**GROUP PROJECT FRONT SHEET**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | **BTEC Level 5 HND Diploma in Computing** | | |
| **Unit number and title** | Semi Project | | |
| **Submission date** | 10/04/2023 | **Date Received 1st submission** | 12/04/2023 |
| **Re-submission Date** |  | **Date Received 2nd submission** |  |
| **Student Name** | Tran Hoang Anh | **Student ID** | BH00113 |
| **Class** | IT0501 | **Assessor name** | Ngo Thi Mai Loan |
| **Student declaration**  I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice. | | | |
|  |  | **Student’s signature** | Tran Hoang Anh |

|  |
| --- |
|  |

**Grade**

|  |  |  |
| --- | --- | --- |
| **❒ Summative Feedback: ❒ Resubmission Feedback:** | | |
| **Grade:** | **Assessor Signature:** | **Date:** |
| **Signature & Date:** | | |

Content

[**I.** **User Requirements** 5](#_Toc131962171)

[**1.** **Functional and Non-Functional Requirements** 5](#_Toc131962172)

[**2.** **Use-Case diagram** 6](#_Toc131962173)

[**3.** **Use case description** 8](#_Toc131962174)

[**II.** **System Designs** 14](#_Toc131962175)

[**1.** **Sitemap** 14](#_Toc131962176)

[**2.** **Entity Relationship Diagram (ERD)** 14](#_Toc131962177)

[**3.** **ERD Documentation** 15](#_Toc131962178)

[**III.** **Implementation** 17](#_Toc131962179)

[**1.** **Sample Source Code** 17](#_Toc131962180)

[**2.** **Web Screenshot** 29](#_Toc131962181)

[**3.** **Test Case** 38](#_Toc131962182)

[**IV.** **Conclusion** 43](#_Toc131962183)

[**1.** **Advantages** 43](#_Toc131962184)

[**2.** **Disadvantages** 43](#_Toc131962185)

[**3.** **Lesson Learnt** 43](#_Toc131962186)

[**4.** **Future improvement** 44](#_Toc131962187)

[**V.** **Appendix** 44](#_Toc131962188)

[**1.** **Group member list** 44](#_Toc131962189)

[**2.** **Task** 44](#_Toc131962190)

[**3.** **Link to GitHub** 46](#_Toc131962191)

[Figure 1 : Use-Case admin/teacher 7](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835131)

[Figure 2 : Use-Case Students 7](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835132)

[Figure 3 : Sitemap 14](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835133)

[Figure 4 : Entity Relationship Diagram (ERD) 14](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835134)

[Figure 5 : Project structure 18](#_Toc131835135)

[Figure 6 : Models : User 19](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835136)

[Figure 7 : Models:Class 19](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835137)

[Figure 8 : Models : RequestEditScore 20](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835138)

[Figure 9 : Models : Score 20](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835139)

[Figure 10 : Models : Subject 21](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835140)

[Figure 11 : Models : TeacherSubject 21](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835141)

[Figure 12 : Controller : login 23](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835142)

[Figure 13 : Controller : score 24](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835143)

[Figure 14 : Controller : class 24](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835144)

[Figure 15 : Controller : Subject 26](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835145)

[Figure 16 : Controller : student 27](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835146)

[Figure 17 : Controller : teacher 28](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835147)

[Figure 18 : Web Screenshot:Login 29](#_Toc131835148)

[Figure 19 : Web Screenshot:Admin/Teacher/Manage student 29](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835149)

[Figure 20 : Web Screenshot:Admin/Teacher/Add students 30](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835150)

[Figure 21 : Web Screenshot:Admin/Teacher/Manage teacher 30](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835151)

[Figure 22 : Web Screenshot:Admin/Teacher/Add Teacher 31](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835152)

[Figure 23 : Web Screenshot:Admin/Teacher/Manage subject 31](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835153)

[Figure 24 : Web Screenshot:Admin/Teacher/Add subject 32](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835154)

[Figure 25 : Web Screenshot:Admin/Teacher/Manage score/Request to correct points 32](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835155)

[Figure 26 : Web Screenshot:Admin/Teacher/Manage score/Scoreboard by subject 33](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835156)

[Figure 27 : Web Screenshot:Admin/Teacher/Manage score/Scoreboard by class 33](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835157)

[Figure 28 : Web Screenshot:Admin/Teacher/Manage score/Scoreboard by students 34](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835158)

[Figure 29 : Web Screenshot:Admin/Teacher/Manage score/Scoreboard by term 34](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835159)

[Figure 30 : Web Screenshot:Admin/Teacher/Manage score/Add score 35](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835160)

[Figure 31 : Web Screenshot:Admin/Teacher/Manage class 35](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835161)

[Figure 32:Web Screenshot:Admin/Teacher/Manage class/Add class 36](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835162)

[Figure 33 : Web Screenshot:Student/view subject 36](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835163)

[Figure 34 : Web Screenshot:Student/view core/personal scoreboard 37](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835164)

[Figure 35 : Web Screenshot:Student/view sore/scoreboard by term 37](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835165)

[Figure 36 : Web Screenshot:Student/request to correst points 38](file:///C:\Users\ducph\Desktop\Cô%20Loan\ASM\Group%20Project%20Frontsheet.docx#_Toc131835166)

[Figure 37 : Web Screenshot:Student/view class 38](#_Toc131835167)

1. **User Requirements**
2. **Functional and Non-Functional Requirements**
3. **Functional Requirements**
   1. **Teacher's role**

* Can login to the system as soon as the website is opened
* Can add/edit/delete student accounts
* Can add/update/delete subjects
* Can add/update/delete teacher's account
* Can add/update/delete student scores
  + View a student's score correction request and also edit and decline a student's request
  + Display scoreboard by subject
  + Display scoreboard by class
  + Display scoreboard by term
  + Display points by period
* Can add/update/delete class
  1. **Student's role**
* Can login to the system as soon as you enter the website with the username and password registered with the teacher.
* Can see the list of subjects
* Can be viewed Scoreboard
  + According to personally
  + By semester
* Can see the class list

1. **Non-Functional Requirements**
   1. **Operational**

The platform is highly versatile and accessible through a variety of web browsers, such as Google Chrome, Firefox, Microsoft Edge, and Coc Coc. It is compatible with a range of devices including those running Windows, MacOS, Kali, and even Android and IOS, provided there is an internet connection and an available web browser.

Furthermore, the platform offers a customizable display feature that allows for various screen sizes, ensuring that users can comfortably use the platform regardless of their device's dimensions. To enhance its functionality, the platform is built on the PHP (Laravel) framework and utilizes a MySQL database. This ensures that the platform is not only user-friendly but also highly efficient and reliable.

* 1. **Performance**

Our student score management program is designed to handle a substantial amount of traffic while providing real-time updates for student information and scores. To ensure optimal performance, we have implemented a capacity to support up to 1000 student users concurrently accessing the program. This means that even during peak usage periods, our program will remain responsive and reliable with minimal latency, ensuring a seamless user experience.

We understand that speed is essential, especially when dealing with real-time data, and our program is optimized to provide a response time of up to 1ms. This means that updates and modifications will be reflected instantly, ensuring that users have access to the latest information without any delays.

Our team has put in extensive efforts to optimize the program's performance to provide the smoothest possible user experience, making it easy to navigate and use.

* 1. **Security**

Our system is designed to provide the utmost security and privacy for our users. The platform will be hosted on the cloud using the trusted and reliable service of Cloudflare. To ensure secure access to the platform, we support authentication through HTTPS/SSH and MFA security protocols, which offer advanced protection.

All user information and data transmitted within the platform will be encrypted to prevent any potential security breaches. We understand the importance of user privacy, and our commitment to data protection is reflected in our platform's security features.

To ensure that the system remains secure, our team will conduct regular security checks and update the system with the latest security protocols. In the event of any security issues, we will immediately test and fix the errors to ensure that our platform remains secure and reliable for our users.

In summary, our website platform prioritizes the privacy and security of our users, ensuring that all user information is encrypted and stored securely. With our commitment to regular security updates and testing, our users can rest assured that our platform remains protected against potential security threats.

* 1. **Political Culture**

The system will use the main language is English. However, the platforms will help you translate into Vietnamese.

* Website policies will be updated without regard to sexist, racist, or regionalist rules.
* The launcher uses 3 main colors: #1f1f20 - #db3170 - #ffffff

