CS551 Advanced Software Engineering

Third 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/.

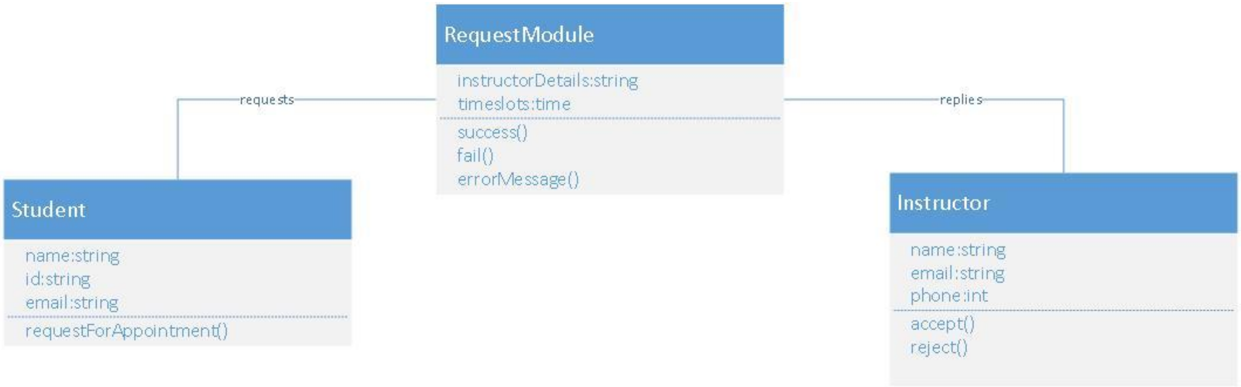
IV. DESIGN

The portal has 3 logins Administrator, Instructor and Student.

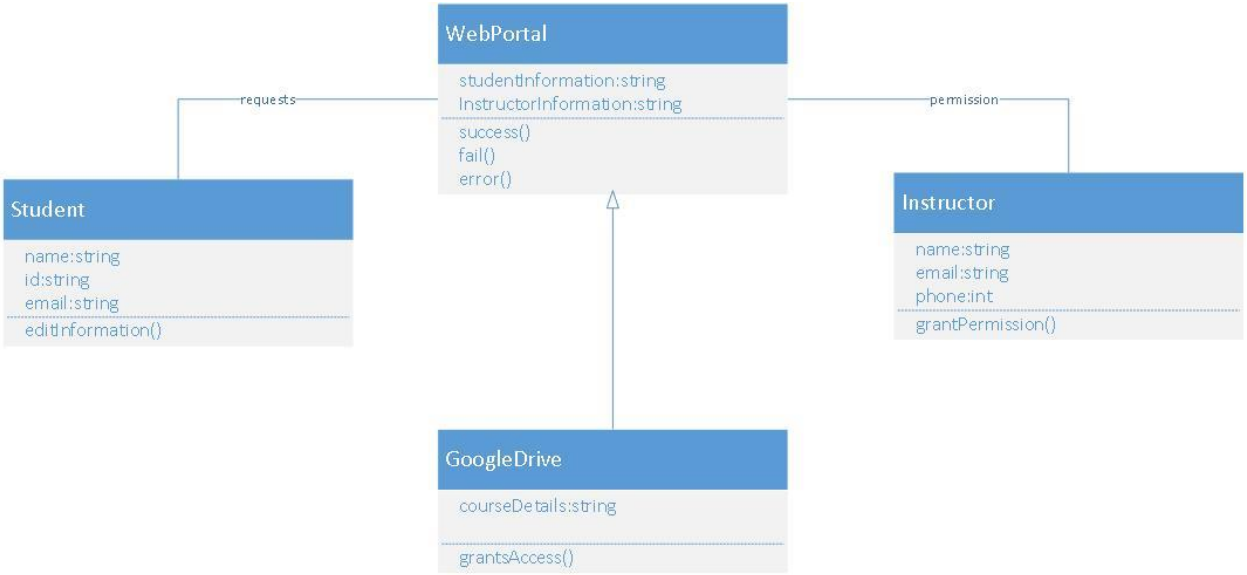
The project is divided into 5 modules

1. Requests Module.
2. Attendance Module.
3. Performance Module.
4. Course Tracking.
5. Google Drive Integration

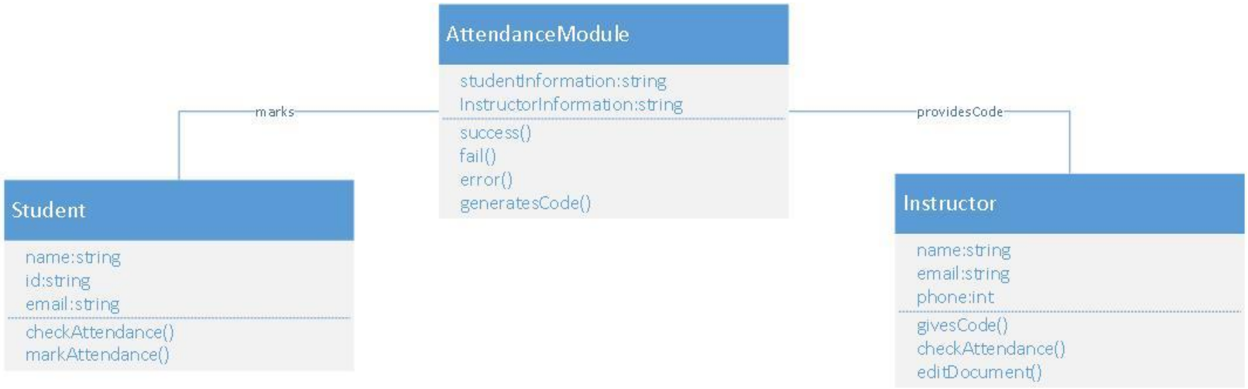
CLASS DIAGRAM:



Request Module

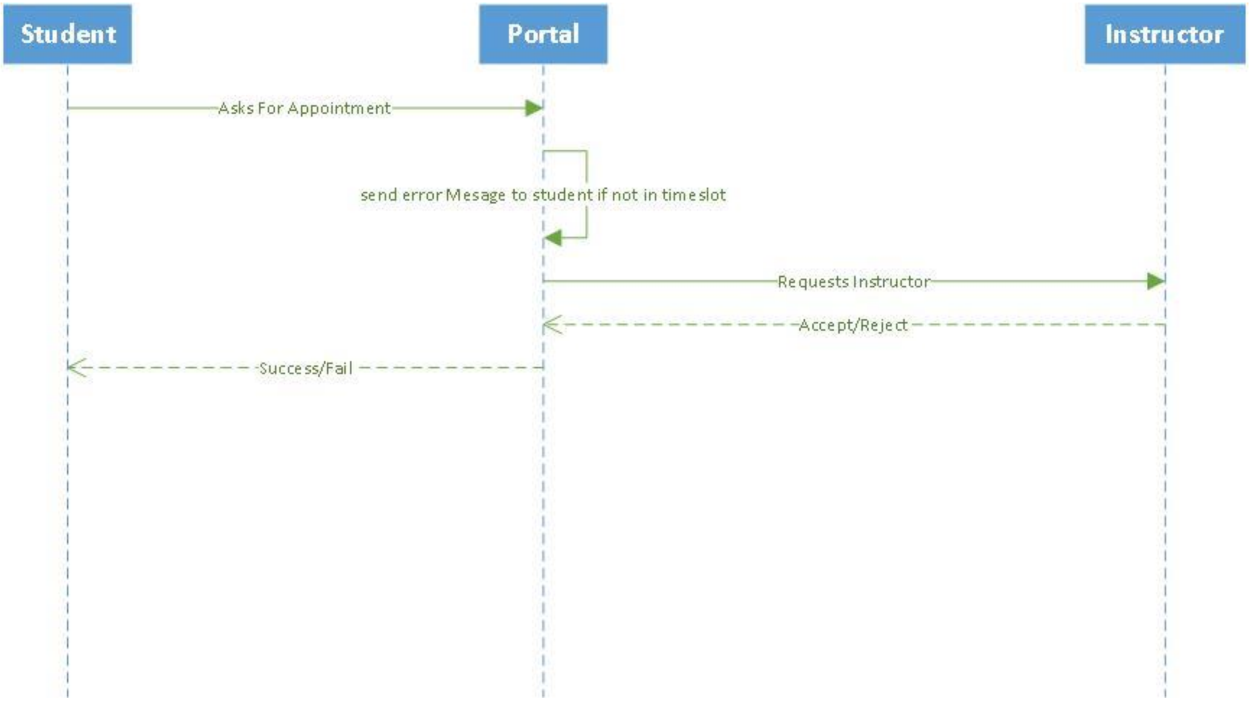


Google Drive Integration

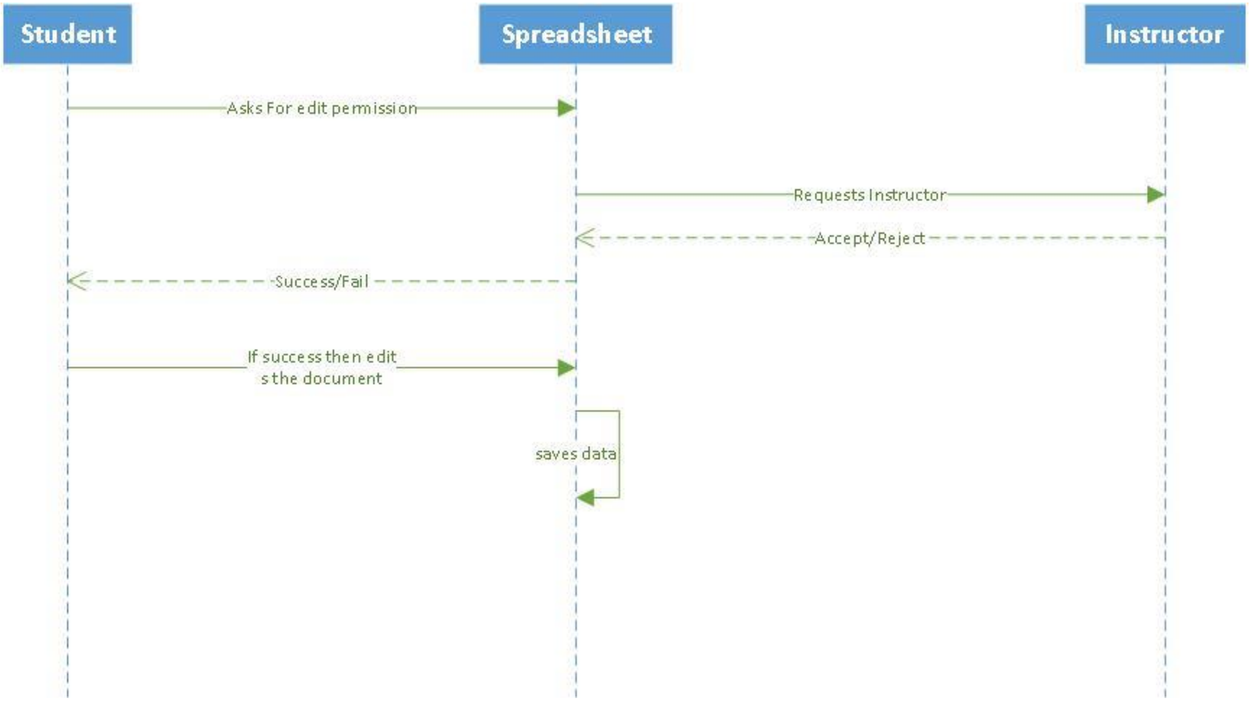


Attendance Module

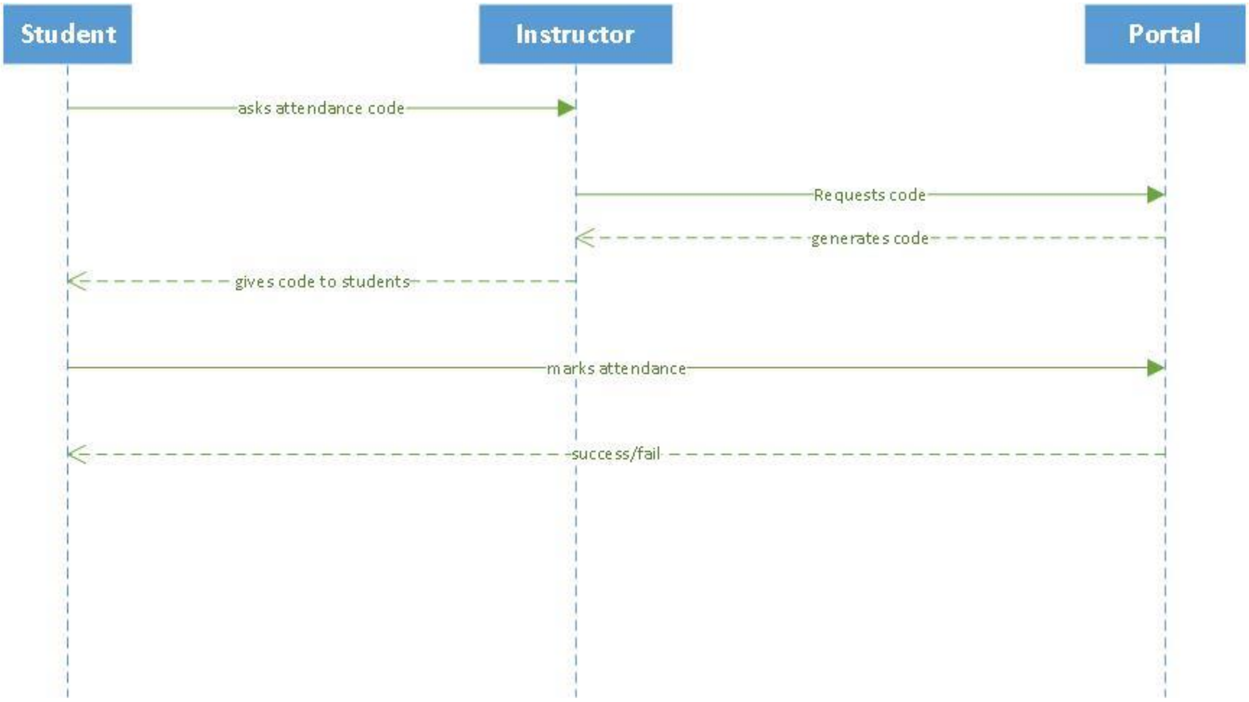
SEQUENCE DIAGRAM:



