

STUDENT REGISTRATION SYSTEM

PART 4: USER MANUAL

COURSE: SOFTWARE ENGINEERING

Fall 2019

CLASS: MONDAY MORNING (LABORATORY)

SATURDAY MORNING (THEORY)

INSTRUCTOR: Ms. N.T.T.LOAN

**Member:**

Trịnh Minh Đăng ITITIU16022

Nguyễn Đức Phi Hồng ITITIU17022

Võ Huy Thành ITITIU17026

|  |
| --- |
| **MEMBER CONTRIBUTION** |

|  |  |  |
| --- | --- | --- |
| **NAME** | **ID** | **CONTRIBUTION** |
| **Nguyễn Đức Phi Hồng** | **ITITIU17022** | III. TEST CASE |
| **Trịnh Minh Đăng** | **ITITIU16022** | I. INSTALL MANUAL |
| **Võ Huy Thành** | **ITITIU17026** | II. USER MANUAL |

**OUTLINE**

|  |
| --- |
|  |

I. INSTALL MANUAL

II. USER MANUAL

III. TEST CASE

**I. INSTALL MANUAL**

**Installation Steps (Configuration)**

1. Download XAMPP

2. Put “onlinecourse” file inside root directory (htdocs folder in xampp folder)

3.Open XAMPP

4.Start XAMPP MySQL and Apache

5. Database Configuration

**Database Configuration**

Open phpmyadmin

Create Database onlinecourse

Import database onlinecourse.sql (available inside zip package)

**Login Details**

To Login as admin put inside browser “http://localhost/onlinecourse/admin”

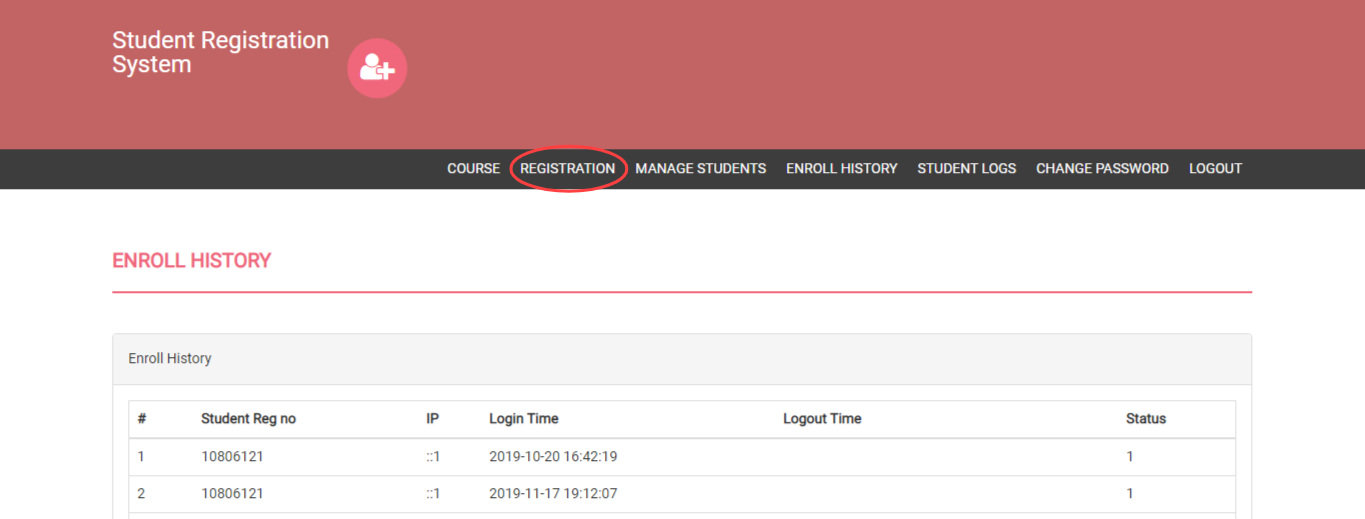
To Login as Student put inside browser “http://localhost/onlinecourse/”

**II. USER MANUAL**

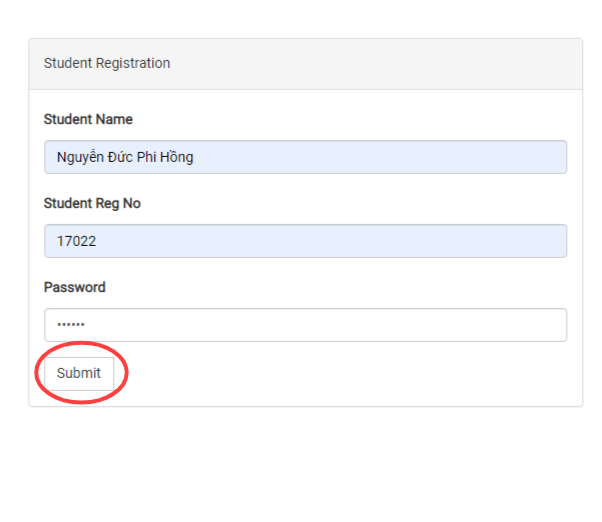
**2.1 Register Account**

Step 1: Login as an admin

Step 2: Admin can registry a student by click on ‘Registration’ button at the menu bar. The Student Registration page will then appear.

****

Step 3: Student Registration page will appear and admin can enter information and click on ‘Sign up’ button to finish registration.



**2.2 Login/Logout Account**

To Login as admin put inside browser “http://localhost/onlinecourse/admin”

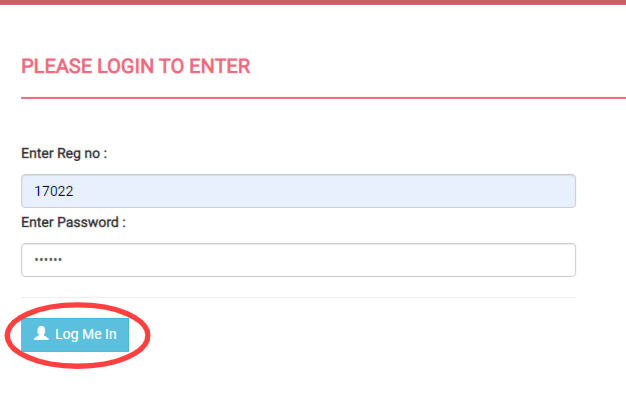
To Login as Student put inside browser “http://localhost/onlinecourse/”

