Senior Design Project Report

CSE/EEE/ETE 499A

Online Classroom Management System

Agent

Submitted By

1311316042 - Nabila Ahmed

1320116040 - Ahnaf Tahmid Chowdhury

1230843042 - Mohammad Lummim Sarker

Supervisor

Mohammad Rezaul Islam – IMR

Lecturer, Department of Electrical and Computer Engineering

ELECTRICAL AND COMPUTER ENGINEERING

NORTH SOUTH UNIVERSITY

SUMMER 2017

Agreement Form

We take great pleasure in submitting our senior design project report on Agent. This report is prepared as a requirement of the Capstone Design Project CSE/EEE/ETE 499 A & B which is a two semester long senior design course. This course involves multidisciplinary teams of students who build and test custom designed systems, components or engineering processes. We would like to request you to accept this report as a partial fulfillment of Bachelor of Science degree under Electrical and Computer Engineering Department of North South University.

Declared By:

……………………………………………
Name: Nabila Ahmed
ID: 1311316042

……………………………………………
Name: Ahnaf Tahmid Chowdhury
ID: 1320116040

……………………………………………
Name: Mohammad Lummim Sarker
ID: 1230843042

Approved By:

……………………………

Supervisor
Mohammad Rezaul Islam
Lecturer, Department of Electrical and Computer Engineering
North South University, Dhaka, Bangladesh

....

Dr. Rezaul Bari
Chairman, Department of Electrical and Computer Engineering
North South University, Dhaka, Bangladesh

Agent

Agent is an Online Classroom Management System. It is a real time web application implemented with new technologies. Due to the new web technologies, the application is fast and efficient. Agent has brought together most of the functionalities the Engrade, Piazza and Google Classroom have. Other than that it also introduced some new features that will make teacher's and student's life easier. As the new features, it has grouping system inside a class, submission window with the end time and an advanced marksheet system. Most important thing is if it goes to the production level it will be open for all. User just needs to register to open an account that's it and the account will be completely free.

Table of Contents

List of Figures:

CHAPTER 1

INTRODUCTION

1.1 Project Details:

Classroom management system is a system which makes facultys' and students' university life easier and efficient. These systems usually help to improve interaction between teachers and students, keep students up to date with notices, marks and help faculties to share resources. Agent is also a classroom management system but with many other advanced facilities. It is a web application (client-server software application) that is stored on a remote server and delivered over the internet through a web interface. It uses database to store all the information about users (faculty and student). The database is stored at the server side, which runs on a particular port and users interact with the server through a web browser. From client side user sends a Hypertext Transfer Protocol (HTTP) request to the server side, the server side listens to the request and responds to the client with an Hypertext Markup Language (HTML) page which then gets rendered by the browser.

1.2 Background:

In the market, there are lots of classroom management and also 'gradebook' solutions such as Engrade, Piazza, Google Classroom etc. These software sets come up with what a teacher prefers for the students to manage the classroom activity. People pay and get the expensive use of these software to the upgraded level. While few free solutions do not even provide the up to the mark features for classroom management. Statistical data can not be gathered from those solutions and selection of class activities are not even dynamic. Few provides accept, store, manage, tract student data and their information but with a very limited use only. By looking all the web-based applications over the internet, we found that features are not extendable and not even dynamic. The aim and the outcome is to reduce effort in student management and save time by providing certain solutions on grading, grouping and managing the class.

1.3 Motivation:

Web-based applications are now a days very handier and accessible. So solving problems and managing data are very common for the industries and institutions. Considering on the point of a student from the North South University, it becomes very difficult to manage all the courses since they have to log into different applications to get connected with the courses. Also North South University is growing fast and with the era, it’s becoming more digital.

On the other side, the faculties of this university deal so many hassles in managing the student activity. No statistical data, number of students performing differently can not be viewed categorically. Noticing and publishing projects and assignments are through different procedures as it increase hassles and sometimes ambiguous. As the number of students in the university hits big, producing enormous problem in advising and seat allocation for the students of the university and creates lag for the students to graduate in time due to lack of seats in a classroom. Now North South has its own online advising system and online student attendance system, so we were thinking let’s make North South’s own classroom management system, which will decrease the gap between admin, faculty and students and also, focusing North South University, we also planned to develop a solution for seat allocation of a new course section opening.

1.4 Project Goal:

Agent has the following features:

- Faculties can share resources (slides, links), announcement and notices.
- Customize way for faculties to arrange exam’s marks.
- Faculties have the facility to choose best exams and assign weight.
- Faculty and student can see the class performance statistics in pie chart.
- Faculties can open submission window with an end time.
- Group management system inside a class.
- Student can see his/her mark of an exam with the highest mark.

- Students can submit the assignment through the system.
- Students can get faculties' office hour and can fix an appointment with the teacher.
- Students can also see faculty's background.
- There are also discussion part for both class and individual group.
- It is completely free of cost.
- Students can do preregistration.
- Chairman can add courses in preregistration.
- Chairman can view the details of a preregistration course.

CHAPTER 2

TECHNICAL DESIGN

2.1 Existing Solution:

There are many web-based classroom management systems such as Engrade, Piazza, Google Classroom, Turnitin, Learnboost, Gradebooster etc. Engrade, Piazza helps to post resources, notices, track, grades, display information to the students. These are for the non-premium users. For more information about the students activity, needs to pay over $100 for the advance features. Sharing notices and managing with simple features, the Google classroom managements system only shows information to the students from the instructor. Learnboost and Gradewizards are about grading and notice posting online. They do not offered premium edition as they are limited to their functionality. To share resources Thinkwave offers a free 25mb of data space and also for extra 100mb need to pay. When it comes to submitting assignments and projects, Turnitin roles a big role now a days but not managing the class activity so well. Verifying through email, hidden post in classes by the students, displaying information can be solved but not by the only one application alone. Users must go through many sites as premium non-premium or a even 30 day trial to avail all maximum features.

2.2 Proposed Solution:

Agent has three portals: faculty, student and chairman.The main idea is to exclude the admin portal so that the administration part is excluded from the educational part. So, we have focused only on the educational part of the system. Now, in chairman portal of the system, it has some extra feature like pre-registration.Other than that, a faculty is just a subset of the chairman portal.

In faculty portal, faculty can share his office hours and office room with students. She can add and delete courses in her course list. Along with each course, Agent provides a different portal to manage that single course separately. In that portal faculty can share notes, resources, notices and marks(individual, average, highest) with the students who are enrolled in that course. The most important part of the faculty portal is that it makes marks calculation automated. Also, faculties now can see each class performance in a graphical manner.

Student portal is fully associated and integrated with the faculty portal. A student can access notes, resources, notices and marks of the respective course. Student have also access for pre-registration. Students can register for their desired subject primarily so that chairman can decision based on the demand of a subject.

