Software Maintenance Report

for

STUDENT LEARNING MANAGEMENT SYSTEM-SLMS

**Version 2.0**

**Group No.: 1**

|  |  |  |
| --- | --- | --- |
| **ISLAM MUKTARUL** | **1171300761** | **Team leader** |
| **ISLAM MURADUL** | **1161303822** | **Team member** |
| **KHALEDUZZAMAN** | **1171300038** | **Team member** |
| **POH LEONG CHIEN CALVIN** | **1161103393** | **Team member** |

**Date: 21/9/2020**

# Table of Contents

**Contents**

[**Table of Contents**](#_qhrh9fis6csk) **2**

[**Revisions**](#_g6f4yoe9ibuo) **4**

[**System Overview**](#_85kx3wwka4hc) **5**

[Description](#_5lkhsdc4nsv0) 5

[Assignment of Modules and Tasks](#_kpqi3wqdx8u) 5

[Assumptions and Dependencies](#_mmc1rp4bzfg0) 5

[Use Case Diagram](#_9wubzxix86u2) 5

[**Software Documentation**](#_2s8eyo1) **6**

Class diagram 6

[Sequence](#_hwpa2qt99783) diagram 6

[Architecture](#_eo1tk2cny6yq) diagram 6

Data dictionary 6

[Sample Data](#_pfjmv6smmyyp) 6

[**Modules Documentation**](#_rt5xii5yxwzf) **7**

[Implementation Screens](#_d4mr97rlpdrt) 7

[Test Cases](#_xhuq3z2xodpl) 7

[Error/Defect Log](#_cujq7ystqxfx) 7

[**Change Requests**](#_pjcnvxetsi1i) **8**

[List of Changes](#_2jxsxqh) 8

[Evaluation of Change Requests](#_3c8wlofui24r) 8

[Implementation of Change Requests](#_oiz60f3pnwys) 8

[**Updated Documentation**](#_1y810tw) **9**

Class diagram 9

Data dictionary 9

[**Regression Testing**](#_hu042338nw0g) **10**

[Test Cases](#_d6u9cnk4j1mh) 10

[**Configuration Management**](#_woh4iqva6h5i) **11**

[Update of Repository](#_tg5bdo16oge9) 11

[Version Control](#_eamwvo877f9k) 11

[**Final Release Version**](#_i43z813bslne) **12**

[Release Notes](#_u9mx1c6lg2wh) 12

[**<Other relevant documentation (create your own heading)>**](#_66saqftosw1d) **13**

# Revisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| SLMS version 2.0 | Islam Muktarul  Islam Muradul  Khaluduzzaman | This version provides a smooth interaction with the user compared to the primary one, which needed some bug fixes | 20/09/2020 |

# System Overview

## Description

A Student Learning Management System (SLMS) is a software application that can be used to plan, implement and assess a specific learning process. It can be used for eLearning practices and, in its most common form, consists of two elements: a server that performs the base functionality and a user interface that is operated by instructors, students and administrators.

With the help of SLMS:

* Delivering learning materials through an SLMS keeps the content centralized so all the students have a single source of content, instructions and questions. No excuses of old books or past years material. Everybody is up-to-date.
* Easy tracking and reporting of students progress.

Functions of the system:

* The system provides unique sets of ID for three kinds of users (admin, student, and lecturer).
* The system provides different interfaces for different users to login.
* The system stores and retrieves data to and from the database.
* The system allows the admin to alter student, lecture, and subject details.
* The system allows lecturers to distribute learning materials among a group of students, and update student’s progress.
* The system allows students to enrol subjects, and view learning materials, attendance, test results, and coursework marks of a particular subject.

## Assignment of Modules and Tasks

Modules of the system

* The program is separated into five classes namely Admin, Student, Lecturer, Subject and ClassTiming.
* A database naming slms is created.
* Seven stand-alone methods are written in the main.cpp file to initiate the program.

|  |  |
| --- | --- |
| **Modules** | **Assigned to member** |
| Admin class, database creation, and all the stand-alone methods | Islam Muktarul, ID-1171300761 |
| Lecturer and Student classes, and database creation | Islam Muktarul, ID-1161303822 |
| Subject class | Khaleduzzaman, ID-1171300038 |
| ClassTiming class | POH LEONG CHIEN CALVIN, ID-1161103393 |

## Assumptions and Dependencies

* The class diagram of the program might be changed in order to retrieve data smoothly from the database.
* A few tables in the database (slms) may need to be discarded which will not be required for the system.

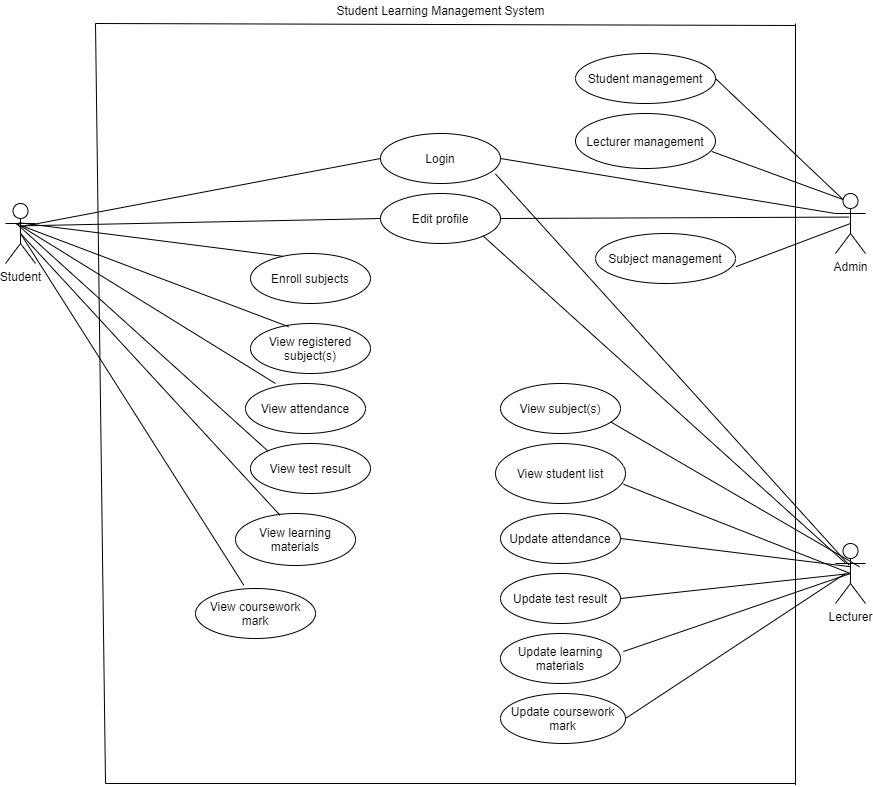
## Use Case Diagram

User Story

Admin perspective: As an admin, I should have the control over students, lecturers, and subjects data, so that the institution runs in an organized manner.

Student perspective: As a student, I would like the system to provide me the learning materials including web links, so that I can acquire knowledge from online.

Lecturer perspective: As a lecturer, I want smooth navigation from menu to menu, so that it saves time. In addition, I want a student to see only his/her own test result and coursework marks, rather than the whole marking lists of all students in class, this will be beneficial in psychological terms to the students who scores less marks.



Use case description table:

|  |  |
| --- | --- |
| Use Case ID/Name | SLMS\_UC1 /Login |
| Precondition | Student must have valid ID |
| Postcondition | System displays relevant homepage |
| Main Flow | * Student enters studentID * Student keys inpassword * System validates studentID * System displays relevanthomepage * Use-caseend |
| Alternate Flow |  |
| Exception Flow | System failed to validate. |

|  |  |
| --- | --- |
| Use Case ID/Name | SLMS\_UC2 /Edit Profile |
| Precondition | Student must be logged in |
| Postcondition | Information of student is updated |
| Main Flow | * Student navigates to editprofile * Student entersinformation * Student submits tosystem * System displays newly updatedinformation * Use-caseend |
| Alternate Flow |  |
| Exception Flow |  |

|  |  |
| --- | --- |
| Use Case ID/Name | SLMS\_UC3 /Enroll Subject(s) |
| Precondition | * Student must be loggedin * Subject enrollment isavailable |
| Postcondition | Student selects required subjects |
| Main Flow | * Student navigates to subjectenrollment * Student selects subject toregister * Student submitsselection * System displays selectedsubject * Use-caseend |
| Alternate Flow |  |
| Exception Flow | System displays error due to student exceeds credit hour limitation. |

|  |  |
| --- | --- |
| Use Case ID/Name | SLMS\_UC4 /View Registered Subject(s) |
| Precondition | * Student must be loggedin. * Student must have registeredsubject |
| Postcondition | System displays list of registered subjects |
| Main Flow | * Student navigates to view registeredsubject * Student may exit fromsystem * Use-caseend |
| Alternate Flow |  |
| Exception Flow |  |

|  |  |
| --- | --- |
| Use Case ID/Name | SLMS\_UC5 /View Attendance |
| Precondition | * Student must be loggedin * Student must have registeredsubject * Student selects a particularsubject |
| Postcondition | System shows percentage of attendance for a particular subject |
| Main Flow | * Student navigates to list of registeredsubjects * Student select a subject to view itsattendance * System shows percentage of attendance for the particular subject * Student may exitsystem * Use-caseend |
| Alternate Flow |  |
| Exception Flow |  |

|  |  |
| --- | --- |
| Use Case ID/Name | SLMS\_UC6 /View Test Result |
| Precondition | * Student must be loggedin * Student must have registeredsubject * Student selects a particularsubject |
| Postcondition | System displays test result for a particular subject |
| Main Flow | * Student navigates to list of registeredsubjects * Student selects a subject to view its testresult * System display test result for the particularsubject * Student may exitsystem * Use-caseend |
| Alternate Flow |  |
| Exception Flow |  |

|  |  |
| --- | --- |
| Use Case ID/Name | SLMS\_UC7 /View Learning Materials |
| Precondition | * Student must be loggedin * Student must have registeredsubject * Student selects a particularsubject |
| Postcondition | The lecture notes will be displayed |
| Main Flow | * Student selects a week to view the informationupdated by thelecturer * System displays learning materials based on week selected * Student may exitsystem * Use-caseend |
| Alternate Flow |  |
| Exception Flow | .A lecture note may not be downloaded due to error in the file |

|  |  |
| --- | --- |
| Use Case ID/Name | SLMS\_UC8 /View Coursework Mark |
| Precondition | * Student must be loggedin * Student must have subject registered * Student selects a particularsubject |
| Postcondition | System displays coursework mark for a particular subject |
| Main Flow | * Student navigates to list of registeredsubjects * Student selects a subject to view its courseworkmarks * System shows coursework mark for subjectchosen * Student may exitsystem * Use-caseend |
| Alternate Flow |  |
| Exception Flow |  |

