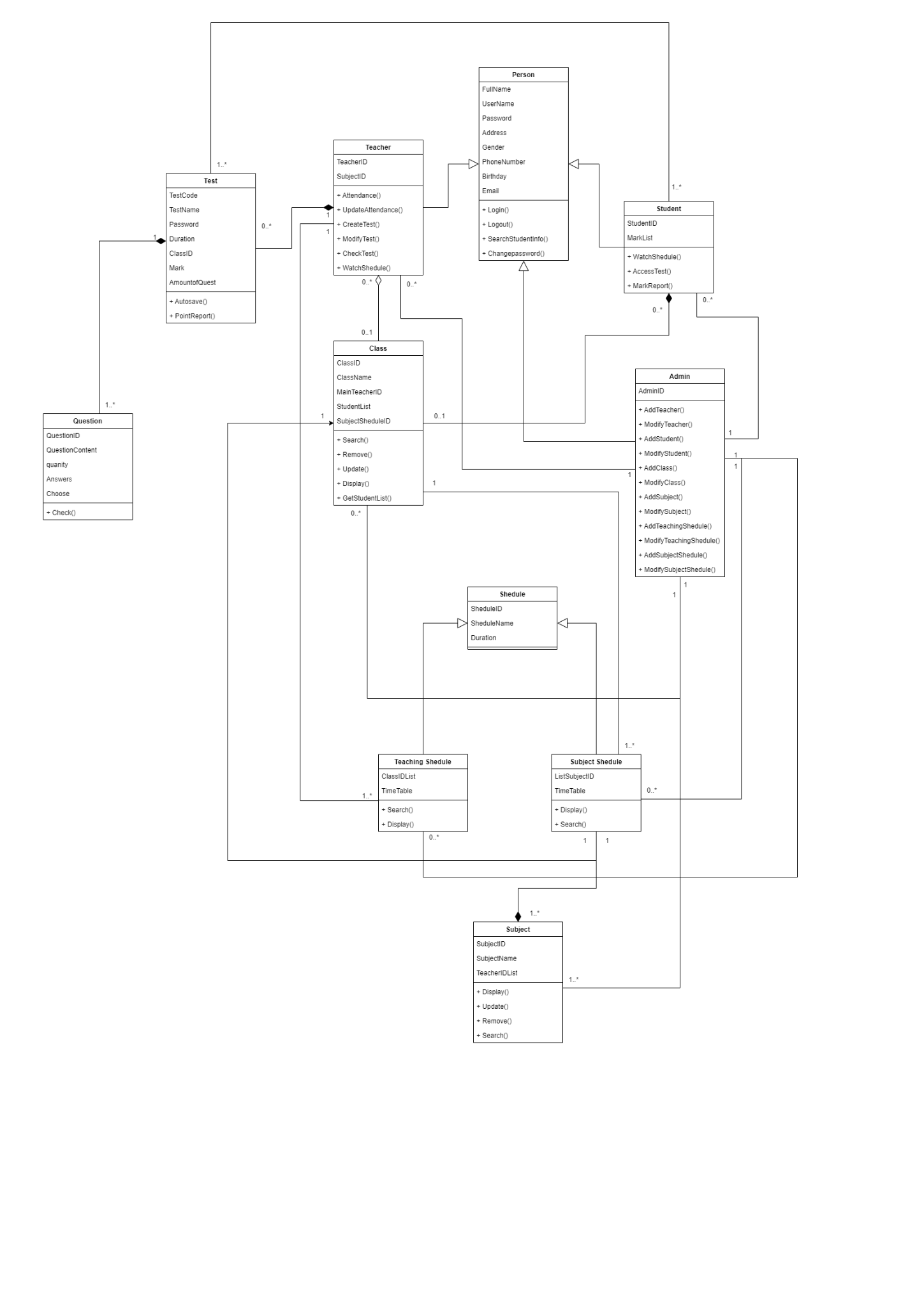
| Name | Checkpoint. |
| --- | --- |
| ID | UC\_005 |
| Description | Học sinh xem điểm của mình trong hệ thống quản lý trường học. |
| Actor | Học sinh. |
| Organizational Beneﬁts | Học sinh có thể tự theo dõi và kiểm tra điểm số của mình trong hệ thống quản lý trường học.  Học sinh có thể đánh giá tiến độ học tập của mình dựa trên kết quả điểm số đã được hiển thị trong hệ thống.  Xem điểm giúp học sinh nắm bắt được tình hình học tập của mình và có thể cải thiện kết quả học tập trong tương lai. |
| Trigger | Học sinh muốn xem điểm của mình trong hệ thống quản lý trường học. |
| Preconditions | Học sinh đã đăng nhập vào hệ thống quản lý trường học.  Học sinh có quyền truy cập vào thông tin điểm của mình trong hệ thống. |
| Postconditions | Success: Học sinh có quyền truy cập vào thông tin điểm của mình trong hệ thống. |
| Main Course | 1. Học sinh truy cập vào tính năng xem điểm của mình trong hệ thống quản lý trường học. 2. Hệ thống hiển thị giao diện xem điểm, cho phép học sinh xem thông tin điểm của mình, chẳng hạn điểm số của từng môn học, điểm trung bình, điểm tổng kết, v.v. 3. Học sinh kiểm tra và đánh giá thông tin điểm của mình dựa trên kết quả hiển thị trên giao diện xem điểm. 4. Học sinh có thể in hoặc tải về bản sao của điểm của mình để lưu trữ hoặc chia sẻ với phụ huynh hoặc giáo viên nếu cần thiết. |