Student and admin have similar login/logout method so we will do the Student Login/Logout

**\*Login**

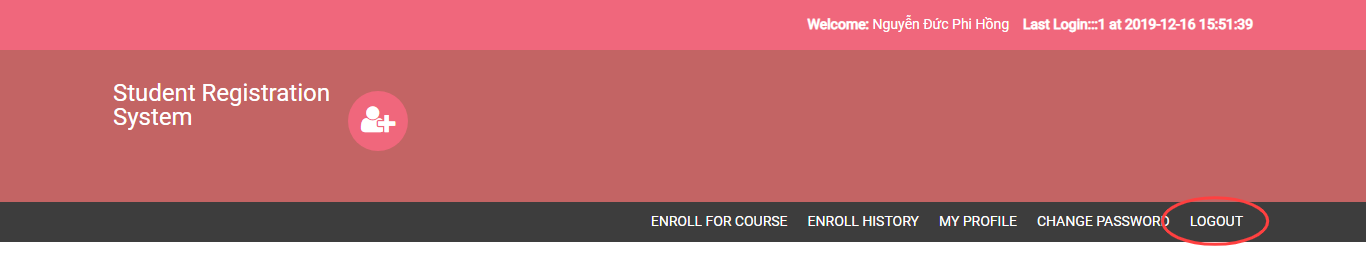
Step 1: Enter web browser

Step 2: Enter information and click ‘Log in’ button to log in.



**\*Logout**

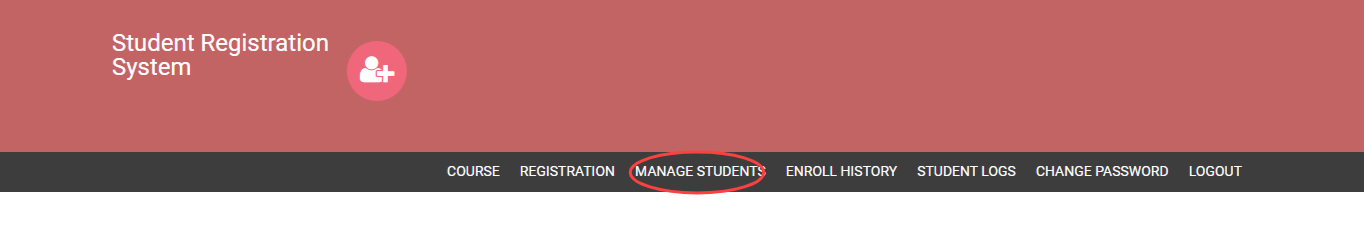
Step 1: Click ‘Log out’ button in the menu bar



