CS551 Advanced Software Engineering

Second Increment Report (PG - 7)

Title: Campus Network

Submitted By: Sumanth Gunda (Class ID: 21) (TL)

Goutham Donthu (Class ID: 14)

Kothuri UmaMaheshwarRao(Class ID:28)

Jonnalagadda Rajesh (Class ID: 23)

I. INTRODUCTION

To get an appointment from the Instructor students are using mail to interact with the respective faculty. There is no facility to the students to check their attendance. Instructors do not have the smart facility to assess the student's profile. Faculty are using mails to notify any deadlines to the students. We are designing a web based portal which provides all the above facilities in an efficient way.

II. GOAL OF THE PROJECT

Our project provides a web based application which includes major options like Appointment scheduler, Attendance tracker, Performance tracker, Course tracker. Instructors and Students will no longer interact with the mails. They are provided with an interface which makes the work simple, efficient and saves the time.

III. EXISTING SERVICES

1. **Service Name - Google Maps API**

   Description: - To track the student position before marking Attendance.

   URL: https://developers.google.com/maps/documentation/javascript/tutorial.

2. **Service Name - Google Chart API**

   Description: - To visualize the attendance and performance report of students.

   URL: https://google-developers.appspot.com/chart/interactive/docs/quick_start.

3. **Service Name - Google Drive API**

   Description: - To help create docs inside the web portal.

   URL: https://developers.google.com/drive/web/.