|  |  |
| --- | --- |
| Use Case ID/Name | SLMS\_UC9 /Login |
| Precondition | - Lecturer must have validID |
| Postcondition | System displays relevant homepage |
| Main Flow | * Lecturer enters ID * Lecturer keys inpassword. * System validates ID andpassword * System displays relevanthomepage * Use-caseend |
| Alternate Flow |  |
| Exception Flow | System failed to validate. |

|  |  |
| --- | --- |
| Use Case ID/Name | SLMS\_UC10 /Edit Profile |
| Precondition | - Lecturer must be loggedin |
| Postcondition | Information of lecturer is updated |
| Main Flow | * Lecturer navigates to editprofile * Lecturer entersinformation * Lecturer submits tosystem * System displays updatedinformation * Use-caseend |
| Alternate Flow |  |
| Exception Flow |  |

|  |  |
| --- | --- |
| Use Case ID/Name | SLMS\_UC11 /View Subject |
| Precondition | * Lecturer must be loggedin * Lecturer must be assigned tosubject |
| Postcondition | System displays the list of assigned subject |
| Main Flow | * Lecturer navigates to view subjectlist * System displays assignedsubjects * Lecturer may exit fromsystem * Use-caseend |
| Alternate Flow |  |
| Exception Flow |  |

|  |  |
| --- | --- |
| Use Case ID/Name | SLMS\_UC12 /View Student List |
| Precondition | * Lecturer must loggedin * Lecturer select a particularsubject |
| Postcondition | System displays list of students for a particular subject |
| Main Flow | * Lecturer selects a particular subject to view studentlist * System obtains list of students for that particularsubject * Lecturer may exit thesystem * Use-caseend |
| Alternate Flow |  |
| Exception Flow |  |

|  |  |
| --- | --- |
| Use Case ID/Name | SLMS\_UC13 /Update Attendance |
| Precondition | * Lecturer must be loggedin * Lecturer must select a particularsubject |
| Postcondition | System displays attendance of students for that particular subject |
| Main Flow | * Lecturer selects astudent * Lecturer updates the attendance list of thestudent * System displays the updated list ofattendance * Lecturer may exitsystem * Use-endend |
| Alternate Flow |  |
| Exception Flow |  |

|  |  |
| --- | --- |
| Use Case ID/Name | SLMS\_UC14 /Update Test Result |
| Precondition | * Lecturer must be loggedin * Lecturer selects a particularsubject |
| Postcondition | System displays updated test result |
| Main Flow | * Lecturer selects astudent * Lecturer submits updated test result for thatstudent * System displays updated testresult * Lecturer may exitsystem * Use-caseend |
| Alternate Flow |  |
| Exception Flow |  |

|  |  |
| --- | --- |
| Use Case ID/Name | SLMS\_UC15 /Update Learning Materials |
| Precondition | * Lecturer must be loggedin * Lecturer select a particularsubject |
| Postcondition | System displays updated learning materials |
| Main Flow | * Lecturer updates the learning materials according to currentweek * System displays updated learning materials for current week * Lecturer may exitsystem * Use-caseend |
| Alternate Flow |  |
| Exception Flow |  |

|  |  |
| --- | --- |
| Use Case ID/Name | SLMS\_UC16 /Update Coursework Mark |
| Precondition | * Lecturer must be loggedin * Lecturer selects a particularsubject |
| Postcondition | System displays updated coursework mark |
| Main Flow | * Lecturer inserts coursework mark for eachstudent * Lecturer submits updated coursework marks to the system * System displays updated coursework marks for each student under a particularsubject * Lecturer may exitsystem * Use-caseend |
| Alternate Flow |  |
| Exception Flow | System display error after inserting invalid number (negative) for marks |

|  |  |
| --- | --- |
| Use Case ID/Name | SLMS\_UC17 /Login |
| Precondition | - Admin must have validID |
| Postcondition | System displays relevant homepage |
| Main Flow | * Admin enters adminID * Admin keys inpassword. * System validates adminID * System displays relevanthomepage * Admin may exit from system * Use-caseends |
| Alternate Flow |  |
| Exception Flow | System failed to validate. |

|  |  |
| --- | --- |
| Use Case ID/Name | SLMS\_UC18 /Edit Profile |
| Precondition | - Admin must be loggedin |
| Postcondition | Information of admin is updated |
| Main Flow | * Admin navigates to edit profile * Admin entersinformation * Admin submits tosystem * System displays updatedinformation * Admin may exit fromsystem * Use-caseends |
| Alternate Flow |  |
| Exception Flow |  |

|  |  |
| --- | --- |
| Use Case ID/Name | SLMS\_UC19 / Manages Student |
| Precondition | - Admin must be loggedin. |
| Postcondition | System displays updated details of student |
| Main Flow | * Admin opts to student details updatepage * Admin controls details ofstudent * System store or remove student details in and fromdatabase * Admin may exit fromsystem * Use caseends |
| Alternate Flow |  |
| Exception Flow |  |

|  |  |
| --- | --- |
| Use Case ID/Name | SLMS\_UC20 / Manage Lecturer |
| Precondition | - Admin must be loggedin. |
| Postcondition | System displays updated details oflecturer |
| Main Flow | * Admin opts to lecturer details updatepage * Admin controls details oflecturer * System store or remove lecturer details in and from database * Admin may exit fromsystem * Use caseends |
| Alternate Flow |  |
| Exception Flow |  |

|  |  |
| --- | --- |
| Use Case ID/Name | SLMS\_UC21 / Manage Subject |
| Precondition | - Admin must be loggedin. |
| Postcondition | System displays subjects for the trimester that are made available. |
| Main Flow | * Admin opts to edit subject list for thetrimester * Admin updates the subjects to be available for thestudents toenroll. * Admin assigns lecturers tosubjects. * System store subject list indatabase * Admin may exit fromsystem * Use-caseends |
| Alternate Flow |  |
| Exception Flow |  |

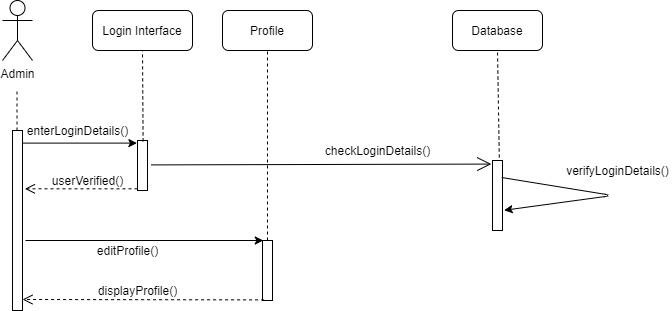
# Software Documentation

## Class Diagram:

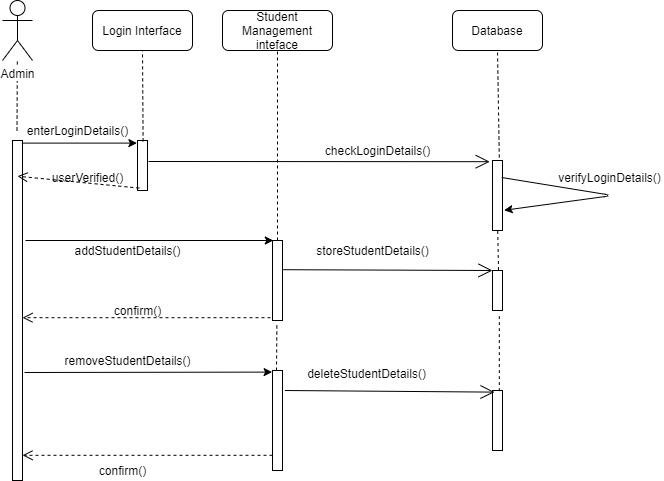
****

## Sequence Diagram

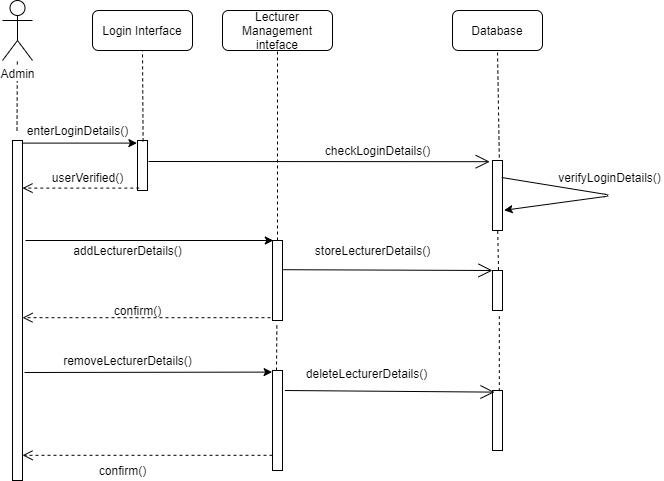
Admin interaction with the system(I):

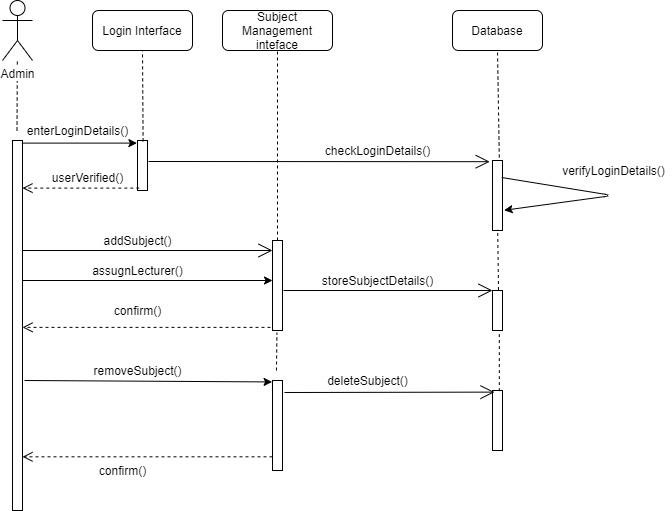


Admin interaction with the system(II):