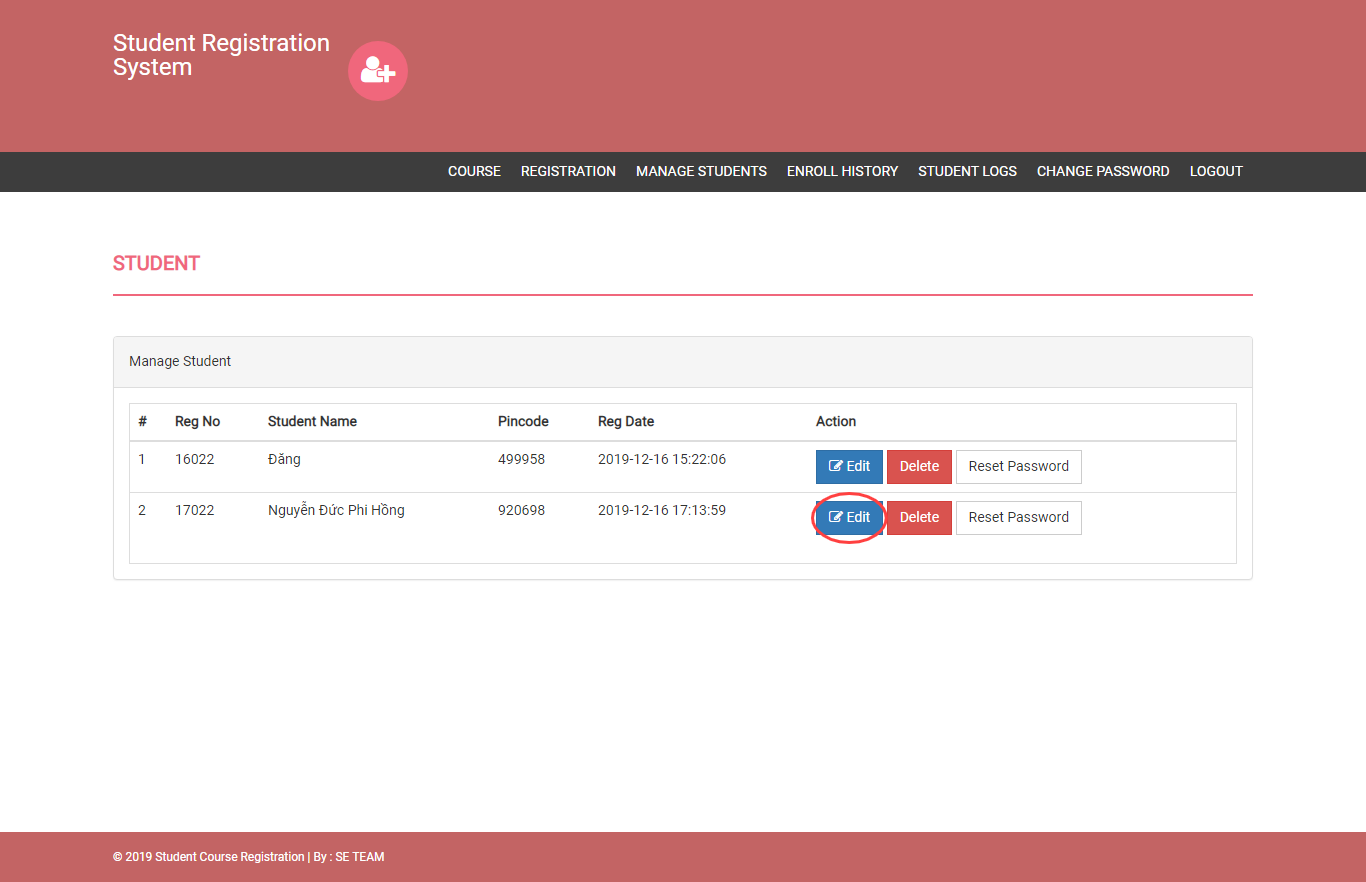
**2.3 View/Edit Student**

Step 1: Login as admin

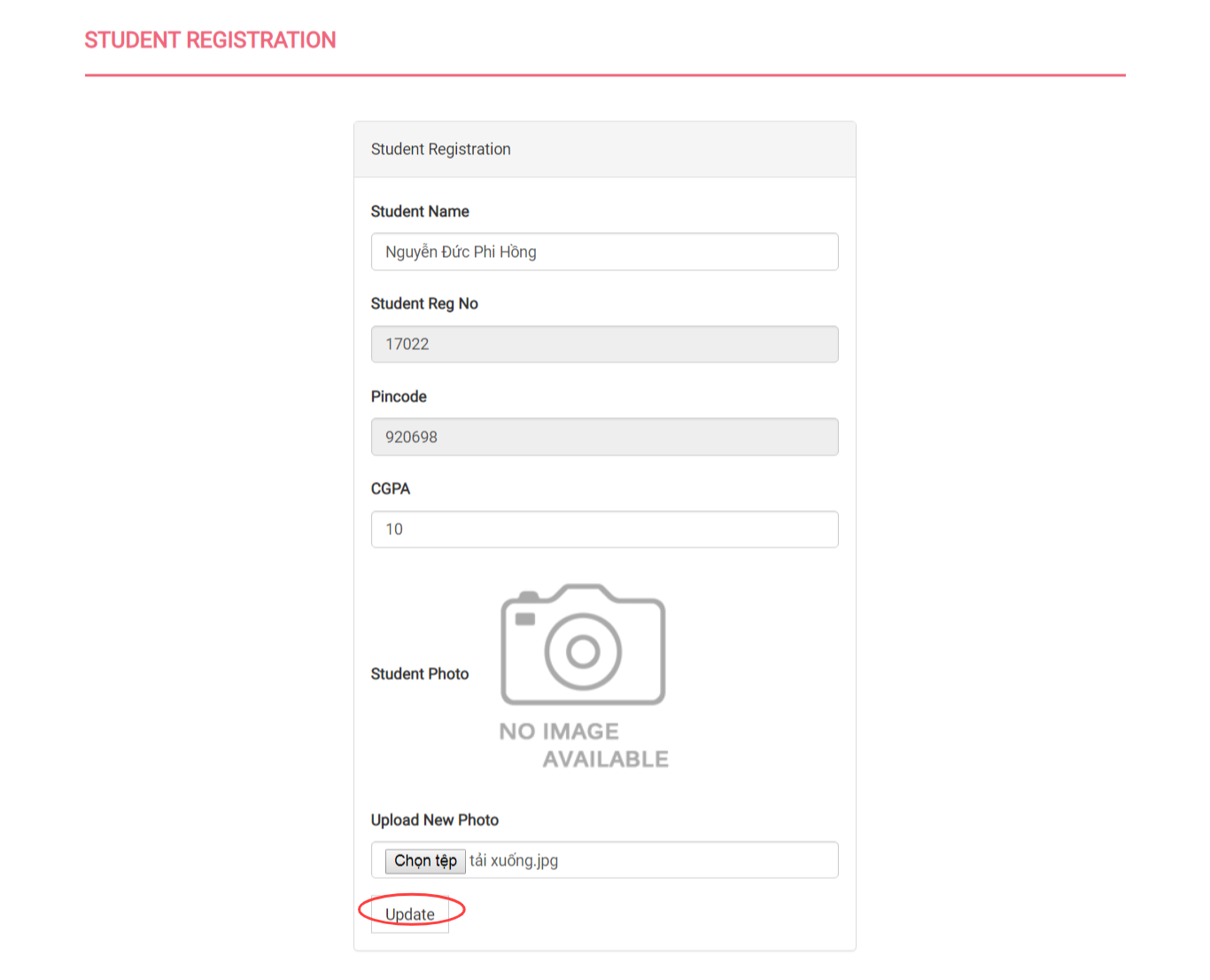
Step 2: Admin can view students list by click on ‘Manage Student’ button at the menu bar. The Manage Student page will then appear.



Step 3: Manage Student page will appear and admin can click on “Edit” to edit a student information



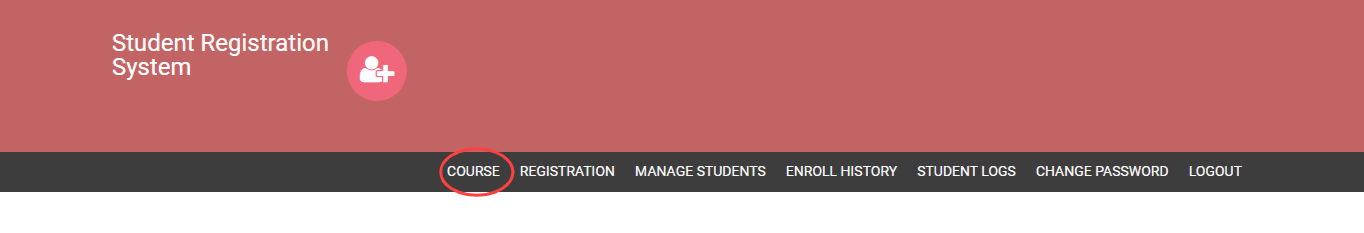
Step 3: Student Profile page will appear and admin can input new information of student and click on ‘Update’ button to finish edit.