IV. DESIGN

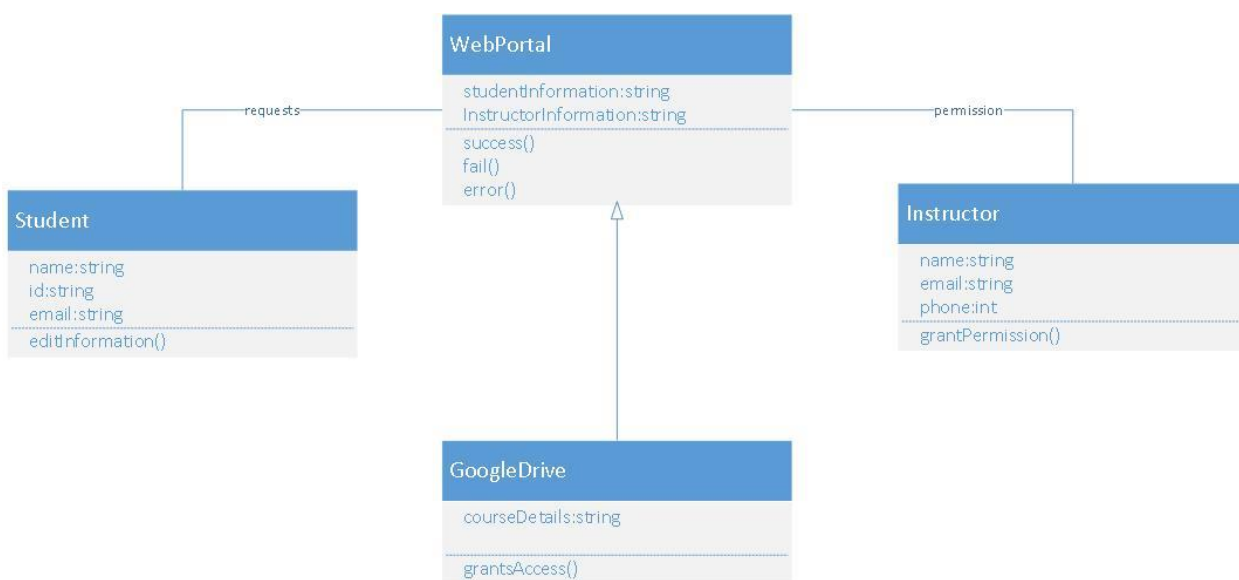The portal has 3 logins Administrator, Instructor and Student.
The project is divided into 5 modules
1. Requests Module.
2. Attendance Module.