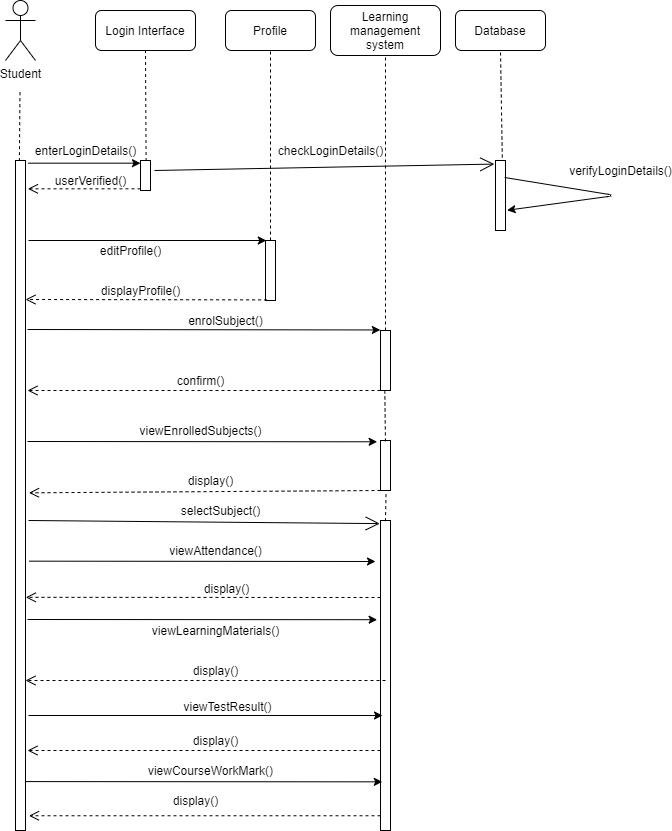


Admin interaction with the system(III):

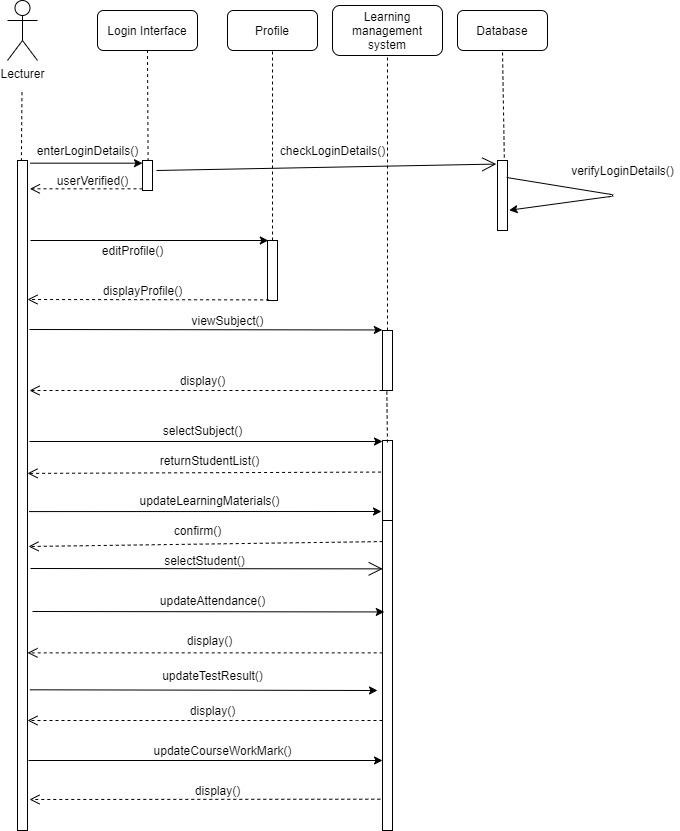


Admin interaction with the system(IV):

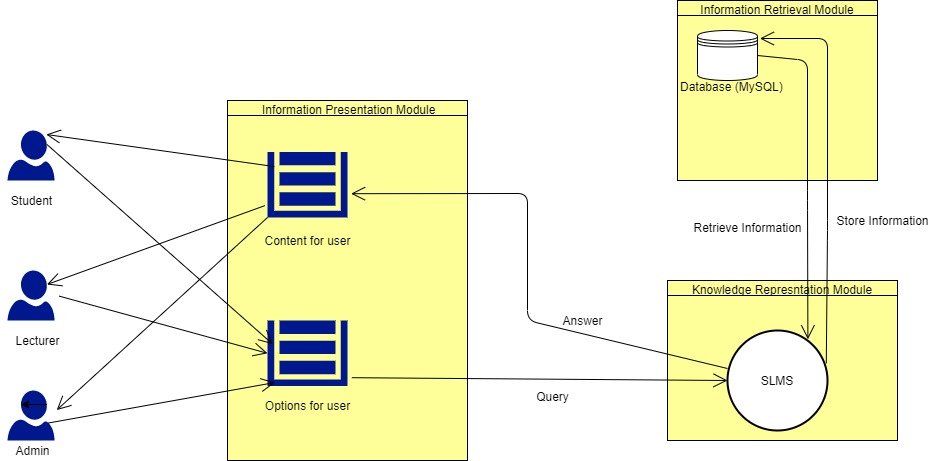
Student interaction with the system:



Lecturer interaction with the system:



2.3 Architecture Diagram

****

Description:

The users such as Admin, Student, and Lecturer will interact with Student Learning Management System (SLMS) via their corresponding interfaces. A user may select an option from the interface that will call a given function in the SLMS. If any data is needed to be stored or retrieved, the SLMS will interact with the database (MySQL). Finally, the system will return information of the selected option via the interface, where the users can interact to view the information

* 1. Data Dictionary

Table: Admin

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Field Type | Field Size | Key | Format | FK Reference |
| Admin\_Id | Int | 10 | Primary Key | 701 |  |
| First\_Name | Varchar | 40 |  | Zakir |  |
| Last\_Name | Varchar | 40 |  | Hossain |  |
| Email | Varchar | 40 |  | zakir@gmail.com |  |
| Password | Varchar | 20 |  | abcd |  |
| Contact\_No | Int | 15 |  | 1234565678 |  |
| Address | Varchar | 100 |  | Kl,cyberjaya |  |

Table: Lecturer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Field Type | Field Size | Key | Format | FK Reference |
| Lecturer\_ID | Int | 10 | Primary Key | 12001 |  |
| First\_Name | Varchar | 40 |  | Sharaf |  |
| Last\_Name | Varchar | 40 |  | horaini |  |
| Email | Varchar | 40 |  | sharaf@yahoo.com |  |
| Password | Varchar | 20 |  | 1234 |  |
| Contact\_No | Int | 15 |  | 12345456791 |  |
| Address | Varchar | 100 |  | Mimi residential area |  |
| Admin\_Id | Int | 10 | Foreign Key | 701 | ADMIN |

Table: Teach

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Field Type | Field Size | Key | Format | FK Reference |
| Student\_ID | Int | 10 | Primary/ Foreign Key | 12345 | STUDENT |
| Lecturer\_ID | Int | 10 | Primary/ Foreign Key | 12345 | LECTURER |

Table: Student

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Field Type | Field Size | Key | Format | FK Reference |
| Student\_ID | Int | 10 | Primary Key | 18001 |  |
| First\_Name | Varchar | 40 |  | Rana |  |
| Last\_Name | Varchar | 40 |  | Khan |  |
| Email | Varchar | 40 |  | rana@gmail.com |  |
| Password | Varchar | 20 |  | qwerty |  |
| Contact\_No | Int | 15 |  | 0181456789 |  |
| Address | Varchar | 100 |  | langkawi |  |
| Admin\_Id | Int | 10 | Foreign Key | 701 | ADMIN |

Table: Enroll

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Field Type | Field Size | Key | Format | FK Reference |
| Student\_ID | Int | 10 | Primary/ Foreign Key | 18001 | STUDENT |
| Subject\_Code | Int | 10 | Primary/ Foreign Key | 12001 | SUBJECT |

Table: Subject

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Field Type | Field Size | Key | Format | FK Reference |
| Subject\_Code | Int | 10 | Primary Key | 8001 |  |
| Subject\_Name | Varchar | 30 |  | Programming fundamental |  |
| Credit\_Hour | Int | 5 |  | 4 |  |
| Admin\_ID | Int | 10 | Foreign Key | 701 | ADMIN |
| Lecturer\_ID | Int | 10 | Foreign Key | 12001 | LECTURER |

Table: Schedule

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Field Type | Field Size | Key | Format | FK Reference |
| Subject\_Code | Int | 10 | Primary/ Foreign Key | 18001 | SUBJECT |
| Timing\_Code | Int | 10 | Primary/ Foreign Key | 30001 | ClASS TIMING |

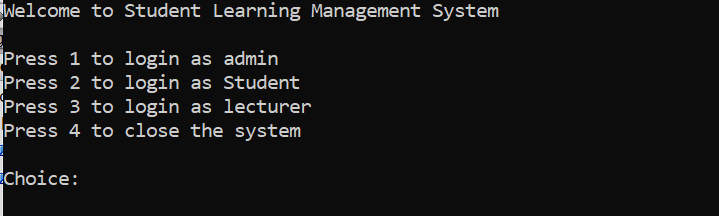
Table: Class Timing

Table: ClassTiming

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Field Type | Field Size | Key | Format | FK Reference |
| Timing\_Code | Int | 10 | Primary Key | 30001 |  |
| Time | Varchar | 20 |  | 12:00 |  |
| Day | Varchar | 20 |  | Monday |  |
| Class\_Room\_Number | Varchar | 20 |  | Tcp2101 |  |

# Modules Documentation

## Implementation Screens



This the system home page where each user can login to their own interface.



Admin login page, where choice given to the admin to login or go back to home page of the system.



This is the homepage for the admin. The admin can view his profile, can edit his profile, and also can manage student, lecturer and subject data.



This is the manage student display, where Admin can add and remove student from database, also can edit and view student details.



This is the manage lecturer display, where Admin can add and remove lecturer from database, also can edit and view student details.



This is the manage Subject screen, where Admin can add and remove subject from database and also can assign subject to lecturer. Admin also can edit subject details.



This screen is the lecturer login. Lecturer is given choice to enter into his homepage or go back to system home display.



This is lecturer homepage. From this interface lecturer can view and edit his profile. Lecturer can view how many subjects assign to him/her by admin and also can check student list for each subject.

Lecturer can update learning materials, attendance, test result, coursework for each subject assign to him/her, where each module will the ask the lecturer to input subject code for which he/she want to update.



This screen is the Student login. Lecturer is given choice to enter into his homepage or go back to system home display.



This is the Student homepage. Student can view and edit his profile. Student can enroll subject where he/she can add or remove subject. Student can view registered subjects and can also view learning materials, attendance, test result, coursework, class schedule by giving the subject code for the subject that he/she wants to view these modules.