2.3 Solution Assessment:

Compared to other products, Agent has separate portals to manage different works. This makes the working environment more organized and more efficient. It is completely free and fast since it is made by modern technique. Faculty found most of the things automated, for example, mark calculation and adding the list of student names. Also, it’s always comfortable to work with some system which is solely focused on your workstation. As a chairman, the chairman has enough data to understand how much student is going to enroll in the coming semester. One of the best feature of agent is, it is user need oriented rather than software service oriented.

2.4 Design Alternative:

For designing Agent(the software), we had other options too. Here is a list of probable options we had for building this.

Rails(Ruby)
Spring(Java)
Laravel(PHP)

For backend we have chosen NodeJS over PHP, Ruby or Java. PHP/Ruby have a very rich library.Since, PHP/Ruby are interpreted language it will take more time and interfacing overhead. Now, NodeJS is an interpreted language too. But, it works on V8 engine which internally uses C++ as a core language. Now, for Java the main problem is concurrency. If we need threading, then in Java it takes 256KB of stack memory each time we run only one threading. But it doubles up the memory if we are using 64 bit operating system. In general, people use at least 5 - 6 threading per application. But, in NodeJS we can easily handle the situation by using asynchronus system call. For this reason, it runs blazingly fast.

For DOM manipulation and front end work we used Jquery and Ajax. We used Jquery for DOM manipulation and Ajax for real time rendering. We also used using Bootstrap a twitter made HTML/CSS/JS framework which had make things easier and elegant for the system.

2.5 Technical Design:

Technical design is the backbone of any project-based work. It is a logical design which is an important step before making more detailed architecture. It guides an implementation, injects technical leadership, discipline and craftsmanship to a solution assembly. In this design, engineers draw the base of the software and fix which language, framework and other important stuff they are gonna use in the project.

In our technical design, we decided to use **Model-View-Controller**(MVC) as the software architectural or design pattern. It divides a given application into three interconnected parts. This is done to separate internal representations of information from the ways information is presented to, and accepted from, the user.

- Model is a place to define data structures and methods to interact with the data store.
- View is a place to manage everything the end user sees on his or her screen.

- Controller is a place to take user requests, bring data from the model and pass it back to the view.

The Model-View-Controller pattern has proven to be one of the most effective patterns for web development due to its clear separation of the data, logic, and presentation layers. Figure-1 shows the modular view of MVC.

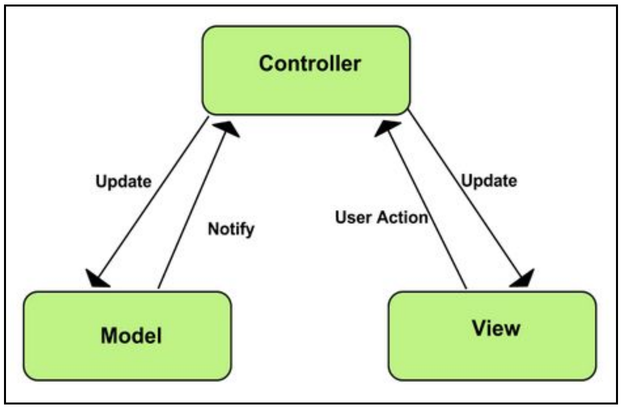


Figure-1: Model-View-Controller modular view

Since we already decided to use MVC design pattern, now to support it we had to select a framework that can easily give our design a tangible shape. Express is a framework that provides a boilerplate for MVC design. So we decided to use Express framework, which is one of the best frameworks for Node. It has great support and a bunch of helpful features.

So far we have discussed the core part of our technical design. Now let’s see what extra design we added for better user experience.

For better user experience we added an extra layer between view and controller. The layer is called javascript in form of Ajax. Due to Ajax, a web page can update without reloading the page. With Ajax, Web applications can send data to and retrieve from a server asynchronously (in the background) without interfering with the display and behavior of the existing page. Figure-2 shows the complete technical design of Agent.

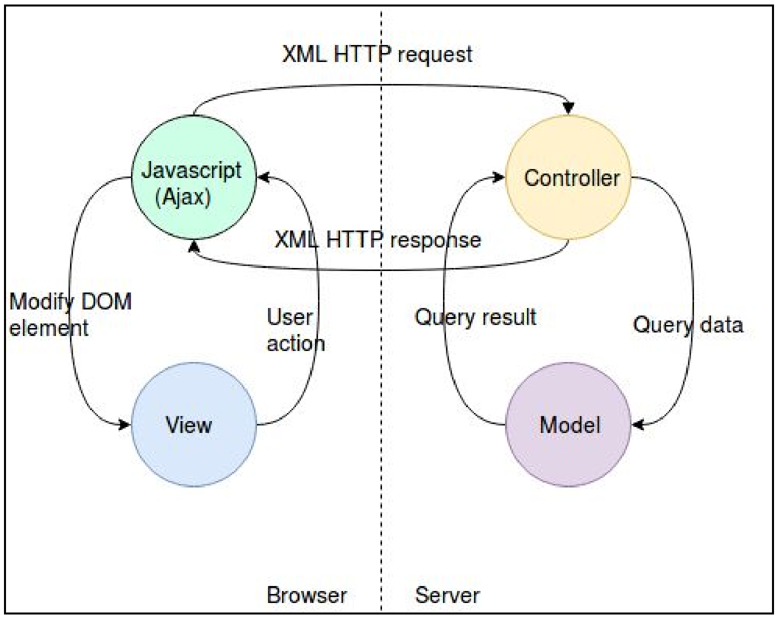


Figure-2: Technical design

As shown in the above figure, javascript (Ajax) performs the main connection part between browser and server in our application. When user performs an action like click a button, the action is caught by the Ajax. Ajax then sends XML HTTP request to the server. The controller in the server captures the request, performs necessary changes in the database and send the changes as a response to the Ajax. Ajax then modifies the DOM elements of the HTML page and on the browser, user can view the changes.

2.6 Required Skills

The following are the stacks we used to build the whole project.

- HTML, CSS, Bootstrap
- Javascript, JQuery, AJAX
- We used Pug as the template language
- Node.js, Express framework
- MongoDB

CHAPTER 3

Naming Conventions and Definitions

3.1 Definition of Key Terms:

While working we used a lot of conventions so that one new coder can easily pick up the system as soon as s/he starts to contribute to the system. We devide the whole thing into four parts

1. Model
2. Methods
3. Handler Functions
4. Template Variables

Database Schema:

CousrseModel - data structure for courses.

ExamModel - data structure for exams.

FileModel - data structure for files.

GroupModel - data structure for groups.

MarkSheetModel - data structure for a marksheets.

PreRegistrationModel - data structure for pre-registrations.

SubmissionModel - data structure for submissions.

UserModel - data structure for users.

Methods**:**

Get* - method for retrieving any object by it's ID.

Put - method for storing data in database.

List* - method for retrieving all documents of an object.

Handler Functions:

Handle* - Handles PUT requests.

Serve* - Handles GET requests.

***Values -** Stores values of a parsed form.

Template variables:

Tpl*Values - For passing values to a template.

Tpl* - For executing and persing templates, for serving a page.