Request Module

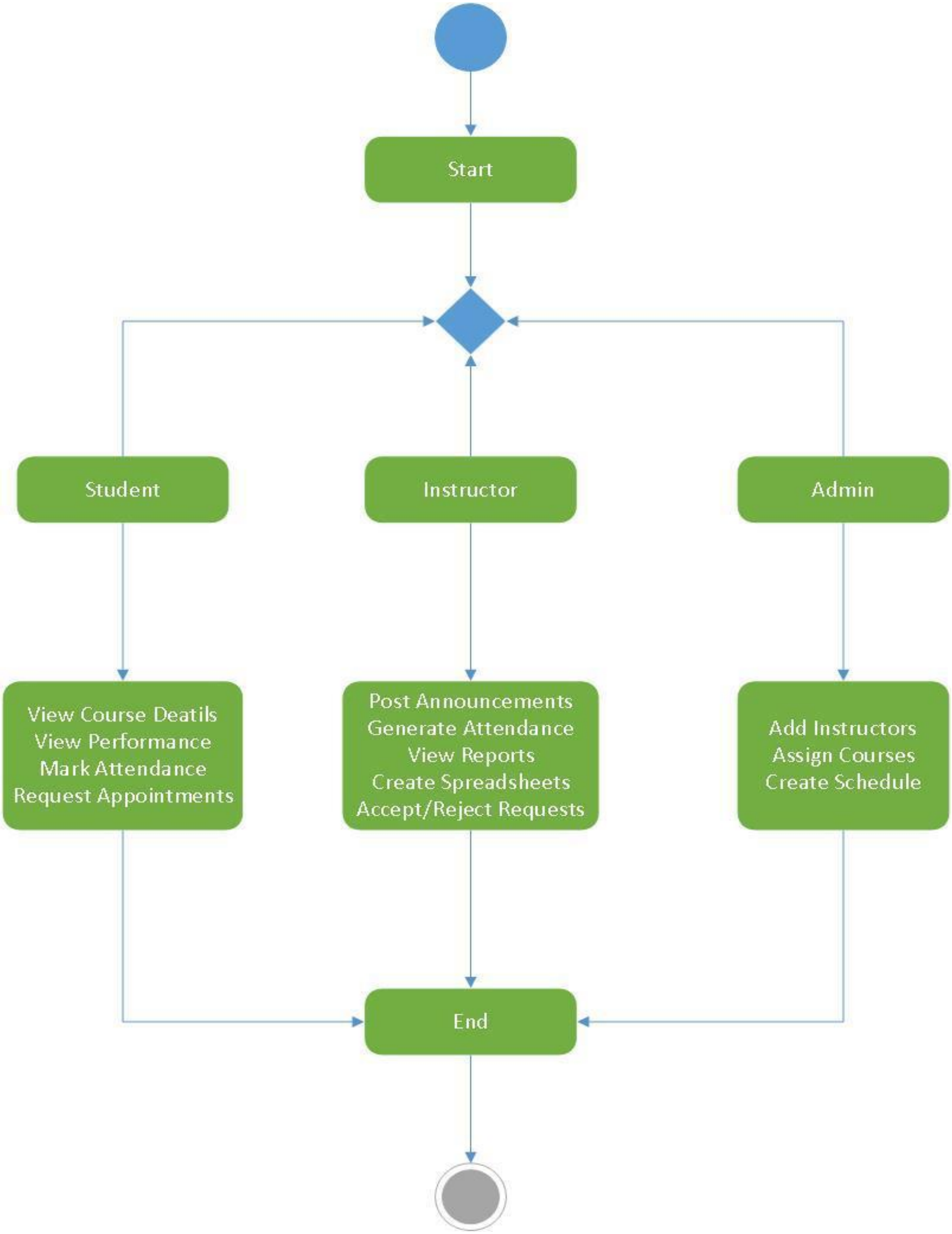


Google Drive Integration



Attendance Module

ACTIVITY DIAGRAM:

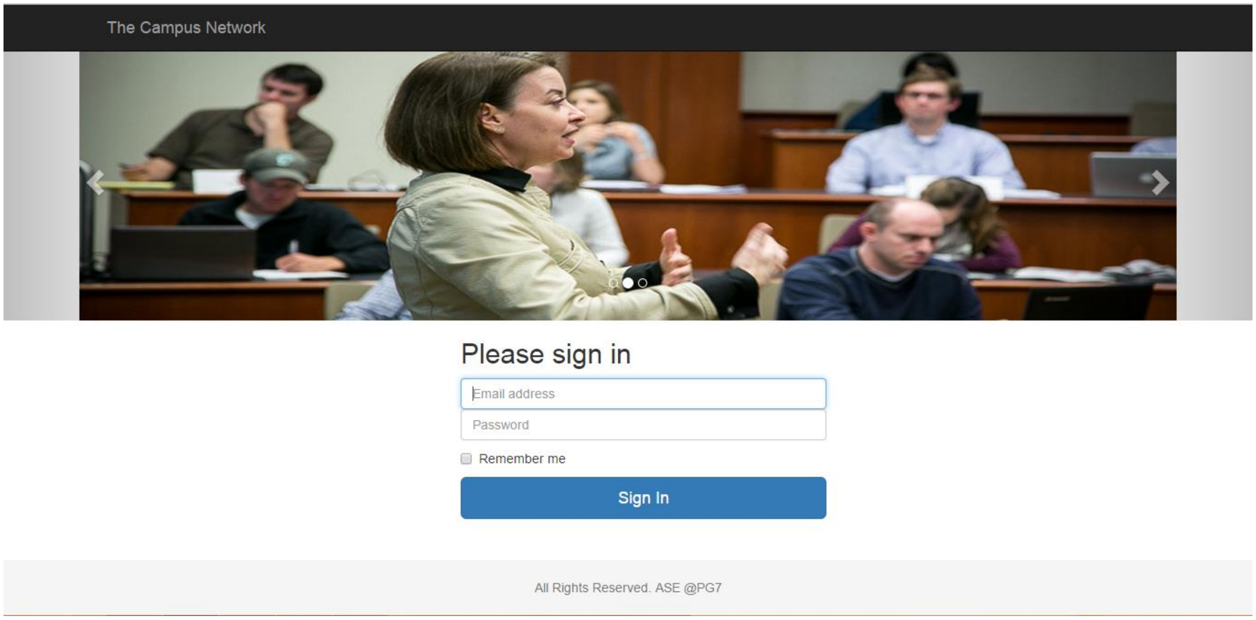


V. IMPLEMENTATION

The Phase 3 of the Project Implements

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

Screenshots:



The Campus Network

Please sign in

Email address

Password

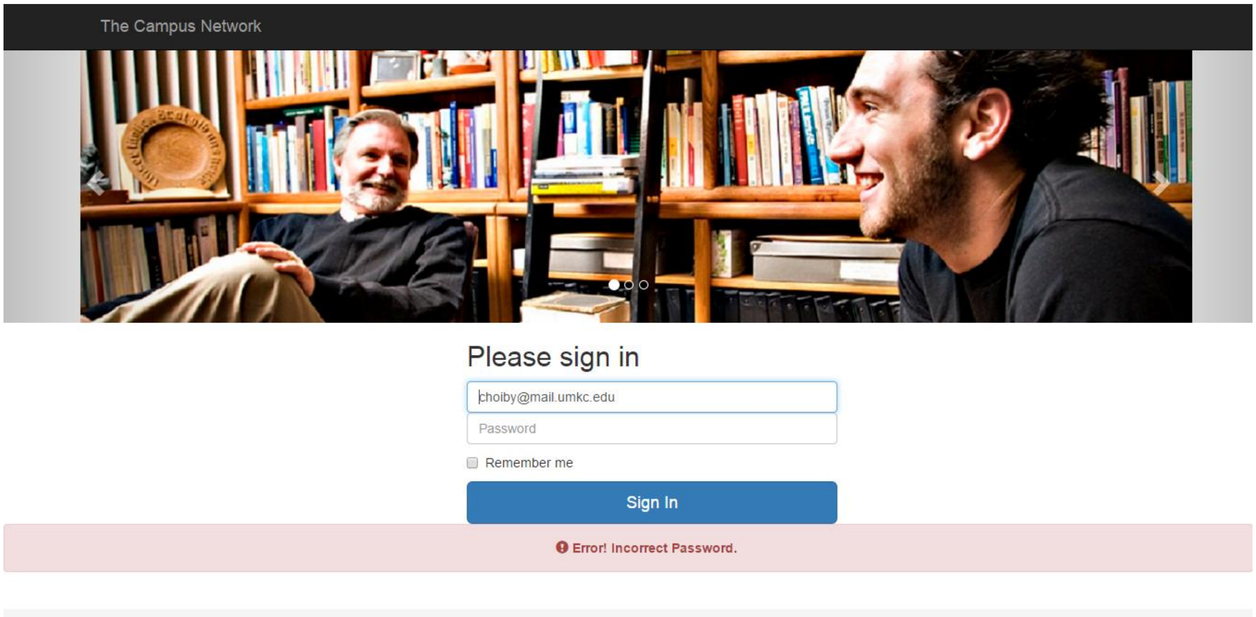
□ Remember me

Sign In

All Rights Reserved. ASE @PG7

Fig 1: Login Page

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

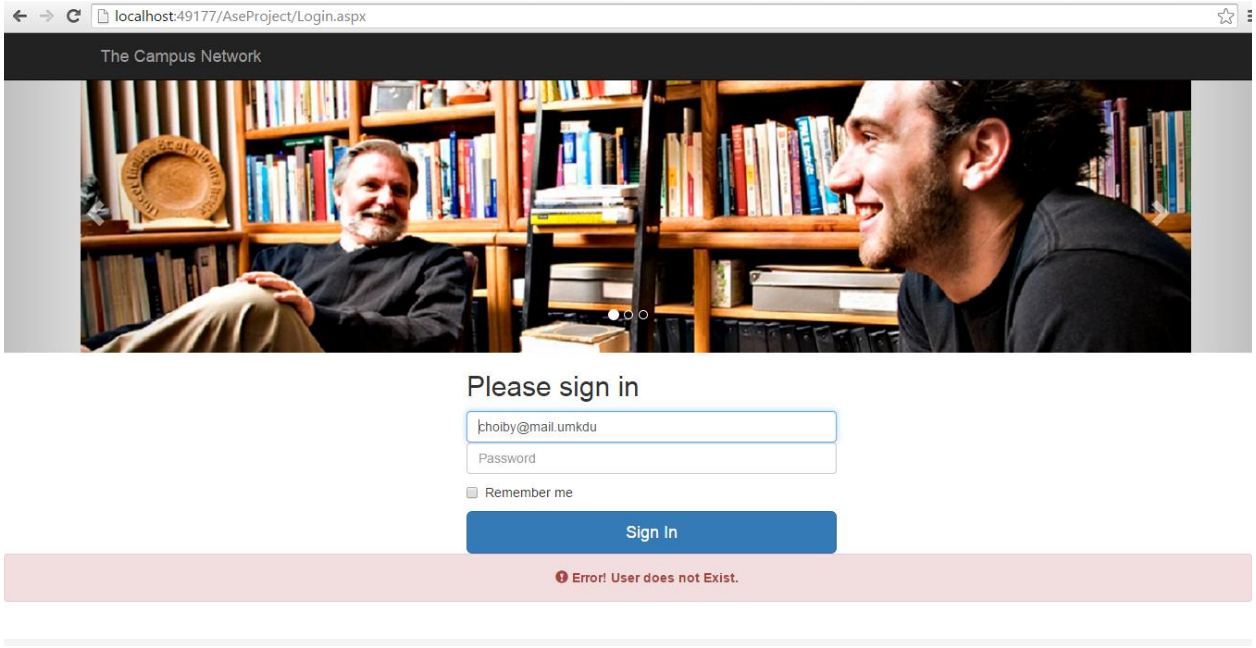


choiby@mail.umkc.edu

Error! Incorrect Password.