1. **Use-Case diagram**

* **Use-Case admin/teacher**

Ảnh có chứa biểu đồ

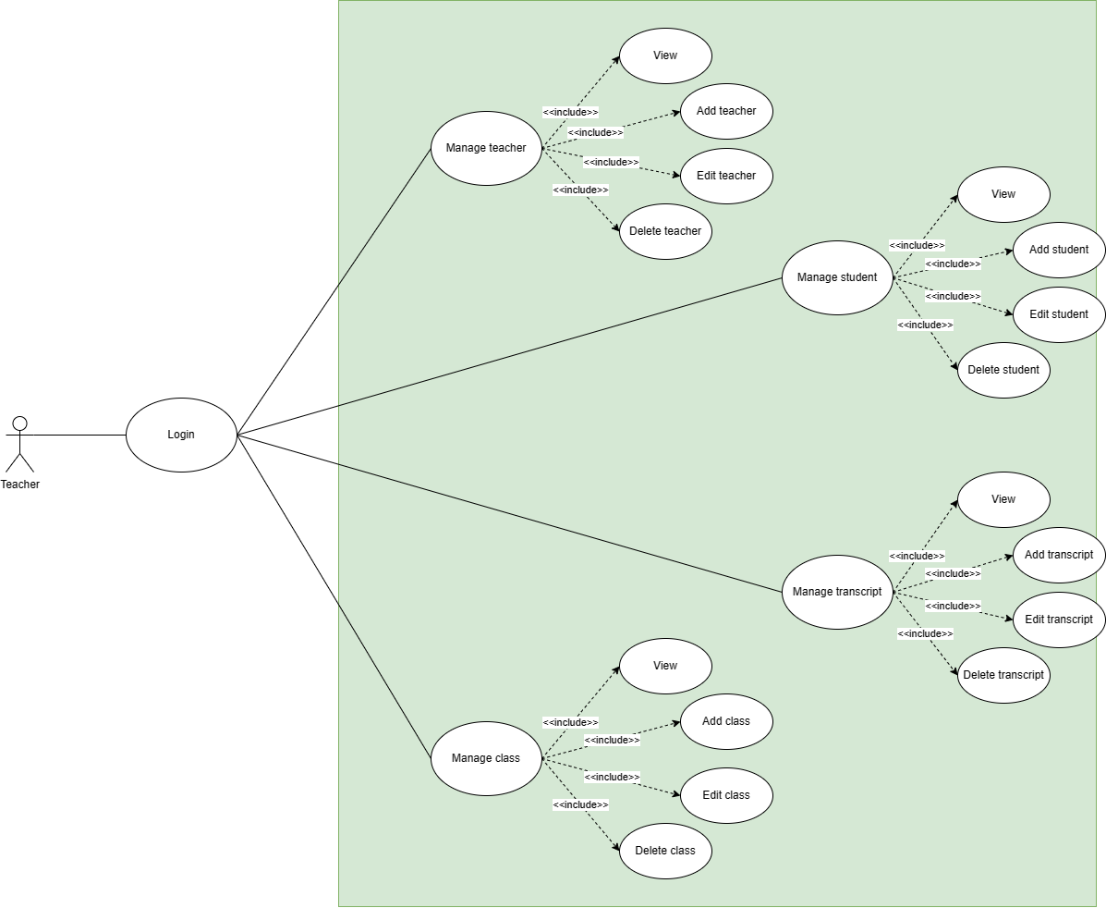
Mô tả được tạo tự động****

Figure 1 : Use-Case admin/teacher

* **Use-Case Students**

**Ảnh có chứa biểu đồ

Mô tả được tạo tự động**

Figure 2 : Use-Case Students

1. **Use case description**

**#UC01:Logout**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Log out | **Code** | UC01 |
| **Description** | Allow users to log out of the system | | |
| **Role** | Teacher | **Trigger** | User presses logout button |
| **Pre-condition** | User has successfully logged into the system | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Log out successfully** | | | |
| 1 | Users click on the logout button right below the Side bar |  | |
|  | | 2 | Log out and return the user to the login page |

**#UC02: Add Student**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Add Student | **Code** | UC02 |
| **Description** | Allow users to log out of the system | | |
| **Role** | Teacher | **Trigger** | Actor presses Add button |
| **Pre-condition** | The teacher has successfully logged into the system. | | |
| **Post condition** | Go to the manage student page. | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Add Student** | | | |
| 1 | Actor clicks on student button, on admin page |  | |
|  | | 2 | Fill full the information |
| 3 | Click the add student button |  | |
|  | | 4 | Load into manage student page. |

**System Message**

|  |  |
| --- | --- |
| **MS01** | “Added” Message when the actor clicks the Add button. |

**#UC3: Edit Student**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Edit Student | **Code** | UC03 |
| **Description** | Allow the Teacher to edit/delete/add score the Student | | |
| **Role** | Teacher | **Trigger** | Teachers presses Edit, Delete and Add score button |
| **Pre-condition** | Teacher has successfully edited, deleted and added score the student. | | |
| **Post condition** | Return to the manage student page. | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Edit Student** | | | |
| 1 | Teacher clicks on student button, on admin page |  | |
| 2 | Click the List of students button |  | |
| 3 | Click Edit, Delete or Add score button |  | |
|  | | 4 | Fill full the information. |
|  | | 5 | Load into manage student page. |

**System Message**

|  |  |
| --- | --- |
| **MS01** | “Added, Deleted or Added score” Message when the Teacher clicks the Add/Update/Delete button. |

**#UC4: Add Class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Add Class | **Code** | UC04 |
| **Description** | Allow the Teacher to add the class | | |
| **Role** | Teacher | **Trigger** | Teacher presses Add button |
| **Pre-condition** | Teacher has successfully added the class | | |
| **Post condition** | Go to the manage student page. | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Add Class** | | | |
| 1 | Teacher clicks on class button, on admin page |  | |
|  | | 2 | Fill full the information |
| 3 | Click the add button |  | |
|  | | 4 | Load into manage student page. |

**System Message**

|  |  |
| --- | --- |
| **MS01** | “Added” Message when the Teacher clicks the Add button. |

**#UC05: Edit Class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Edit Class | **Code** | UC05 |
| **Description** | Allow the Teacher to edit/delete list the class | | |
| **Role** | Teacher | **Trigger** | Teacher presses list/edit/delete button |
| **Pre-condition** | Teacher has successfully edited/deleted list the class | | |
| **Post condition** | Go to the manage student page. | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Edit Class** | | | |
| 1 | Teacher clicks on class button, on admin page |  | |
|  | | 2 | Fill full the information |
| 3 | Click the update button |  | |
|  | | 4 | Load into manage student page. |

**System Message**

|  |  |
| --- | --- |
| **MS01** | “Updated/Deleted” Message when the Teacher clicks the Add/Update/Delete button. |

**#UC06: View Class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | View Class | **Code** | UC05 |
| **Description** | Allow the Teacher to view list the class (eg: fullname, student ID,…) | | |
| **Role** | Teacher/Student | **Trigger** | Teacher/Student presses list button |
| **Pre-condition** |  | | |
| **Post condition** | Go to the Class list page. | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: View Class** | | | |
| 1 | Teacher/Student clicks on class button |  | |
|  | | 2 | Load list on the shown class page |

**#UC07: Add Subject**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Add Subject | **Code** | UC06 |
| **Description** | Allow the Teacher to add the subject | | |
| **Role** | Teacher | **Trigger** | Teacher presses Add button |
| **Pre-condition** | Teacher has successfully added the subject | | |
| **Post condition** | Go to the manage student page. | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Add Subject** | | | |
| 1 | Teacher clicks on class button, on admin page |  | |
|  | | 2 | Fill full the information |
| 3 | Click the add button |  | |
|  | | 4 | Load into manage student page. |

**System Message**

|  |  |
| --- | --- |
| **MS01** | “Added” Message when the Teacher clicks the Add button. |

**#UC08: Edit Subject**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Edit Subject | **Code** | UC07 |
| **Description** | Allow the Teacher to update/delete the subject | | |
| **Role** | Teacher | **Trigger** | Teacher presses update/delete button |
| **Pre-condition** | Teacher has successfully updated/deleted the subject | | |
| **Post condition** | Go to the manage student page. | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Edit Subject** | | | |
| 1 | Teacher clicks on subject button, on admin page |  | |
|  | | 2 | Fill full the information |
| 3 | Click the update/delete button |  | |
|  | | 4 | Load into manage student page. |

**System Message**

|  |  |
| --- | --- |
| **MS01** | “Updated/Deleted” Message when the Teacher clicks the Update/Delete button. |

**#UC09: View Subject**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | View Subject | **Code** | UC09 |
| **Description** | Allow to view list the subject(e.g.: name, ID, semester) | | |
| **Role** | Teacher/Student | **Trigger** |  |
| **Pre-condition** |  | | |
| **Post condition** | Go to the Subject list page. | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: View Subject** | | | |
| 1 | Teacher/Student clicks on Subject button |  | |
|  | | 2 | Load list on the shown subject page |

**#UC010: Edit Transcript**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Edit Transcript | **Code** | UC07 |
| **Description** | Allow the Teacher to edit the score (from subject, class and term)and reply the feedbacks of student | | |
| **Role** | Teacher | **Trigger** | Teacher presses Checkpoint button |
| **Pre-condition** | Teacher has successfully edited/delete the score | | |
| **Post condition** | Go to the manage student page. | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Edit Transcript** | | | |
| 1 | Teacher clicks on transcript button, on admin page |  | |
|  | | 2 | Fill full the information |
| 3 | Click the edit/delete button |  | |
|  | | 4 | Load into manage student page. |

**System Message**

|  |  |
| --- | --- |
| **MS01** | “Added/Deleted” Message when the Teacher clicks the Edit/Delete button. |

**#UC10: View Transcript**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | View Transcript | **Code** | UC10 |
| **Description** | Allow to view list the score( from personal score and score from term) and feedback | | |
| **Role** | Teacher/Student | **Trigger** |  |
| **Pre-condition** |  | | |
| **Post condition** | Go to the score list page. | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: View Subject** | | | |
| 1 | Teacher/Student clicks on Transcript button |  | |
|  | | 2 | Load list on the shown score page from personal scoreboard or scoreboard by term |

1. **System Designs**
   * + 1. **Sitemap**

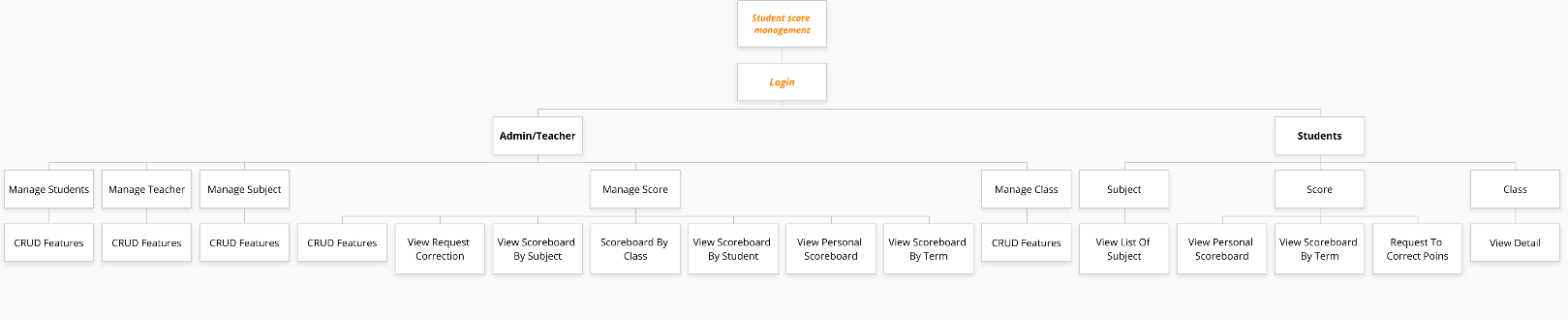
The full version and details of our sitemap can be accessed through [this URL](https://www.gloomaps.com/zk9gMfzmjQ)