| Alternate Courses | * Nếu học sinh không tìm thấy thông tin điểm của mình, họ có thể kiểm tra lại thông tin nhập vào hoặc liên hệ với giáo viên hoặc quản trị viên hệ thống để được hỗ trợ. * Nếu học sinh muốn xem điểm của các học kỳ trước đó, họ có thể sử dụng tính năng xem điểm theo học kỳ hoặc năm học để có thông tin chi tiết hơn. |
| Exceptions | * Nếu học sinh không đăng nhập vào hệ thống, họ sẽ không thể truy cập vào tính năng xem điểm của mình. * Nếu học sinh không có quyền truy cập vào thông tin điểm trong hệ thống, họ sẽ không thể xem được điểm của mình và sẽ cần liên hệ với giáo viên hoặc quản trị viên để yêu cầu quyền truy cập. * Nếu hệ thống quản lý trường học gặp lỗi kỹ thuật, ví dụ như mất kết nối, lỗi hệ thống, học sinh sẽ không thể xem được điểm của mình trong thời điểm đó và cần thử lại sau. * Nếu thông tin điểm của học sinh chưa được cập nhật đầy đủ trong hệ thống, học sinh sẽ không thể xem được điểm mới nhất và cần liên hệ với giáo viên hoặc quản trị viên để cập nhật thông tin điểm. |