## Test Cases

Test Case ID: TC1, Test Case Name: Login

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Admin enters ID | Valid (701) | Admin logged in | Admin  logged  in | Pass | Admin has  access to the  system |
| **2** | Adminenters ID | Invalid (700) | Prompt invalid user ID | None | Fail | Admin can re-enter ID |
| **3** | Admin enters password | Valid(123@abc) | Admin logged in | Admin  logged  in | Pass | Admin has  access to the  system |
| **4** | Admin enters  password | Invalid(1 2abc e) | Prompt invalid user password | None | Fail | Admin can re-enter password |
| **5** | Student enters ID | Valid(18001) | Student  logged  in | Student  logged  in | Pass | Student has  access to the  system |
| **6** | Student enters ID | Invalid(18000) | Prompt  invalid user  ID | None | Fail | Admin can re-enter ID |
| **7** | Student enters password | Valid(123@abc) | Student logged in | Student  logged  in | Pass | Student has  access to the  system |
| **8** | Student enters password | Invalid(1 2abc e) | Prompt invalid user password | None | Fail | Student can re-enter password |
| **9** | Lecturer enters ID | Valid(12001) | Lecturer  logged  in | Lecturer  logged  in | Pass | Lecturer has  access to the  system |
| **10** | Lecturer enters ID | Invalid(12000) | Prompt  invalid user  ID | None | Fail | Lecturer can re-enter ID |
| **11** | Lecturer enters password | Valid(123@abc) | Lecturer logged in | Lecturer  logged  in | Pass | Lecturer has  access to the  system |
| **12** | Lecturer enters password | Invalid(1 2abc e) | Prompt invalid user password | None | Fail | Lecturer can re-enter password |

Test Case ID: TC2, Test Case Name: Manage Student

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Admin enters choice from menu | Valid (3) | Menu to add/remove/etc student is shown | Menu to add/remove/etc student is shown | Pass | Admin has control on students data |
| **2** | Admin enters choice from menu | Invalid (a) | Prompt invalid choice | None | Fail | Admin can re-enter choice |

Test Case ID: TC3, Test Case Name: Manage Lecturer

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Admin enters choice from menu | Valid (4) | Menu to add/remove /etc lecturer is shown | Menu to add/remove/etc lecturer is shown | Pass | Admin has control on lecturers data |
| **2** | Admin enters choice from menu | Invalid (9) | Prompt invalid choice | None | Fail | Admin can re-enter choice |

Test Case ID: TC4, Test Case Name: Manage Subject

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Admin enters choice from menu | Valid (5) | Menu to add/remove /etc subject is shown | Menu to add/remove/etc subject is shown | Pass | Admin has control on subjects data |
| **2** | Admin enters choice from menu | Invalid (9) | Prompt invalid choice | None | Fail | Admin can re-enter choice |

Test Case ID: TC5, Test Case Name: Edit Profile

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Admin/Student/ Lecturer enters choice from menu | Valid (2) | Fields to edit are shown | Fields to edit are shown | Pass | Admin/Student/ Lecturer can change personal data |
| **2** | Admin/Student/ Lecturer enters choice from menu | Invalid (@) | Prompt invalid choice | None | Fail | Admin can re-enter choice |

Test Case ID: TC6, Test Case Name: Enroll Subjects

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Student enters choice from menu | Valid (3) | Menu to add/remove subject is shown | Menu to add/remove subject is shown | Pass | Student can add or drop subjects |
| **2** | Student enters choice from menu | Invalid (csa) | Prompt invalid choice | None | Fail | Student can re-enter choice |

Test Case ID: TC7, Test Case Name: View registered subjects

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Student enters choice from menu | Valid (4) | Registered subjects are shown | Registered subjects are shown | Pass | Student can know the details of the registered subjects |
| **2** | Student enters choice from menu | Invalid (44) | Prompt invalid choice | None | Fail | Student can re-enter choice |

Test Case ID: TC8, Test Case Name: View attendance

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Student enters subject code from his/her subject list | Valid (8001) | Percentage of attendance in that particular subject is shown | Percentage of attendance in that particular subject is shown | Pass | Student can only able to his own attendance |
| **2** | Student enters subject code from his/her subject list | Invalid (8000) | Prompt “Not found…” | None | Fail | System returns to previous menu |

Test Case ID: TC9, Test Case Name: View test result

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Student enters subject code from his/her subject list | Valid (8015) | Test result in that particular subject is shown | Test result in that particular subject is shown | Pass | Student can only able to his own test result |
| **2** | Student enters subject code from his/her subject list | Invalid (12001) | Prompt “Not found…” | None | Fail | System returns to previous menu |

Test Case ID: TC10, Test Case Name: View coursework mark

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Student enters subject code from his/her subject list | Valid (8020) | Course marks in that particular subject is shown | Course marks in that particular subject is shown | Pass | Student can only able to his own course marks |
| **2** | Student enters subject code from his/her subject list | Invalid (7000) | Prompt “Not found…” | None | Fail | System returns to previous menu |

Test Case ID: TC11, Test Case Name: View learning materials

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Student enters subject code from his/her subject list | Valid (8035) | Learning materials of that particular subject is shown | Learning materials of that particular subject is shown | Pass | Student can see learning materials of several weeks in the same page |
| **2** | Student enters subject code from his/her subject list | Invalid (1000) | Prompt “Not found…” | None | Fail | System returns to previous menu |

Test Case ID: TC12, Test Case Name: View assigned subjects

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Lecturer enters choice from menu | Valid (3) | Assigned subjects are shown to lecturer | Assigned subjects are shown to lecturer | Pass | Lecturer can know the details of the assigned subjects |
| **2** | Lecturer enters choice from menu | Invalid (a) | Prompt invalid choice | None | Fail | Lecturer can re-enter choice |

Test Case ID: TC13, Test Case Name: View student list

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Lecturer enters subject code from his/her subject list | Valid (8035) | List of students enrolled in that particular subject is shown | List of students enrolled in that particular subject is shown | Pass | Lecturer can know the details of his/her class students |
| **2** | Lecturer enters subject code from his/her subject list | Invalid (1000) | Prompt “Not found…” | None | Fail | System returns to previous menu |

Test Case ID: TC14, Test Case Name: Update attendance

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Lecturer enters subject code from his/her subject list | Valid (8005) | Student ID’s are prompted one by one to update attendance percentage | Student ID’s are prompted one by one to update attendance percentage | Pass | Lecturers have to calculate attendance percentage manually |
| **2** | Lecturer enters subject code from his/her subject list | Invalid (705) | Prompt “Not found…” | None | Fail | System returns to previous menu |

Test Case ID: TC15, Test Case Name: Update test result

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Lecturer enters subject code from his/her subject list | Valid (8006) | Student ID’s are prompted one by one to update test score | Student ID’s are prompted one by one to update test score | Pass | Lecturers have to calculate test result percentage manually |
| **2** | Lecturer enters subject code from his/her subject list | Invalid (7050) | Prompt “Not found…” | None | Fail | System returns to previous menu |

Test Case ID: TC16, Test Case Name: Update coursework mark

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Lecturer enters subject code from his/her subject list | Valid (8025) | Student ID’s are prompted one by one to update coursework mark | Student ID’s are prompted one by one to update coursework mark | Pass | Lecturers have to calculate coursework percentage manually |
| **2** | Lecturer enters subject code from his/her subject list | Invalid (709) | Prompt “Not found…” | None | Fail | System returns to previous menu |

Test Case ID: TC17, Test Case Name: Update learning materials

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Lecturer enters subject code from his/her subject list | Valid (8003) | Lecturer is asked to input week number and upload corresponding materials | Lecturer is asked to input week number and upload corresponding materials | Pass | Lecturers can upload multiple materials for a particular week |
| **2** | Lecturer enters subject code from his/her subject list | Invalid (8000) | Prompt “Not found…” | None | Fail | System returns to previous menu |

## Error/Defect Log

Table 3.1: Error/Defect Log

|  |  |  |
| --- | --- | --- |
| No. | Error/Defect | Description/Remark |
| 1 | Input of subject name | When admin tries to enter a subject name containing white spaces, the input curser jumps into another input variable. |
| 2 | View subject details | Admin cannot see the subject details |
| 3 | View student list | Subject details are not displayed to lecturer so that the lecturer can input subject code to view the student list registered to a particular subject. |
| 4 | Removal of subject | When admin tries to remove a subject from database, if the admin puts a wrong input of subject code, it shows removal successful |
| 5 | Input of contact number | If user enters input of string type to add the contact number, the program stops working. |
| 6 | View registered subjects | In a run, if a student add subjects to the subject list, the student cannot find the subjects as registered in the next run of the program |
| 7 | View assigned subjects | In a run, if a subject is assigned to a lecturer, the lecturer cannot find the subjects as assigned in the next run of the program |
| 8 | View class schedule | In a run, if a student adds a subject to the subject list, the student cannot view the class schedule in the next run of the program |
| 9 | Menu choice | When a user is given a choice to choose from menu options, if the user inputs string type instead of integer, the program terminates. |
| 10 | Input of address | When user tries to enter address containing white spaces, the input curser jumps into another input variable. |

# Change Requests

## List of Changes

**Change Request**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project: Student learning management system** |  | **Date: 21/9/2020** |  |  |
| **Change Requestor:** |  | **Group No: 1** |  |  |
| **Change Category:** |  |  |  |  |
| □ Corrective | □Adaptive | | □ Perfective | □Preventive |
|  | | |  |  |
| **Does this Change Affect (tick all that apply):** | | |  |  |
| □Source Code | □ Design | | □ Requirements | □Documentation |

□ Others (Please specify: )

**Describe the Change Being Requested: Class separation from one single file to multiple .cpp and .h files.**

**Describe the Reason for the Change: For a better object oriented programming and to increase the efficiency of the program**

**Describe all Alternatives Considered: N/A**

**Describe any Technical Changes Required to Implement this Change: Two variables are replaced in Subject class.**

**Describe Risks to be Considered for this Change: Overall design might be compromised**

**Estimate Resources and Costs Needed to Implement this Change: None**

**Describe the Implications to Quality: None**

3

**Change Request**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project: Student learning management system** |  | **Date: 21/9/2020** |  |  |
| **Change Requestor:** |  | **Group No: 1** |  |  |
| **Change Category:** |  |  |  |  |
| □Corrective | □Adaptive | | □ Perfective | □Preventive |
|  | | |  |  |
| **Does this Change Affect (tick all that apply):** | | |  |  |
| □Source Code | □ Design | | □ Requirements | □ Documentation |

□ Others (Please specify: )

**Describe the Change Being Requested: Inputs are validated**

**Describe the Reason for the Change: The users cannot input wrong values and the program don’t terminate**

**Describe all Alternatives Considered: N/A**

**Describe any Technical Changes Required to Implement this Change: Some try and catch method are used and also a function is made to validate the input.**

**Describe Risks to be Considered for this Change: System might not work properly**

**Estimate Resources and Costs Needed to Implement this Change: None**

**Describe the Implications to Quality: None**

**Change Request**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project: Student learning management system** |  | **Date: 21/9/2020** |  |  |
| **Change Requestor:** |  | **Group No: 1** |  |  |
| **Change Category:** |  |  |  |  |
| □Corrective | □Adaptive | | □ Perfective | □Preventive |
|  | | |  |  |
| **Does this Change Affect (tick all that apply):** | | |  |  |
| □Source Code | □ Design | | □ Requirements | □Documentation |