Figure 3 : Sitemap

* + - 1. **Entity Relationship Diagram (ERD)**

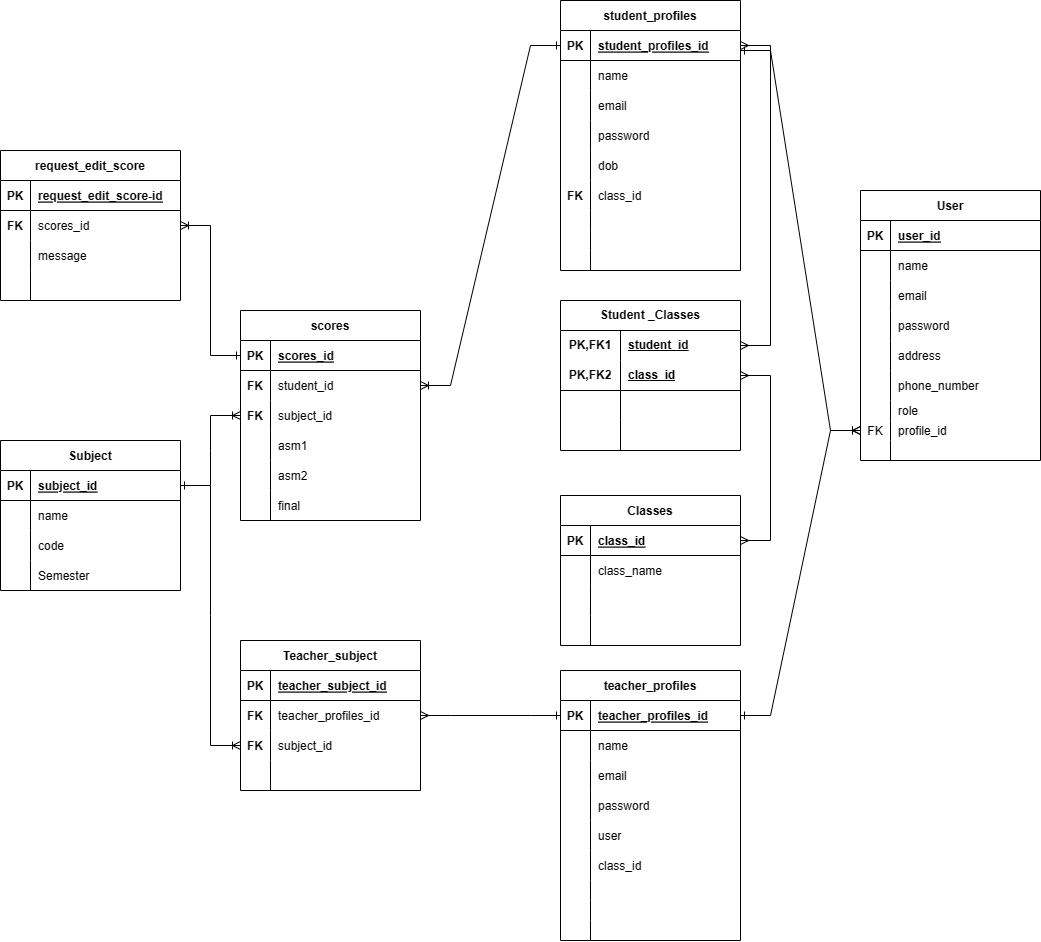
****

Figure 4 : Entity Relationship Diagram (ERD)

* + - 1. **ERD Documentation**

1. **User**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** | **Explanation** |
| user\_id | bigint(20) | Primary key | ID of User |
| name | Varchar(255) | Not Null | Identification of a specific user, account name of a user in the system |
| Email | Varchar(255) | Not null | Email of a user |
| Role | Varchar(255) | Foreign Key | Role of the account in system |
| Password | Varchar(255) | Not Null | Password of a user account. It is used with username to login to the system |
| address | Varchar(255) | Not null | Address of a user |
| Phone\_number | Varchar(255) | Not Null | Phone\_number of a user |
| profile\_id | bigint(20) | Not null | Profile ID of teacher or student |

1. **Student\_profiles**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** | **Explanation** |
| Student\_profiles\_id | bigint(20) | Primary key | ID of Student\_profiles |
| dob | datetime | Not Null | Class name |
| code | Varchar(255) | Not Null | ID of User |
| Class\_id | bigint(20) | Foreign Key | ID of Class |

1. **Teacher\_profiles**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** | **Explanation** |
| Teacher\_profiles\_id | bigint(20) | Primary key | ID of Teacher\_profiles |

1. **Class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** | **Explanation** |
| class\_id | bigint(20) | Primary key | ID of Class |
| class\_name | Varchar(255) | Not Null | Class name |
| user\_id | bigint(20) | Foreign Key | ID of User |

1. **Teacher\_subject**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** | **Explanation** |
| teacher\_subject\_id | bigint(20) | Primary key | ID of Teacher\_subject |
| user\_id | bigint(20) | Foreign Key | ID of User |
| subject\_id | bigint(20) | Foreign Key | ID of Subject |

1. **Subject**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** | **Explanation** |
| subject\_id | bigint(20) | Primary key | ID of subject |
| subject\_name | Varchar(255) | Not Null | Subject Name |
| semester | Int(10) | Not Null | Semester |

1. **Scores**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** | **Explanation** |
| score\_id | bigint(20) | Primary key | ID of Score |
| user\_id | bigint(20) | Foreign Key | ID of User |
| subject\_id | bigint(20) | Foreign Key | ID of Subject |
| asm1 | Double(8,2) | Not Null | Student's asm1 score |
| asm2 | Double(8,2) | Not Null | Student's asm2 score |
| final | Double(8,2) | Not Null | Final score |

1. **Request\_edit\_score**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** | **Explanation** |
| request\_edit\_score\_id | bigint(20) | Primary key | ID of request\_edit\_score |
| score\_id | bigint(20) | Foreign Key | ID of Score |
| message | Varchar(255) | Not Null | Message describing why score correction is requested |

1. **Implementation**
2. **Sample Source Code**
3. **Project structure**

As the project is developed using the Laravel framework, it adheres to the Model-View-Controller (MVC) architectural pattern. This design pattern separates the application logic into three interconnected components to improve flexibility, scalability, and maintainability.The project’s structure follows a well-defined and organized approach to development. The MVC structure comprises three core components: the Model, the View, and the Controller. The structure of the project is:

Ảnh có chứa văn bản

Mô tả được tạo tự động

Figure 5 : Project structure

1. **Project elements**

When working with Laravel, it's essential to focus on the main components of the framework, including the Model, Controller, View, and Route. These components play a critical role in building a robust and scalable application, making it easier to manage, maintain, and extend the codebase.Below I will present all the ingredients we use:

* 1. **Model**
* **Ảnh có chứa văn bản

  Mô tả được tạo tự độngModels : User**

Figure 6 : Models : User

* **Models:Class**Ảnh có chứa văn bản

  Mô tả được tạo tự động

Figure 7 : Models:Class

* **Ảnh có chứa văn bản

  Mô tả được tạo tự độngModels : RequestEditScore**

Figure 8 : Models : RequestEditScore

* **Ảnh có chứa văn bản

  Mô tả được tạo tự độngModels : Score**

Figure 9 : Models : Score

* Ảnh có chứa văn bản

  Mô tả được tạo tự động**Models : Subject**

Figure 10 : Models : Subject

* **Models : TeacherSubject**Ảnh có chứa văn bản

  Mô tả được tạo tự động

Figure 11 : Models : TeacherSubject

* 1. **Controller**
* **Controller : login**

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

Figure 12 : Controller : login

* **Controller : score**

Figure 13 : Controller : score

|  |  |
| --- | --- |
|  | **Ảnh có chứa văn bản  Mô tả được tạo tự động** |

* **Controller : class**

Figure 14 : Controller : class

|  |  |
| --- | --- |
| **Ảnh có chứa văn bản  Mô tả được tạo tự động** | **Ảnh có chứa văn bản  Mô tả được tạo tự động** |