CHAPTER 4

Product Use Cases

4.1 Use Case Diagram:

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. It is the simplest representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. Figure-3 shows the use case diagram of agent.

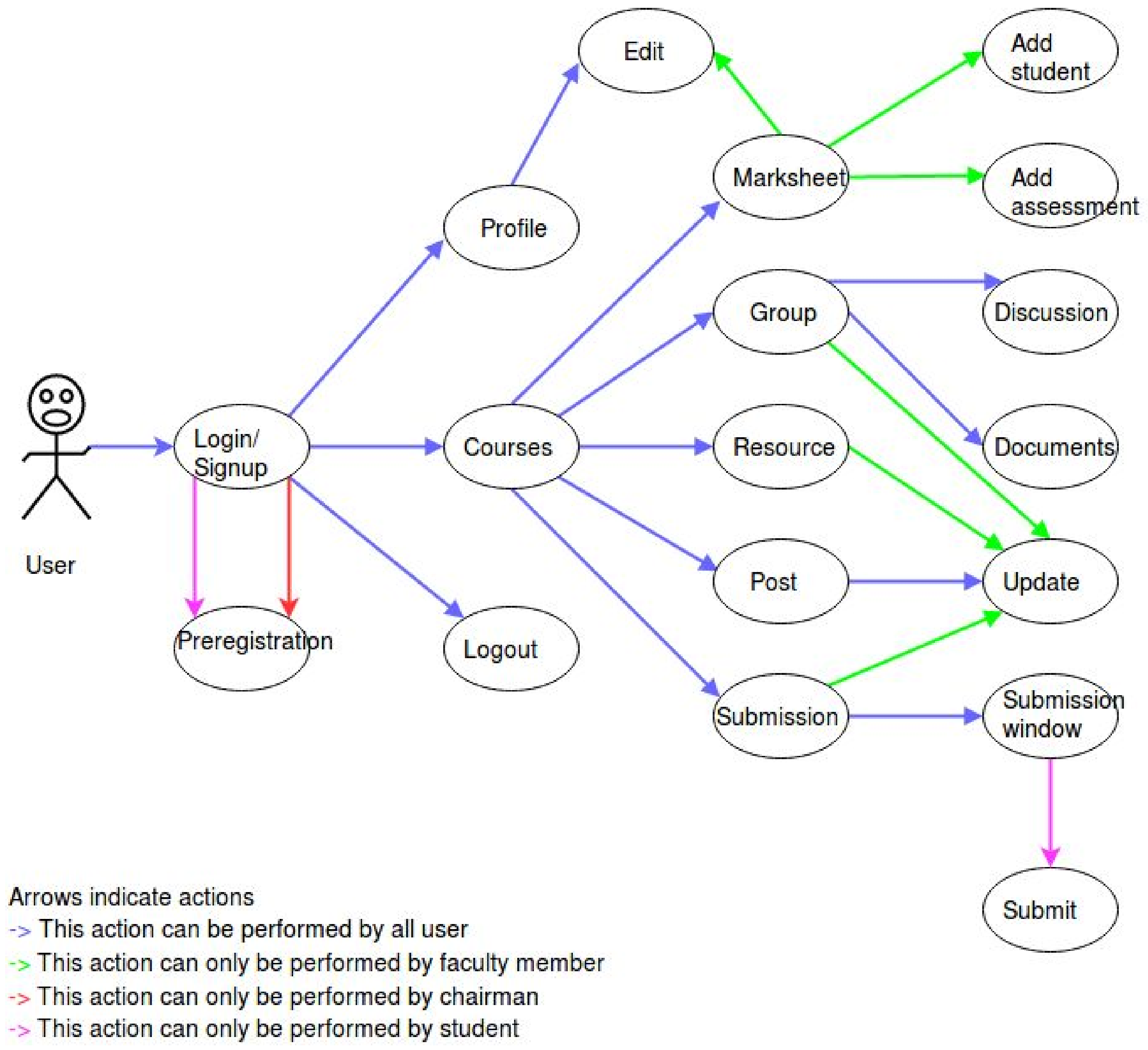
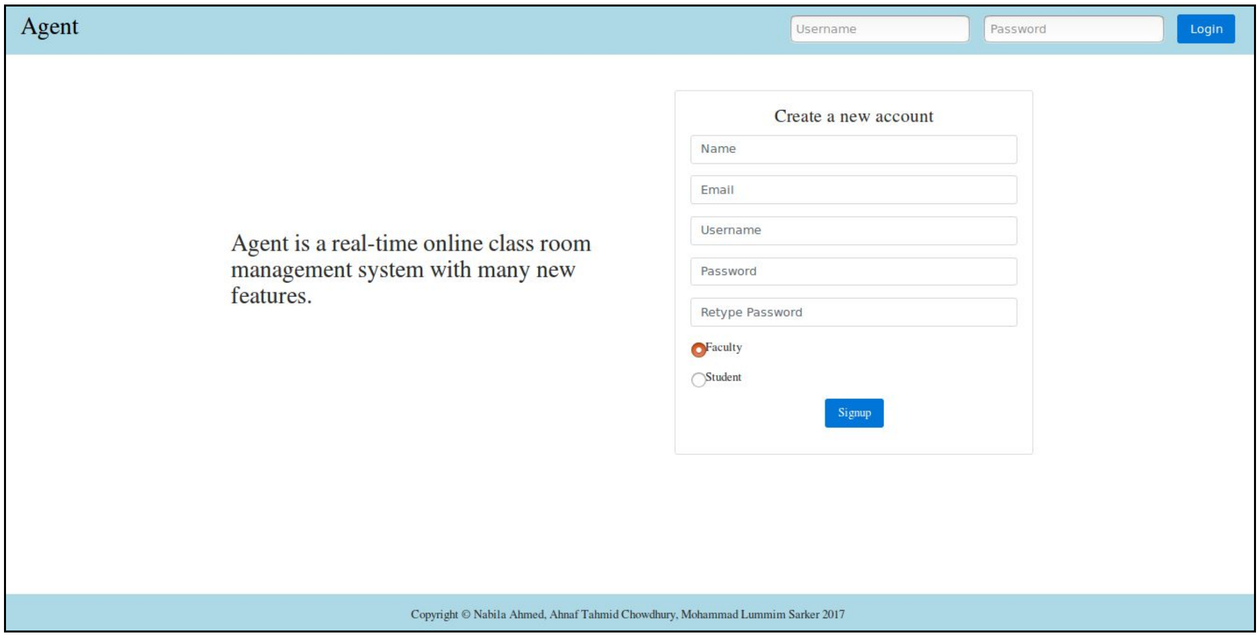

User
Login/ Signup
Preregistration
Profile
Edit
Courses
Logout
Marksheet
Add student
Add assessment
Group
Discussion
Resource
Documents
Post
Update
Submission
Submission window
Submit
Arrows indicate actions
-> This action can be performed by all user
-> This action can only be performed by faculty member
-> This action can only be performed by chairman
-> This action can only be performed by student

Figure-3: Use case diagram

4.2 Ideal Flow:

Landing Page

Landing page is any web page that a visitor can arrive at or "land" on. On hitting the agent URL, the server sends the landing page to the browser, the browser then displays the page to the visitor. Figure-4 shows the landing page of Agent.