Fig 2: Login failed because of incorrect password

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

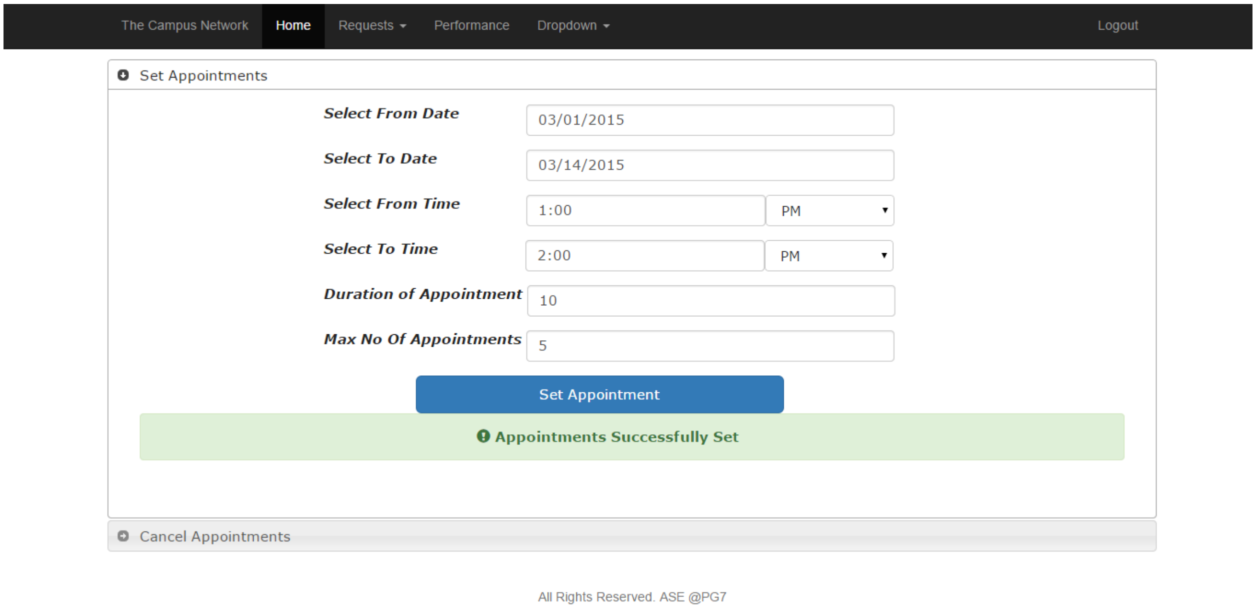

localhost:49177/AseProject/Login.aspx

choiby@mail.umkdu

Error! User does not Exist.

Fig 3: Login Failed because of wrong user name

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



Select From Date: 03/01/2015

Select To Date: 03/14/2015

Select From Time: 1:00 PM

Select To Time: 2:00 PM

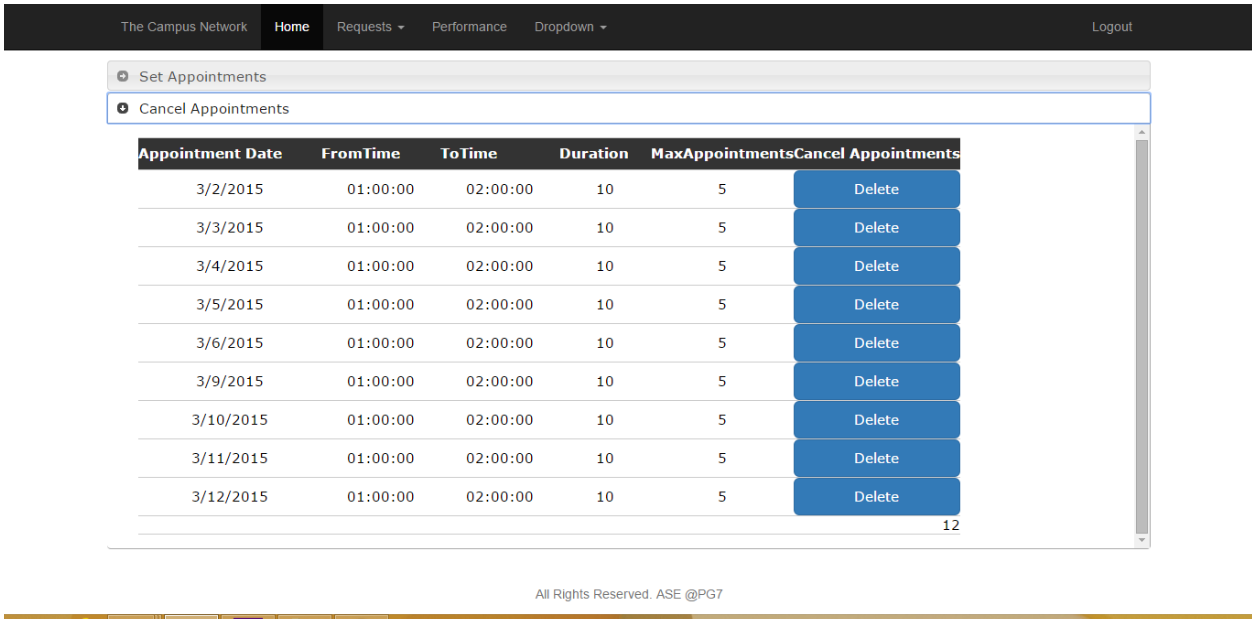
Duration of Appointment: 10

Max No Of Appointments: 5

Appointments Successfully Set

Fig 4: Instructor set's Appointment

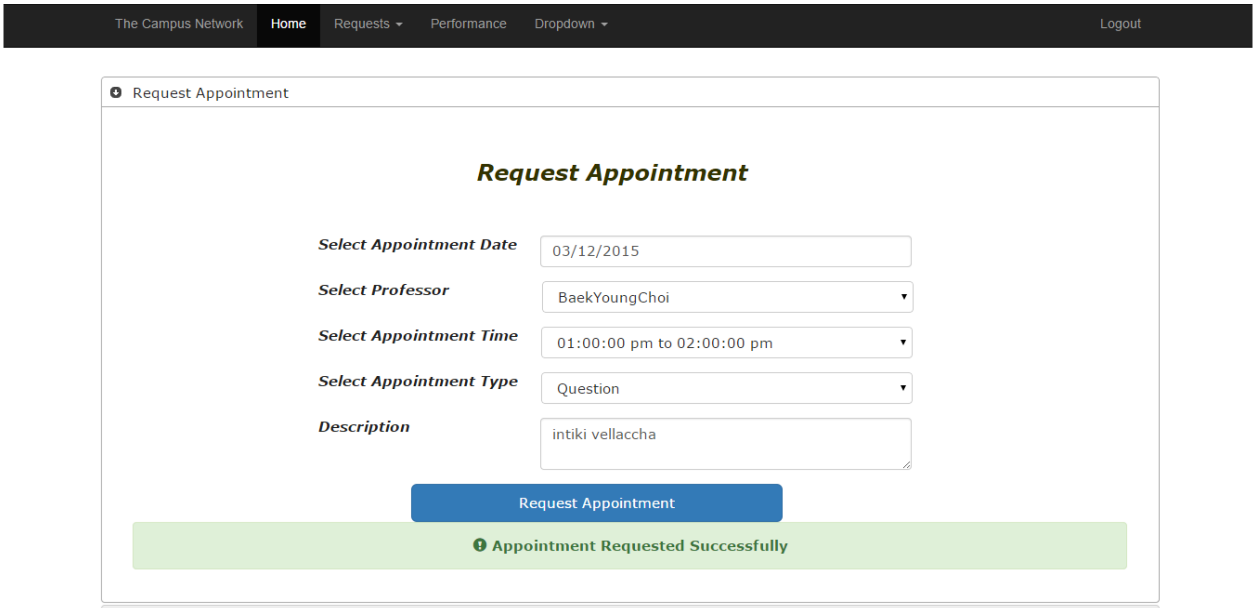
Here Instructor can set the time for their appointments



Appointment Date	FromTime	ToTime	Duration	MaxAppointments	Cancel Appointments
3/2/2015	01:00:00	02:00:00	10	5	Delete
3/3/2015	01:00:00	02:00:00	10	5	Delete
3/4/2015	01:00:00	02:00:00	10	5	Delete
3/5/2015	01:00:00	02:00:00	10	5	Delete
3/6/2015	01:00:00	02:00:00	10	5	Delete
3/9/2015	01:00:00	02:00:00	10	5	Delete
3/10/2015	01:00:00	02:00:00	10	5	Delete
3/11/2015	01:00:00	02:00:00	10	5	Delete
3/12/2015	01:00:00	02:00:00	10	5	Delete
					12

Fig 5: Instructor can delete Appointments for a day

Here Instructor can delete his appointments for any days.



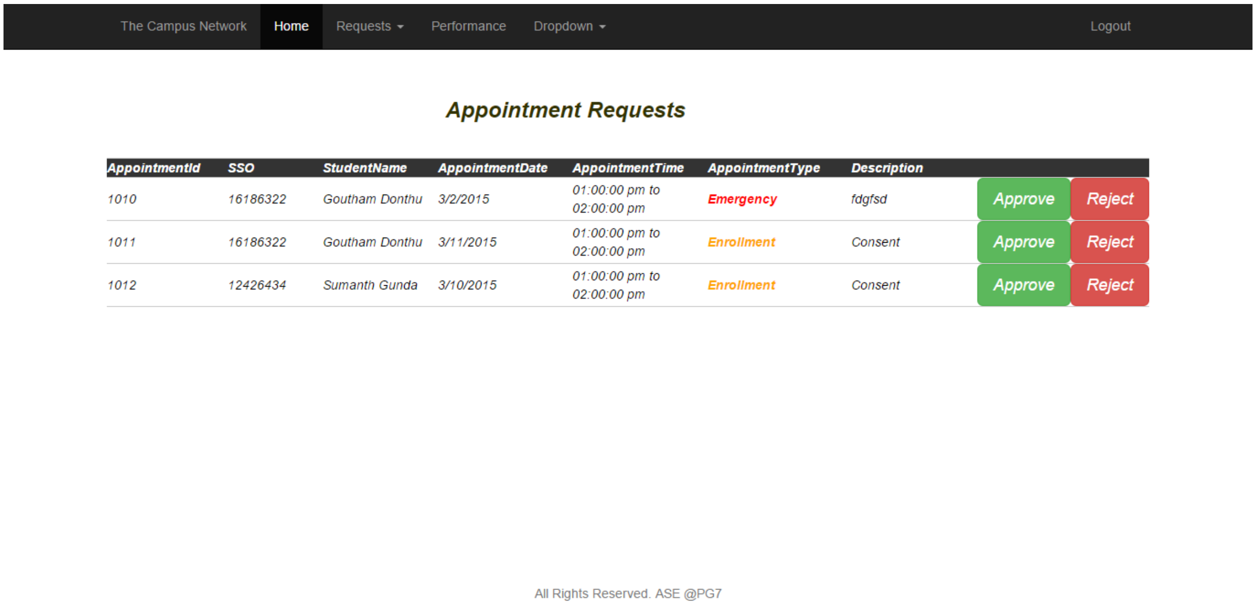
Request Appointment

Select Appointment Date	03/12/2015
Select Professor	BaekYoungChoi
Select Appointment Time	01:00:00 pm to 02:00:00 pm
Select Appointment Type	Question
Description	intiki vellaccha

Appointment Requested Successfully

Fig 6: Student Requesting Appointment

Student can request appointment from any of his Instructor



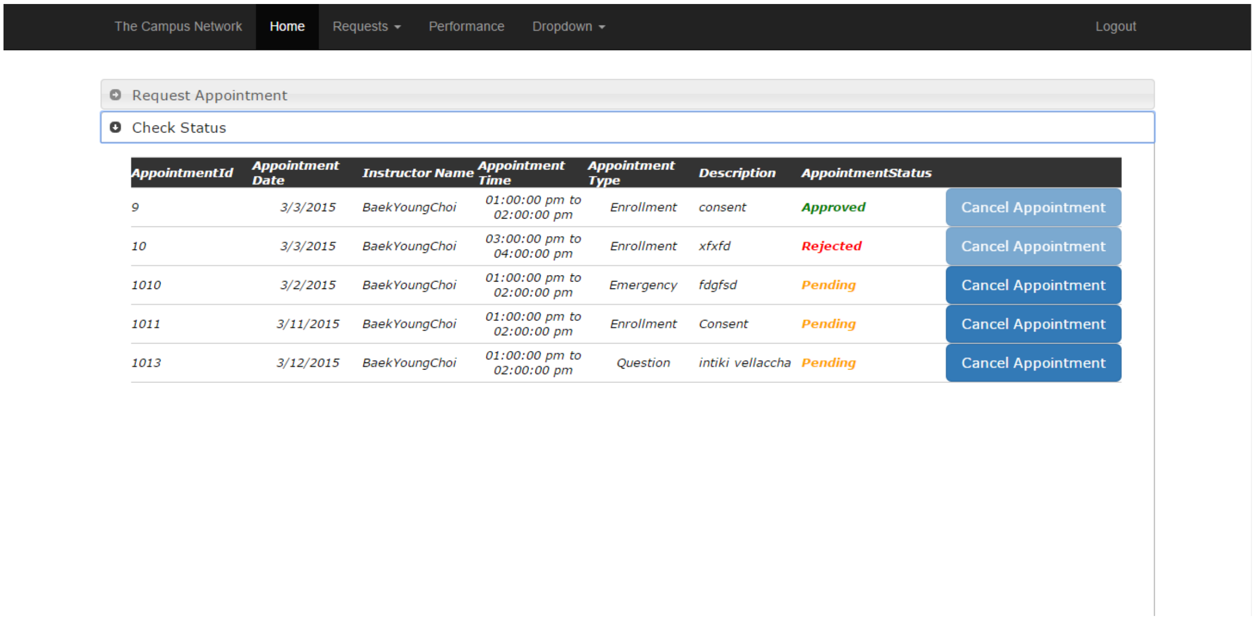
Appointment Requests

AppointmentId	SSO	StudentName	AppointmentDate	AppointmentTime	AppointmentType	Description		
1010	16186322	Goutham Donthu	3/2/2015	01:00:00 pm to 02:00:00 pm	Emergency	fdgfsd	Approve	Reject
1011	16186322	Goutham Donthu	3/11/2015	01:00:00 pm to 02:00:00 pm	Enrollment	Consent	Approve	Reject
1012	12426434	Sumanth Gunda	3/10/2015	01:00:00 pm to 02:00:00 pm	Enrollment	Consent	Approve	Reject