* **Ảnh có chứa văn bản

  Mô tả được tạo tự độngController : Subject**

Figure 15 : Controller : Subject

* **Ảnh có chứa văn bản

  Mô tả được tạo tự độngController : student**

Figure 16 : Controller : student

* **Ảnh có chứa văn bản

  Mô tả được tạo tự độngController : teacher**

Figure 17 : Controller : teacher

1. **Web Screenshot**

* **Login**



Figure 18 : Web Screenshot:Login

* **Admin/Teacher**
  + **Ảnh có chứa bàn

    Mô tả được tạo tự độngManage student**

Figure 19 : Web Screenshot:Admin/Teacher/Manage student

Ảnh có chứa biểu đồ

Mô tả được tạo tự động

Figure 20 : Web Screenshot:Admin/Teacher/Add students

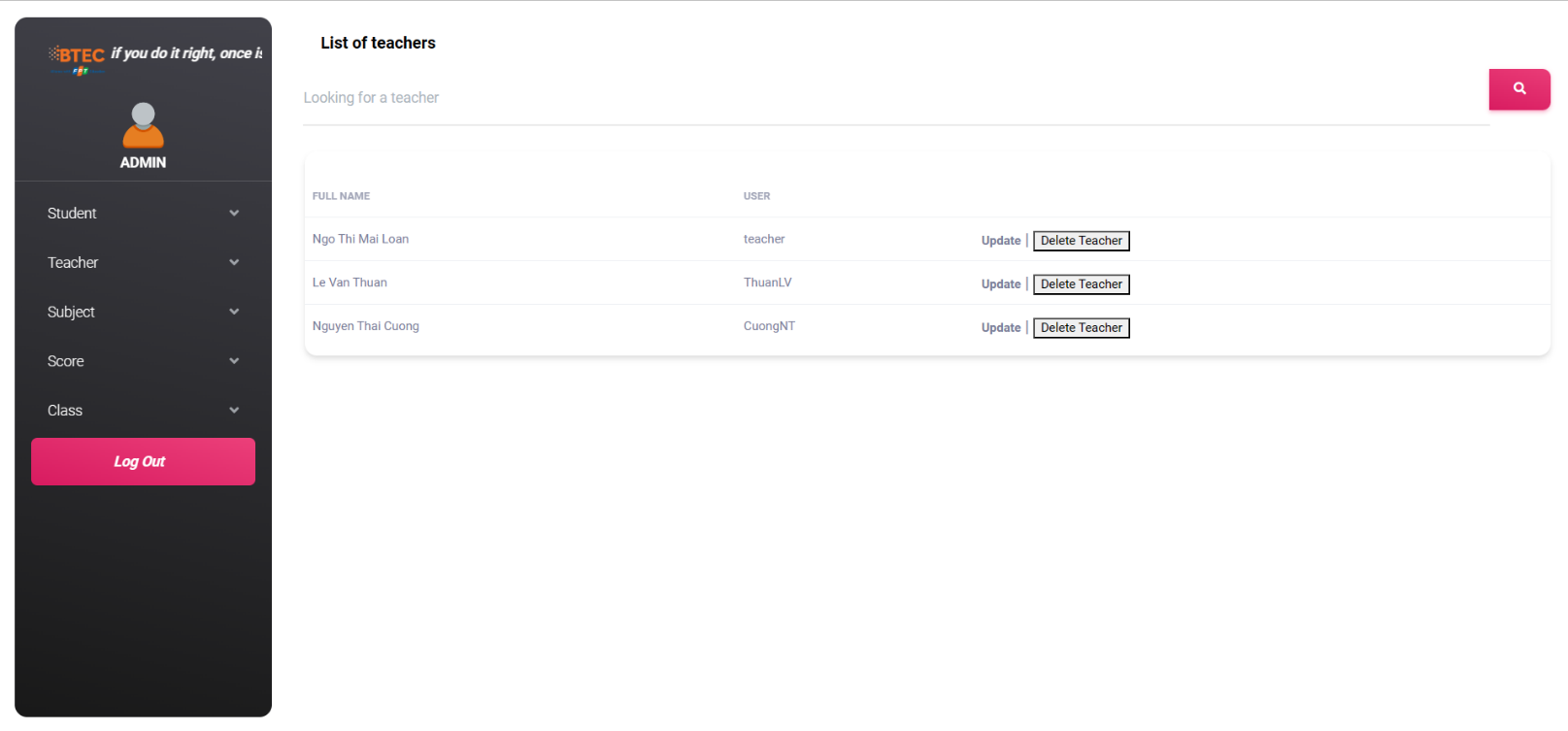
* + **Manage teacher**

Figure 21 : Web Screenshot:Admin/Teacher/Manage teacher

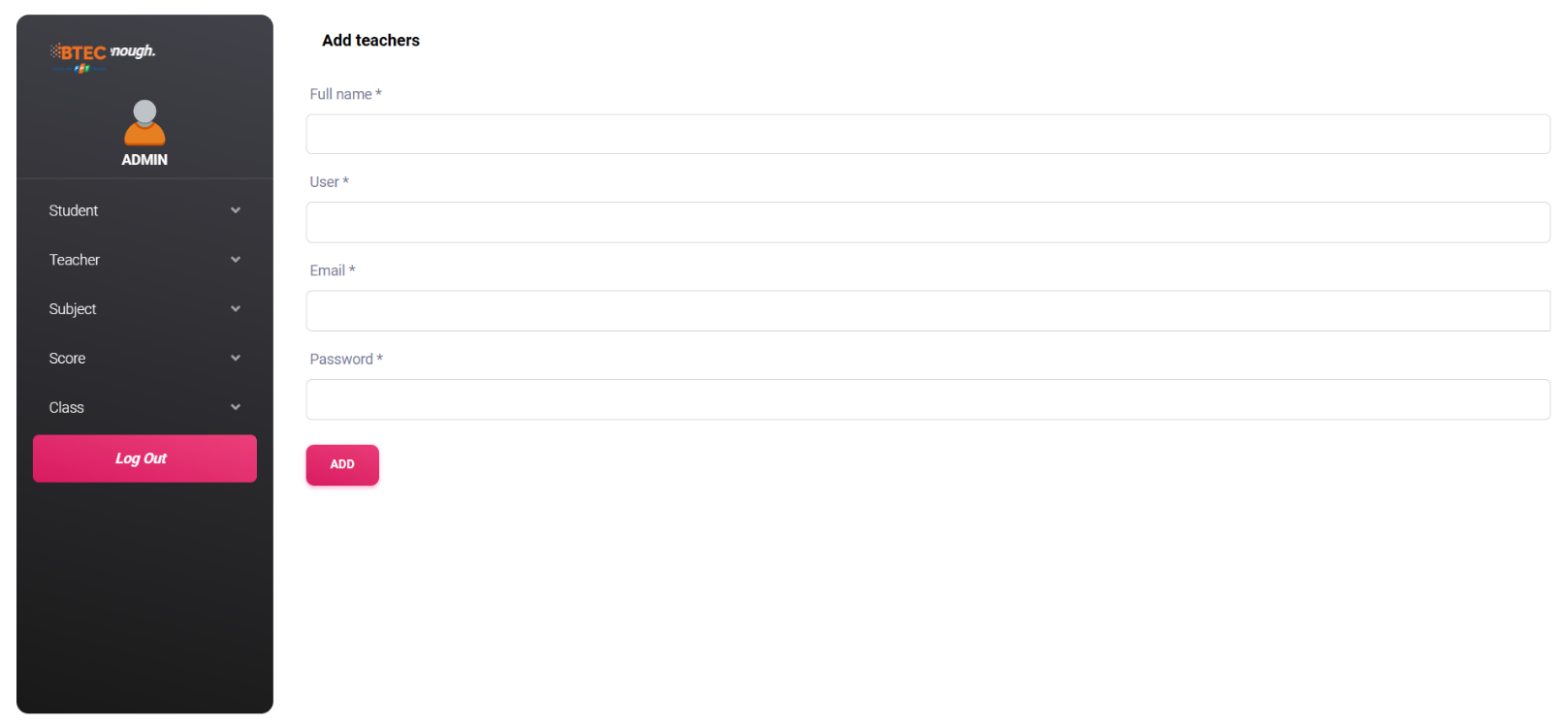
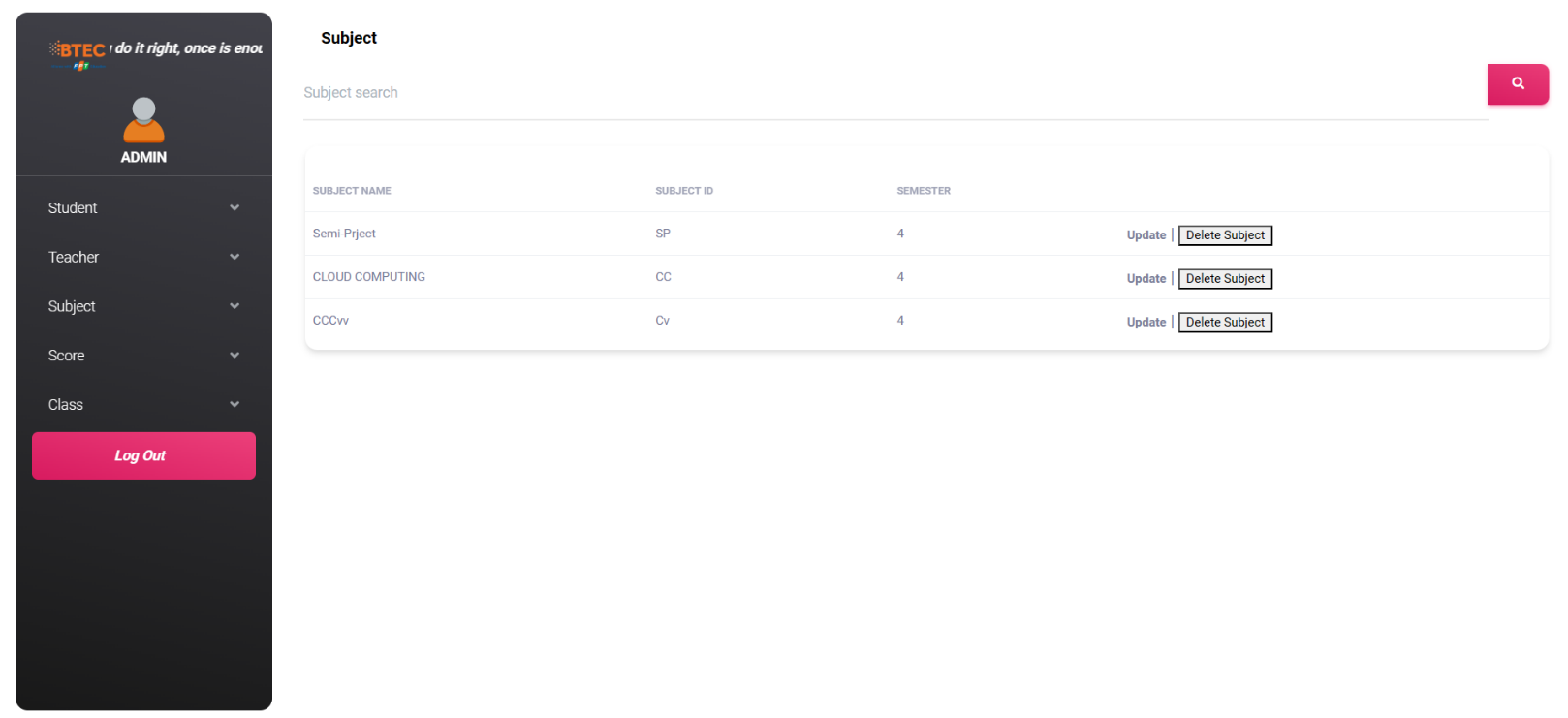