Agent is a real-time online class room management system with many new features.

Copyright © Nabila Ahmed, Ahnaf Tahmid Chowdhury, Mohammad Lummim Sarker 2017

Figure-4: Landing page

As you can see, the landing page is made up of three parts header, body and footer. The header is a navigation bar, which contains the application name and a login form. The body contains a small description of the application and a signup form. The last but not the least is the footer, that contains the application's author's name. The next two segments describe the signup and login parts in more details.

Signup

The very first thing to enter into the system is signup, this action creates an account for the user. For signup, user needs to provide user's name, email, username, password and status(faculty or student). Figure-5 shows the signup form.

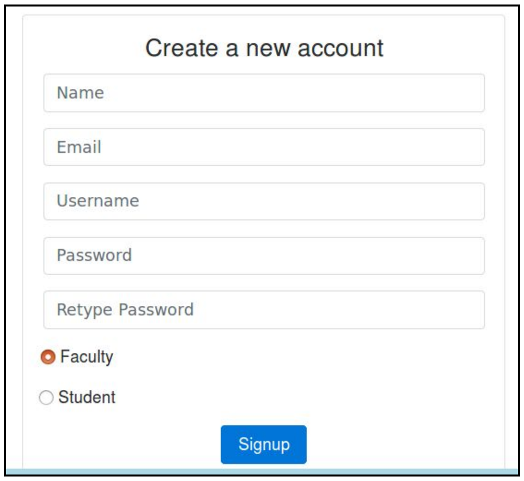
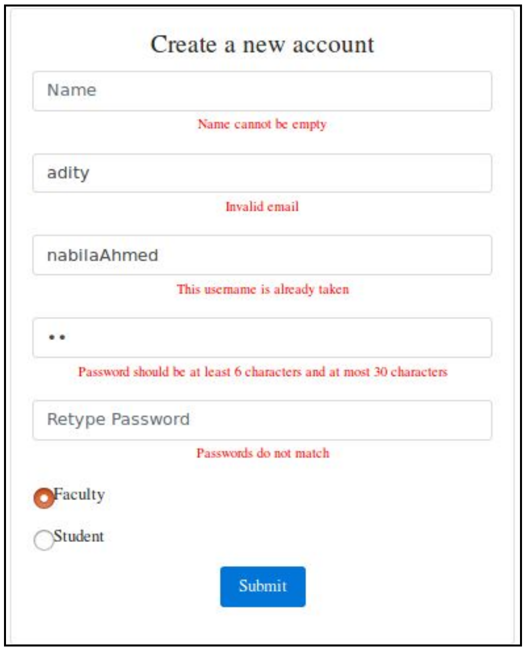


Figure-5: Signup form

During signup, user needs to provide valid information and the inputs should fall into the following rules:

- None of the fields can be empty.
- Email address should be a valid email address.
- Username and email address should not be used before to create any account in Agent.
- Password should be at least 6 and at most 30 characters.
- Password and retype-password should be same.

After clicking the Signup button the form is submitted to the server and the server goes through all the inputs. If any of the inputs break the rule, the server sends the corresponding alerts to the user. Figure-6 shows how the alerts are shown on the browser.


Create a new account

Name

Name cannot be empty

adity

Invalid email

nabilaAhmed

This username is already taken

••

Password should be at least 6 characters and at most 30 characters

Retype Password

Passwords do not match

[✓ Faculty]

Student

Submit

Figure-6: Signup validation alerts

But if all the inputs are well formed, the server stores the user's information into the database and thus a new account for the user is created. The user is notified about the successful registration by a flash message send by the server. Figure-7 shows the successful flash message.

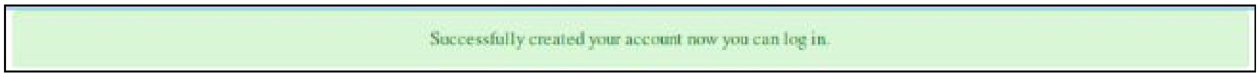

Successfully created your account now you can log in.

Figure-7: Flash message for successful registration

Login

Once signup, user can login anytime user wants. For login user needs to provide the same username and password, those are used during registration. Figure-8 shows the login form.

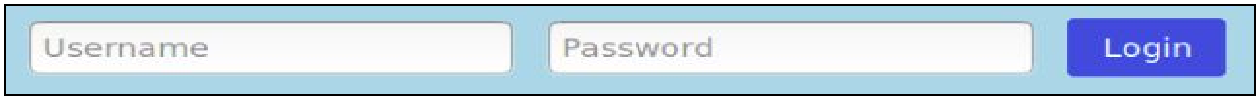


Figure-8: Login form

On clicking the Login button, the form is submitted to the server. Server then checks whether this username and password pair belongs to a registered user or not. If there is no such user, the server notifies the trying user about the error by a flash message. Figure-9 shows the flash message on failed login.

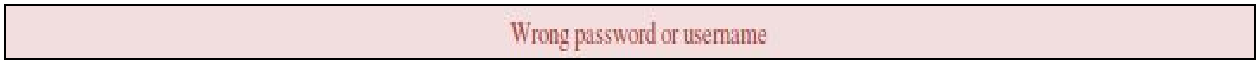
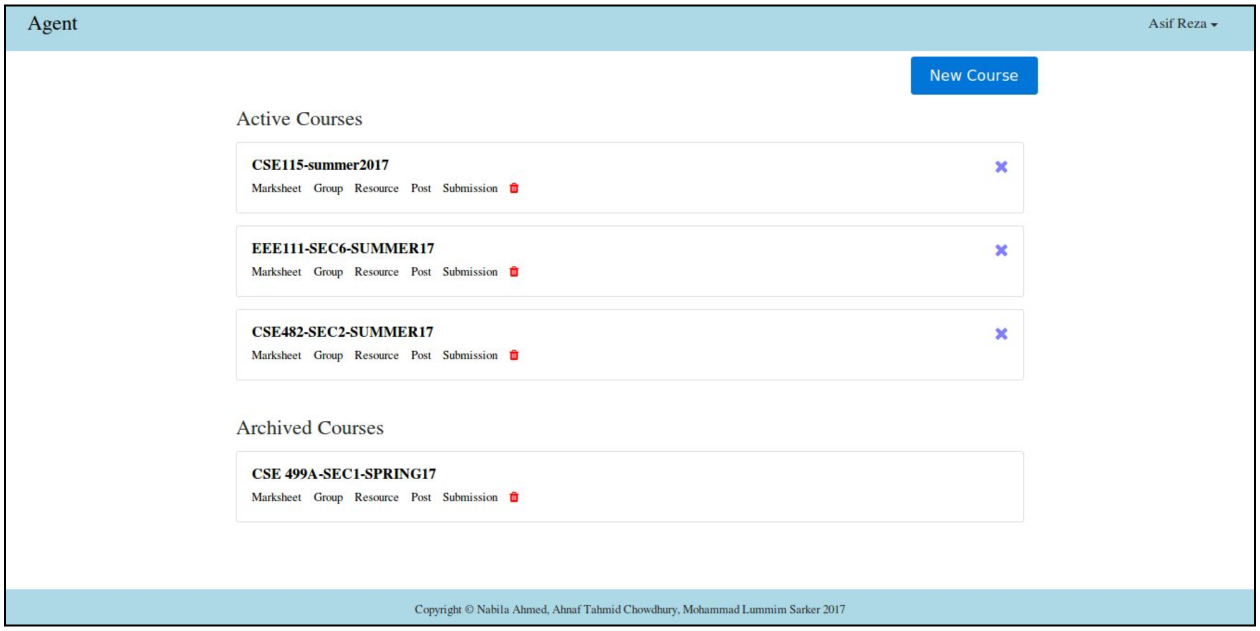