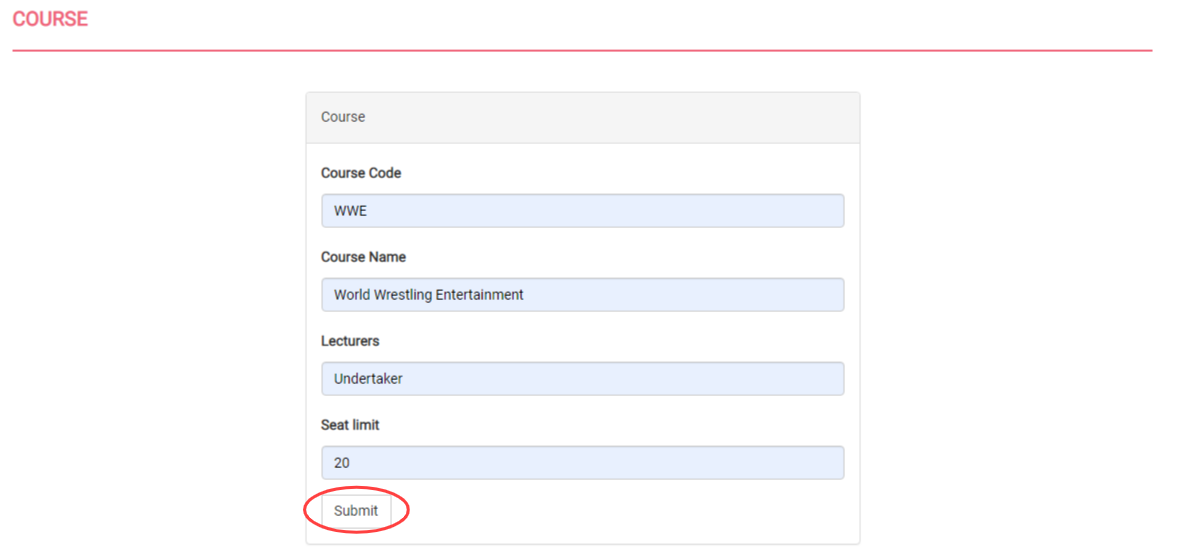
**2.4 Add course**

Step 1: Login as admin

Step 2: Admin can add course by click on ‘Course’ button at the menu bar. The Course page will then appear.

****

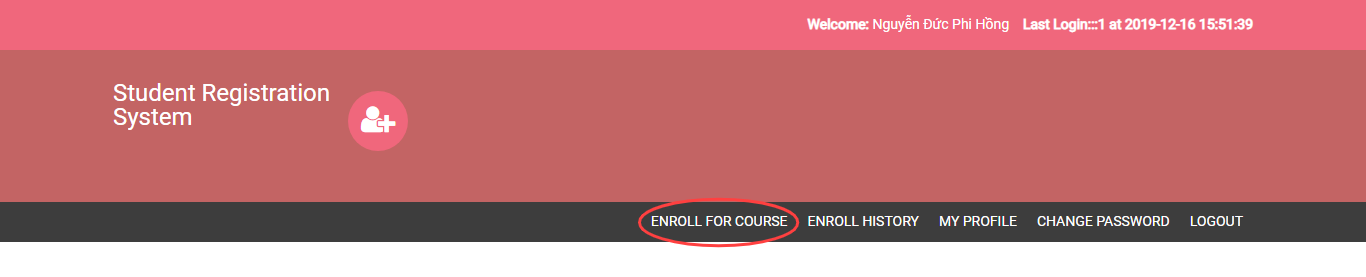
Step 3: Course page will appear and admin can input new information of student and click on ‘Submit’ button to add course.



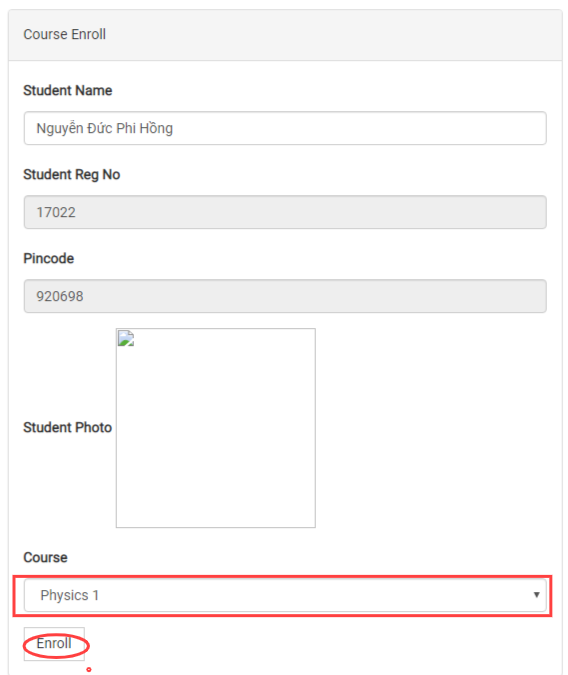
**2.5 Add course (Student)**

Step 1: Login as student

Step 2: Student can enroll for course by click on ‘Enroll for course’ button at the menu bar. The enroll for course page will then appear.



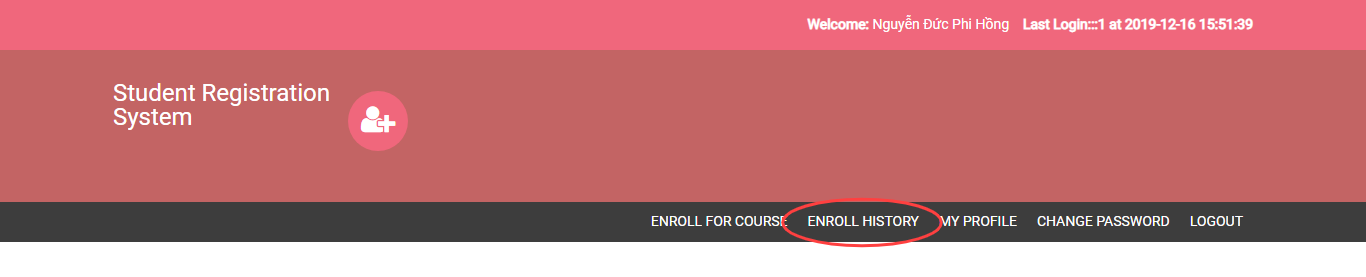
Step 3: Enroll for course page will appear and Student can choose course to enroll and click on ‘Enroll’ button to enroll for course.