* 1. Activity Diagram
  2. Search student
  3. Create Class
  4. Create a test

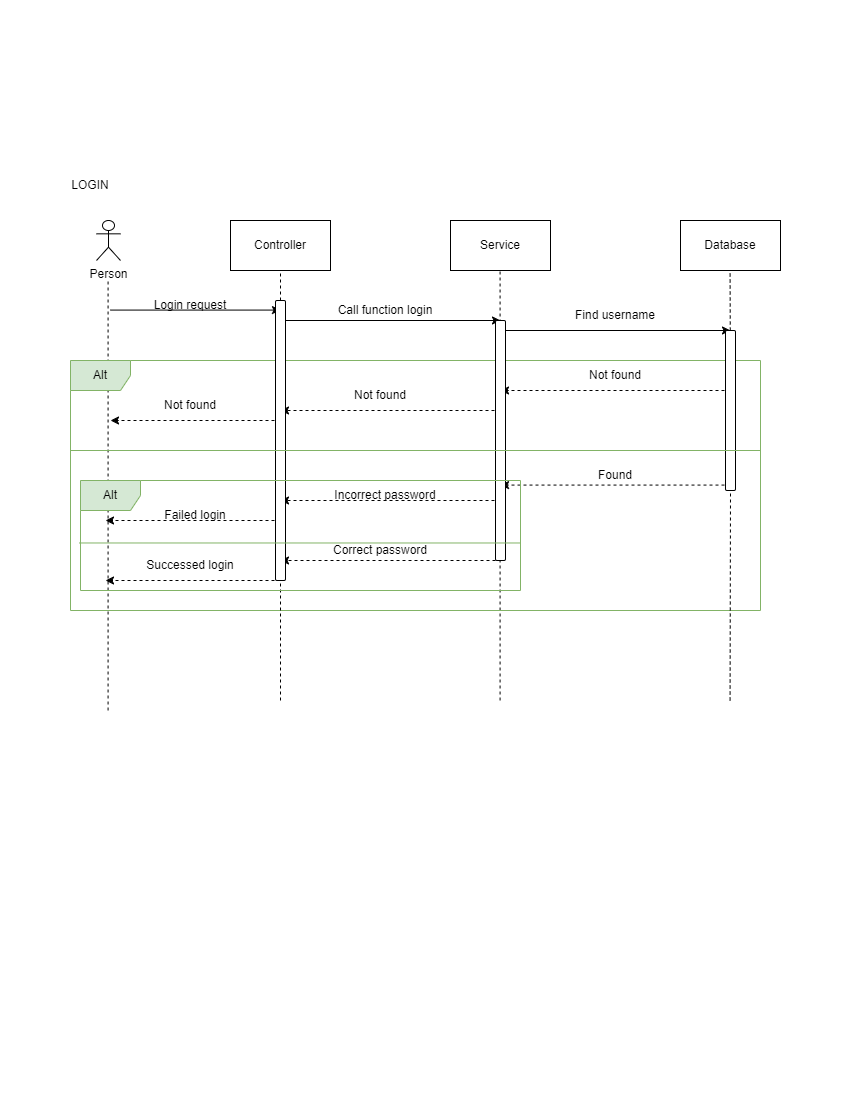
1. Design Details
   1. UI Design
   2. Login



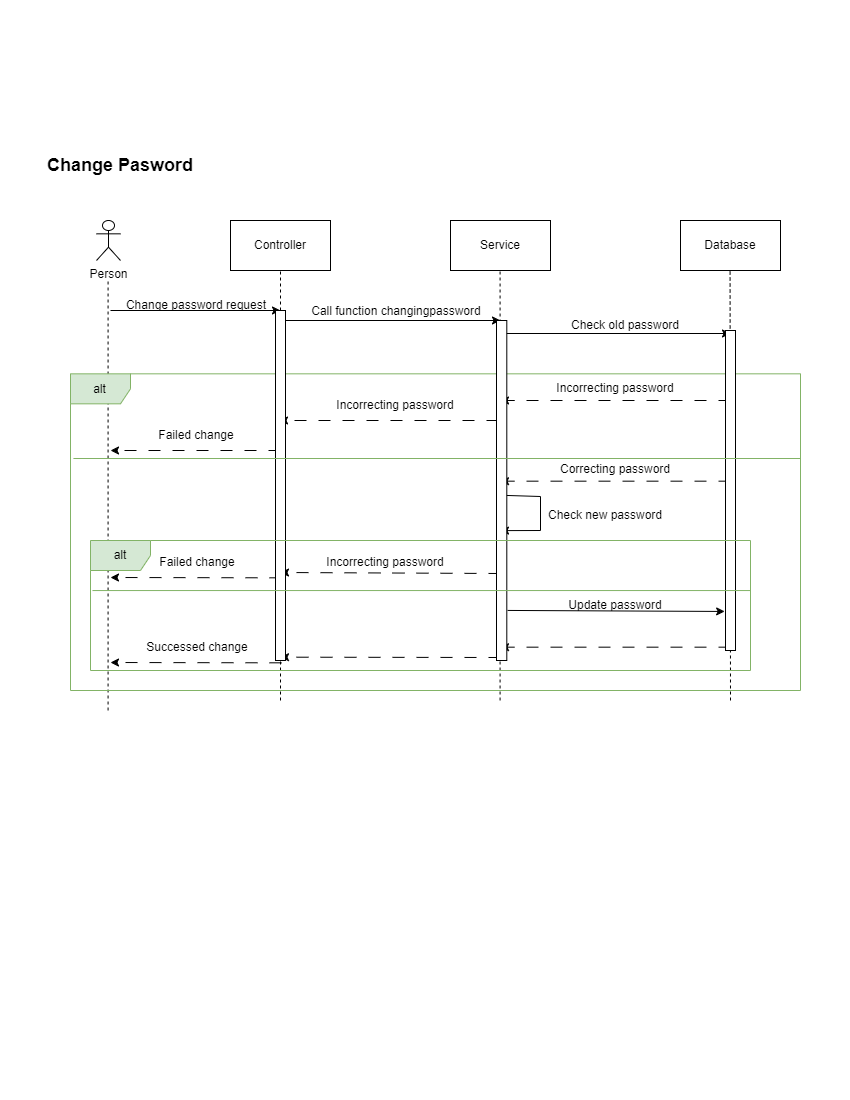
* 1. Main Interface
  2. Change Password
  3. Teacher’s Information
  4. Teaching Schedule
  5. Test Interface
  6. Create A Test
  7. Create Questions (after create a test)
  8. Modify Test
  9. Preview Test
  10. Attendance
  11. Modify the attendance
  12. Code Design (Class Diagram)