Wrong password or username

Figure-9: Flash message for failed login

But if a user is found with those username and password, the server extracts all the information about the user from database and takes user to the user's home page. Figure-10 shows the home page of a user.



Active Courses

CSE115-summer2017
Marksheet Group Resource Post Submission

EEE111-SEC6-SUMMER17
Marksheet Group Resource Post Submission

CSE482-SEC2-SUMMER17
Marksheet Group Resource Post Submission

Archived Courses

CSE 499A-SEC1-SPRING17
Marksheet Group Resource Post Submission

Copyright © Nabila Ahmed, Ahnaf Tahmid Chowdhury, Mohammad Lummim Sarker 2017

Figure-10: Home page

Home Page

The differences between the home page and the landing page are in two places. Instead of a login form, the home page contains a drop down menu. Figure-11 shows the contents of the drop down menu. And the body contains a list of courses user has access, divided into active and archived courses.

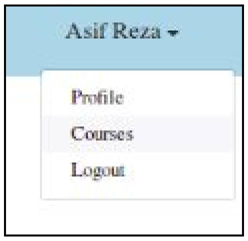
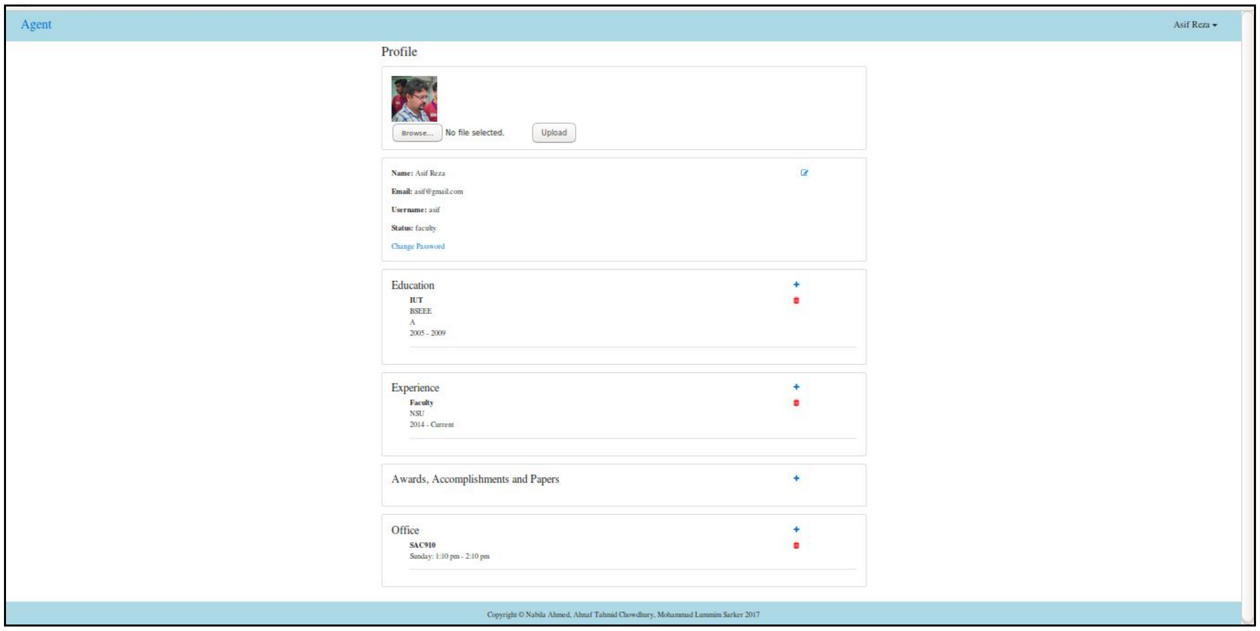


Figure-11: Drop down menu

Profile

On clicking the profile from the drop down menu, the server takes user to the user’s profile page. Figure-12 shows the profile page. Where user can perform the following tasks:

- Change profile picture
- Edit name
- Edit ID(only visible in student account)
- Change Password
- Add or delete education
- Add or delete experience(only visible in faculty account)
- Add or delete awards, accomplishment, and papers.
- Add or delete office hours (only visible in faculty account)



Profile

No file selected.

Name: Asif Reza

Email: asif@gmail.com

Username: asif

Status: faculty

Change Password

Education

IUT
BSEEE
A
2005 - 2009

Experience

Faculty
NSU
2014 - Current

Awards, Accomplishments and Papers

Office

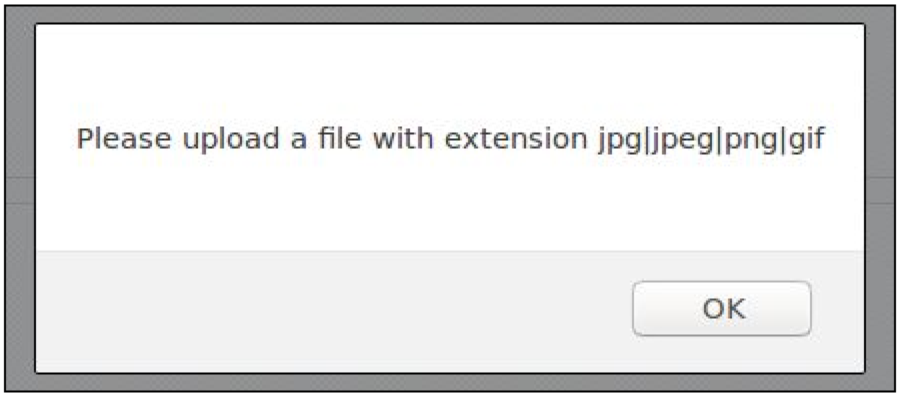
SAC910
Sunday: 1:10 pm - 2:10 pm

Copyright © Nabila Ahmed, Ahnaf Tahmid Chowdhury, Mohammad Lummim Sarker 2017

Figure-12: Profile page

Change Profile Picture

To change profile picture, user needs to click the browse button. It then opens a window ask user to choose the image from user's local directory. After choosing the image file, user has to upload it by clicking the upload button. If the selected file is not an image file, server does not accept it and in return shows error as an alert on user's browser. The error is shown in figure-13. Otherwise, the server successfully updates the profile picture in the database and the change can see on the browser.