□ Others (Please specify: )

**Describe the Change Being Requested: The database connection is improved**

**Describe the Reason for the Change: Data manipulated inside the system were not store and retrieve properly.**

**Describe all Alternatives Considered: N/A**

**Describe any Technical Changes Required to Implement this Change: a method is written in the main.cpp file to copy the data from the database into the system, Sql commands are used to store data from the system into the database.**

**Describe Risks to be Considered for this Change: Redundancy in database**

**Estimate Resources and Costs Needed to Implement this Change: None**

**Describe the Implications to Quality: None**

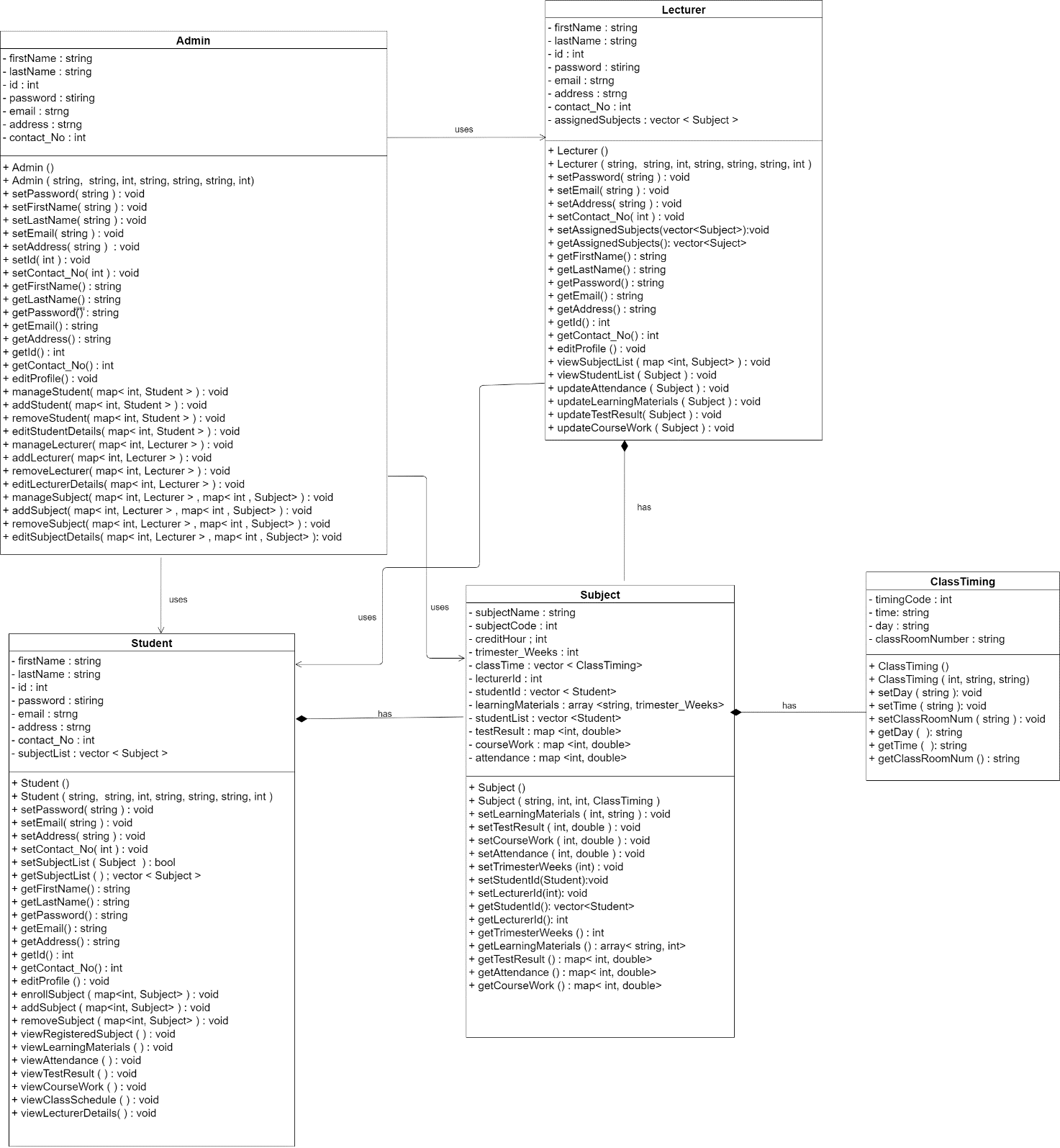
## Implementation of Change Requests

Modules/files that undergone implementation changes:

* Admin.h
* Admin.cpp
* Student.h
* Student.cpp
* Lecturer.h
* Lecturer.cpp
* Subject.h
* Subject.cpp
* main.cpp

# Updated Documentation

## Class diagram



## Data Dictionary:

## Table: Admin

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Field Type | Field Size | Key | Format | FK Reference |
| Admin\_Id | Int | 10 | Primary Key | 701 |  |
| First\_Name | Varchar | 40 |  | Zakir |  |
| Last\_Name | Varchar | 40 |  | Hossain |  |
| Email | Varchar | 40 |  | zakir@gmail.com |  |
| Password | Varchar | 20 |  | abcd |  |
| Contact\_No | Int | 15 |  | 1234565678 |  |
| Address | Varchar | 100 |  | Kl,cyberjaya |  |

Table: Lecturer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Field Type | Field Size | Key | Format | FK Reference |
| Lecturer\_ID | Int | 10 | Primary Key | 12001 |  |
| First\_Name | Varchar | 40 |  | Sharaf |  |
| Last\_Name | Varchar | 40 |  | horaini |  |
| Email | Varchar | 40 |  | sharaf@yahoo.com |  |
| Password | Varchar | 20 |  | 1234 |  |
| Contact\_No | Int | 15 |  | 12345456791 |  |
| Address | Varchar | 100 |  | Mimi residential area |  |
| Admin\_Id | Int | 10 | Foreign Key | 701 | ADMIN |

Table: Student

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Field Type | Field Size | Key | Format | FK Reference |
| Student\_ID | Int | 10 | Primary Key | 18001 |  |
| First\_Name | Varchar | 40 |  | Rana |  |
| Last\_Name | Varchar | 40 |  | Khan |  |
| Email | Varchar | 40 |  | rana@gmail.com |  |
| Password | Varchar | 20 |  | qwerty |  |
| Contact\_No | Int | 15 |  | 0181456789 |  |
| Address | Varchar | 100 |  | langkawi |  |
| Admin\_Id | Int | 10 | Foreign Key | 701 | ADMIN |

Table: Enroll

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Field Type | Field Size | Key | Format | FK Reference |
| Student\_ID | Int | 10 | Primary/ Foreign Key | 18001 | STUDENT |
| Subject\_Code | Int | 10 | Primary/ Foreign Key | 12001 | SUBJECT |

Table: Subject

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Field Type | Field Size | Key | Format | FK Reference |
| Subject\_Code | Int | 10 | Primary Key | 8001 |  |
| Subject\_Name | Varchar | 30 |  | Programming fundamental |  |
| Credit\_Hour | Int | 5 |  | 4 |  |
| Admin\_ID | Int | 10 | Foreign Key | 701 | ADMIN |
| Lecturer\_ID | Int | 10 | Foreign Key | 12001 | LECTURER |

Table: Class Timing

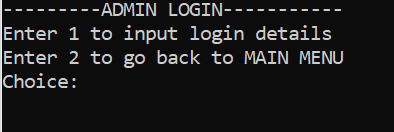
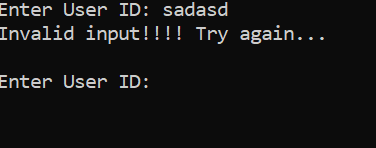
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Field Type | Field Size | Key | Format | FK Reference |
| Timing\_Code | Int | 10 | Primary Key | 30001 |  |
| Time | Varchar | 20 |  | 12:00 |  |
| Day | Varchar | 20 |  | Monday |  |
| Class\_Room\_Number | Varchar | 20 |  | Tcp2101 |  |

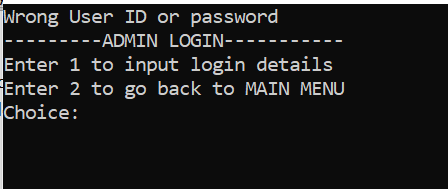
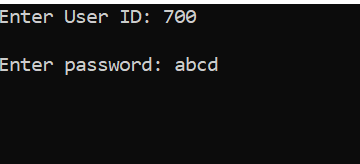
# Regression Testing

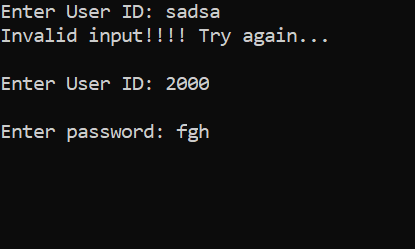
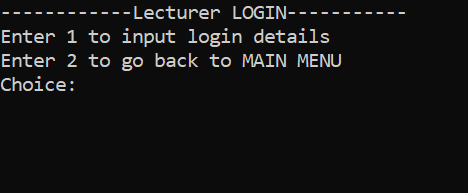
## Test Cases

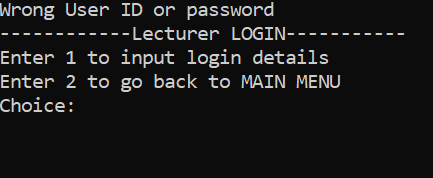
Test Case ID: TC1, Test Case Name: Login