Figure 22 : Web Screenshot:Admin/Teacher/Add Teacher

* + **Manage subject**

Figure 23 : Web Screenshot:Admin/Teacher/Manage subject



Ảnh có chứa văn bản

Mô tả được tạo tự động

Figure 24 : Web Screenshot:Admin/Teacher/Add subject

* + **Manage score**

Figure 25 : Web Screenshot:Admin/Teacher/Manage score/Request to correct points

Ảnh có chứa văn bản

Mô tả được tạo tự động

Ảnh có chứa văn bản

Mô tả được tạo tự động

Figure 26 : Web Screenshot:Admin/Teacher/Manage score/Scoreboard by subject

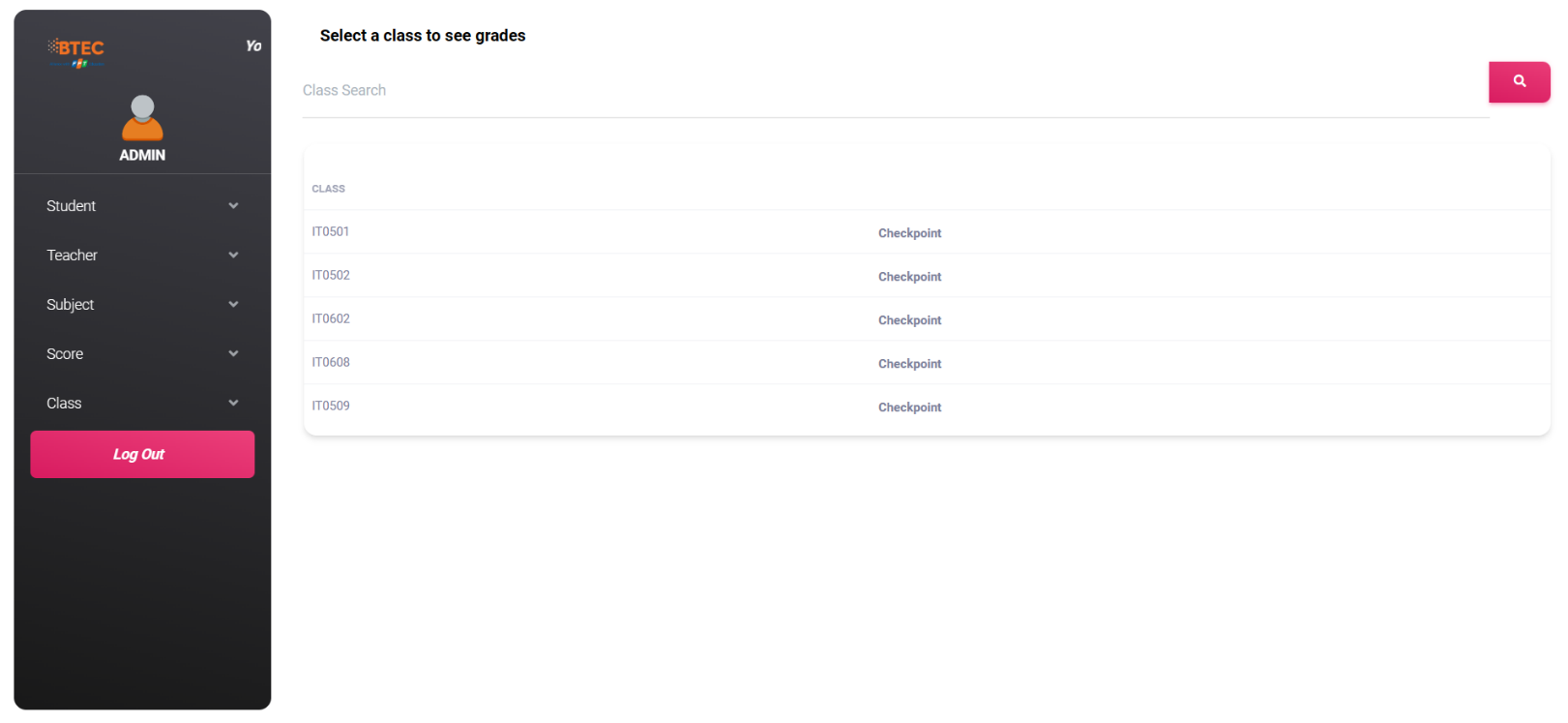


Figure 27 : Web Screenshot:Admin/Teacher/Manage score/Scoreboard by class

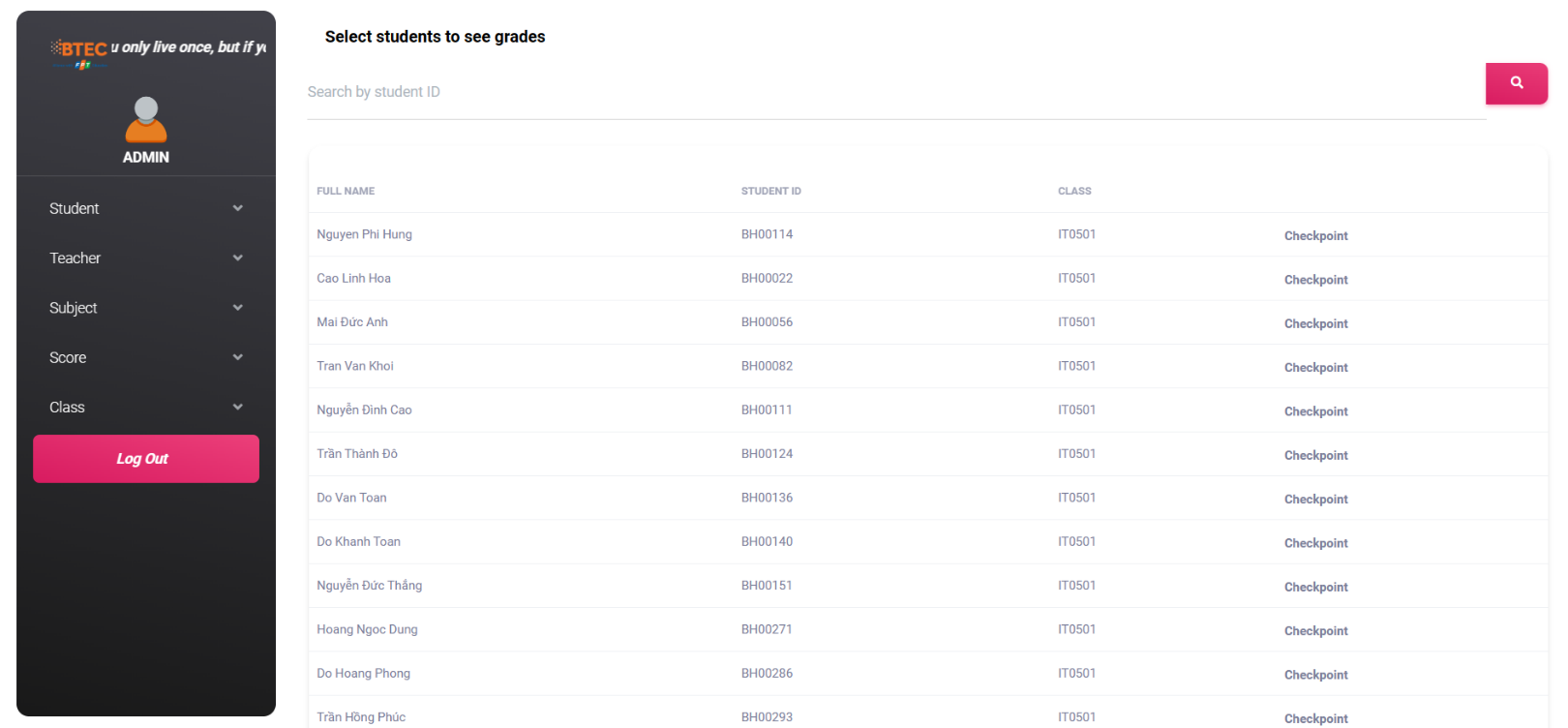