Please upload a file with extension jpg|jpeg|png|gif

Figure-13: Dialog box pop up on uploading non-image file

Change Name or ID

To change name or ID user just needs to click the edit button. A prompt window pops up and waits for user input. The window is shown in figure-14. Once user provides input and press ok, the server records the changes and that change appears on the browser.

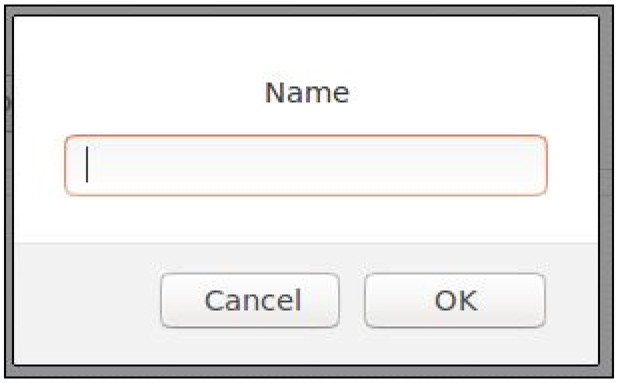


Figure-14: Prompt window during changing name or ID.

Change Password

To change password user needs to click on the change password link. This brings user in a new page as shown in figure-15. In this page, user has to provide the current password, so that no one else other than the valid user can change account's password. Along with the current password, user needs to provide the new password. Again the previous rules are applied on password, those are password should be at least 6 and at most 30 characters, and password and retype-password should be same. Disobey any of these rules show alerts on user's browser.

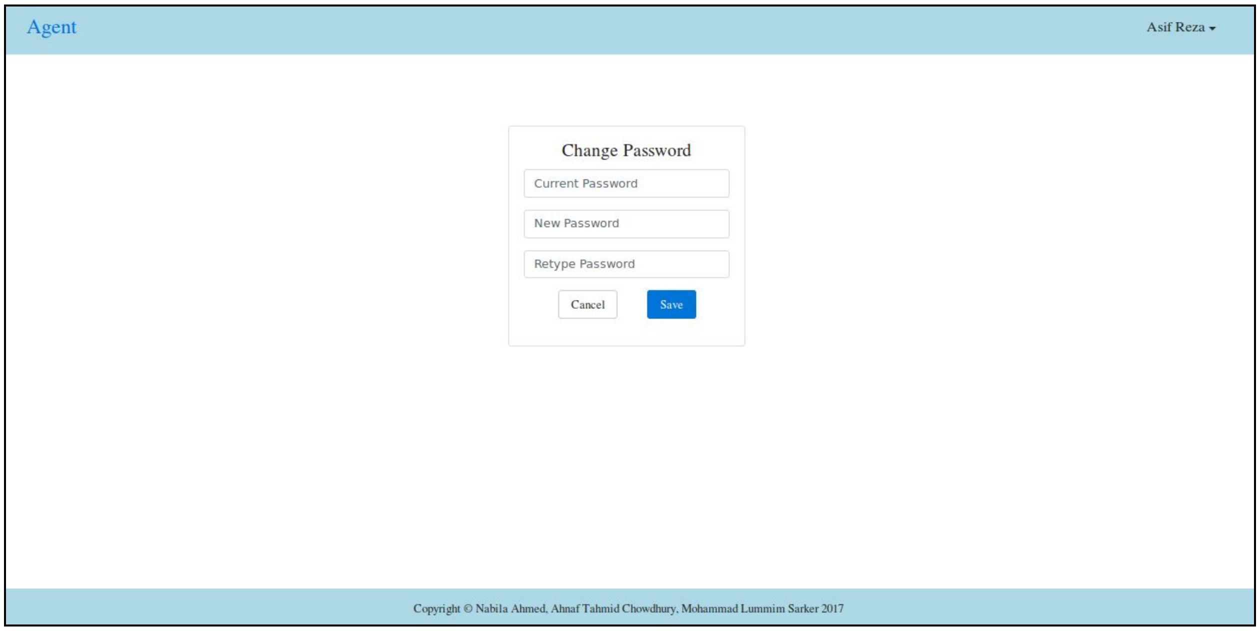


Figure-15: Change password page

Add New Item

To add any new item, user just needs to click the '+' button of the corresponding field as shown in figure-12. On clicking any of the add buttons, a dialog prompt slides down and fade in from the top of the page. The dialog contains a form and the form contains information related to the field. The forms against the fields are shown in figure-16. Whatever the form is, user needs to fill up the form and submits to the server. Server then stores the information into the database and on the web browser, user can see a new item appear in the field.

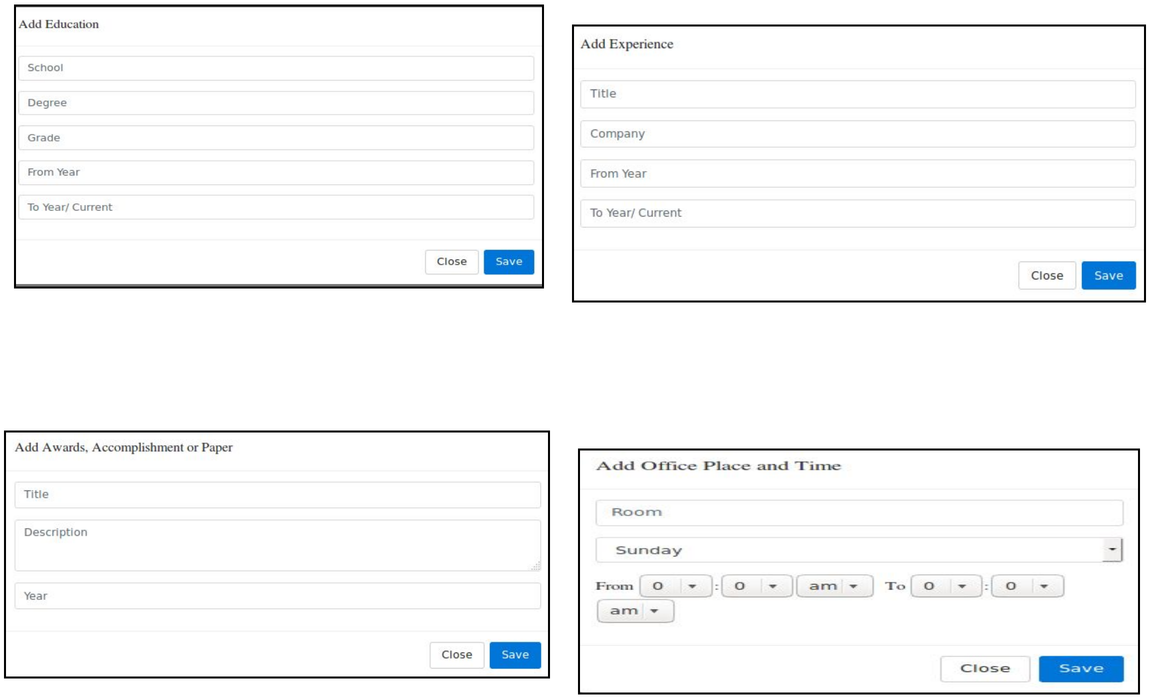


Figure-16: The dialog boxes to add items in profile.