3. Performance Module.
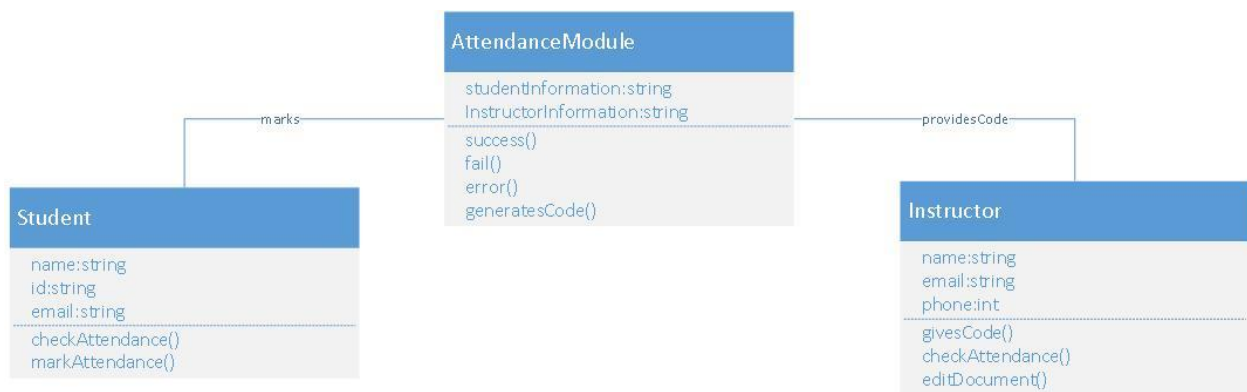4. Course Tracking.
5. Google Drive Integration
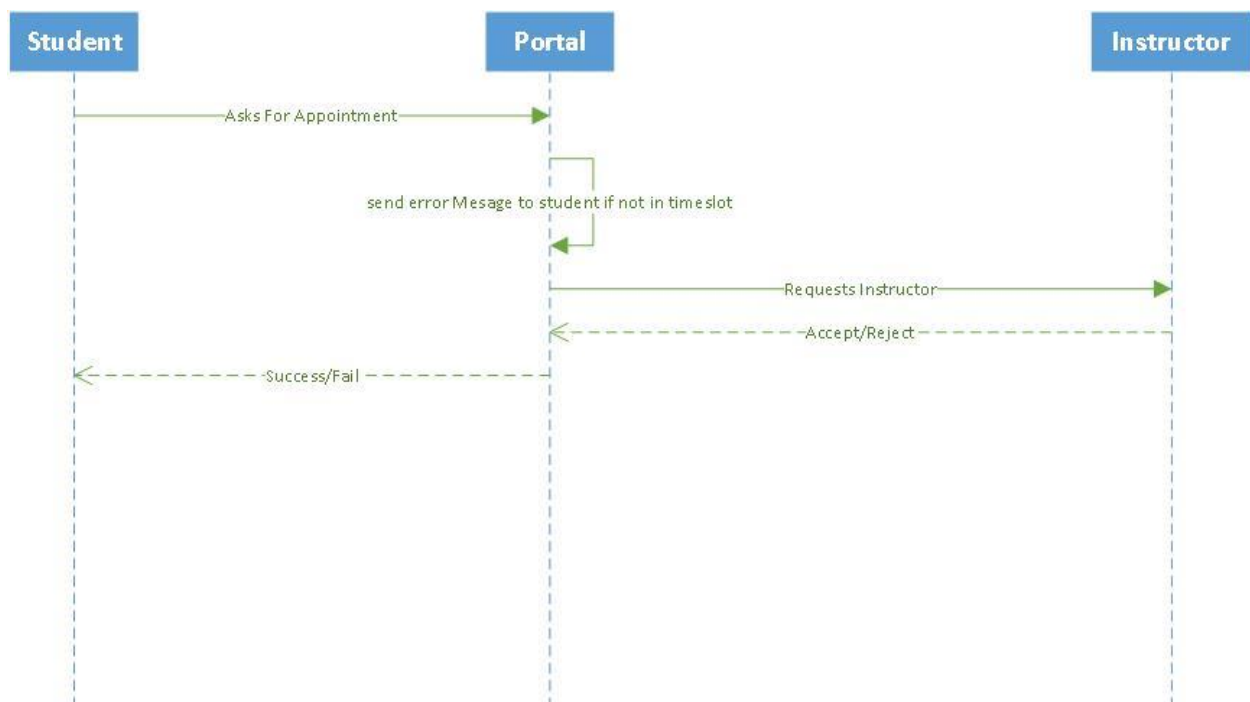
CLASS DIAGRAM:
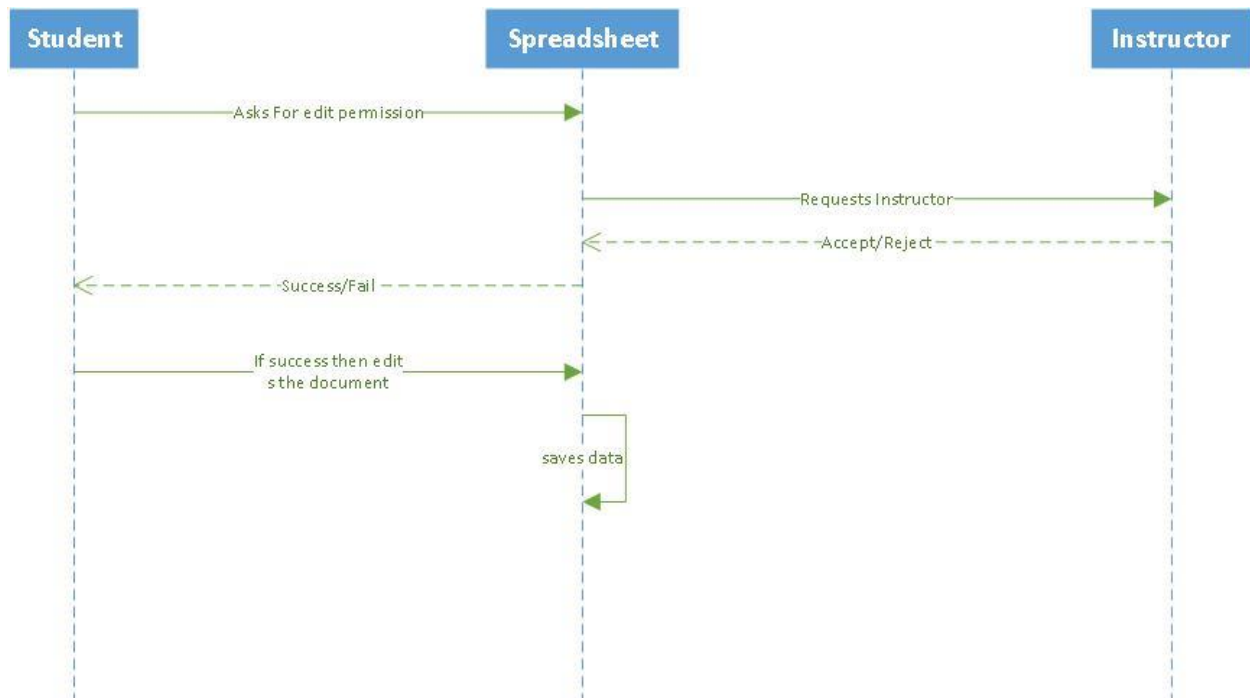