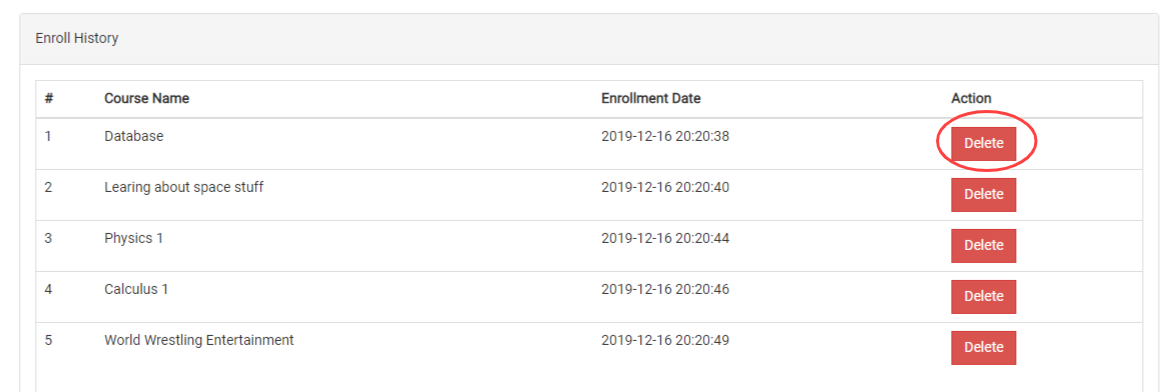
**2.6 View/Remove course (Student)**

Step 1: Login as student

Step 2: Student can view enrolled course by click on ‘Enroll History’ button at the menu bar. The Enroll History page will then appear



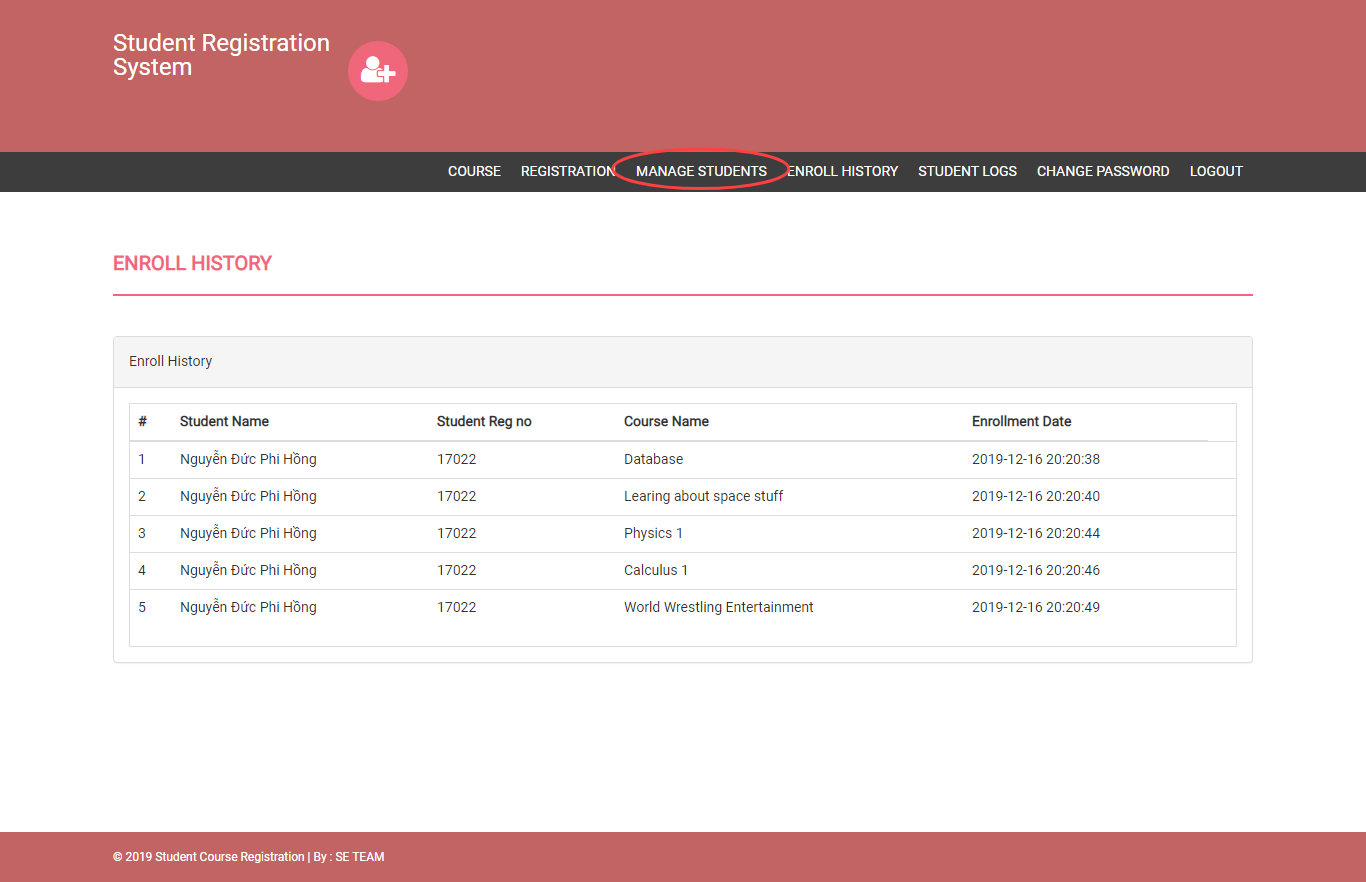
Step 3: Enroll History page will appear and Student can click on ‘Delete’ button to remove course.



**2.7 View student’s course**

Step 1: Login as admin

Step 2: Admin can view student’s enrolled course by click on ‘Enroll History’ button at the menu bar. The Enroll History page will then appear