Delete Item

To delete any item, user just needs to click its corresponding trash button as shown in figure-12. The server deletes that item and the item no longer appears on the browser.

Courses

On clicking the courses from the drop down menu as shown in figure-11, the same home page appears as shown in figure-10. Here the courses are divided into active and archived courses. Active courses mean user is still engaged with the courses and archived courses mean user no longer works on those courses. Every course has its marksheet, group, resource, post and submission links. Those links are common in both faculty and student account. Other than these, the faculty has some extra power.

Close Course

Only faculty can convert an active course into an archived course by clicking the 'x' button of that course. Once the course is archived, it can not active again. The warning is provided as an alert on clicking the close button. Figure-17 shows the warning.

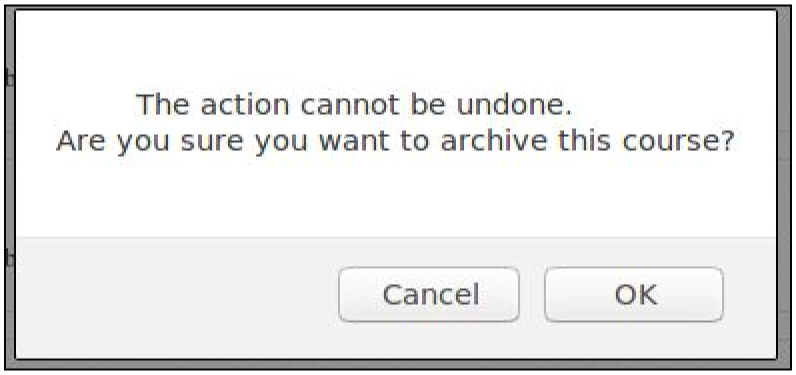

The action cannot be undone.
Are you sure you want to archive this course?

Figure-17: Warning during closing a course

Faculty can also create a new course and can delete an existing course.

Create New Course

To create a course user needs to click the new course button of the homepage. A dialog prompt slides down and fades in from the top of the page. Figure-18 shows the dialog. The dialog contains a form, user needs to fill up the form and submit to the server. Server then stores the course information into the database and a new course appears in the active list. The courses are sorted in the order of time, i.e recent courses appear at the top.

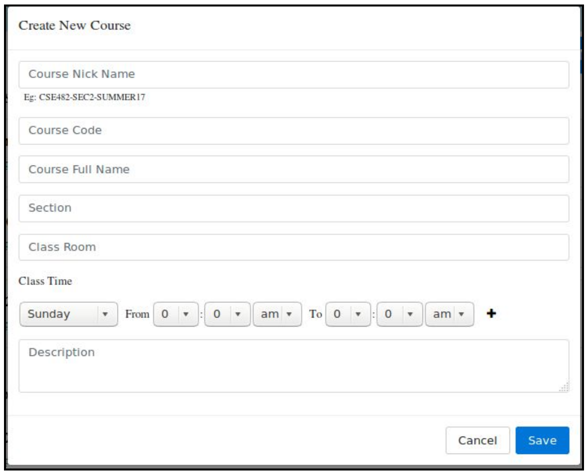
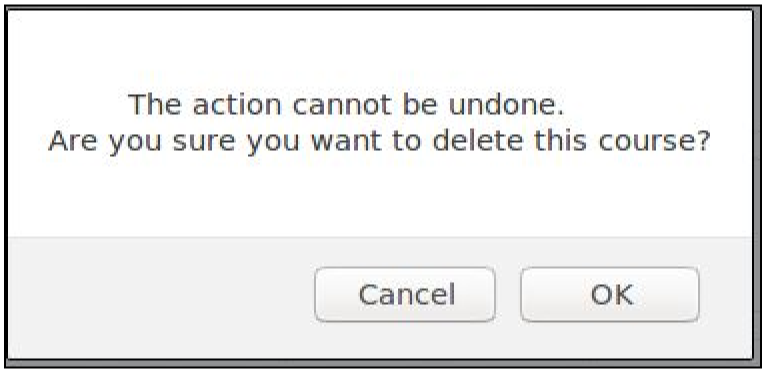


Figure-18: Dialog box to create a new course

Delete Course

To delete a course user just needs to click the trash button of the course. The action can not be undone. So user gets a warning message as an alert before deleting the course. The warning message is shown in figure-19. If the user press ok, server deletes that course and the course no longer appears in the list.



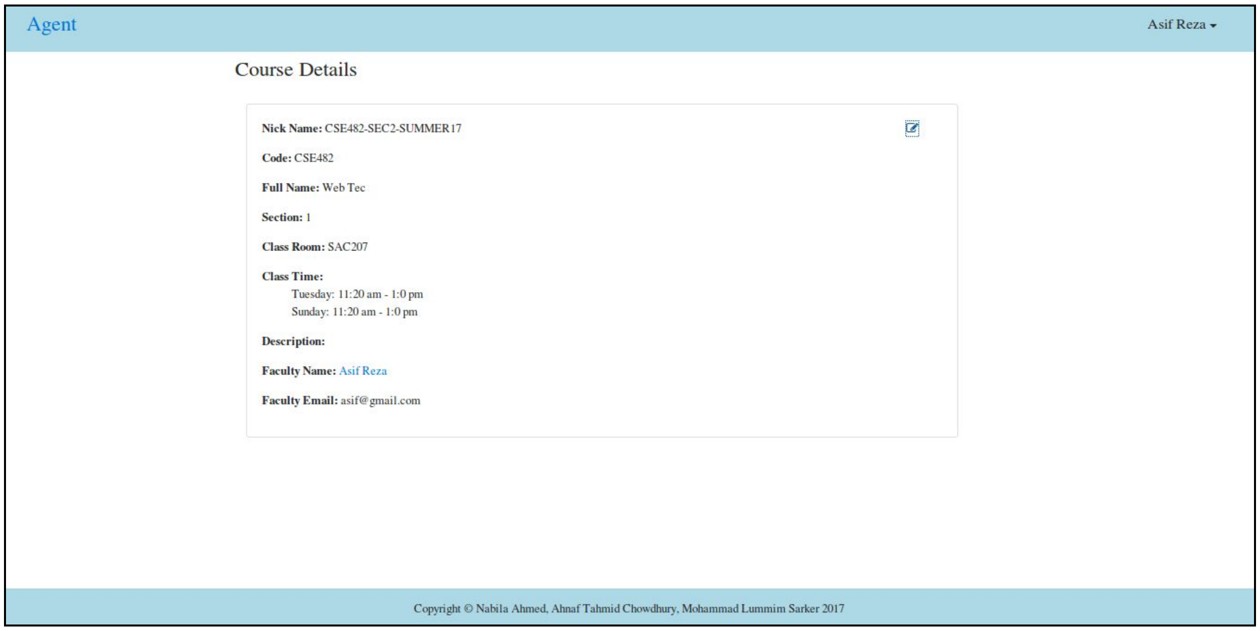
The action cannot be undone.
Are you sure you want to delete this course?