All Rights Reserved. ASE @PG7

Fig 7: Instructor's view of Appointments

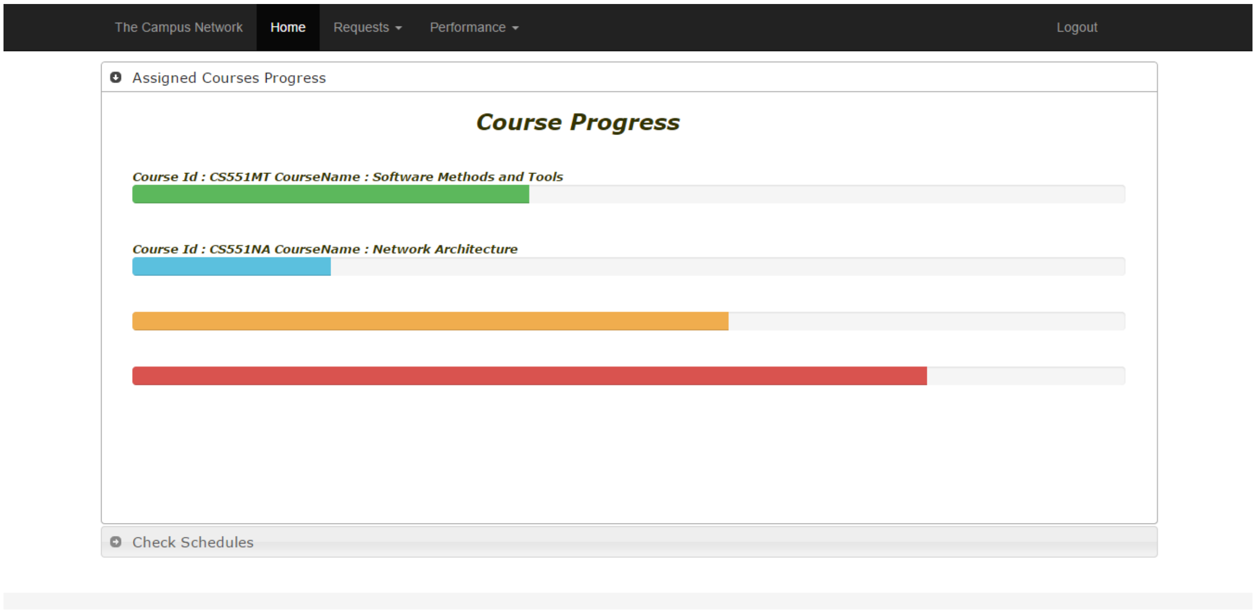
This is the page showing all the appointment requests came to the Instructor by students



AppointmentId	Appointment Date	Instructor Name	Appointment Time	Appointment Type	Description	AppointmentStatus	
9	3/3/2015	BaekYoungChoi	01:00:00 pm to 02:00:00 pm	Enrollment	consent	Approved	Cancel Appointment
10	3/3/2015	BaekYoungChoi	03:00:00 pm to 04:00:00 pm	Enrollment	xfxfd	Rejected	Cancel Appointment
1010	3/2/2015	BaekYoungChoi	01:00:00 pm to 02:00:00 pm	Emergency	fdgfsd	Pending	Cancel Appointment
1011	3/11/2015	BaekYoungChoi	01:00:00 pm to 02:00:00 pm	Enrollment	Consent	Pending	Cancel Appointment
1013	3/12/2015	BaekYoungChoi	01:00:00 pm to 02:00:00 pm	Question	intiki vellaccha	Pending	Cancel Appointment

Fig 8: Student's view of his Appointment Status

Student viewing status for all his Appointments



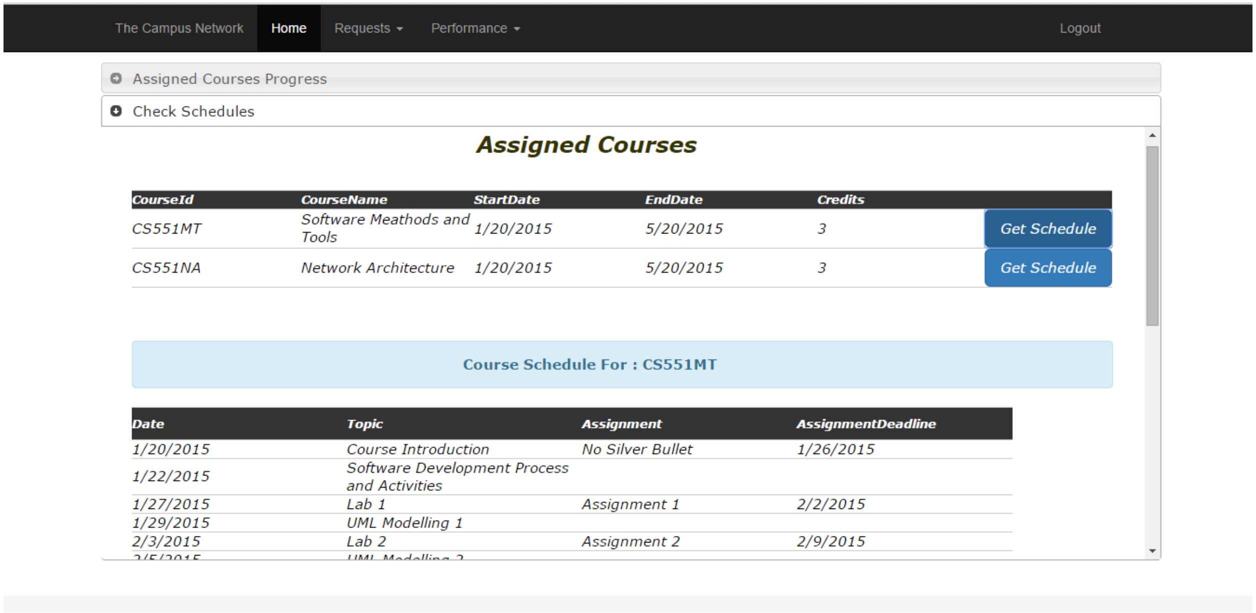
Course Progress

Course Id : CS551MT CourseName : Software Methods and Tools

Course Id : CS551NA CourseName : Network Architecture

Fig 9: showing course progress for Instructor

This page showing how much course has been completed dynamically



Assigned Courses

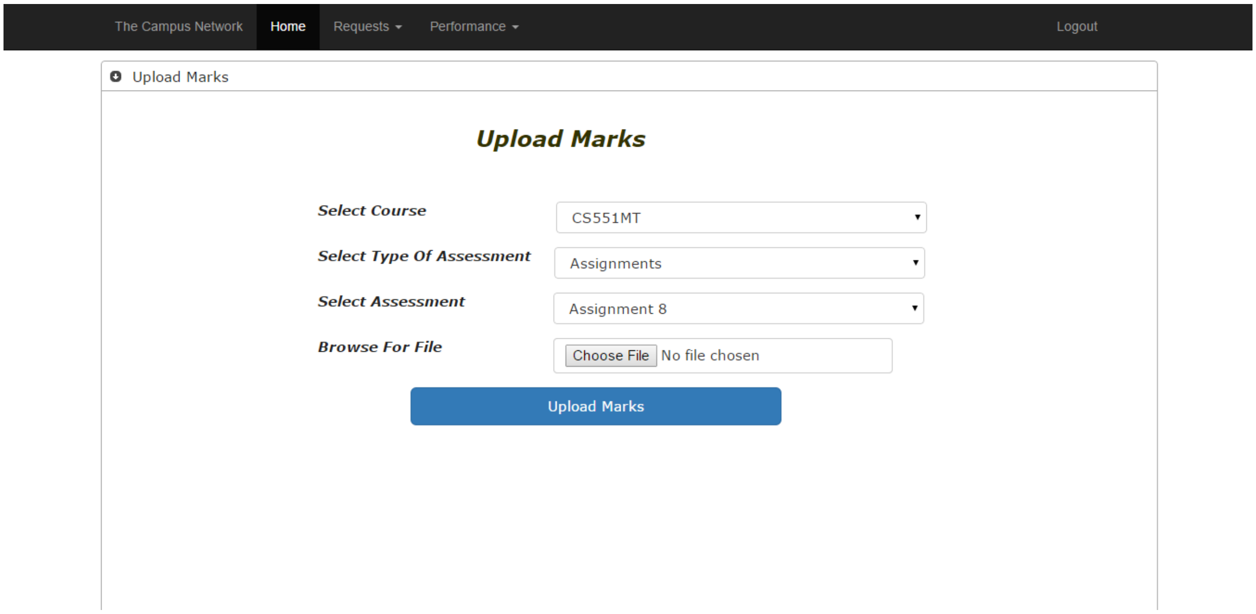
CourseId	CourseName	StartDate	EndDate	Credits	
CS551MT	Software Meathods and Tools	1/20/2015	5/20/2015	3	Get Schedule
CS551NA	Network Architecture	1/20/2015	5/20/2015	3	Get Schedule

Course Schedule For : CS551MT

Date	Topic	Assignment	AssignmentDeadline
1/20/2015	Course Introduction	No Silver Bullet	1/26/2015
1/22/2015	Software Development Process and Activities		
1/27/2015	Lab 1	Assignment 1	2/2/2015
1/29/2015	UML Modelling 1		
2/3/2015	Lab 2	Assignment 2	2/9/2015
2/5/2015	UML Modelling 2		

Fig 10: Assigned courses for Instructor

Here Instructor can see what all courses assigned for him and details of those courses

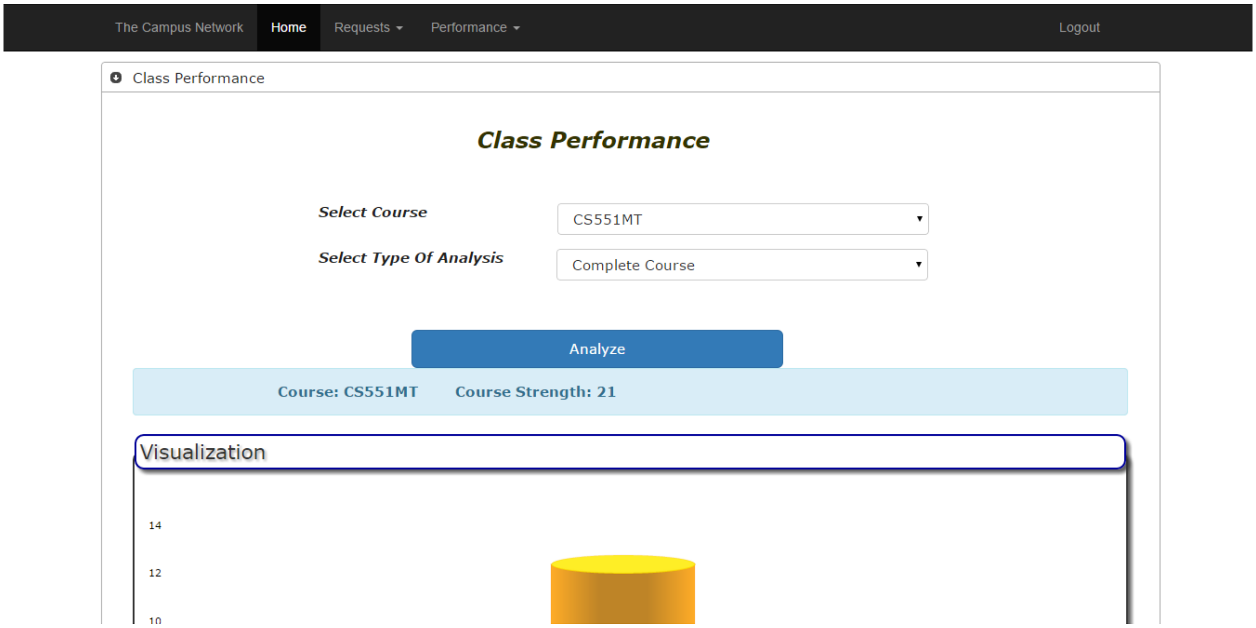


Upload Marks

Select Course	CS551MT
Select Type Of Assessment	Assignments
Select Assessment	Assignment 8
Browse For File	Choose File No file chosen