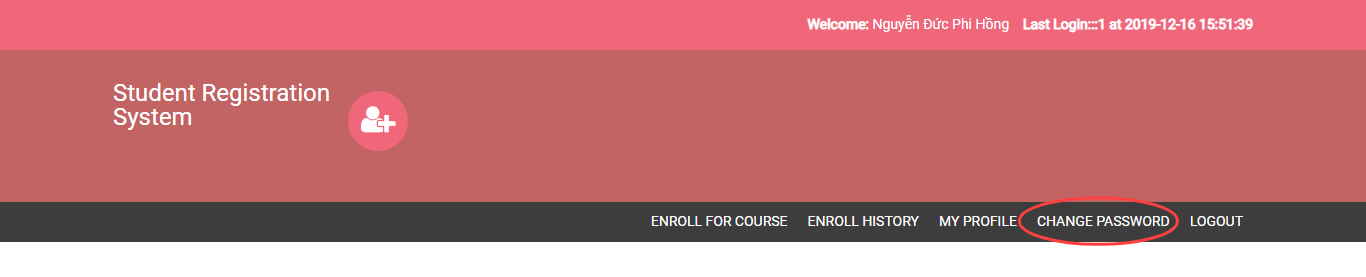


**2.8 Change password**

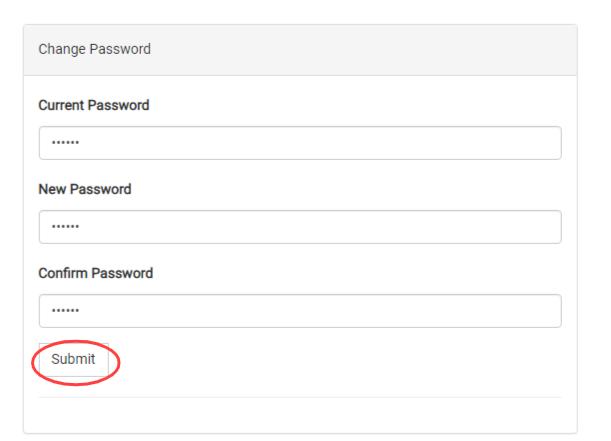
Student and admin have similar password method so we will do the Student Change password

Step 1: Login as student

Step 2: Student can change their account’s password by click on ‘Change password’ button at the menu bar. The Change password page will then appear



Step 3: Enter current password, new password and confirm it. Then click Submit to change password



**III. TEST CASE**

**3.1 Register Account**

**3.1.1 Simple Test**

|  |  |
| --- | --- |
| **Test case** #: 01  **System**: Student Registration System  **Designed by**: SE Team  **Executed by**: SE Team  **Short Description**: Test the Register Account service | **Test case name**: Register Account  **Subsystem**: Account  **Design Date**:19/10/18  **Execution Date**:19/12/18 |

|  |
| --- |
| **Pre-conditions:**  The system displays the Register Account page |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| **1** | Enter name “Phi Hong” |  | Pass |  |
| **2** | Enter Student Reg No “17022” |  | Pass |  |
| **3** | Enter password “123456” |  | Pass |  |
| **4** | Click “Submit” | Succesfull message “Student Registered Successfully !!” appear | Pass |  |
| **5** | **Check Post-condition 1** |  | Pass |  |

|  |
| --- |
| **Post-conditions:** 1. A new account with: Name: “Phi Hong”  Student Reg No: “17022”  Password: “123456”are saved in database |

**3.1.2 Complex Test**

|  |  |
| --- | --- |
| **Test case** #: 01  **System**: Student Registration System  **Designed by**: SE Team  **Executed by**: SE Team  **Short Description**: Test the Register Account service | **Test case name**: Register Account  **Subsystem**: Account  **Design Date**:19/10/18  **Execution Date**:19/12/18 |
| **Pre-conditions:**  The user must be logging in as admin  The system displays the Register Account page | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| **1** | Enter name “Phi Hong” |  | Pass |  |
| **2** | Enter Student Reg No “17022” |  | Pass |  |
| **3** | Enter password “123456” |  | Pass |  |
| **4** | Click “Submit” | Succesfull message “Student Registered Successfully !!” appear | Pass |  |
| **5** | **Check Post-condition 1** |  | Pass |  |
| **6** | Repeat step 1 with name “Huy Thanh” |  | Pass |  |
| **7** | Repeat step 2 with Student Reg No “17022” | The system displays the message warning that the Student Reg No is not valid | Pass |  |
| **8** | Repeat step 2 with Student Reg No “17026” |  | Pass |  |
| **9** | Click “Submit” | The system displays the message warning that the password is empty. | Pass |  |
| **10** | Repeat step 3 with password “159753” |  | Pass |  |
| **11** | Click “Submit” | Return to Homepage with the account is automatically logged in. | Pass |  |
| **12** | **Check Post-condition 2** |  | Pass |  |

|  |
| --- |
| **Post-conditions:**  1. A new account with: Name: “Phi Hong”  Student Reg No: “17022”  Password: “123456”are saved in database  2. A new account with: Name: “Huy Thanh”  Student Reg No: “17026”  Password: “159753”are saved in database |

**3.2 Login/Logout Account**

\*The Logout function is too simple, so my team decide to skip this function

**3.2.1 Simple Test**

|  |  |
| --- | --- |
| **Test case** #: 02  **System**: Student Registration System  **Designed by**: SE Team  **Executed by**: SE Team  **Short Description**: Test the Login Account service | **Test case name**: Login Account  **Subsystem**: Account  **Design Date**:19/10/18  **Execution Date**:19/12/18 |

|  |
| --- |
| **Pre-conditions:**  -The user has not logged in before.  -The user has an activate account: Student Reg No – “17022”,  Password – “123456”.  -The system displays the Login box. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| **1** | Enter Student Reg No “17022” |  | Pass |  |
| **2** | Enter password “123456” |  | Pass |  |
| **3** | Click “Login” button | Return to Homepage with the account is logged in | Pass |  |
| **4** | **Check Post-condition 1** |  | Pass |  |

|  |
| --- |
| **Post-conditions:**  1. User has successfully login |

**3.2.2 Complex Test**

|  |  |
| --- | --- |
| **Test case** #: 02  **System**: Student Registration System  **Designed by**: SE Team  **Executed by**: SE Team  **Short Description**: Test the Login Account service | **Test case name**: Login Account  **Subsystem**: Account  **Design Date**:19/10/18  **Execution Date**:19/12/18 |

|  |
| --- |
| **Pre-conditions:**  -The user has not logged in before.  -The user has an activate account: Student Reg No – “17022”,  Password – “123456”.  -The system displays the Login box. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| **1** | Enter Student Reg No “17022” |  | Pass |  |
| **2** | Enter password “123456” |  | Pass |  |
| **3** | Click “Login” button | Return to Homepage with the account is logged in | Pass |  |
| **4** | **Check Post-condition 1** |  | Pass |  |
| **5** | Repeat step 1,2,3 using wrong Student Reg No “dsg48fgs”. | The system displays an alert about wrong Student Reg No or password. | Pass |  |
| **6** | Repeat step 1,2,3 using wrong password “itu212”. | The system displays an alert about wrong email or password. | Pass |  |
| **7** | Repeat step 1,2,3 without performing anything | The system displays an alert prompting input email/password. | Pass |  |
| **8** | Repeat step 1,2,3 | Return to Home page with the account is logged in. | Pass |  |
| **9** | **Check Post-conditions 1** |  | Pass |  |