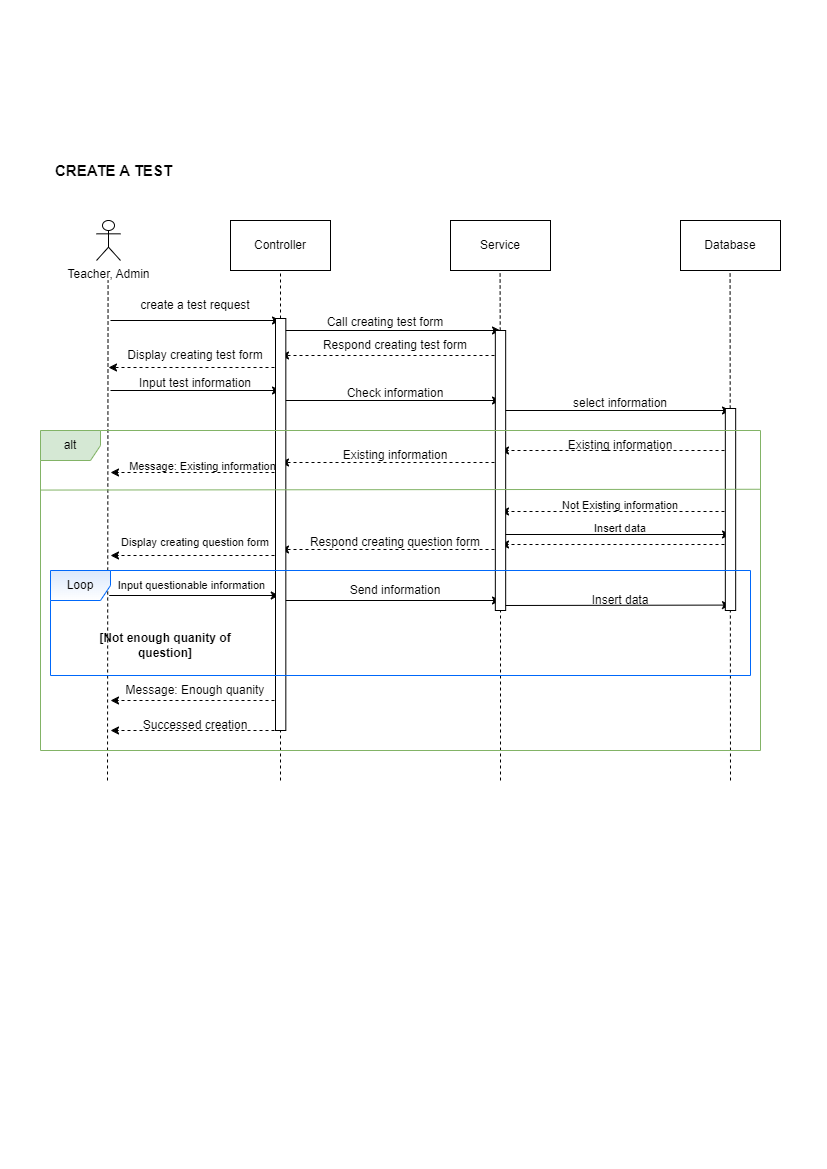


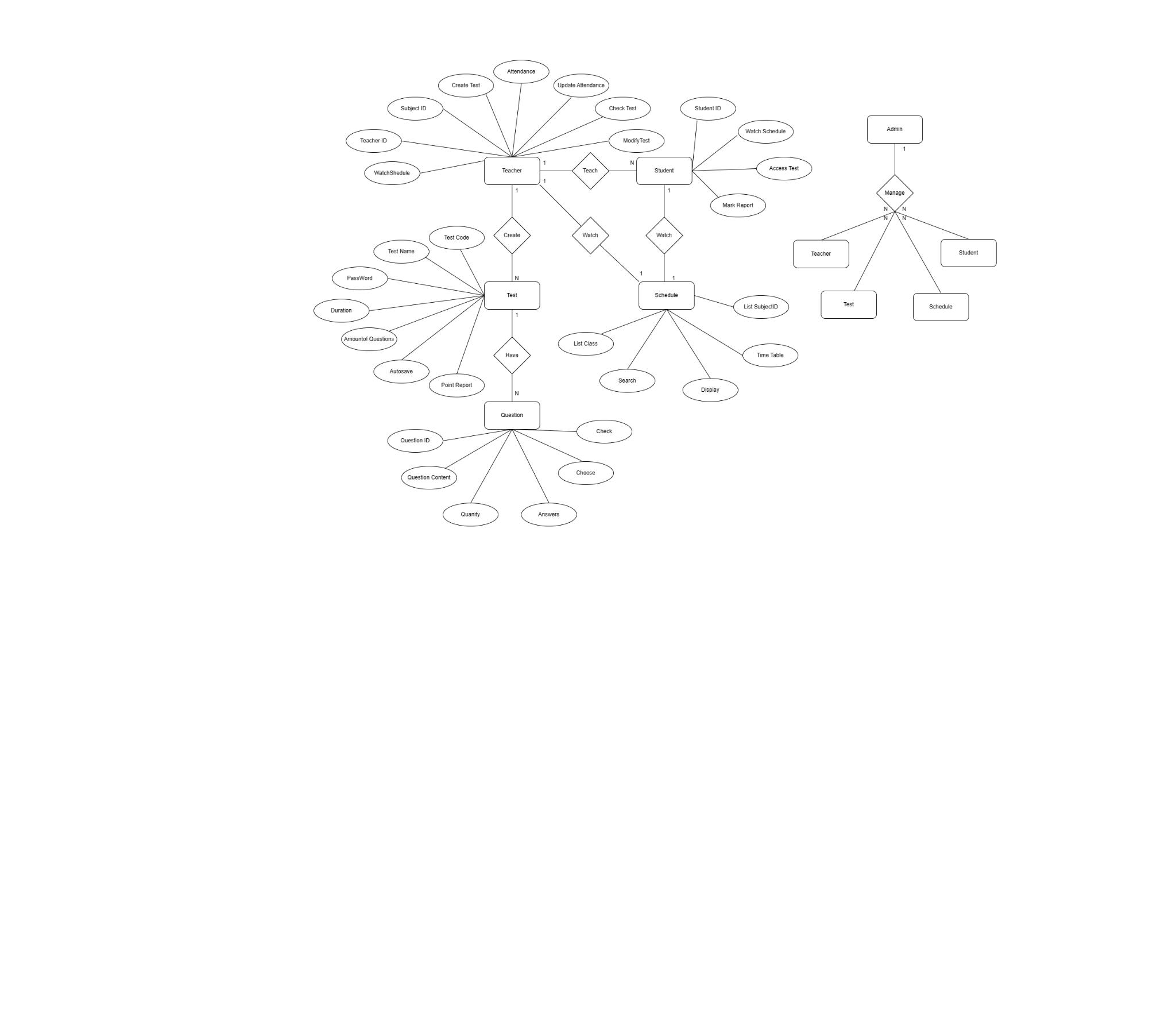
* 1. Sequence Diagram
  2. Login



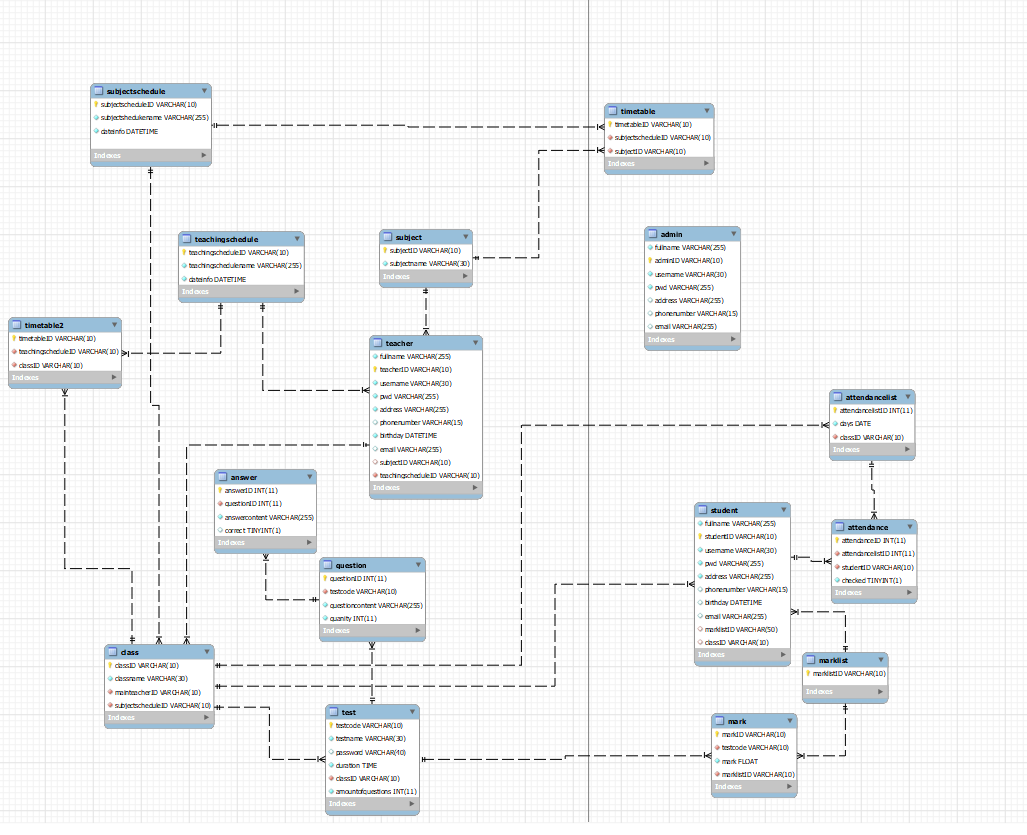
* 1. Change password



* 1. Create a test
  2. Database Design
  3. Entity Relationship Diagram



* 1. Database Design Details



| class | | | |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraints | Description |
| classID | varchar(10) | primary key not null unique |  |
| classname | varchar(30) | not null |  |
| mainteacherID | varchar(10) | not null unique |  |
| subjectscheduleID | varchar(50) | not null | ID’s Schedule |