Fig 11: Uploading marks

Here Instructor can upload student’s marks



Class Performance

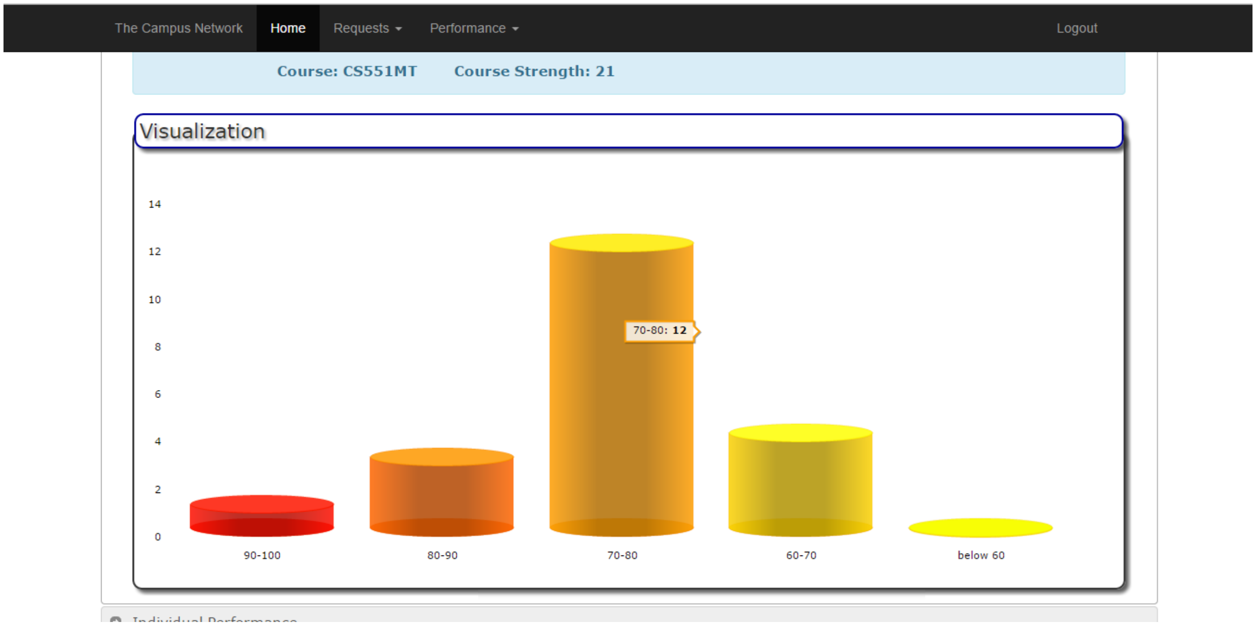
Select Course: CS551MT

Select Type Of Analysis: Complete Course

Course: CS551MT Course Strength: 21

Fig 12: Seeing class performance

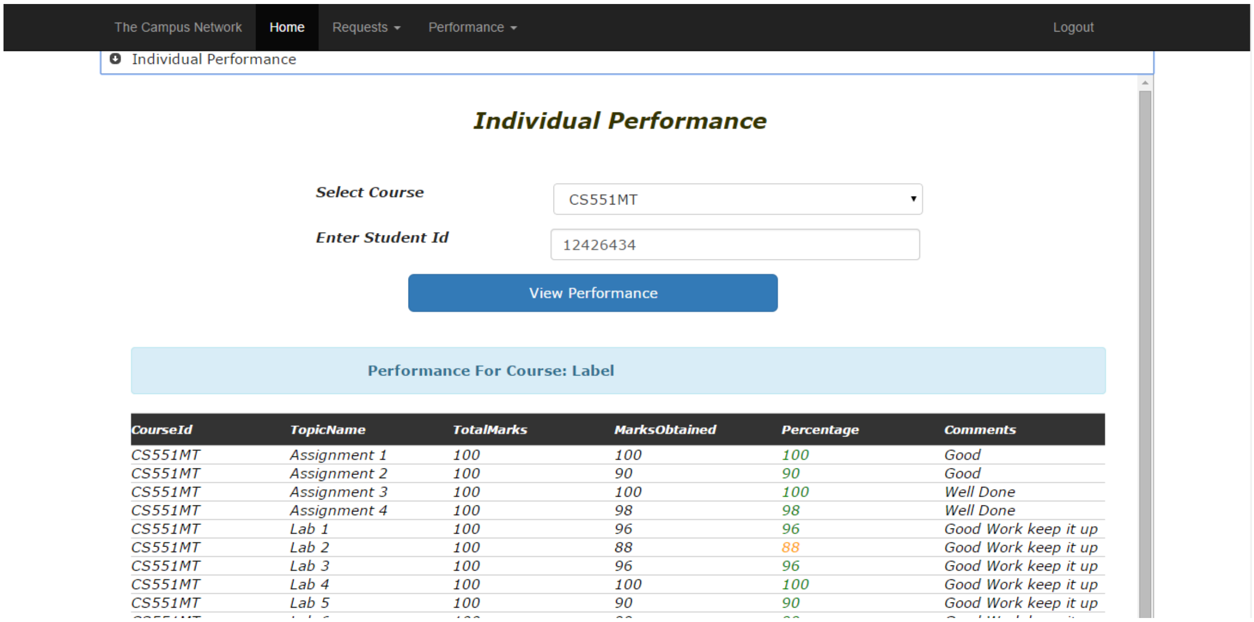
Here Instructor can see the performance of class



Course: CS551MT Course Strength: 21

Fig 13: class performance

Class performance is graphically represented



Individual Performance

Select Course: CS551MT

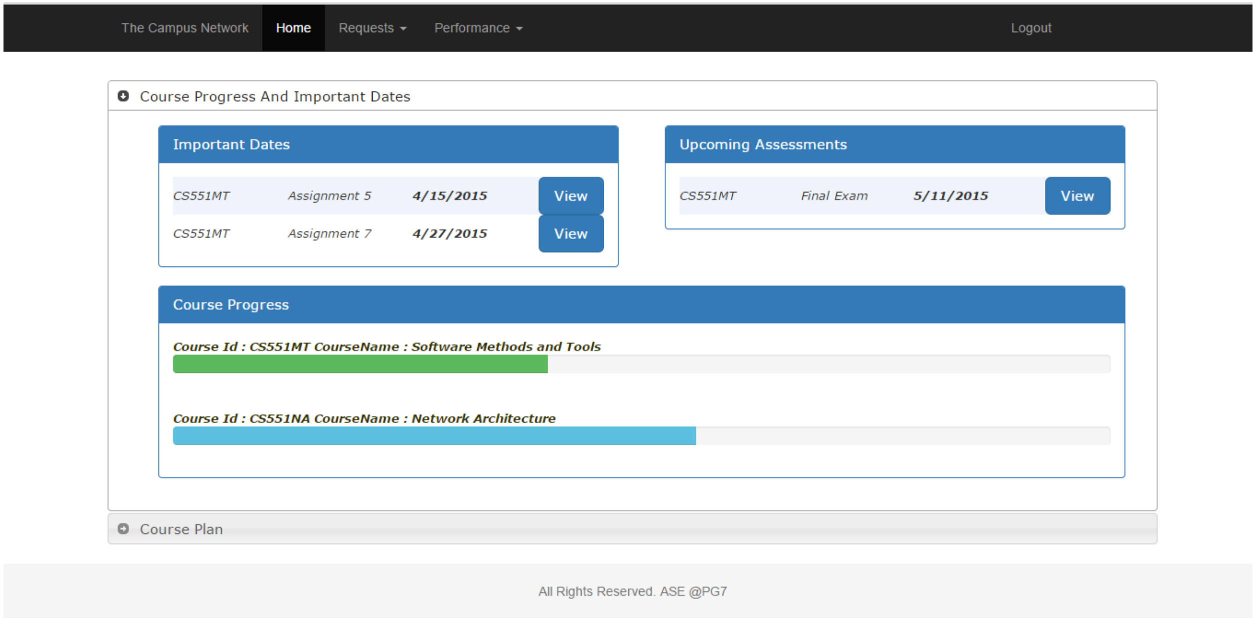
Enter Student Id: 12426434

Performance For Course: Label

CourseId	TopicName	TotalMarks	MarksObtained	Percentage	Comments
CS551MT	Assignment 1	100	100	100	Good
CS551MT	Assignment 2	100	90	90	Good
CS551MT	Assignment 3	100	100	100	Well Done
CS551MT	Assignment 4	100	98	98	Well Done
CS551MT	Lab 1	100	96	96	Good Work keep it up
CS551MT	Lab 2	100	88	88	Good Work keep it up
CS551MT	Lab 3	100	96	96	Good Work keep it up
CS551MT	Lab 4	100	100	100	Good Work keep it up
CS551MT	Lab 5	100	90	90	Good Work keep it up

Fig 14: check individual performance

Instructor can see individual student performance



Important Dates			
CS551MT	Assignment 5	4/15/2015	View
CS551MT	Assignment 7	4/27/2015	View

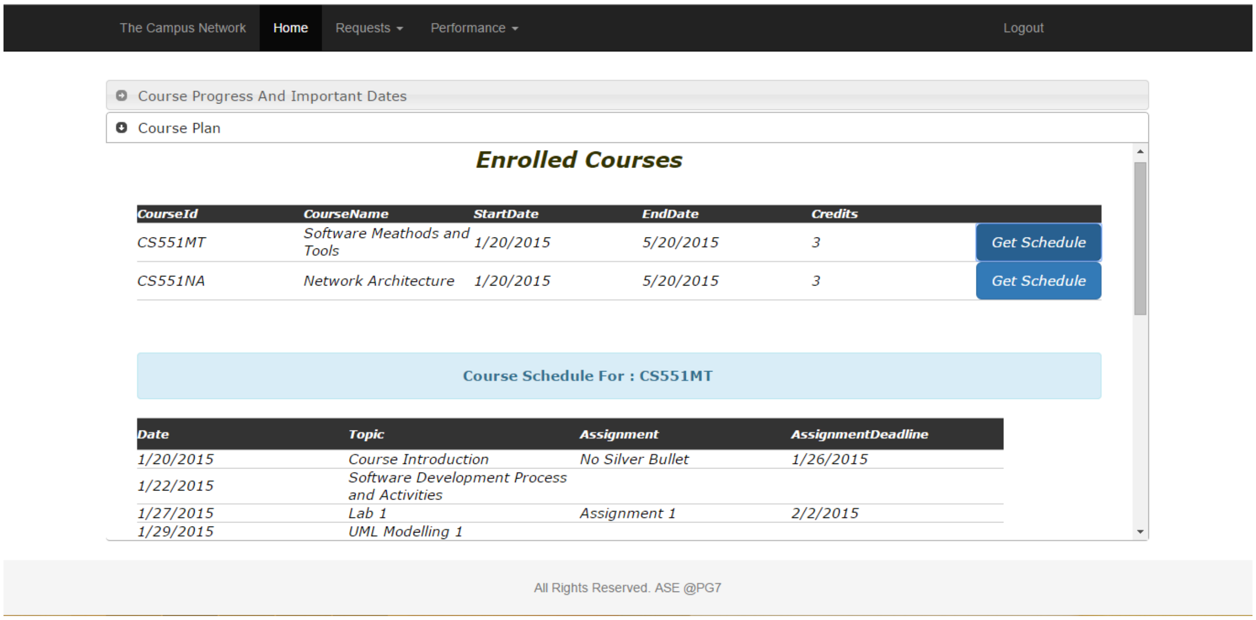
Upcoming Assessments			
CS551MT	Final Exam	5/11/2015	View

Course Id : CS551MT CourseName : Software Methods and Tools

Course Id : CS551NA CourseName : Network Architecture

Fig 15: Student's view of course progress

This is the page showing course progress and all important assessments for students



Enrolled Courses

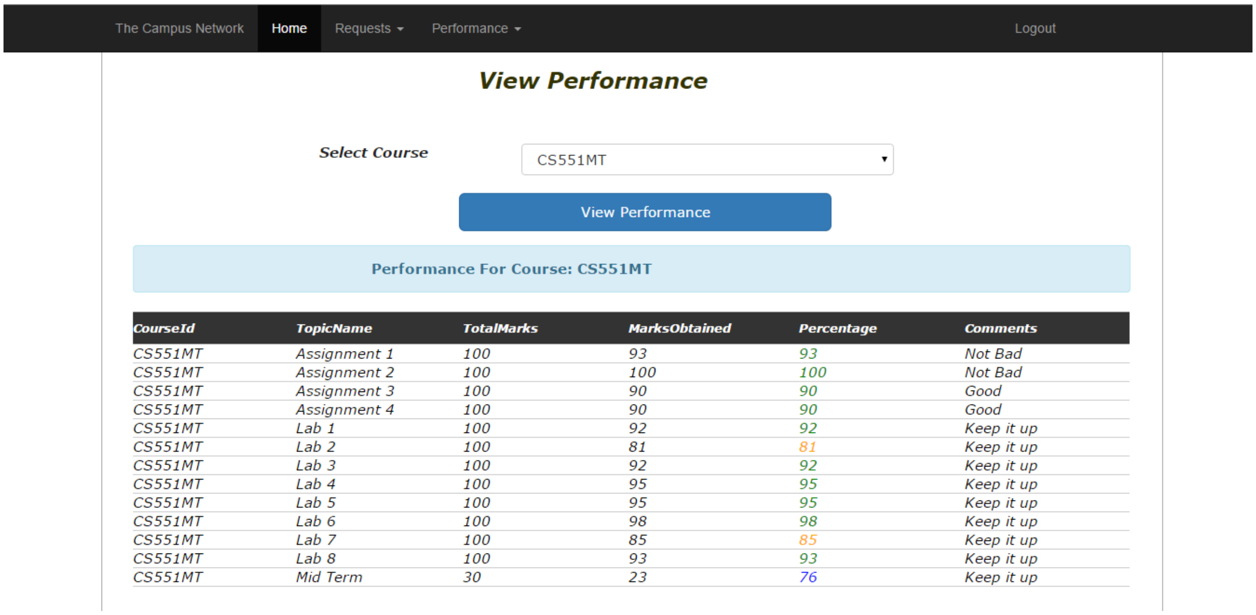
CourseId	CourseName	StartDate	EndDate	Credits	
CS551MT	Software Meathods and Tools	1/20/2015	5/20/2015	3	Get Schedule
CS551NA	Network Architecture	1/20/2015	5/20/2015	3	Get Schedule