Request Module
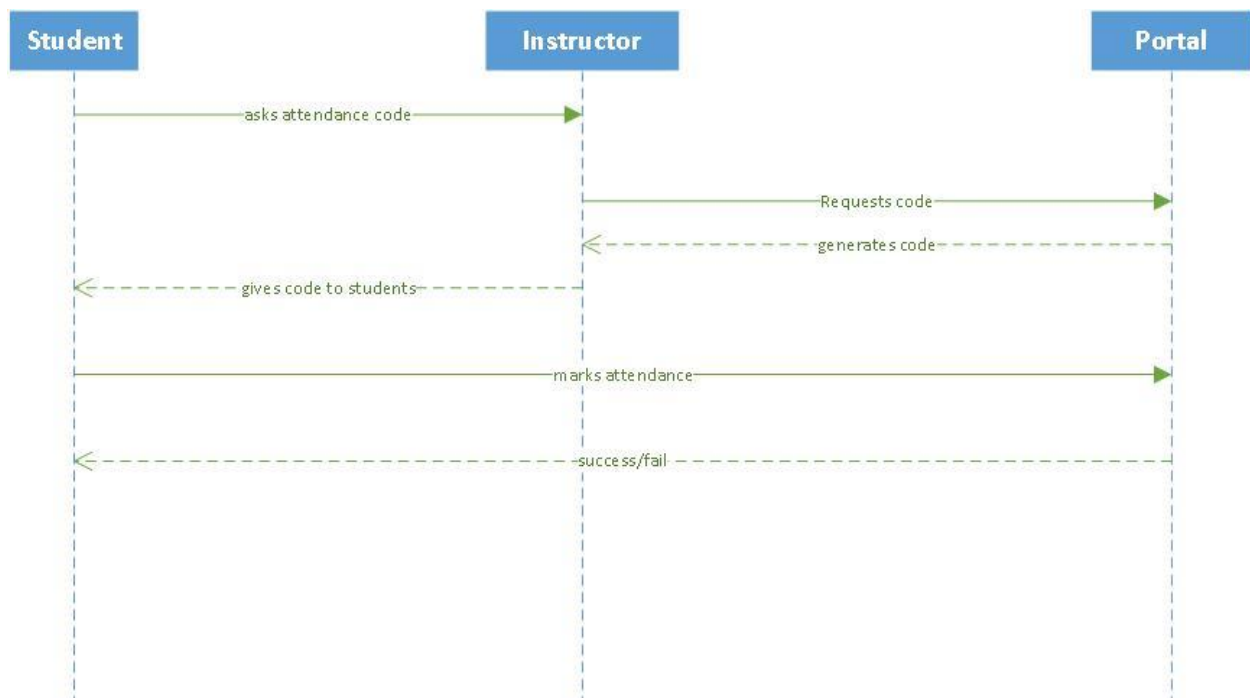

Google Drive Integration

AttendanceModule

studentInformation:string
InstructorInformation:string

success()
fail()
error()
generatesCode()

marks

providesCode

Student

name:string
id:string
email:string

checkAttendance()
markAttendance()

Instructor

name:string
email:string
phone:int

givesCode()
checkAttendance()
editDocument()

Attendance Module

SEQUENCE DIAGRAM:

Student	Portal	Instructor

Asks For Appointment

send error Mesage to student if not in time slot

Requests Instructor

Accept/Reject

Success/Fail

Request Module

Google Drive Integration

Attendance Module

ACTIVITY DIAGRAM:
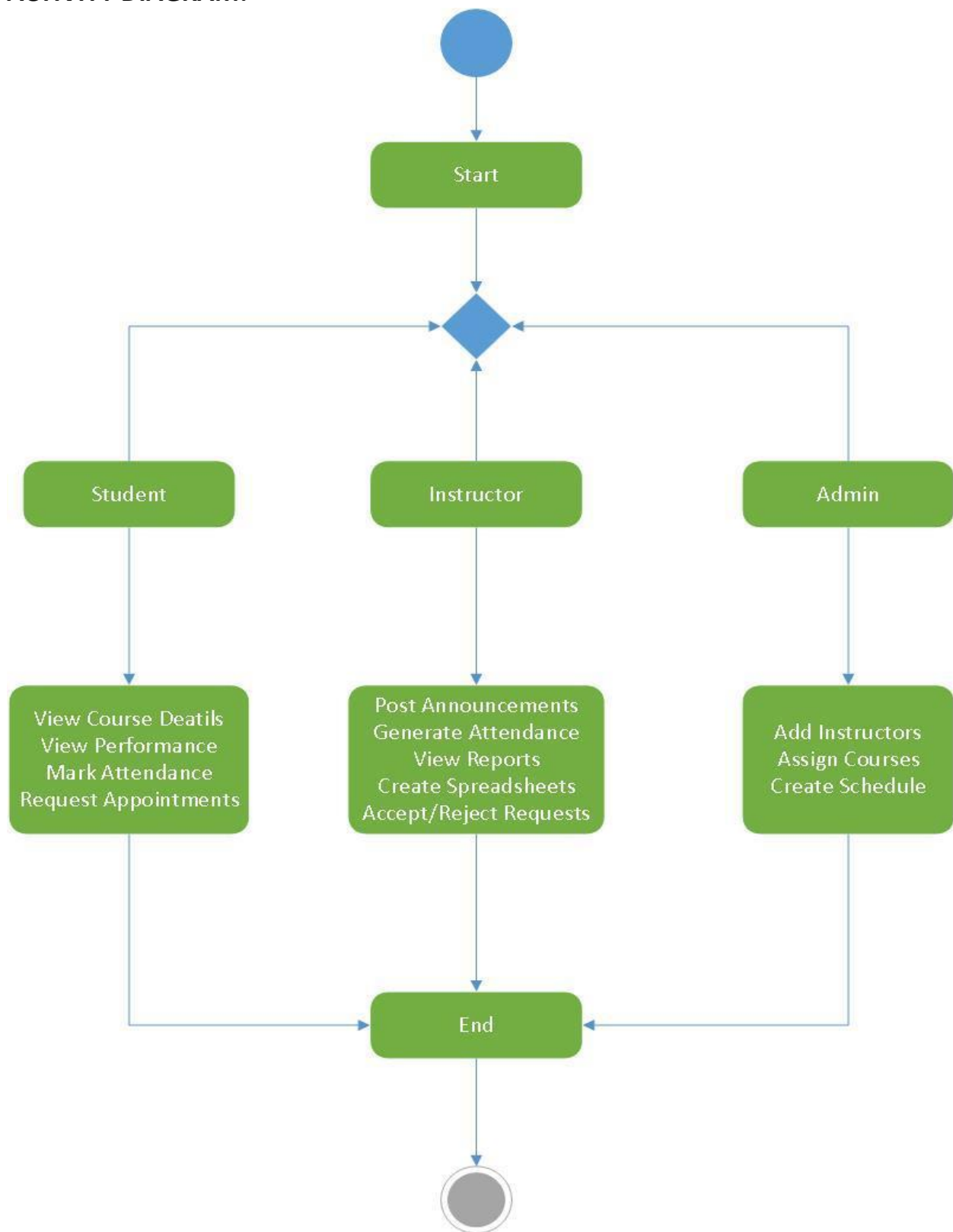
V. IMPLEMENTATION

The Phase 2 of the Project Implements

1. Home page for students
2. Home page for Instructor
3. Instructor page in Request module
4. Student page in Request module
5. Validation of all pages
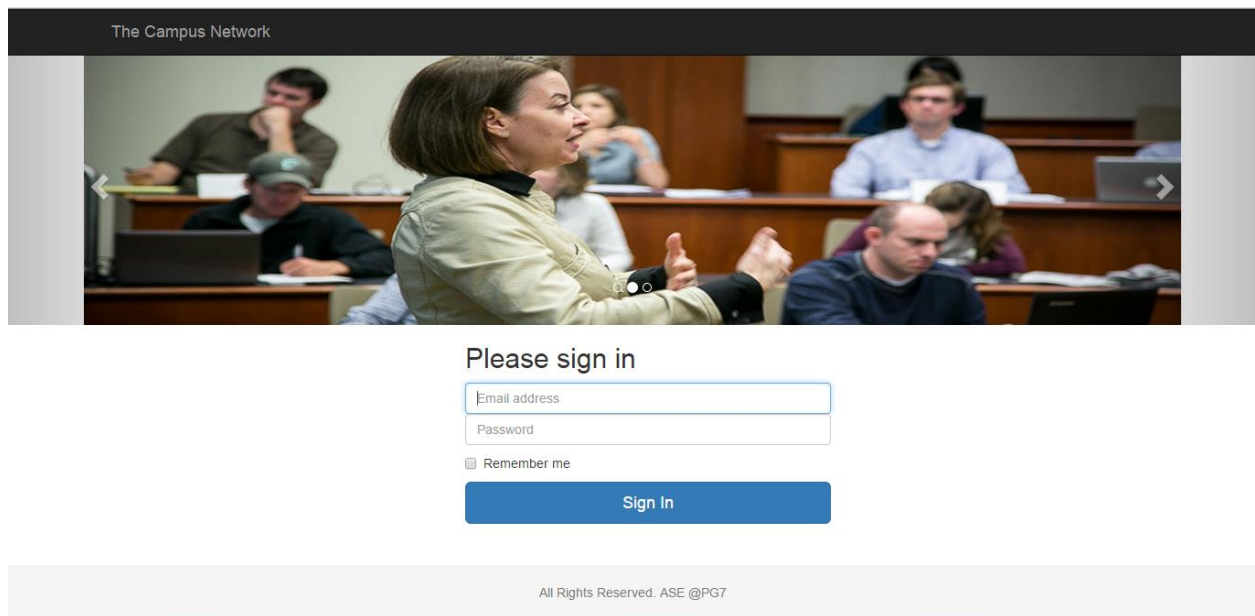6. Database design and Executing scripts

Screenshots:

Fig 1: Login Page

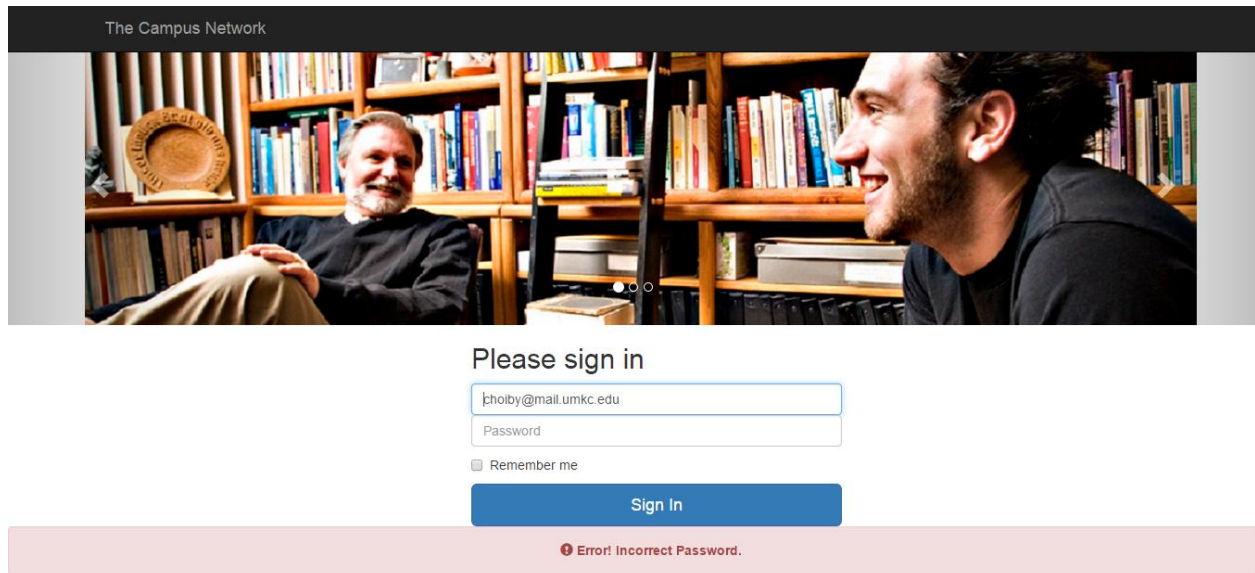This page will allow both students and Instructor to login to the portal.

Fig 2: Login failed because of incorrect password

Here Instructor failed to enter correct password so login failed and error message is shown
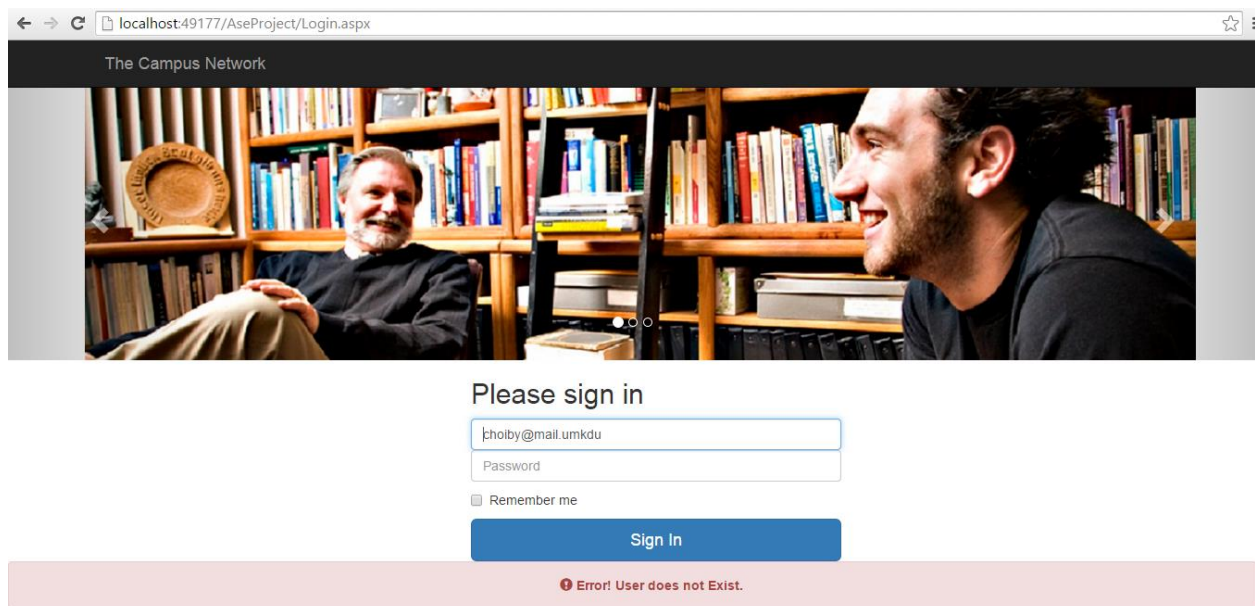
Fig 3: Login Failed because of wrong user name

Here Instructor failed to enter correct ID so login failed and error message is shown

Fig 4: Instructor set's Appointment

Here Instructor can set the time for their appointments

Fig 5: Instructor can delete Appointments for a day

Here Instructor can delete his appointments for any days.