| teacher | | | |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraints | Description |
| teacherID | varchar(255) | unique not null primary key |  |
| fullname | varchar(50) | not null |  |
| username | varchar(10) | unique not null |  |
| pwd | varchar(100) | not null |  |
| address | varchar(255) | not null |  |
| phonenumber | varchar(15) | varchar(15) |  |
| birthday | datetime | not null |  |
| email | varchar(255) |  |  |
| subjectID | varchar(10) |  |  |
| teachingscheduleID | varchar(10) | not null |  |

| student | | | |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraints | Description |
| studentID | varchar(255) | unique not null primary key |  |
| fullname | varchar(30) | not null |  |
| username | varchar(10) | unique not null |  |
| pwd | varchar(255) | not null |  |
| address | varchar(255) | not null |  |
| phonenumber | varchar(15) | not null unique |  |
| birthday | datetime |  |  |
| email | varchar(255) | not null unique |  |
| marklistID | varchar(50) | not null unique |  |
| classID | varchar(10) |  |  |

| admin | | | |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraints | Description |
| adminID | varchar(255) | unique not null primary key |  |
| fullname | varchar(30) | not null |  |
| username | varchar(10) | unique not null |  |
| pwd | varchar(255) | not null |  |
| address | varchar(255) |  |  |
| phonenumber | varchar(15) |  |  |
| birthday | datetime |  |  |
| email | varchar(255) |  |  |

| mark | | | |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraints | Description |
| markID | varchar(10) | primary key |  |
| testcode | varchar(50) | not null unique |  |
| mark | varchar(255) | not null |  |
| marklistID | varchar(255) | not null |  |

| marklist | | | |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraints | Description |
| marklistID | varchar(10) | primary key | Save mark |

| attendance | | | |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraints | Description |
| attendanceID | int | primary key auto\_increment |  |
| attendancelistID | int | not null |  |
| studentID | varchar(10) | not null |  |
| checked | varchar(1) | not null | “0”: absent or “1”:present |

| attendancelist | | | |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraints | Description |
| attendancelistID | int | primary key auto\_increment |  |
| days | date | not null |  |
| classID | varchar(10) | not null |  |

| subjectschedule | | | |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraints | Description |
| subjectscheduleID | varchar(10) | primary key |  |
| subjectshedulename | varchar(255) | not null |  |
| dateinfo | datetime | not null |  |

| timetable | | | |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraints | Description |
| timetableID | varchar(10) | primary key |  |
| subjectscheduleID | varchar(10) | not null |  |
| subjectID | varchar(10) | not null |  |
| slot | int | not null |  |

| teachingschedule | | | |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraints | Description |
| teachingscheduleID | varchar(10) | primary key |  |
| teachingschedulename | varchar(255) | not null |  |
| dateinfo | datetime | not null |  |

| timetable2 | | | |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraints | Description |
| timetableID | varchar(10) | primary key |  |
| teachingscheduleID | varchar(10) | not null |  |
| classID | varchar(10) | not null |  |
| slot | int | not null |  |

1. Test
   1. Login test
   2. Login test 1

| Test Case Number | 001 |
| --- | --- |
| Test Case Name | LoginTest1() |
| Test Case Description | This test case checks username and password correct or not |
| Preconditions | Username and password must exist in the database |
| Test Case Input | “hoangquoc”, “12345678”  “hoanglong001”,”012264235844” |
| Test Case Expected Output | Succeed sign in |
| Test Case Steps | Open signin form  Find user name in database and if it exists then check password  Check password in signin form  Compare output with expected output |
| Default Value Preserving |  |

* 1. Login test 2

| Test Case Number | 002 |
| --- | --- |
| Test Case Name | LoginTest2() |
| Test Case Description | This test case checks username and password correct or not |
| Preconditions | Username and password must exist in the database |
| Test Case Input | "vanhau112001", "12345678"  "thanhthuan152004", "ddf12345678" "congbao2292004", "12345678" |
| Test Case Expected Output | Succeed sign in |
| Test Case Steps | Open signin form.  Find the user name in the database and if it exists then check the password.  Check password in signin form.  Compare output with expected output. |
| Default Value Preserving |  |