Course Schedule For : CS551MT

Date	Topic	Assignment	AssignmentDeadline
1/20/2015	Course Introduction	No Silver Bullet	1/26/2015
1/22/2015	Software Development Process and Activities		
1/27/2015	Lab 1	Assignment 1	2/2/2015
1/29/2015	UML Modelling 1		

Fig 16: Student's view of his enrolled courses

Student viewing what courses he enrolled and can get schedule for his course



View Performance

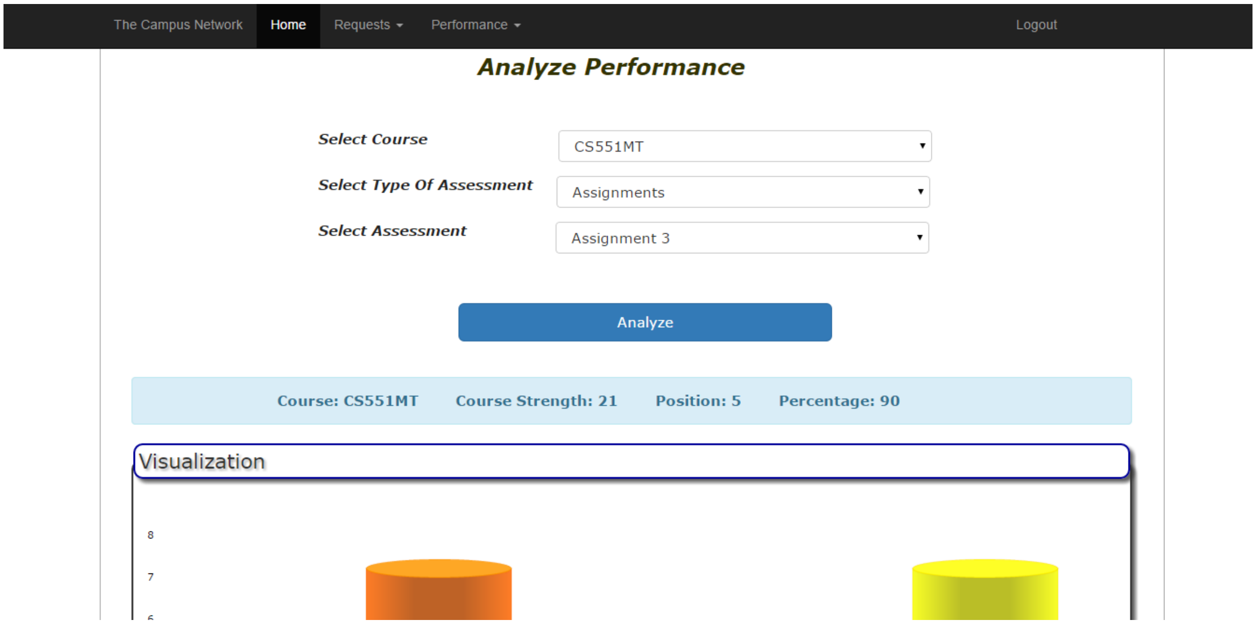
Select Course: CS551MT

Performance For Course: CS551MT

CourseId	TopicName	TotalMarks	MarksObtained	Percentage	Comments
CS551MT	Assignment 1	100	93	93	Not Bad
CS551MT	Assignment 2	100	100	100	Not Bad
CS551MT	Assignment 3	100	90	90	Good
CS551MT	Assignment 4	100	90	90	Good
CS551MT	Lab 1	100	92	92	Keep it up
CS551MT	Lab 2	100	81	81	Keep it up
CS551MT	Lab 3	100	92	92	Keep it up
CS551MT	Lab 4	100	95	95	Keep it up
CS551MT	Lab 5	100	95	95	Keep it up
CS551MT	Lab 6	100	98	98	Keep it up
CS551MT	Lab 7	100	85	85	Keep it up
CS551MT	Lab 8	100	93	93	Keep it up
CS551MT	Mid Term	30	23	76	Keep it up

Fig 17: Student can view his performance

Student can check all his scores and respective comments by Instructor



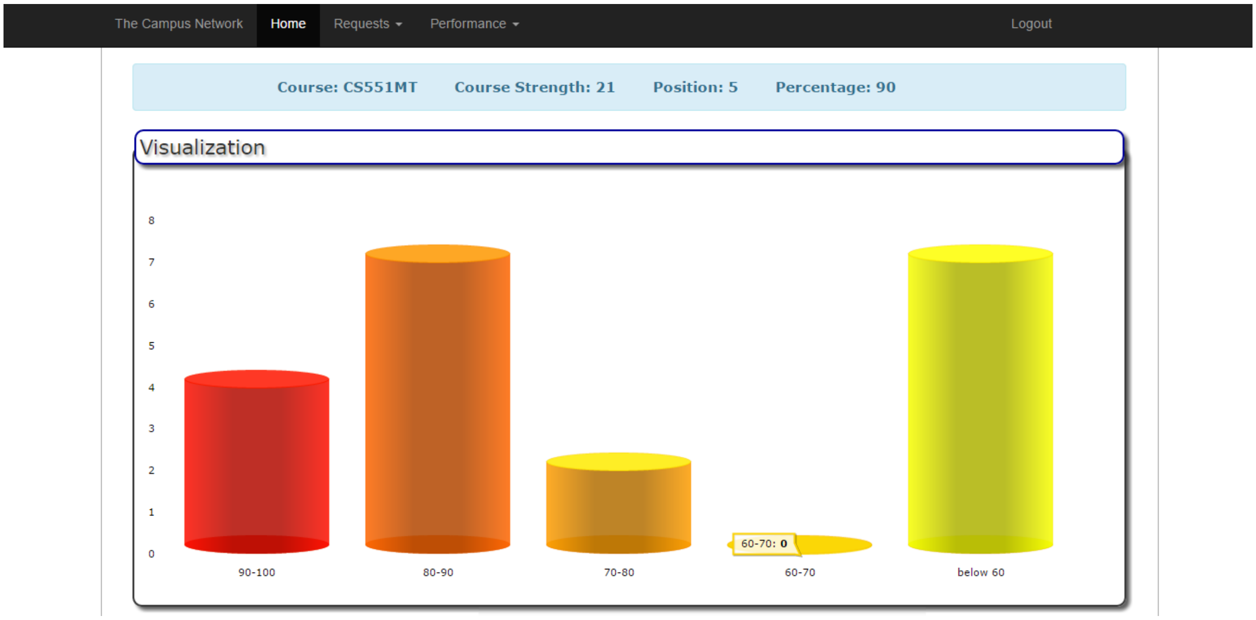
Analyze Performance

Select Course	CS551MT
Select Type Of Assessment	Assignments
Select Assessment	Assignment 3

Course: CS551MT Course Strength: 21 Position: 5 Percentage: 90

Fig 18: Student can analyze his performance

Student can analyze his performance in class and he can check his position in particular class



Course: CS551MT Course Strength: 21 Position: 5 Percentage: 90

Fig 19: Graphical representation of student’s performance

This is the graphical representation of all students’ performance

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

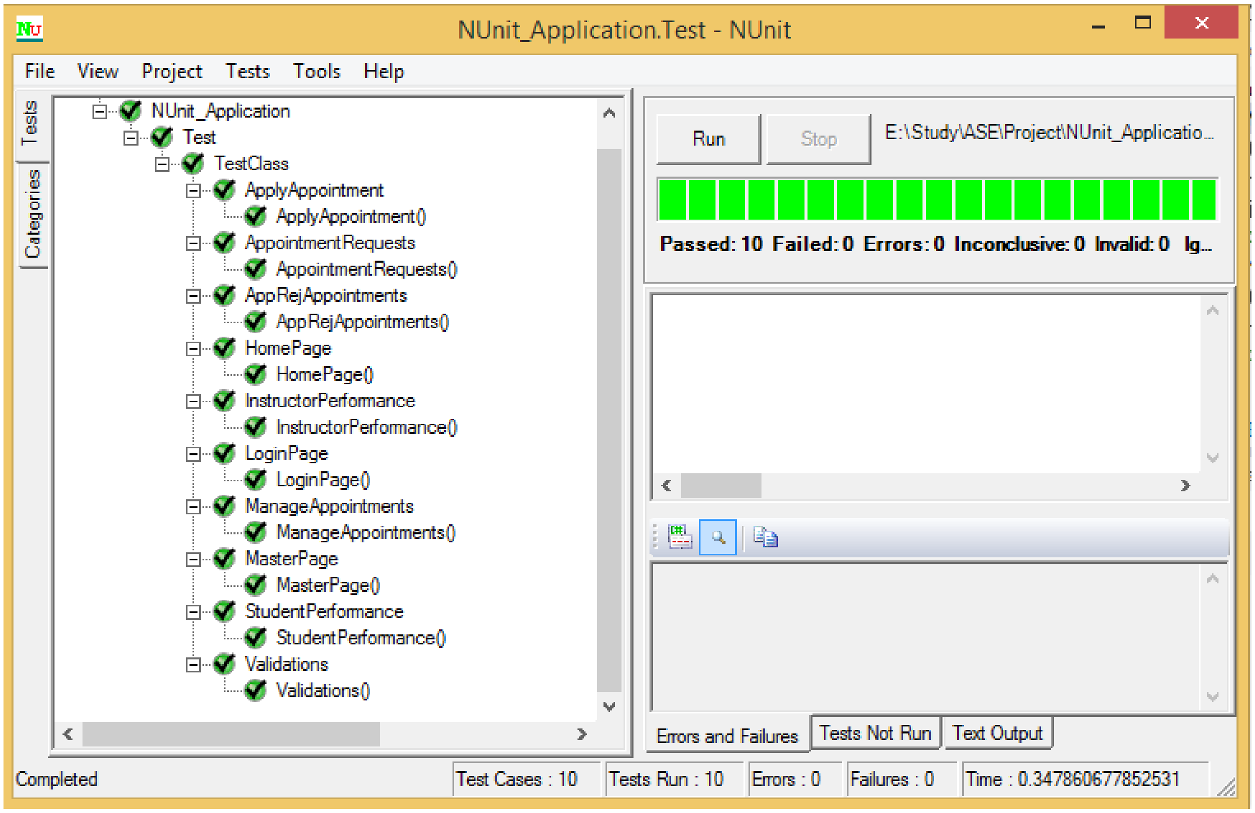
- **InstructorPerformance()**

  This function tests for all the validations in performance module on the instructor login.

- **AppoinmentRequest()**

  This function tests for all the validations in th performance module on the student module.

Screen shot for NUnit Testing:



```
NUnit_Application
  Test
    TestClass
      ApplyAppointment
        ApplyAppointment()
      AppointmentRequests
        AppointmentRequests()
      AppRejAppointments
        AppRejAppointments()
      HomePage
        HomePage()
      InstructorPerformance
        InstructorPerformance()
      LoginPage
        LoginPage()
      ManageAppointments
        ManageAppointments()
      MasterPage
        MasterPage()
      StudentPerformance
        StudentPerformance()
      Validations
        Validations()
```

```
E:\Study\ASE\Project\NUnit_Applicatio...
Passed: 10  Failed: 0  Errors: 0  Inconclusive: 0  Invalid: 0  Ig...
```

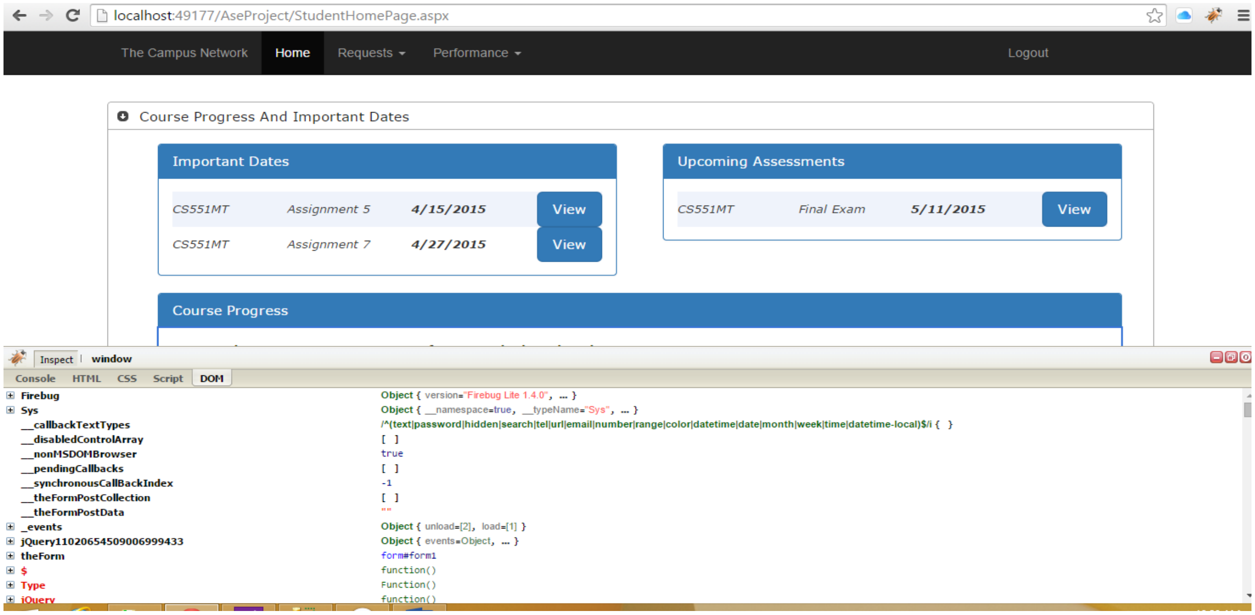
```
Completed    Test Cases : 10    Tests Run : 10    Errors : 0    Failures : 0    Time : 0.347860677852531
```

Implementation of Test Cases

VII. PERFORMANCE TESTING

- **Firebug Testing:**
  It is a web browser extension that will facilitate the live debugging and also monitoring of any websites

We have tested student home page and apply appointments page using firebug for this Increment.