Figure 28 : Web Screenshot:Admin/Teacher/Manage score/Scoreboard by students

Figure 29 : Web Screenshot:Admin/Teacher/Manage score/Scoreboard by term

Ảnh có chứa văn bản

Mô tả được tạo tự động

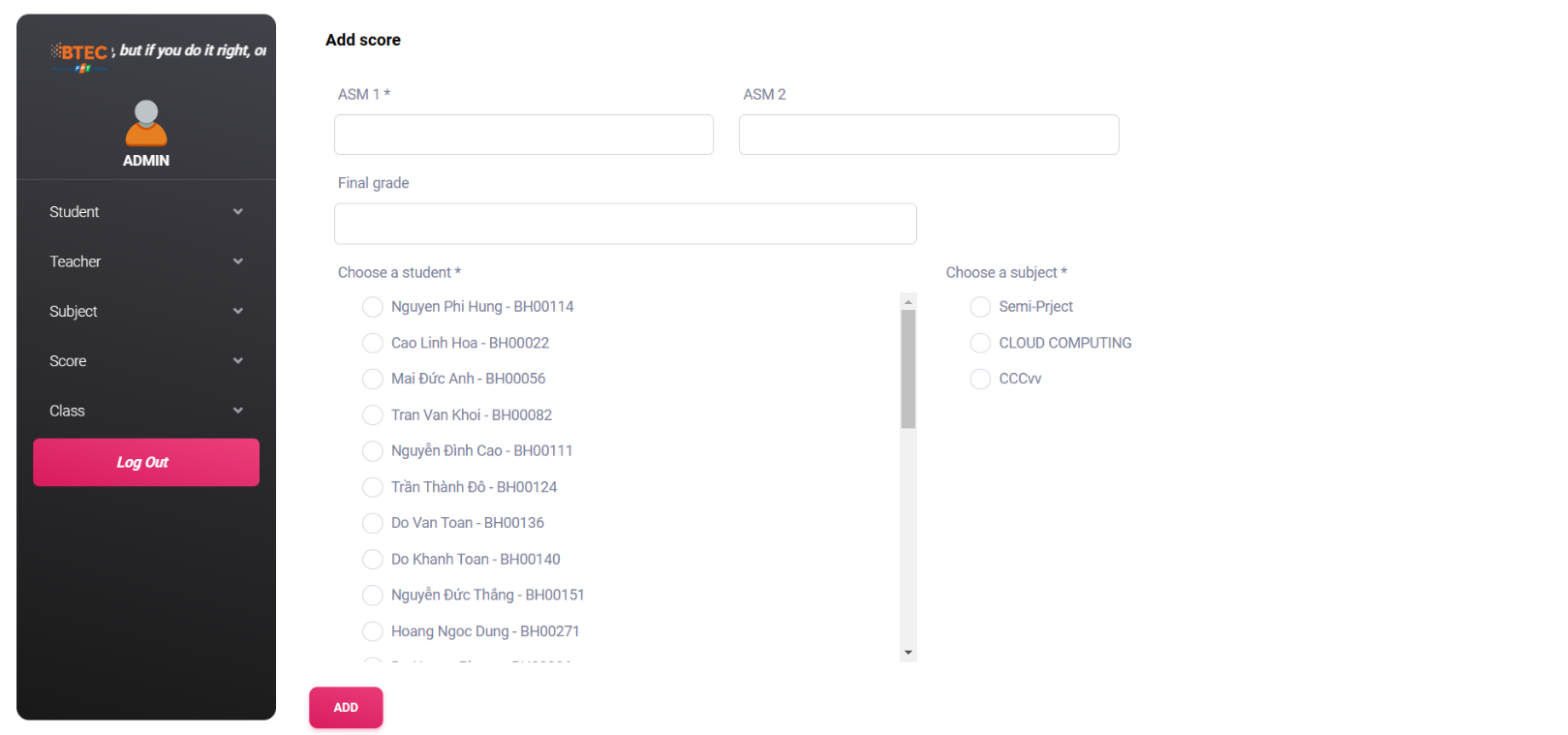


Figure 30 : Web Screenshot:Admin/Teacher/Manage score/Add score

* + **Ảnh có chứa bàn

    Mô tả được tạo tự độngManage class**

Figure 31 : Web Screenshot:Admin/Teacher/Manage class

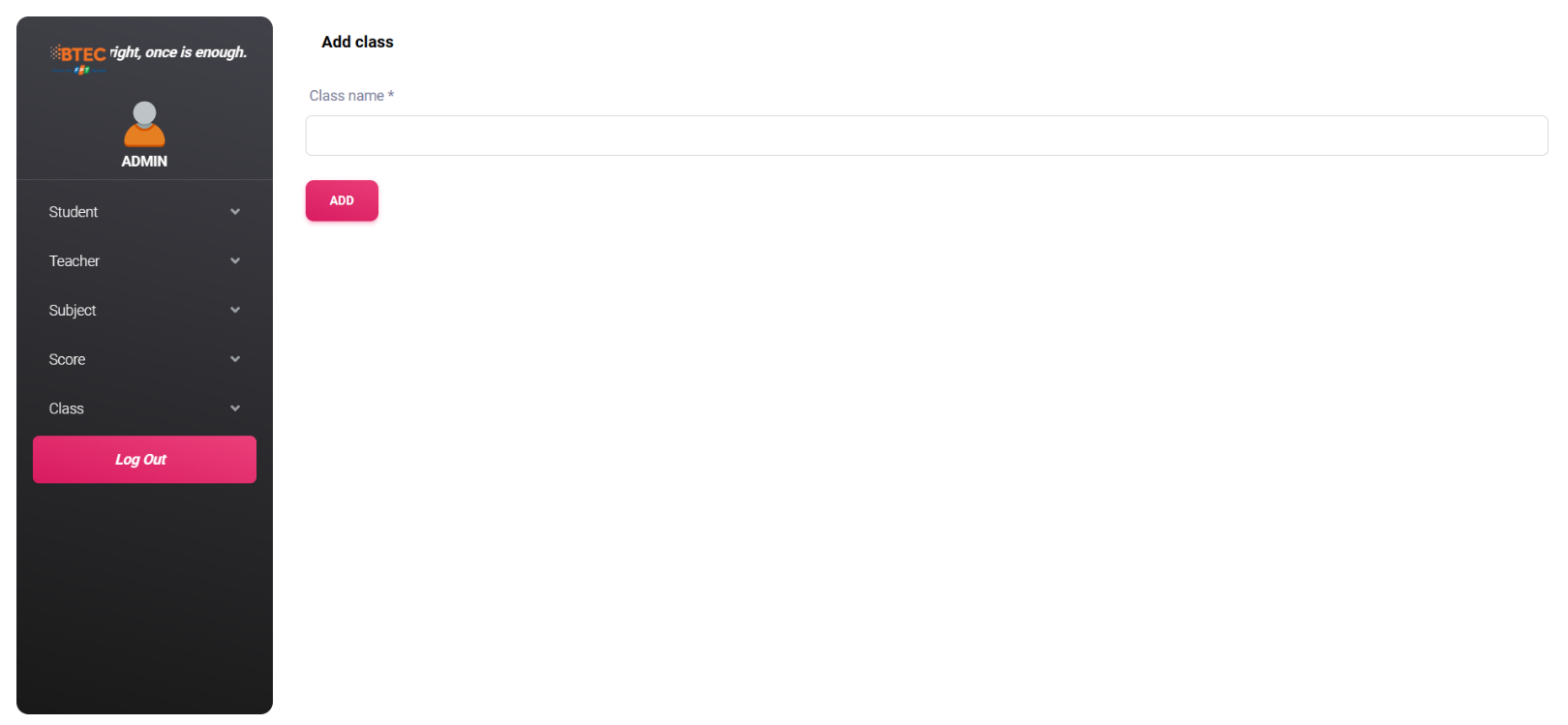


Figure 32:Web Screenshot:Admin/Teacher/Manage class/Add class

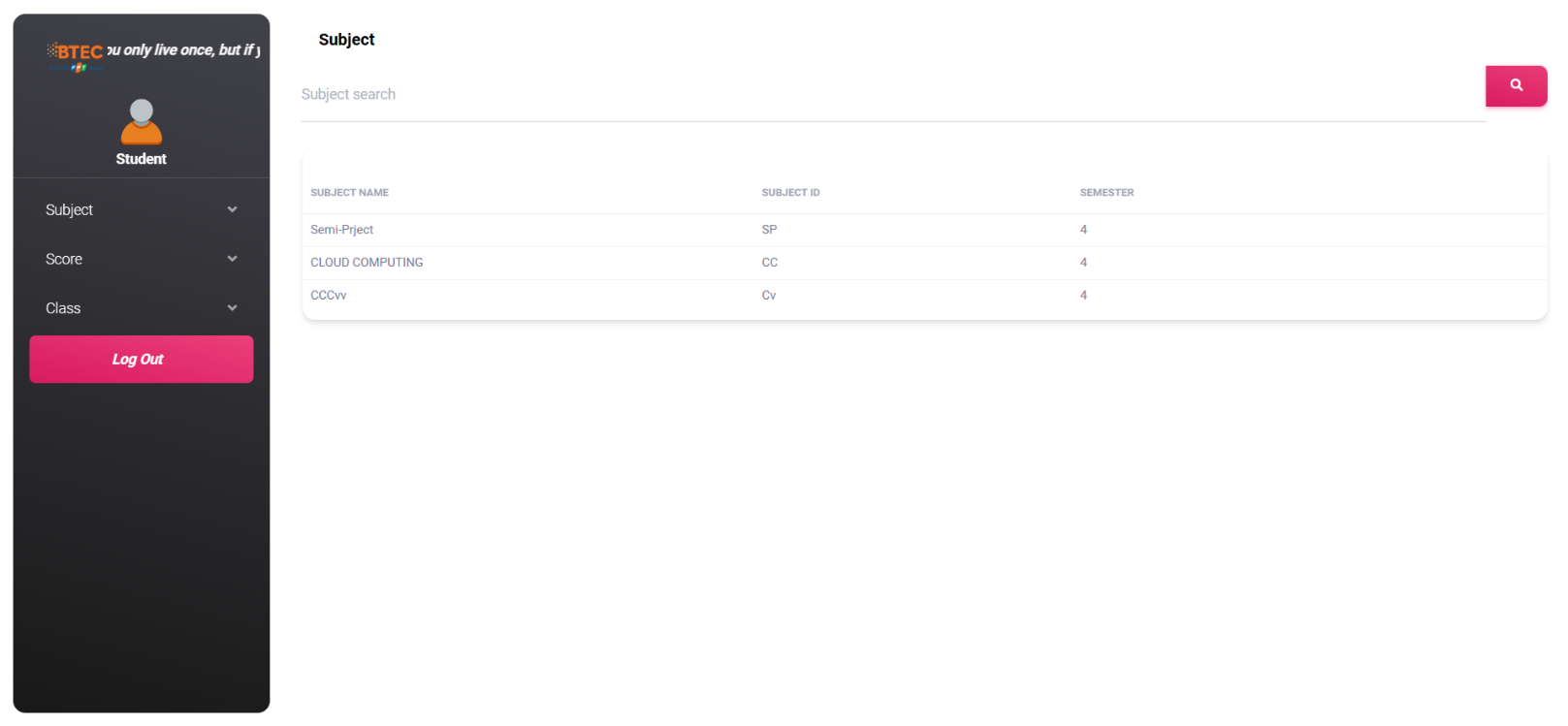
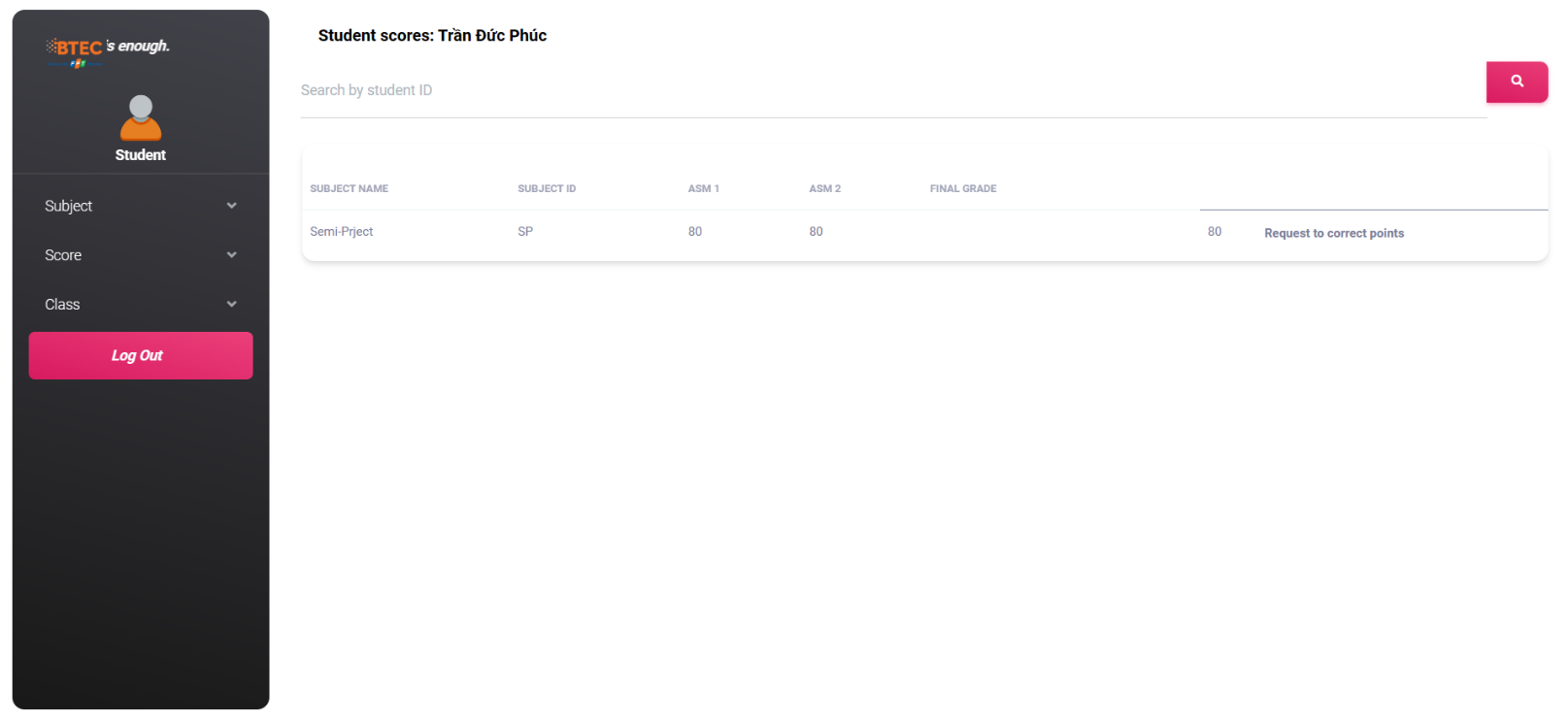
* **Student**
  + **View subject**