* 1. Attendance test
  2. Attendance test 1

| Test Case Number | 001 |
| --- | --- |
| Test Case Name | AttendanceTest1() |
| Test Case Description | Attendance student in certain classes |
| Preconditions | Logged in with the teacher user and that class must have students. |
| Test Case Input | “CLASS001”  “CLASS002” |
| Test Case Expected Output | Load the attendance list of class and save it. |
| Test Case Steps | Using the select command to find students in this class through classID.  Loading all students name that are found in checkboxlist  Attendance.  Saving by the inserting value command of mysql.  Compare output with expected output. |
| Default Value Preserving |  |

* 1. Attendance test 2

| Test Case Number | 002 |
| --- | --- |
| Test Case Name | AttendanceTest1() |
| Test Case Description | Attendance student in certain classes |
| Preconditions | Logged in with the teacher user and that class must have students. |
| Test Case Input | “CLASS003”  “CLASS004” |
| Test Case Expected Output | Load the attendance list of class and save it. |
| Test Case Steps | Using the select command to find students in this class through classID.  Loading all students name that are found in checkboxlist  Attendance.  Saving by the inserted value command of mysql.  Compare output with expected output. |
| Default Value Preserving |  |

* 1. Modify attendance test
  2. Modify attendance test 1

| Test Case Number | 001 |
| --- | --- |
| Test Case Name | Modifyattendancetest1() |
| Test Case Description | Modify attendance list that teacher has attendance |
| Preconditions | Logged in with the teacher user and that students in class had attendance before. |
| Test Case Input | “CLASS001”,”2023-4-24” |
| Test Case Expected Output | Load the attendance list of this class and can modify this list. |
| Test Case Steps | Using the select command to find students in this class through classID.  Loading all students name that are found on this date in checkboxlist  Modify attendance list.  Saving by the updating value command of mysql.  Compare output with expected output. |
| Default Value Preserving |  |

* 1. Modify attendance test 2

| Test Case Number | 002 |
| --- | --- |
| Test Case Name | Modifyattendancetest2() |
| Test Case Description | Modify attendance list that teacher has attendance |
| Preconditions | Logged in with the teacher user and that students in class had attendance before. |
| Test Case Input | “CLASS002”,”2023-4-24” |
| Test Case Expected Output | Load the attendance list of this class and can modify this list. |
| Test Case Steps | Using the select command to find students in this class through classID.  Loading all students' names that are found on this date in checkboxlist.  Modify attendance list.  Saving by the updating value command of mysql.  Compare output with expected output. |
| Default Value Preserving |  |

* 1. Get Customer by phone test
  2. Get Customer by phone test 1

| Test Case Number | 001 |
| --- | --- |
| Test Case Name | GetCustomerByPhoneTest1() |
| Test Case Description | Search for customer in the system |
| Preconditions | Logged in |
| Test Case Input | “0836984311”  “0836984312”  “0836984313” |
| Test Case Expected Output | A customer |
| Test Case Steps | New CustomerDAL  Call GetByPhone method  Compare output with expected output |
| Default Value Preserving |  |

* 1. Get Customer by phone test 2

| Test Case Number | 002 |
| --- | --- |
| Test Case Name | GetCustomerByPhoneTest2() |
| Test Case Description | Search for customer in the system |
| Preconditions | Logged in |
| Test Case Input | “”  “0836984310” |
| Test Case Expected Output | Null |
| Test Case Steps | New CustomerDAL  Call GetByPhone method  Compare output with expected output |
| Default Value Preserving |  |

1. TaskAssign (to each team member)

| Group  6 | High School Management | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| No | Task name | Description | Start Date | End Date | Member | Self  assessment |
| 1 | Write report |  | 20 April | 26 April | Quốc, Thành và Dũng |  |
| 2 | Code |  | 1 April | 26 April | Quốc |  |
| 3 | Code |  | 9 April | 26 April | Thành |  |
| 4 | Code |  | 9 April | 26 April | Dũng |  |
| 5 | Control |  | 1 April | 26 April | Quốc |  |