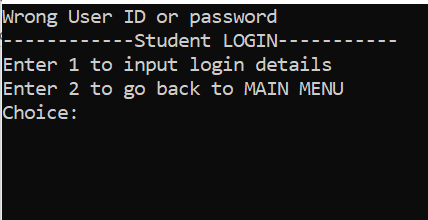
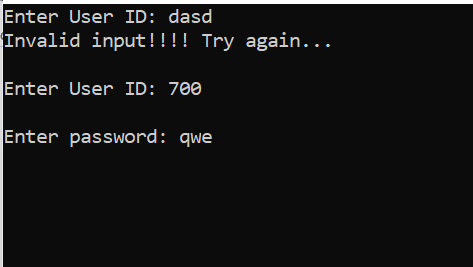
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Admin enters ID | Valid (701) | Admin logged in | Admin  logged  in | Pass | Admin has  access to the  system |
| **2** | Adminenters ID | Invalid (700) | Prompt invalid user ID | None | Fail | Admin can re-enter ID |
| **3** | Admin enters password | Valid(123@abc) | Admin logged in | Admin  logged  in | Pass | Admin has  access to the  system |
| **4** | Admin enters  password | Invalid(1 2abc e) | Prompt invalid user password | None | Fail | Admin can re-enter password |
| **5** | Student enters ID | Valid(18001) | Student  logged  in | Student  logged  in | Pass | Student has  access to the  system |
| **6** | Student enters ID | Invalid(18000) | Prompt  invalid user  ID | None | Fail | Admin can re-enter ID |
| **7** | Student enters password | Valid(123@abc) | Student logged in | Student  logged  in | Pass | Student has  access to the  system |
| **8** | Student enters password | Invalid(1 2abc e) | Prompt invalid user password | None | Fail | Student can re-enter password |
| **9** | Lecturer enters ID | Valid(12001) | Lecturer  logged  in | Lecturer  logged  in | Pass | Lecturer has  access to the  system |
| **10** | Lecturer enters ID | Invalid(12000) | Prompt  invalid user  ID | None | Fail | Lecturer can re-enter ID |
| **11** | Lecturer enters password | Valid(123@abc) | Lecturer logged in | Lecturer  logged  in | Pass | Lecturer has  access to the  system |
| **12** | Lecturer enters password | Invalid(1 2abc e) | Prompt invalid user password | None | Fail | Lecturer can re-enter password |





**



Test Case ID: TC2, Test Case Name: Manage Student

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Admin enters choicefrom menu | Valid (3) | Menu to add/remove/etc student is shown | Menu to add/remove/etc student is shown | Pass | Admin has control on students data |
| **2** | Admin enters choice from menu | Invalid (a) | Prompt invalid choice | None | Fail | Admin can re-enter choice |

Test Case ID: TC3, Test Case Name: Manage Lecturer

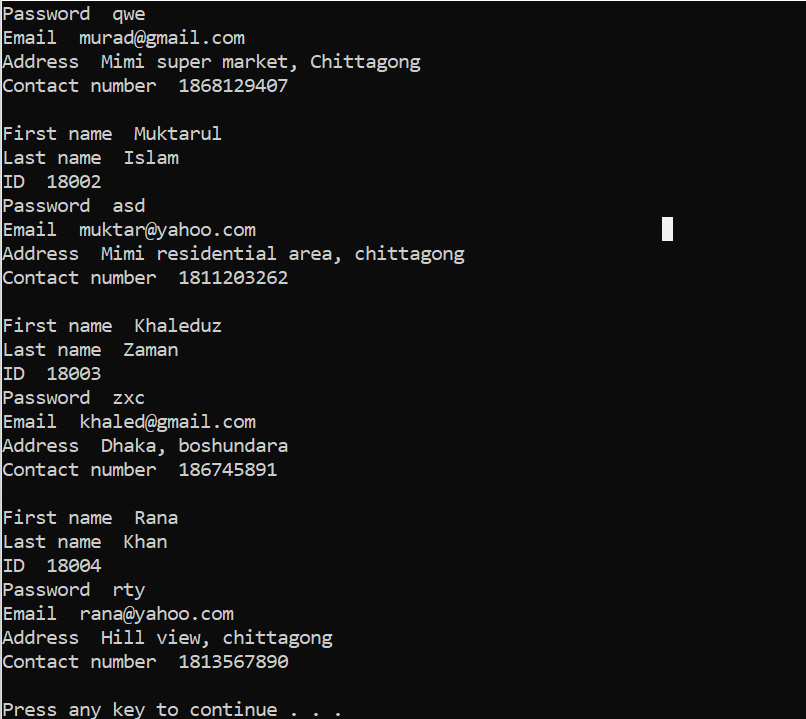
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Admin enters choice from menu | Valid (4) | Menu to add/remove /etc lecturer is shown | Menu to add/remove/etc lecturer is shown | Pass | Admin has control on lecturers data |
| **2** | Admin enters choice from menu | Invalid (9) | Prompt invalid choice | None | Fail | Admin can re-enter choice |

Test Case ID: TC4, Test Case Name: Manage Subject

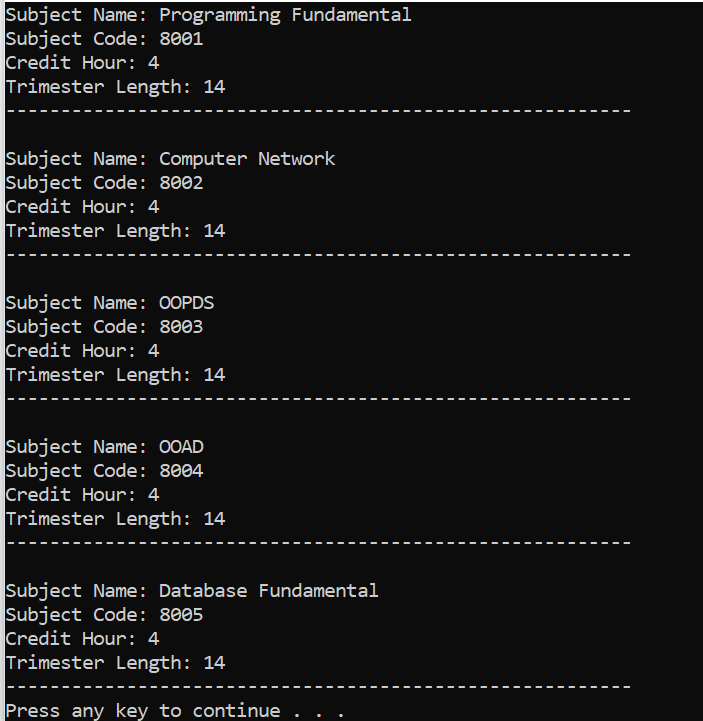
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Admin enters choice from menu | Valid (5) | Menu to add/remove /etc subject is shown | Menu to add/remove/etc subject is shown | Pass | Admin has control on subjects data |
| **2** | Admin enters choice from menu | Invalid (9) | Prompt invalid choice | None | Fail | Admin can re-enter choice |

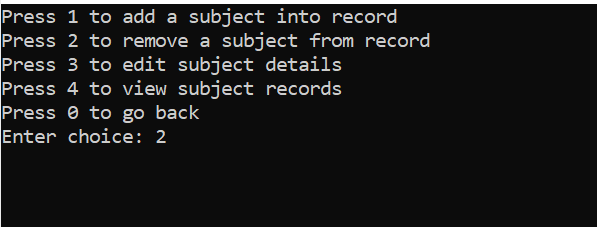


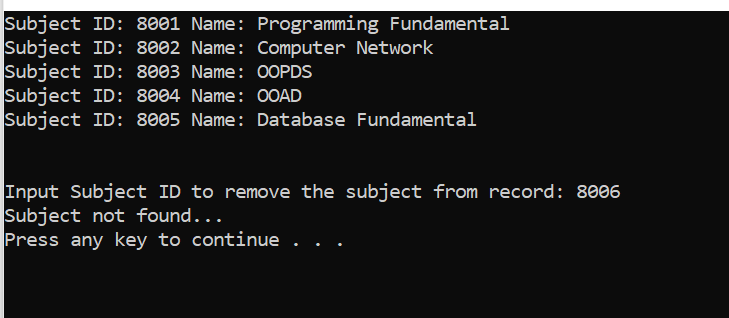






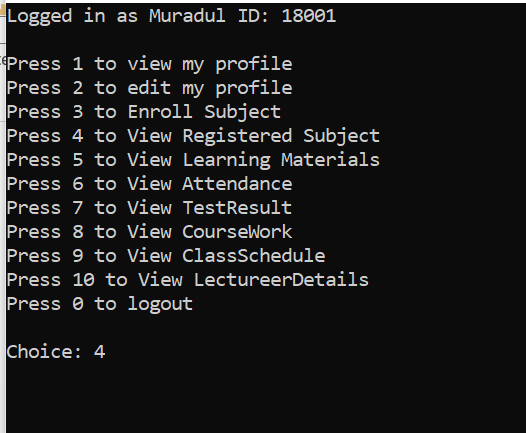


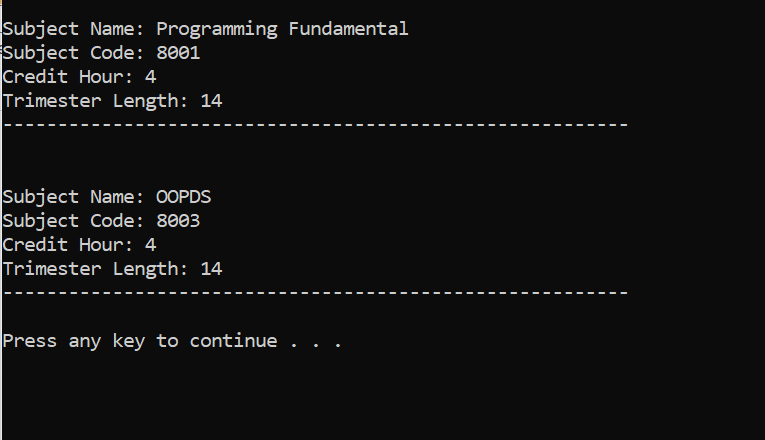




Test Case ID: TC7, Test Case Name: View registered subjects

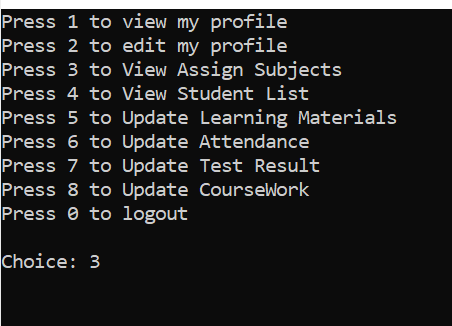
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Student enters choice from menu | Valid (4) | Registered subjects are shown | Registered subjects are shown | Pass | Student can know the details of the registered subjects |
| **2** | Student enters choice from menu | Invalid (44) | Prompt invalid choice | None | Fail | Student can re-enter choice |