Course Progress And Important Dates

Important Dates

CS551MT	Assignment 5	4/15/2015	View
CS551MT	Assignment 7	4/27/2015	View

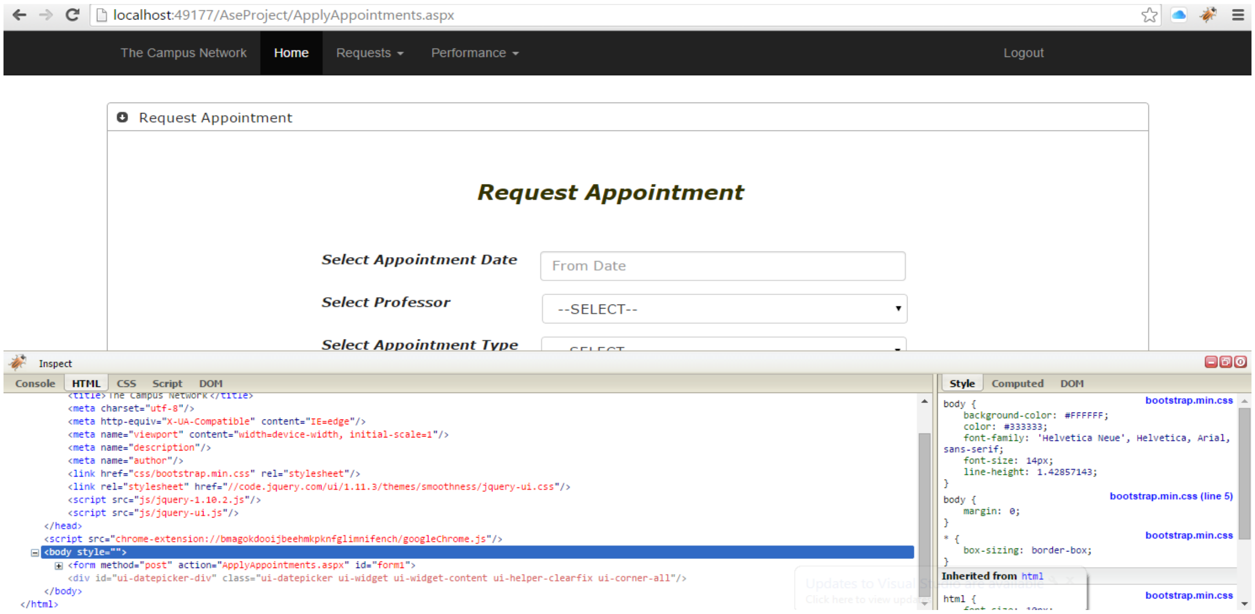
Upcoming Assessments

CS551MT	Final Exam	5/11/2015	View

Course Progress

```
⊞ Firebug                                Object { version="Firebug Lite 1.4.0", … }
⊞ Sys                                    Object { __namespace=true, __typeName="Sys", … }
  __callbackTextTypes                    /^(text|password|hidden|search|tel|url|email|number|range|color|datetime|date|month|week|time|datetime-local)$/i {  }
  __disabledControlArray                 [ ]
  __nonMSDOMBrowser                      true
  __pendingCallbacks                     [ ]
  __synchronousCallBackIndex             -1
  __theFormPostCollection                [ ]
  __theFormPostData                      ""
⊞ _events                                Object { unload=[2], load=[1] }
⊞ jQuery1102065450900699943 3            Object { events=Object, … }
⊞ theForm                                form#form1
⊞ $                                      function()
⊞ Type                                   Function()
⊞ jQuery                                 function()
```

Firebug Testing for Student Home Page



Request Appointment

Request Appointment

Select Appointment Date From Date

Select Professor --SELECT--

Select Appointment Type --SELECT--

```
      <title>The Campus Network</title>
      <meta charset="utf-8"/>
      <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
      <meta name="viewport" content="width=device-width, initial-scale=1"/>
      <meta name="description"/>
      <meta name="author"/>
      <link href="css/bootstrap.min.css" rel="stylesheet"/>
      <link rel="stylesheet" href="//code.jquery.com/ui/1.11.3/themes/smoothness/jquery-ui.css"/>
      <script src="js/jquery-1.10.2.js"/>
      <script src="js/jquery-ui.js"/>
   </head>
   <script src="chrome-extension://bmagokdooijbeehmkpknfglimnifench/googleChrome.js"/>
 ⊟ <body style="">
    ⊞ <form method="post" action="ApplyAppointments.aspx" id="form1">
      <div id="ui-datepicker-div" class="ui-datepicker ui-widget ui-widget-content ui-helper-clearfix ui-corner-all"/>
   </body>
</html>
```

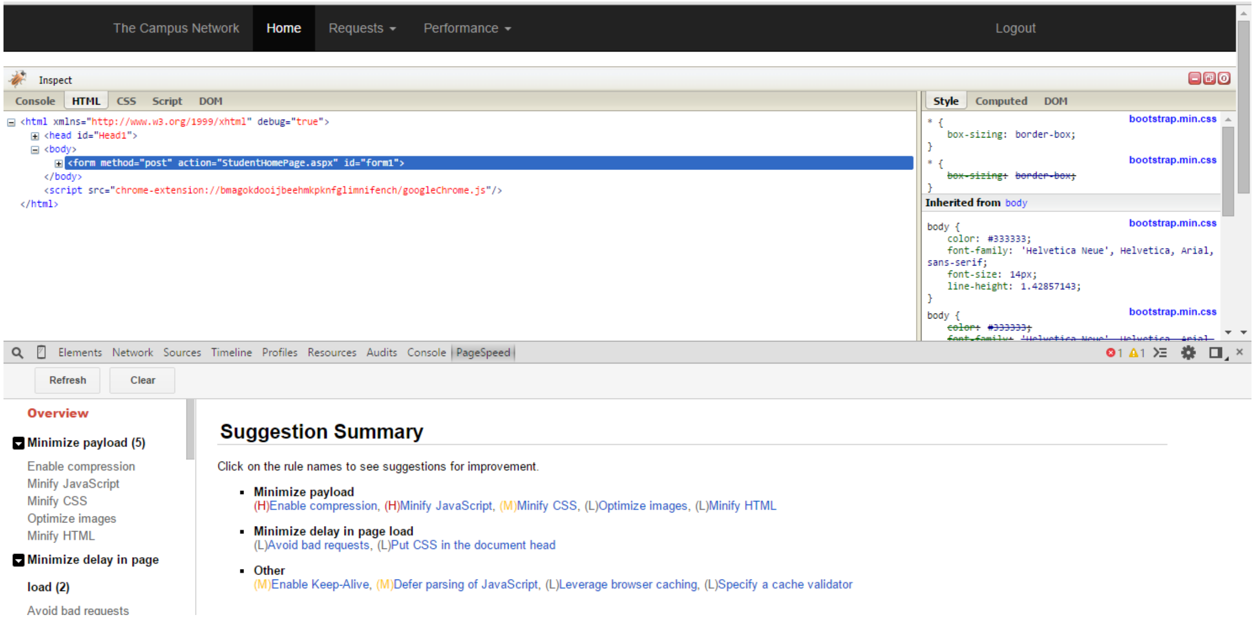
```
body {                                          bootstrap.min.css
    background-color: #FFFFFF;
    color: #333333;
    font-family: 'Helvetica Neue', Helvetica, Arial,
sans-serif;
    font-size: 14px;
    line-height: 1.42857143;
}
body {                                bootstrap.min.css (line 5)
    margin: 0;
}
* {                                             bootstrap.min.css
    box-sizing: border-box;
}
Inherited from html
html {                                          bootstrap.min.css
```

Firebug Testing for Apply Appointments Page

- **Page Speed Insights:**

  It is now available as a google chrome extension to test the load time of the page and also it gives the suggestions for the page so that it can be made faster.

  We have tested Student Home Page and Apply Appointments page using Page Speed Insights to analyze the speed of the pages and get some suggestions



```
<html xmlns="http://www.w3.org/1999/xhtml" debug="true">
  <head id="Head1">
  <body>
    <form method="post" action="StudentHomePage.aspx" id="form1">
  </body>
  <script src="chrome-extension://bmagokdooijbeehmkpknfglimnifench/googleChrome.js"/>
</html>
```

```
* {                                         bootstrap.min.css
    box-sizing: border-box;
}
* {                                         bootstrap.min.css
    ~~box-sizing: border-box;~~
}
Inherited from body
body {                                      bootstrap.min.css
    color: #333333;
    font-family: 'Helvetica Neue', Helvetica, Arial,
sans-serif;
    font-size: 14px;
    line-height: 1.42857143;
}
body {                                      bootstrap.min.css
    ~~color: #333333;~~
```

Overview

Minimize payload (5)

Enable compression
Minify JavaScript
Minify CSS
Optimize images
Minify HTML

Minimize delay in page load (2)

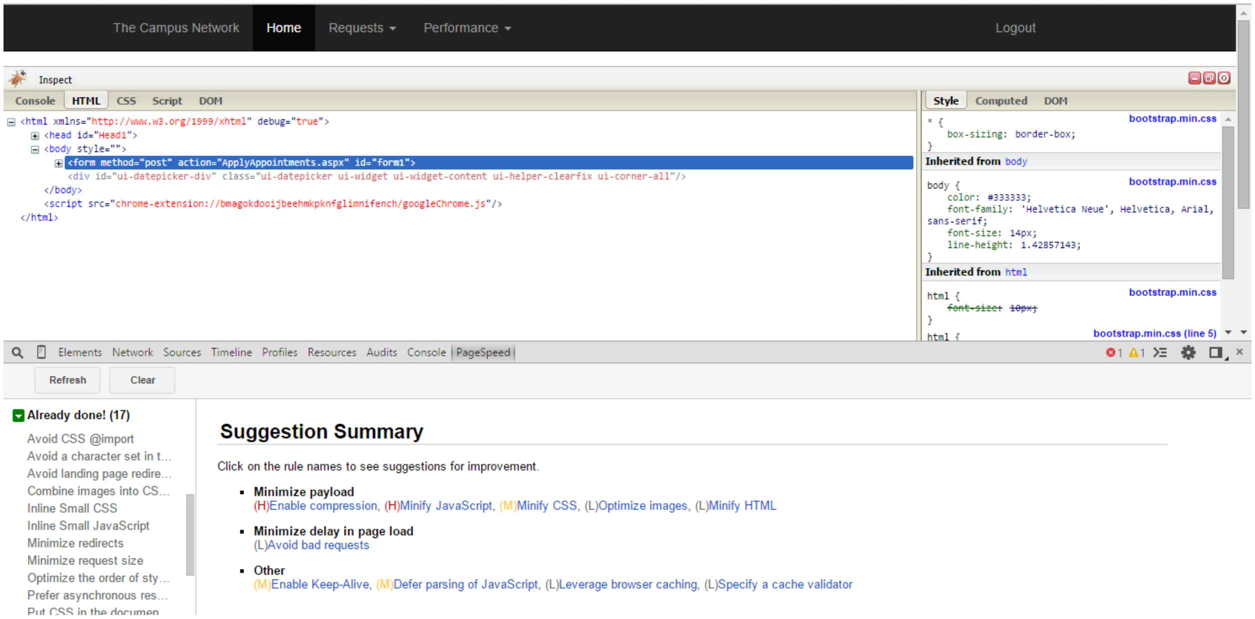
Avoid bad requests

Suggestion Summary

Click on the rule names to see suggestions for improvement.

- **Minimize payload**
  (H)Enable compression, (H)Minify JavaScript, (M)Minify CSS, (L)Optimize images, (L)Minify HTML
- **Minimize delay in page load**
  (L)Avoid bad requests, (L)Put CSS in the document head
- **Other**
  (M)Enable Keep-Alive, (M)Defer parsing of JavaScript, (L)Leverage browser caching, (L)Specify a cache validator

Page Speed Insights for Student Home Page



```
<html xmlns="http://www.w3.org/1999/xhtml" debug="true">
  <head id="Head1">
  <body style="">
    <form method="post" action="ApplyAppointments.aspx" id="form1">
    <div id="ui-datepicker-div" class="ui-datepicker ui-widget ui-widget-content ui-helper-clearfix ui-corner-all"/>
  </body>
  <script src="chrome-extension://bmagokdooijbeehmkpknfglimnifench/googleChrome.js"/>
</html>
```

```
* {                                              bootstrap.min.css
    box-sizing: border-box;
}
Inherited from body
body {                                           bootstrap.min.css
    color: #333333;
    font-family: 'Helvetica Neue', Helvetica, Arial,
sans-serif;
    font-size: 14px;
    line-height: 1.42857143;
}
Inherited from html
html {                                           bootstrap.min.css
    ~~font-size: 10px;~~
}
html {                                 bootstrap.min.css (line 5)
```

Already done! (17)

Avoid CSS @import
Avoid a character set in t…
Avoid landing page redire…
Combine images into CS…
Inline Small CSS
Inline Small JavaScript
Minimize redirects
Minimize request size
Optimize the order of sty…
Prefer asynchronous res…
Put CSS in the documen

Suggestion Summary

Click on the rule names to see suggestions for improvement.