|  |
| --- |
| **Post-conditions:**  1. User has successfully login |

**3.3 View/Edit Student**

**3.3.1 Simple Test**

|  |  |
| --- | --- |
| **Test case** #: 03  **System**: Student Registration System  **Designed by**: SE Team  **Executed by**: SE Team  **Short Description**: Test the View/Edit Student function | **Test case name**: View/Edit student  **Subsystem**: Account  **Design Date**:19/10/18  **Execution Date**:19/12/18 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pre-conditions:**  -The user has already logged in as an admin  -The system displays the menu bar with “Manage Student” Button | | | | |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| **1** | Press “Manage Student” Button | The system displays Manage Student page | Pass |  |
| **2** | **Check Post-conditions 1** |  | Pass |  |
| **2** | Press “Edit” button next to a student | The system displays Student profile page | Pass |  |
| **3** | Change the name of the student in the name bar from “Phi Hong” to “Doraemon” |  | Pass |  |
| **4** | Press “Update Button” | Successful message “Student Record Update Successfully !!” appear | Pass |  |
| **5** | **Check Post-conditions 2,3** |  | Pass |  |

|  |
| --- |
| **Post-conditions:**   1. All students are displayed   2. The page is displayed with new information  3. Changes are saved to the database |

**3.3.2 Complex Test**

|  |  |
| --- | --- |
| **Test case** #: 03  **System**: Student Registration System  **Designed by**: SE Team  **Executed by**: SE Team  **Short Description**: Test the View/Edit Student function | **Test case name**: View/Edit student  **Subsystem**: Account  **Design Date**:19/10/18  **Execution Date**:19/12/18 |

|  |
| --- |
| **Pre-conditions:**  -The user has already logged in as an admin  -The system displays the menu bar with “Manage Student” Button |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| **1** | Press “Manage Student” Button | The system displays Manage Student page | Pass |  |
| **2** | **Check Post-conditions 1** |  | Pass |  |
| **2** | Press “Edit” button next to a student | The system displays Student profile page | Pass |  |
| **3** | Change the name of the student in the name bar from “Phi Hong” to “Doraemon” |  | Pass |  |
| **4** | Press “Update Button” | Successful message “Student Record Update Successfully !!” appear | Pass |  |
| **5** | **Check Post-conditions 2,3** |  | Pass |  |
| **6** | Change the CGPA of the student in the CGPA bar from “10” to “9” |  | Pass |  |
| **7** | Press “Update Button” | Successful message “Student Record Update Successfully !!” appear | Pass |  |
| **8** | **Check Post-conditions 2,3** |  | Pass |  |

|  |
| --- |
| **Post-conditions:**   1. All students are displayed   2. The page is displayed with new information  3. Changes are saved to the database |

**3.4 Add course**

**3.4.1 Simple Test**

|  |  |
| --- | --- |
| **Test case** #: 04  **System**: Student Registration System  **Designed by**: SE Team  **Executed by**: SE Team  **Short Description**: Test the Add course function | **Test case name**: Add course  **Subsystem**: Course  **Design Date**:19/10/18  **Execution Date**:19/12/18 |

|  |
| --- |
| **Pre-conditions:**  -The user has already logged in as an admin  -The system displays the menu bar with “Course” Button |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| **1** | Press “Course” Button | The system displays Manage Student page | Pass |  |
| **2** | Enter course’s information (code,name,unit,seat limit) |  | Pass |  |
| **3** | Press “Submit” button next to a student | Successful message “Course Created Successfully !!” appear | Pass |  |
| **4** | **Check Post-conditions 1,2** |  | Pass |  |

|  |
| --- |
| **Post-conditions:**  1. The system displays a successful Courses box  2. The system add course in database |

**2.4.2 Complex Test**

|  |  |
| --- | --- |
| **Test case** #: 04  **System**: Student Registration System  **Designed by**: SE Team  **Executed by**: SE Team  **Short Description**: Test the Add course function | **Test case name**: Add course  **Subsystem**: Course  **Design Date**:19/10/18  **Execution Date**:19/12/18 |

|  |
| --- |
| **Pre-conditions:**  The system displays the Course page |

|  |
| --- |
| **Pre-conditions:**  -The user has already logged in as an admin  -The system displays the menu bar with “Course” Button |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| **1** | Press “Course” Button | The system displays Manage Student page | Pass |  |
| **2** | Enter course’s information (code,name,unit,seat limit) |  | Pass |  |
| **3** | Press “Submit” button next to a student | Successful message “Course Created Successfully !!” appear | Pass |  |
| **4** | **Check Post-conditions 1,2** |  | Pass |  |
| **5** | Repeat step 1,2,3 without performing anything | The system displays an alert prompting input. | Pass |  |
| **6** | Repeat step 1,2,3 | Successful message “Course Created Successfully !!” appear | Pass |  |
| **7** | **Check Post-conditions 1,2** |  | Pass |  |

|  |
| --- |
| **Post-conditions:**  1. The system displays a successful Courses box  2. The system add course in database |

**3.5 Add course (Student)**

**3.5.1 Simple Test**

|  |  |
| --- | --- |
| **Test case** #: 05  **System**: Student Registration System  **Designed by**: SE Team  **Executed by**: SE Team  **Short Description**: Test the Add course (student) function | **Test case name**: Add course (student)  **Subsystem**: Course  **Design Date**:19/10/18  **Execution Date**:19/12/18 |

|  |
| --- |
| **Pre-conditions:**  -The user has already logged in as an Student  -The system displays the menu bar with “Enroll Course” Button |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| **1** | Press “Enroll for Course” Button | The system displays Enroll Course page | Pass |  |
| **2** | Choose “Database” Course |  | Pass |  |
| **3** | Press “Enroll” button | Successful message “Course Enrolled Successfully !!” appear | Pass |  |
| **4** | **Check Post-conditions 1** |  | Pass |  |