Figure 33 : Web Screenshot:Student/view subject

* + **Score**

Figure 34 : Web Screenshot:Student/view core/personal scoreboard



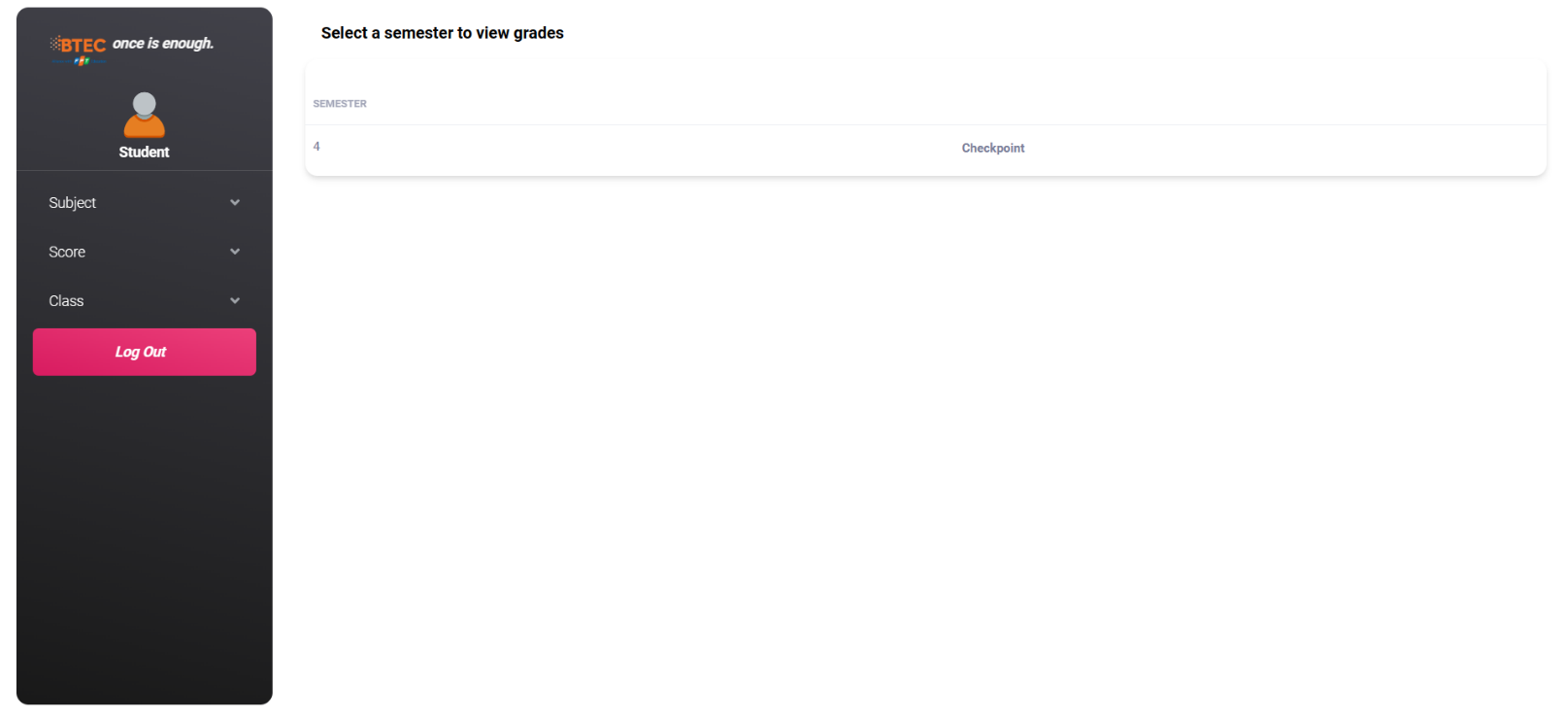


Figure 35 : Web Screenshot:Student/view sore/scoreboard by term

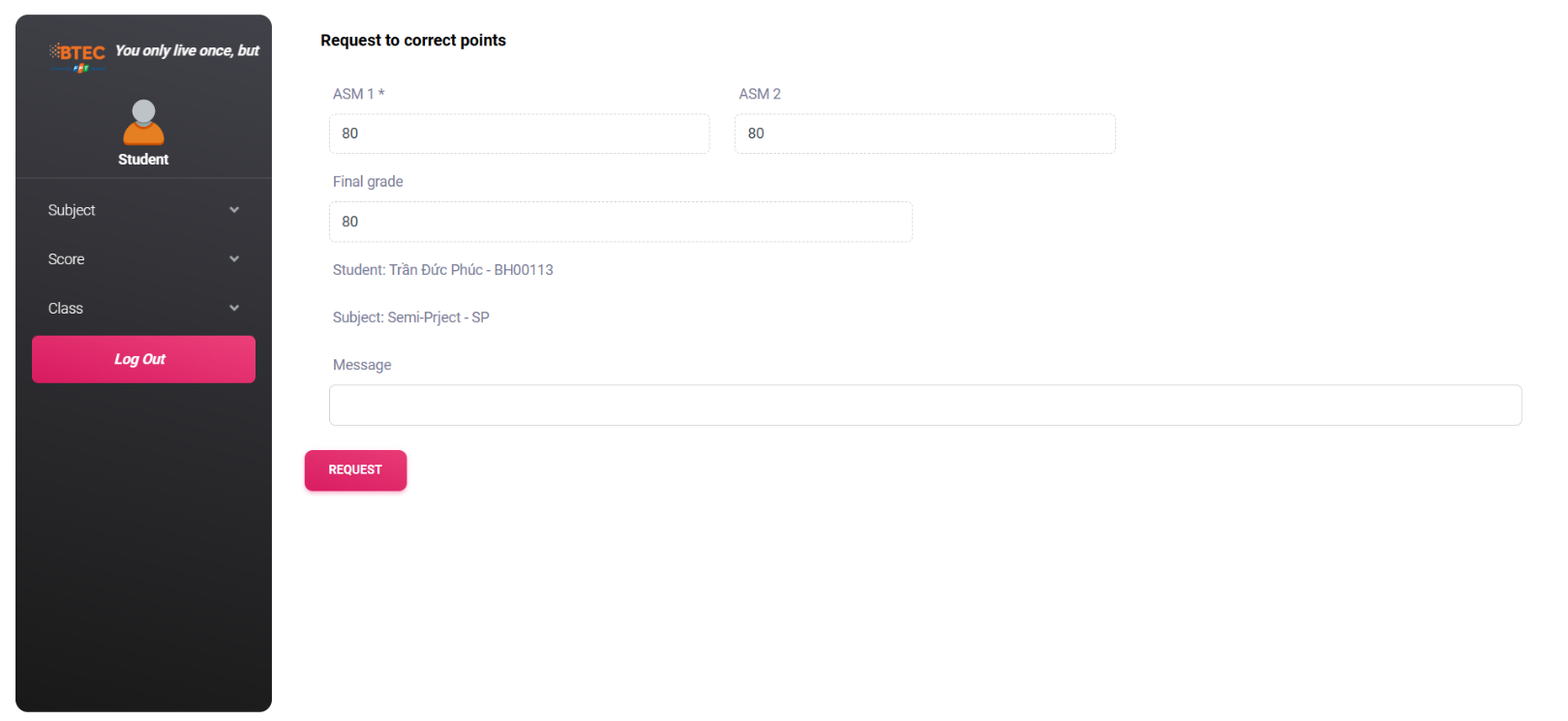


Figure 36 : Web Screenshot:Student/request to correst points

* **View class**

Ảnh có chứa bàn

Mô tả được tạo tự động

Figure 37 : Web Screenshot:Student/view class

1. **Test Case**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Testcase** | **Description** | **Expected result** | **Actual result** | **Status** |
| 01 | Login | User enters User and Password in the form to match the database | Return view home | Return view home | **Pass** |
| 02 | Log Out | The user clicks the logout button on the sidebar | Back to login page | Back to login page | **Pass** |
| 03 | Add student  (Admin/Teacher) | Fill in the form, when the add button is pressed, the system sends the added message, then the information is added to the database | One more student can take many classes,Return view list of studnets | One more student can take many classes,Return view list of studnets | **Pass** |
| 04 | Edit student  (Admin/Teacher) | Edit the information to be corrected in the form, when clicking the update button, the system gives an updated message, then the information is updated on the database | Return view list of studnets | Return view list of studnets | **Pass** |
| 05 | Delete student (Admin/Teacher) | When pressing the Delete button, the system sends a message "Are you sure you want to delete this user account?", when confirming the information will be deleted on the database. | Return view list of studnets | Return view list of studnets | **Pass** |
| 06 | Search student | User enters any related information (name, Id,..) in the dialog box to search | Come view student | Come view student | **Pass** |
| 07 | Add teacher  (Admin/Teacher) | Fill in the information in the form, when the add button is pressed, the system sends an added notification, then the information is added to the database | Add teachers to the system and assign them to each class, Return view list of teacher | Add teachers to the system | **Fail** |
| 08 | Edit teacher  (Admin/Teacher) | Edit the information to be corrected in the form, when clicking the update button, the system gives an updated message, then the information is updated on the database | Return view list of teacher | Return view list of teacher | **Pass** |
| 09 | Delete teacher (Admin/Teacher) | When pressing the Delete button, the system sends a message "Are you sure you want to delete this user account?", when confirming the information will be deleted on the database | Return view list of teacher | Return view list of teacher | **Pass** |
| 10 | Search teacher | User enters any related information (name, Id,..) in the dialog box to search | Come view teacher | Come view teacher | **Pass** |
| 11 | Add subject  (Admin/Teacher) | Fill in the information in the form, when the add button is pressed, the system sends an added notification, then the information is added to the database | Add subjects to the system, sort subjects by class, sort subjects by term, return to the list of subjects | Add a subject to the system, sort subjects by term, return to the list of subjects | **Fail** |
| 12 | Update subject  (Admin/Teacher) | Edit the information to be corrected in the form, when clicking the update button, the system gives an updated message, then the information is updated on the database | Return view list of subject | Return view list of subject | **Pass** |
| 13 | Delete subject (Admin/Teacher) | When the Delete button is pressed, the system sends a message "Are you sure you want to delete this course?", when confirming the information will be deleted on the database. | Return view list of subject | Return view list of subject | **Pass** |
| 14 | Search (subject) | User enters any related information (name, Id,..) in the dialog box to search | Come view subject | Come view subject | **Pass** |
| 15 | Request to correct points(Admin/Teacher) | Showing a request to correct a score from a student | Displayed at request correction page | Displayed at request correction page | **Pass** |
| 16 | Search (Request to correct points) | User enters any related information (name, Id,..) in the dialog box to search | Come view student | Come view student | **Pass** |
| 17 | Add score  (Admin/Teacher) | Fill in the information in the form, when the add button is pressed, the system sends an added notification, then the information is added to the database | Return view Scoreboard by student | Return view Scoreboard by student | **Pass** |
| 18 | Edit score  (Admin/Teacher) | Edit the information to be corrected in the form, when clicking the update button, the system gives an updated message, then the information is updated on the database | Return view Scoreboard by student | Return view Scoreboard by student | **Pass** |
| 19 | Delete score  (Admin/Teacher) | When the Delete button is pressed, the system sends a message "Are you sure you want to delete this student's grades?", when confirming the information will be deleted from the database. | Return view Scoreboard by student | Return view Scoreboard by student | **Pass** |
| 20 | View scoreboard by subject | For the convenience of tracking and editing scores , admin/teacher can view the scoreboard by subject . | View the Scoreboard by subject, select the subject (press the Checkpoint button) then you can see the students with that subject score | View the Scoreboard by subject, select the subject (press the Checkpoint button) then you can see the students with that subject score | **Pass** |
| 21 | Search (scoreboard by subject) | User enters any related information (name, Id,..) in the dialog box to search | Come view subject | Come view subject | **Pass** |
| 22 | View scoreboard by class | For the convenience of tracking and editing scores , admin/teacher can view the scoreboard by class. | View scoreboard by class, select class (click Checkpoint button) to see student's score in that class. | View Gradebook by class, select class (click Checkpoint button) to see student's score in that class. | **Pass** |
| 23 | Search (scoreboard by class) | User enters any related information (name, Id,..) in the dialog box to search | Come view class | Come view class | **Pass** |
| 24 | View scoreboard by student | For the convenience of tracking and editing scores , admin/teacher can view the scoreboard by student. | View Scoreboard by student, select study (click Checkpoint button) to see that student's score. | View Scoreboard by student, select study (click Checkpoint button) to see that student's score. | **Pass** |
| 25 | Search (scoreboard by student) | User enters any related information (name, Id,..) in the dialog box to search | Come view student | Come view student | **Pass** |
| 26 | View scoreboard by term | For the convenience of tracking and editing scores , admin/teacher can view the scoreboard by term . | View scoreboard by term, select the semester (click Test Score button) to see the scores of all students in the selected semester. | View Gradebook by Semester, select the semester (click Test Score button) to see the scores of all students in the selected semester. | **Pass** |
| 27 | Search (scoreboard by term) | User enters any related information (name, Id,..) in the dialog box to search | Come view term | Come view term | **Pass** |