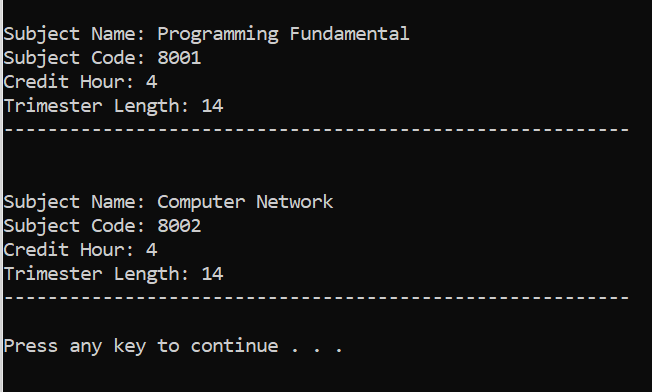




Test Case ID: TC12, Test Case Name: View assigned subjects

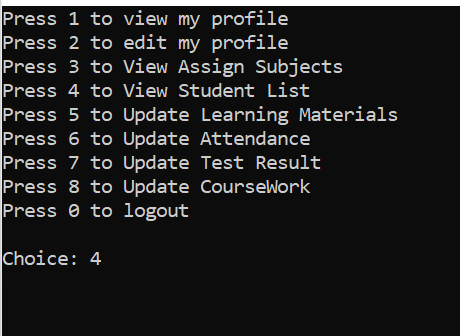
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Lecturer enters choice from menu | Valid (3) | Assigned subjects are shown to lecturer | Assigned subjects are shown to lecturer | Pass | Lecturer can know the details of the assigned subjects |
| **2** | Lecturer enters choice from menu | Invalid (a) | Prompt invalid choice | None | Fail | Lecturer can re-enter choice |

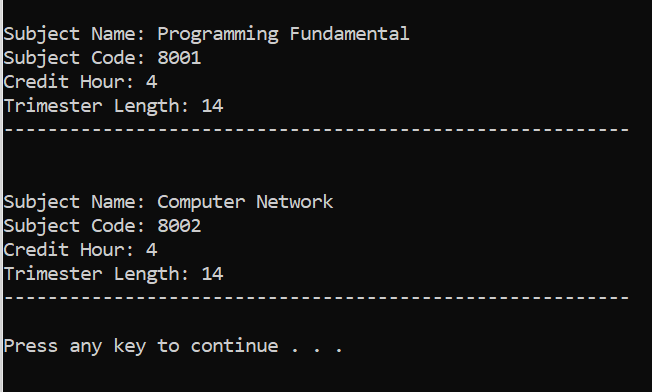




Test Case ID: TC13, Test Case Name: View student list

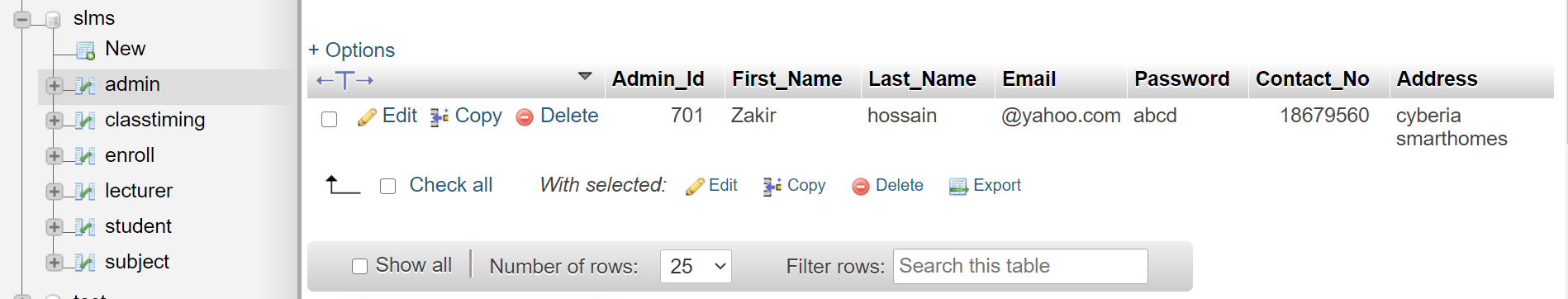
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Actions** | **Input (valid/invalid)** | **Expected**  **Output** | **Actual**  **Output** | **Pass/Fail** | **Remarks** |
| **1** | Lecturer enters subject code from his/her subject list | Valid (8035) | List of students enrolled in that particular subject is shown | List of students enrolled in that particular subject is shown | Pass | Lecturer can know the details of his/her class students |
| **2** | Lecturer enters subject code from his/her subject list | Invalid (1000) | Prompt “Not found…” | None | Fail | System returns to previous menu |



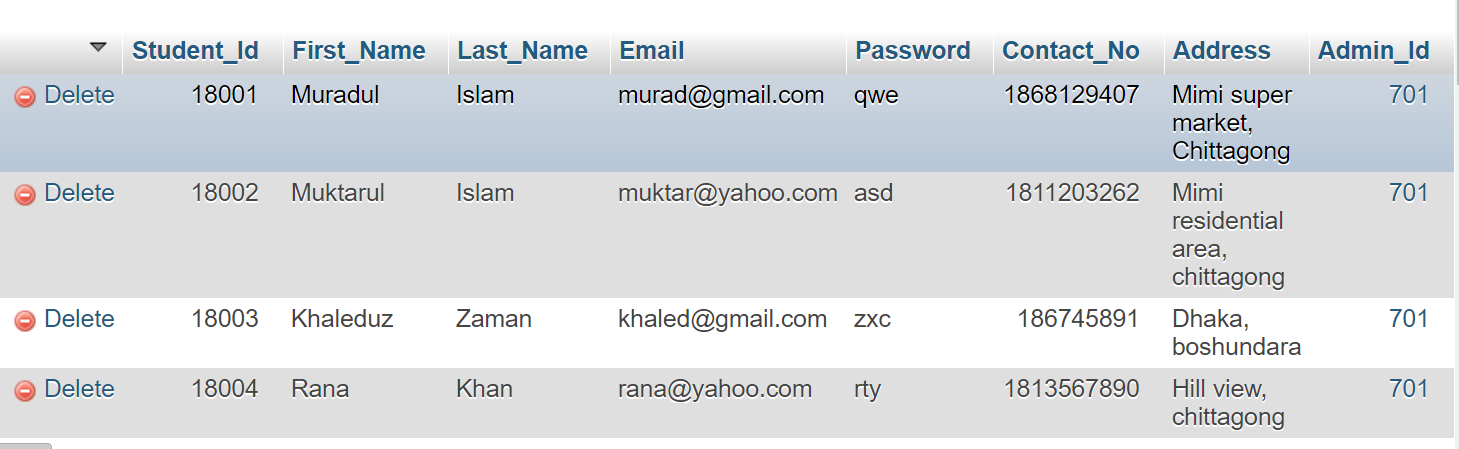


Sample data for the database:

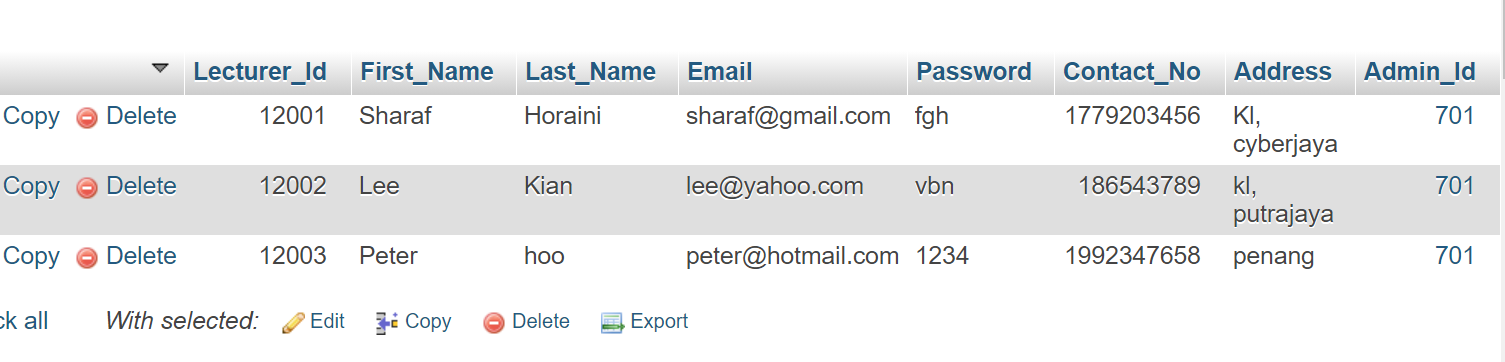
Admin table:



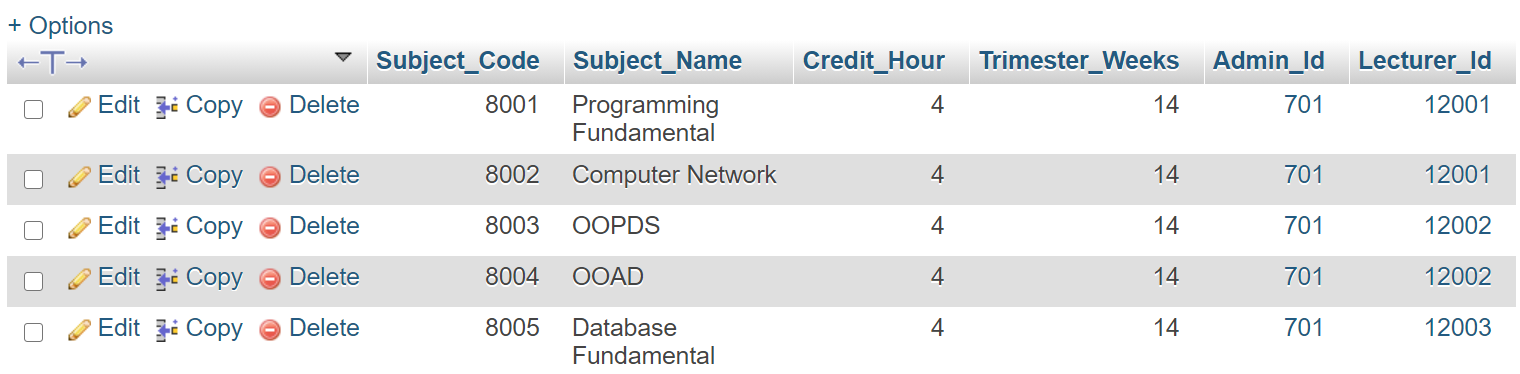
Student table:



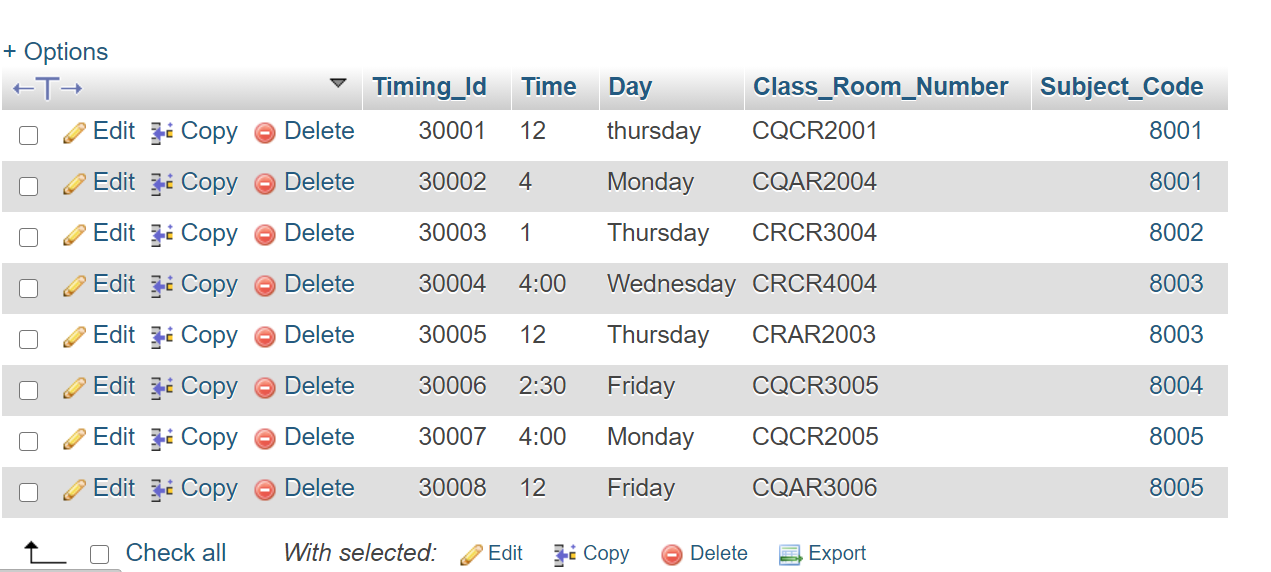
Lecturer table:



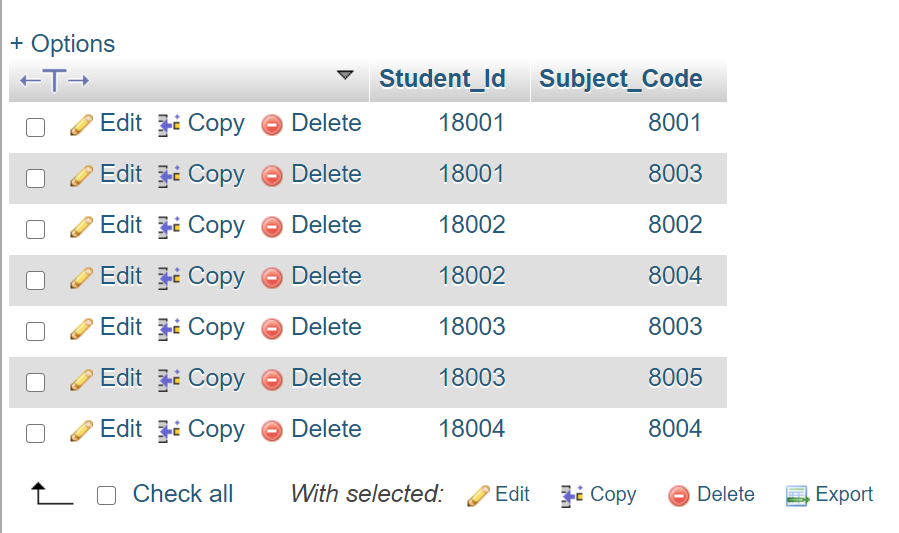
Subject table:



Class Timing table:



Enroll table:



# Configuration Management

## Update of Repository

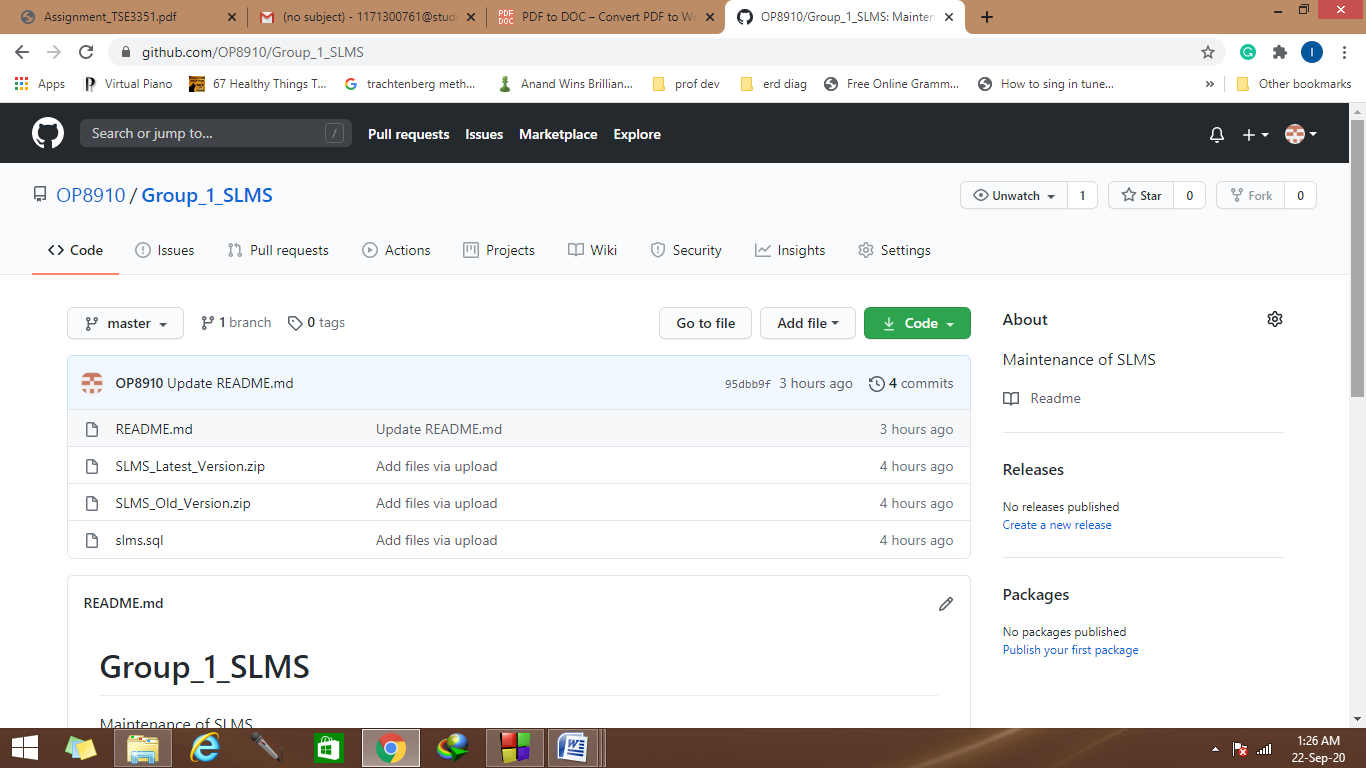
Repository name: Group\_1\_SLMS

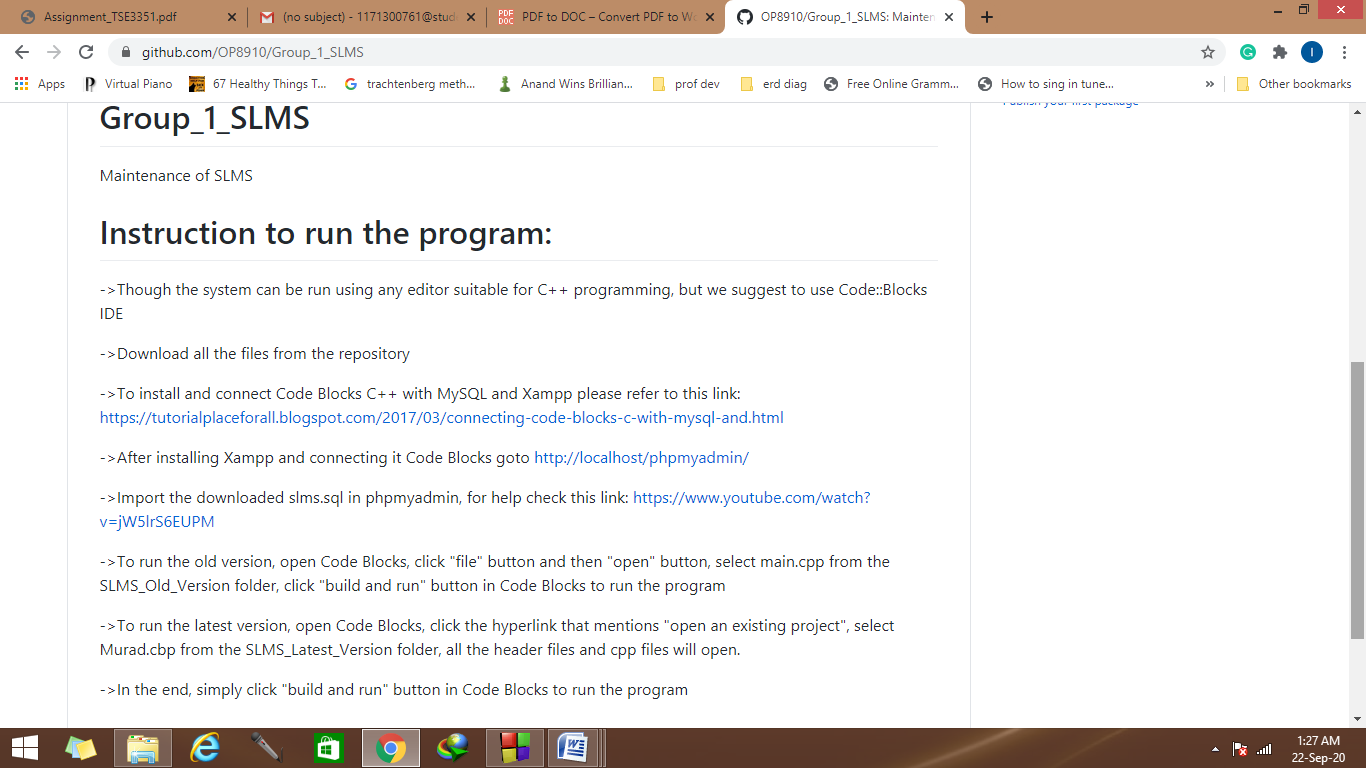
The previous version of the system is named under SLMS\_Old\_Version.zip.

The latest version of the system is named under SLMS\_Latest\_Version.zip.

The database file is named as slms.sql.

## Version Control



**

# Final Release Version

## Release Notes

The latest version 2.0 provides a smooth interaction with the user compared to the older one(version 1.0), which needed some bug fixes. The user can now input details without any constraints, the system has the potential to withstand the invalid inputs. All the manipulated data in the system are stored and retrieved to and from the system.

# <Other relevant documentation (create your own heading)>