|  |
| --- |
| **Post-conditions:**  1. The system add course in database |

**3.5.2 Complex Test**

|  |  |
| --- | --- |
| **Test case** #: 05  **System**: Student Registration System  **Designed by**: SE Team  **Executed by**: SE Team  **Short Description**: Test the Add course (student) function | **Test case name**: Add course (student)  **Subsystem**: Course  **Design Date**:19/10/18  **Execution Date**:19/12/18 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pre-conditions:**  -The user has already logged in as an Student  -The system displays the menu bar with “Enroll Course” Button | | | | |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| **1** | Press “Enroll For Course” Button | The system displays Enroll Course page | Pass |  |
| **2** | Choose “Database” Course |  | Pass |  |
| **3** | Press “Submit” button next to a student | Successful message “Course Created Successfully !!” appear | Pass |  |
| **4** | **Check Post-conditions 1** |  | Pass |  |
| **5** | Choose “Learing about space stuff” Course (which is full) |  | Pass |  |
| **6** | Press “Submit” button next to a student | Fail message “Seat not available for this course. All Seats Are full “appear | Pass |  |

|  |
| --- |
| **Post-conditions:**  1. The system add course in database |

**3.6 View/Remove course (Student)**

**3.6.1 Simple Test**

|  |  |
| --- | --- |
| **Test case** #: 06  **System**: Student Registration System  **Designed by**: SE Team  **Executed by**: SE Team  **Short Description**: Test the View/Remove course (Student)  function | **Test case name**: View/Remove course (Student)  **Subsystem**: Course  **Design Date**:19/10/18  **Execution Date**:19/12/18 |

|  |
| --- |
| **Pre-conditions:**  -The user has already logged in as an Student  -The system displays the menu bar with “Enroll History” Button |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| **1** | Press “Enroll History” Button | The system displays Enroll History page | Pass |  |
| **2** | Press “Delete” button next to a Course | Successful message “Course Remove Successfully !!” appear | Pass |  |
| **3** | **Check Post-conditions 1** |  | Pass |  |

|  |
| --- |
| **Post-conditions:**  1. The system remove course from database |

**3.6.2 Complex Test**

|  |  |
| --- | --- |
| **Test case** #: 06  **System**: Student Registration System  **Designed by**: SE Team  **Executed by**: SE Team  **Short Description**: Test the View/Remove course (Student)  function | **Test case name**: View/Remove course (Student)  **Subsystem**: Course  **Design Date**:19/10/18  **Execution Date**:19/12/18 |

|  |
| --- |
| **Pre-conditions:**  -The user has already logged in as an Student  -The system displays the menu bar with “Enroll History” Button |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| **1** | Press “Enroll History” Button | The system displays Enroll History page | Pass |  |
| **2** | Press “Delete” button next to Database Course | Successful message “Course Remove Successfully !!” appear | Pass |  |
| **3** | **Check Post-conditions 1** |  | Pass |  |
| **4** | Press “Enroll History” Button | The system displays Enroll History page | Pass |  |
| **5** | Press “Delete” button next to Physics 1 Course | Successful message “Course Remove Successfully !!” appear | Pass |  |
| **6** | **Check Post-conditions 1** |  | Pass |  |
| **7** | Press “Enroll History” Button | The system displays Enroll History page | Pass |  |
| **8** | Press “Delete” button next to Calculus 1 Course | Successful message “Course Remove Successfully !!” appear | Pass |  |
| **9** | **Check Post-conditions 1** |  | Pass |  |

|  |
| --- |
| **Post-conditions:**  1. The system remove course from database |

**3.7 View student’s course**

**This use case is simple so it don’t have complex test**

**3.7.1 Simple test**

|  |  |
| --- | --- |
| **Test case** #: 07  **System**: Student Registration System  **Designed by**: SE Team  **Executed by**: SE Team  **Short Description**: Test the View student’s course function | **Test case name**: View student’s course  **Subsystem**: Course  **Design Date**:19/10/18  **Execution Date**:19/12/18 |

|  |
| --- |
| **Pre-conditions:**  -The user has already logged in as an admin  -The system displays the menu bar with “Enroll History” Button |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| **1** | Press “Enroll History” Button | The system displays Enroll History page | Pass |  |
| **2** | **Check Post-conditions 1** |  | Pass |  |

|  |
| --- |
| **Post-conditions:**  1. The system displays a successful Enrolled history box |

**3.8 Change password**

**3.8.1 Simple test**

|  |  |
| --- | --- |
| **Test case** #: 08  **System**: Student Registration System  **Designed by**: SE Team  **Executed by**: SE Team  **Short Description**: Test the Change password function | **Test case name**: Change password  **Subsystem**: Account  **Design Date**:19/10/18  **Execution Date**:19/12/18 |

|  |
| --- |
| **Pre-conditions:**  -The user has already logged in  -The system displays the menu bar with “Change password” Button |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| **1** | Press “Change password” Button | The system displays Change password page | Pass |  |
| **2** | Enter current password, new password and confirm password (same as new password) |  | Pass |  |
| **3** | Press “Submit” button next to a student | Successful message “Change password Successfully !!” appear | Pass |  |
| **4** | **Check Post-conditions 1** |  | Pass |  |

|  |
| --- |
| **Post-conditions:**  1. Changes are saved to the database |

**3.8.2 Complex test**

|  |  |
| --- | --- |
| **Test case** #: 08  **System**: Student Registration System  **Designed by**: SE Team  **Executed by**: SE Team  **Short Description**: Test the Change password function | **Test case name**: Change password  **Subsystem**: Account  **Design Date**:19/10/18  **Execution Date**:19/12/18 |

|  |
| --- |
| **Pre-conditions:**  -The user has already logged in  -The system displays the menu bar with “Change password” Button |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| **1** | Press “Change password” Button | The system displays Change password page | Pass |  |
| **2** | Enter current password, new password and confirm password (same as new password) |  | Pass |  |
| **3** | Press “Submit” button next to a student | Successful message “Change password Successfully !!” appear | Pass |  |
| **4** | **Check Post-conditions 1** |  | Pass |  |
| **5** | Enter current password, new password and confirm password (different from new password) |  | Pass |  |
| **6** | Press “Submit” button next to a student | Successful message “Password and Confirm Password Field do not match !!”appear | Pass |  |
| **7** | Enter current password, new password and confirm password (same as new password) |  | Pass |  |
| **3** | Press “Submit” button next to a student | Successful message “Change password Successfully !!” appear | Pass |  |
| **8** | **Check Post-conditions 1** |  | Pass |  |

|  |
| --- |
| **Post-conditions:**  1. Changes are saved to the database |