| 28 | Add class  (Admin/Teacher) | Fill in the information in the form, when the add button is pressed, the system sends an added notification, then the information is added to the database | Add class to the system, return to the class list page | Add class to the system, return to the class list | **Pass** |
| 29 | Update class  (Admin/Teacher) | Edit the information to be corrected in the form, when clicking the update button, the system gives an updated message, then the information is updated on the database | Return view Class list | Return view Class list | **Pass** |
| 30 | Delete class (Admin/Teacher) | When the Delete button is pressed, the system sends a message "Are you sure you want to delete this course?", when confirming the information will be deleted on the database. | Must leave all students to ask the class to delete the class, if the class is still there, the system will report an error and give a message, if the deletion is successful Return view Class list | Must leave all students to ask the class to delete the class, if the class is still there, the system will report an error and give a message, if the deletion is successful | **Pass** |
| 31 | Search Class | User enters any related information (name, Id,..) in the dialog box to search | Come view Class | Come view Class | **Pass** |
| 32 | Request to correct points (student) | Fill in the information in the form, when the add button is pressed, the system sends an added notification, then the information is added to the database | send request to teacher, go back to score list page | send request to teacher, go back to score list page | **Pass** |

1. **Conclusion**

Upon completing the software implementation, we conducted a thorough critical review of the final product to identify areas for improvement. Our goal is to continually develop new features to enhance the user experience and make the system more intuitive and efficient.Let's take a closer look at the advantages and disadvantages of the website we have created.Let's take a closer look at the advantages and disadvantages of the website we have created

1. **Advantages**
   * User-friendly interface.The website has a user-friendly interface that is easy to navigate and use, providing an enjoyable user experience.
   * The system is scalable and can handle a large number of users simultaneously. Our website has been minimalistic and has the highest level of page performance and smoothness. According to the statistics some users experience when switching between subpages in the website to reach the highest level with a delay of 1 ms.
   * The utilization of the MVC framework ensures a well-defined processing sequence where independent tasks are carried out without interfering with other project components. Additionally, it facilitates the smooth management of processing and applications.
2. **Disadvantages**

* The current system has limited features and additional features are needed to enhance user experience and improve system functionality.
* Our main issue is security. Website data is vulnerable to attacks like XSS or SQL Injection, and although we've made efforts to encrypt it, our security alert tool has repeatedly warned us of potential risks. .Also we are not knowledgeable enough about security
* Laravel programming language is also a hindrance for us, it is one of the relatively new programming languages that we have to learn from a lot of sources and constantly try to make this project a reality. The project was very difficult and I recognized the efforts of all members, after this project I see the importance of cooperation and teamwork.

1. **Lesson Learnt**

In this project we learned a lot of things:

* + We learn a new language, in the world Laravel is a very popular language, it will help us to have a better cv and especially we learn how to handle situations and handling skills this language
  + Also thanks to laravel language, we understand the MVC pattern and pattern and learn how to handle and develop the model.
  + We also learned that creating the correct ERD and database is very important, we learned how to create tables and the association and relationships between tables. Any mistake in the data processing step, we can face changing the whole code.
  + In addition, I find the interface design is also very important, it is the experience of the eyes, the experience of images and information, sometimes it will also decide for users to use and experience.
  + Teamwork is also a challenge and after all we have found a common voice, united and completed the project.

1. **Future improvement**

As mentioned above, laravel is quite a new language for us and the time to work on it is not really long. Our program still lacks a lot of functions. In the future, maybe we will develop and improve some more functions:

* + Improved security, we will try to learn and apply more security methods to protect users' data
  + We will add the function of adding subjects for each student , due to some problems in making the database our program is letting all students take all the courses , if there are many majors . difficult to manage subjects and students
  + In addition, we will also try to develop so that students can pay directly on WebSite
  + Teachers can import data and export data to Excel files to facilitate data entry, as well as control the list.

1. **Appendix**
   * + 1. **Group member list**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Member Name** | **Member ID** | **Role** |
| 1 | Tran Duc Phuc | BH00113 | Leader |
| 2 | Nguyen Quoc Khanh | BH00114 | Member |
| 3 | Tran Hoang Anh | BH00173 | Member |

* + - 1. **Task**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Task** | **Assign** | **Task description** |
| 1 | Front-End (Teacher – Student) | Tran Duc Phuc | User interface design, the complete interface on the Bootstrap platform, combined with JavaScript and HTML/CSS. |
| 2 | Create a project on Laravel | Tran Duc Phuc  Nguyen Quoc Khanh  Tran Hoang Anh | Create project in Laravel, crop pages, split layout and configure route to display to the user interface. |
| 3 | Create ERD | Nguyen Quoc Khanh | an Entity-Relationship Diagram (ERD) is a graphical representation of entities and their relationships with each other. The ERD is used to visualize the data schema of a system and to communicate the structure of a database to stakeholders. |
| 4 | Create UseCase | Tran Duc Phuc | use refers to a description of specific actions and events that the system is designed to handle. |
| 5 | Create a database on migration with controller and models | Nguyen Quoc Khanh  Tran Hoang Anh  Tran Duc Phuc | Initialize the database, and connect to the .env file. Create relationships between tables on models configuring the data roles and composition of each column. |
| 6 | Create controller in login | Tran Duc Phuc | Connect front-end and back-end of the login section. Check the successful login on the website |
| 7 | Manage Student | Nguyen Quoc Khanh | Create and develop more functionality to edit, delete and connect to the database. |
| 8 | Manage Teacher | Nguyen Quoc Khanh | Create and develop more functionality to edit, delete and connect to the database. |
| 9 | Manage Subject | Tran Hoang Anh | Create and develop more functionality to edit, delete and connect to the database. |
| 10 | Manage Score | Tran Duc Phuc | Create, edit, delete and develop additional functionality that requires editing points and connecting to the database. |
| 11 | Manage Class | Tran Duc Phuc | Create, edit, delete and develop more functionality to view class list details and connect to the database. |

* + - 1. **Link to GitHub**

**You can view and download our code via this** [**GitHub link**](https://github.com/duk-fuc/Semi-Project.git)