Fig 6: Student Requesting Appointment

Student can request appointment from any of his Instructor

Fig 7: Instructor's view of Appointments

This is the page showing all the appointment requests came to the Instructor by students

Fig 8: Student's view of his Appointment Status

Student viewing status for all his Appointments

Fig 9: Course Completion Progress

Bar indicates dynamically the progress of course completion

Fig 10: Instructor gets schedule of the course

When Instructor press getschedule button he will get the schedule of that particular course

VI. TESTING

To make sure the quality of the software is good testing is a critical component of the development. Software testing is done to ensure quality and it is also the review of the design and specification.

NUnit Testing:

This tool is used to unit test the application that is developed in C# ASP.Net. It is a Unit Testing Framework when the code that was written in C# can be tested.

The test cases that were designed to test the Campus Network Application are:

- **MasterPage()**

  This function test for the successful loading of the master page when a content page has been inherited from the master page.

- **HomePage()**

  This function tests for successful loading of the dashboard of both the student and instructor after successful loading.

  This function also tests whether the session variables are stored or not.

- **LoginPage()**

  This function tests for successful loading of the login page and also tests for all the validation controls on the page. This function tests whether the appropriate error messages are being displayed.

- **Validations()**

  This function test whether the validations are working properly before the page is sent back to the server.

- **ManageAppointment()**

  This function tests whether approve or reject works properly and the web page redirected to expected page or not

- **ApplyAppointment()**

  This function tests whether after applying appointments Instructor gets the request or not

- **AppoinmentRequest()**

  This function tests whether appointment request came to respective Instructor or not
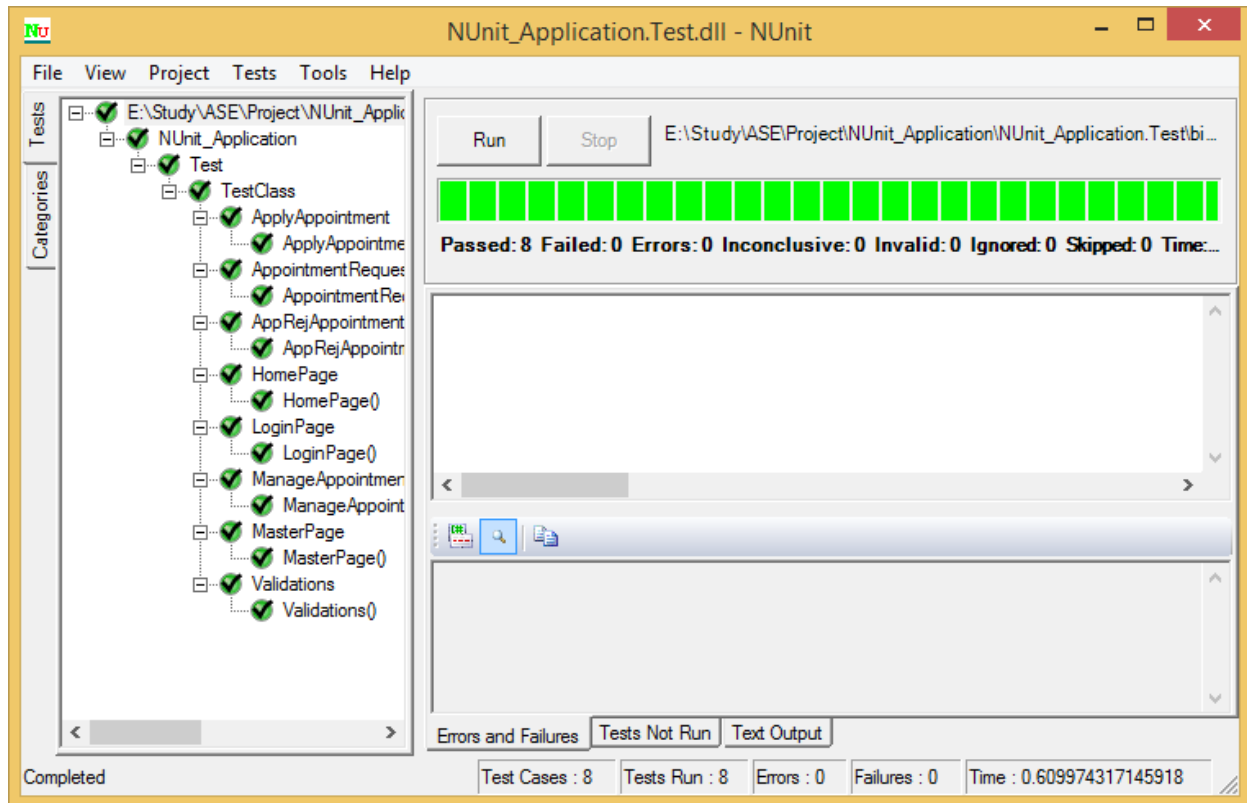
Screen shot for NUnit Testing:

Fig 7: Implementation of Test Cases

VII. DEPLOYMENT

Project Scrum Do Link:

http://www.scrumdo.com/projects/project/campus-network/iteration/121658/board

Screenshots:

GitHub Site URL:

https://github.com/sumanthgundaa/ASE-Project

VIII. PROJECT MANAGEMENT

Implementation Status Report:

Work Completed:

Story 1: Planning

Description:

Whole Team participates in this story. Here we plan the whole structure of project like what features are we providing and how to implement them.

Responsibility:

Here the project scope and appropriate methods for completing the project is determined.

Time Taken: 6hours

Contributions: All team members equally participated in this story.

Story 2: Master Page Design

Description: Here we create Master page for our project

Responsibility: In this we create common structure for our project

Time Taken: 4 hours

Contributions: Sumanth and Goutham.

Story 3: Creating Home Page

Description: Here we design the Home Page for both Student and Instructor. Navigation in the Home Page changes based on the roles of the user.

Responsibility: Home Page will be designed and the Navigation bar will be changed based on the User Login.

Time Taken: 3hours

Contributions: Mahesh and Rajesh.

Story 4: Creating Login Page

Description: Here we design the login page for the portal.

Responsibility: All the Validations should be done from the database and respective stored procedures should be written to validate the user.

Time Taken: 6hours

Contributions: Sumanth and Goutham.

Story 5: Database Design

Description: All the tables will be created and also the stored procedures will be written.

Responsibility: Table Scripts are created and the sample data will be inserted into the database.

Time Taken: 5hours

Contributions: Mahesh and Rajesh.

Story 6: Instructor Home Page

Description: This is the homepage after Instructor login into his account. In this page he can see all the courses taught by him and corresponding schedule.

Responsibility: All Courses taught by the Instructor are displayed in his home page and the schedule is also available

Time Taken: 9hours

Contributions: Mahesh and Rajesh.

Story 7: Student Home Page

Description: This is the homepage for the student after he logged in. He can see what all courses he enrolled and the syllabus of respective courses. He can also view works to be done.

Responsibility: All Courses registered by the student are displayed in his home page

Time Taken: 8hours

Contributions: Mahesh and Rajesh.

Story 8: Instructor Page in Request Module

Description: We create page for instructor such that he can accept or reject the students request for appointment and he can view all the requests

Responsibility: All the requests are displayed in his page and he can accept or reject the request. After the reply from Instructor result is reflected in student page

Time Taken: 9hours

Contributions: Sumanth and Goutham.

Story 9: Student Page in Request Module

Description: Here student can request for an appointment with desired instructor and he can later view the status of the request

Responsibility: Instructors available timings are displayed in his page so that he can seek appointment. After student requests for appointment it is reflected in Instructors page that particular student is asking for his appointment.

Time Taken: 10hours

Contributions: Sumanth and Goutham.

Story 10: Unit Testing for Home Pages of Student and instructor

Description: Here we test the home pages independently such that they are working correctly according to our specifications without any bugs

Responsibility: Unit testing is done on home pages of student and Instructor

Time Taken: 4hours

Contributions: Mahesh and Rajesh.

Story 10: Unit Testing for Appointment Request Module

Description: Here we test the Instructor and Student pages in Appointment Request module such that they are working correctly according to our specifications without any bugs

Responsibility: Here we do the unit testing for student page and Instructors page in appointment request module. Our goal is to run this module without any bugs.

Time Taken: 5hours

Contributions: Sumanth and Goutham.s

Work to be completed:

Description: The Modules that have to be designed on both Instructor and Student side are:

- Performance Module
- Google Drive Integration

Responsibility: Each module has to be created to student and also the Instructor. Each of them has different responsibilities based on the roles.

Time Taken: The Time Taken to develop each module for both student and instructor would be 40hrs. And also the time taken to insert sample data and write stored procedures would be 20hrs.

Note: The time that was written was just an expected time. It may vary in real time.