- **Minimize payload**
  (H)Enable compression, (H)Minify JavaScript, (M)Minify CSS, (L)Optimize images, (L)Minify HTML
- **Minimize delay in page load**
  (L)Avoid bad requests
- **Other**
  (M)Enable Keep-Alive, (M)Defer parsing of JavaScript, (L)Leverage browser caching, (L)Specify a cache validator

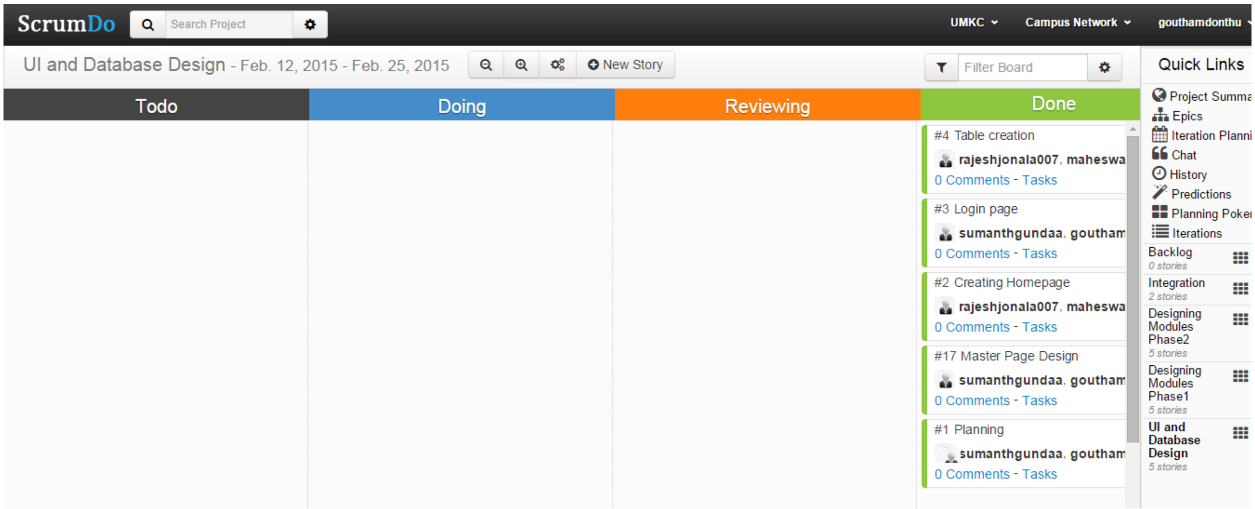
Page Speed Insights for Apply Appointments page

VIII. DEPLOYMENT

Project Scrum Do Link:

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

Screenshots:



UI and Database Design - Feb. 12, 2015 - Feb. 25, 2015

#4 Table creation
rajeshjonala007, maheswa
0 Comments - Tasks

#3 Login page
sumanthgundaa, goutham
0 Comments - Tasks

#2 Creating Homepage
rajeshjonala007, maheswa
0 Comments - Tasks

#17 Master Page Design
sumanthgundaa, goutham
0 Comments - Tasks

#1 Planning
sumanthgundaa, goutham
0 Comments - Tasks

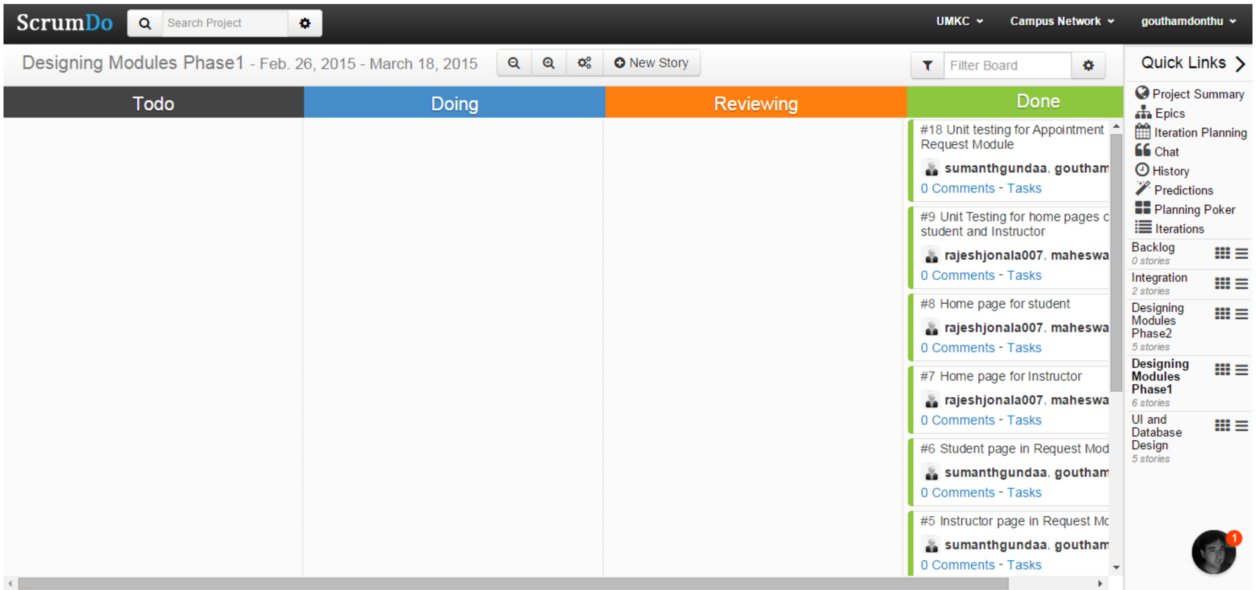
Backlog
0 stories

Integration
2 stories

Designing Modules Phase2
5 stories

Designing Modules Phase1
5 stories

UI and Database Design
5 stories



Designing Modules Phase1 - Feb. 26, 2015 - March 18, 2015

#18 Unit testing for Appointment Request Module
sumanthgundaa, goutham
0 Comments - Tasks

#9 Unit Testing for home pages c student and Instructor
rajeshjonala007, maheswa
0 Comments - Tasks

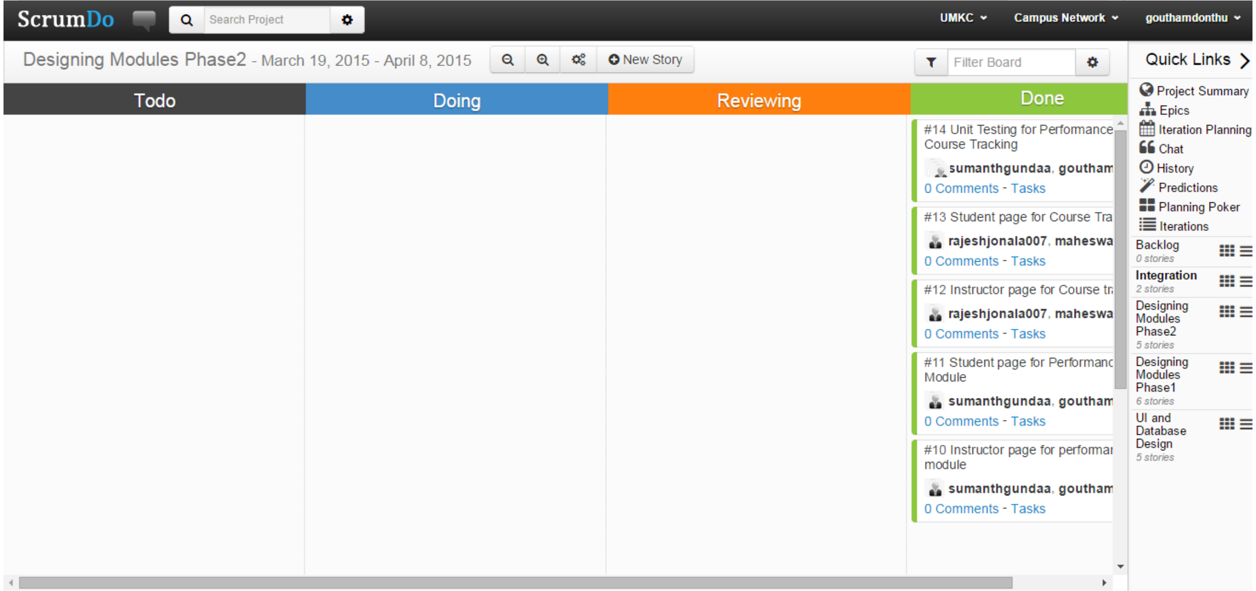
#8 Home page for student
rajeshjonala007, maheswa
0 Comments - Tasks

#7 Home page for Instructor
rajeshjonala007, maheswa
0 Comments - Tasks

#6 Student page in Request Mod
sumanthgundaa, goutham
0 Comments - Tasks

#5 Instructor page in Request Mc
sumanthgundaa, goutham
0 Comments - Tasks

Backlog 0 stories
Integration 2 stories
Designing Modules Phase2 5 stories
Designing Modules Phase1 6 stories
UI and Database Design 5 stories



Designing Modules Phase2 - March 19, 2015 - April 8, 2015

#14 Unit Testing for Performance Course Tracking
sumanthgundaa, goutham
0 Comments - Tasks

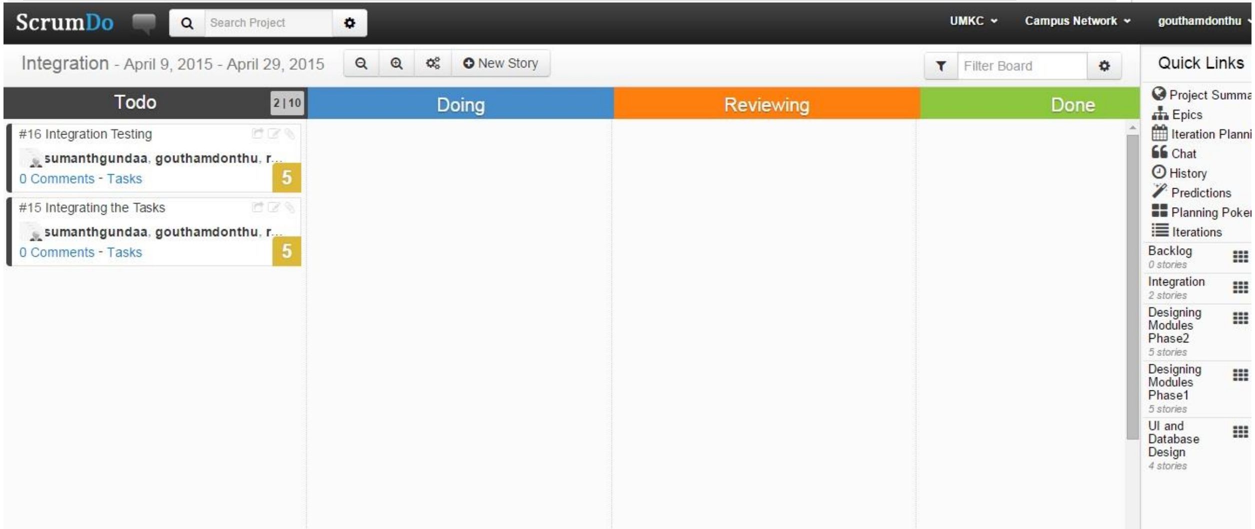
#13 Student page for Course Tra
rajeshjonala007, maheswa
0 Comments - Tasks

#12 Instructor page for Course tr
rajeshjonala007, maheswa
0 Comments - Tasks

#11 Student page for Performanc Module
sumanthgundaa, goutham
0 Comments - Tasks

#10 Instructor page for performar module
sumanthgundaa, goutham
0 Comments - Tasks

Backlog 0 stories
Integration 2 stories
Designing Modules Phase2 5 stories
Designing Modules Phase1 6 stories
UI and Database Design 5 stories



Integration - April 9, 2015 - April 29, 2015

#16 Integration Testing
sumanthgundaa, gouthamdonthu, r...
0 Comments - Tasks
5

#15 Integrating the Tasks
sumanthgundaa, gouthamdonthu, r...
0 Comments - Tasks
5

Backlog 0 stories
Integration 2 stories
Designing Modules Phase2 5 stories
Designing Modules Phase1 5 stories
UI and Database Design 4 stories

GitHub Site URL:

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

IX. 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 11: 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

Story 12: 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 13: Instructor page for performance module

Description: Here Instructor can track all the students performance and he can generate report of all the students

Responsibility: Instructor can track all the students' performance. We are providing data through excel sheet.

Time Taken: 15 hours

Contributions: Sumanth and Goutham

Story 14: Student page for Performance Module

Description: Here student can view his test scores and he can view his position in the class

Responsibility: Students score can be viewed by him and they are graphically represented

Time Taken: 15hours

Contributions: Sumanth and Goutham

Story 15: Instructor page for Course tracking

Description: Instructor can track the status of the course and he can generate a warning message to all the students about any assignment or project deadlines

Responsibility: Instructor can see how much course has been completed and he can see curriculum of course

Time Taken: 13hours

Contributions: Mahesh and Rajesh

Story 16: Student page for Course Tracking

Description: Here student can track the status of the course and they were eventually notified about their tasks

Responsibility: Student can see how much course has been completed and he can see curriculum of course

Time Taken: 16hours

Contributions: Mahesh and Rajesh

Story 17: Unit Testing for Performance and Course Tracking

Description: Here we test the modules independently 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 performance module. Our goal is to run this module without any bugs

Time Taken: 10hours

Contributions: Sumanth and Goutham

Work to be completed:

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

- 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 this 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.