Figure-19: Waring during deleting a course

Close, add and delete course options only visible in faculty account.

Course Details

On clicking on the course name, server takes user to a new page which shows the course details. Figure-20 shows the page. The detail contains all the information about the course like course name, class room, class time etc. User can also edit details by clicking the edit button but the user needs to be a faculty. Or if user wants to get more information about the faculty, can click on the faculty name. It brings user to the faculty's profile.



Course Details

Nick Name: CSE482-SEC2-SUMMER17

Code: CSE482

Full Name: Web Tec

Section: 1

Class Room: SAC207

Class Time:
Tuesday: 11:20 am - 1:0 pm
Sunday: 11:20 am - 1:0 pm

Description:

Faculty Name: Asif Reza

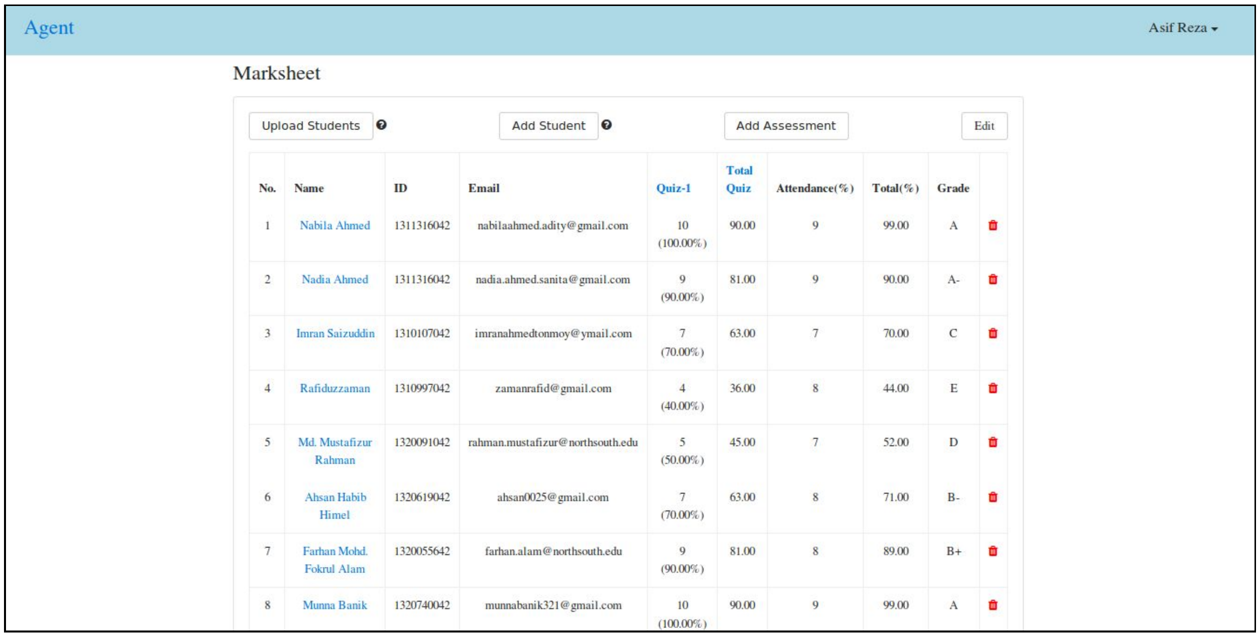
Faculty Email: asif@gmail.com

Copyright © Nabila Ahmed, Ahnaf Tahmid Chowdhury, Mohammad Lummim Sarker 2017

Figure-20: Course detail page

Marksheet

On clicking the marksheet link of any course brings user to the marksheet page of that course. Figure-21 shows the marksheet page. This page is responsible for the very first connection between the faculty and students of that course. The upload student, add student, add assessment, edit and delete student powers are only accessed by the faculty of that course.



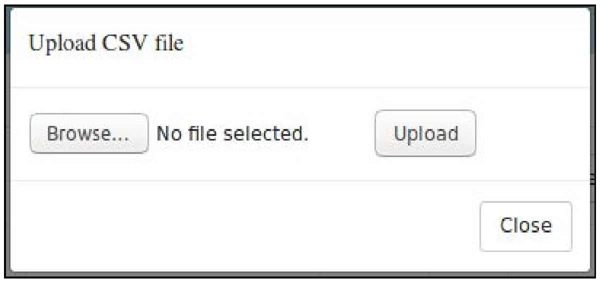
Marksheet

No.	Name	ID	Email	Quiz-1	Total Quiz	Attendance(%)	Total(%)	Grade	
1	Nabila Ahmed	1311316042	nabilaahmed.adity@gmail.com	10 (100.00%)	90.00	9	99.00	A	🗑
2	Nadia Ahmed	1311316042	nadia.ahmed.sanita@gmail.com	9 (90.00%)	81.00	9	90.00	A-	🗑
3	Imran Saizuddin	1310107042	imranahmedtonmoy@ymail.com	7 (70.00%)	63.00	7	70.00	C	🗑
4	Rafiduzzaman	1310997042	zamanrafid@gmail.com	4 (40.00%)	36.00	8	44.00	E	🗑
5	Md. Mustafizur Rahman	1320091042	rahman.mustafizur@northsouth.edu	5 (50.00%)	45.00	7	52.00	D	🗑
6	Ahsan Habib Himel	1320619042	ahsan0025@gmail.com	7 (70.00%)	63.00	8	71.00	B-	🗑
7	Farhan Mohd. Fokrul Alam	1320055642	farhan.alam@northsouth.edu	9 (90.00%)	81.00	8	89.00	B+	🗑
8	Munna Banik	1320740042	munnabanik321@gmail.com	10 (100.00%)	90.00	9	99.00	A	🗑

Figure-21: Marksheet page

Add Student

Faculty can add multiple students at a time by clicking upload student button or can add single student by clicking add student button. Upload student button asks user to upload a CSV(comma separated value) file, The format of data in the file should be like name, ID, email. Figure-22 shows the dialog box to upload CSV file. If user uploads another file rather than CSV file, application shows error on the browser. The error is shown in figure-23.



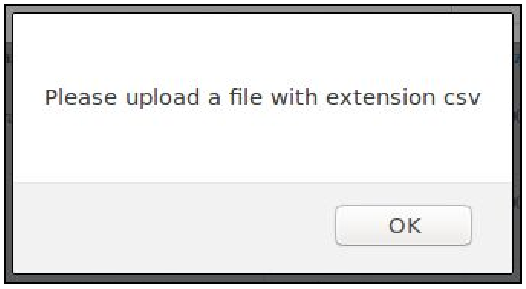
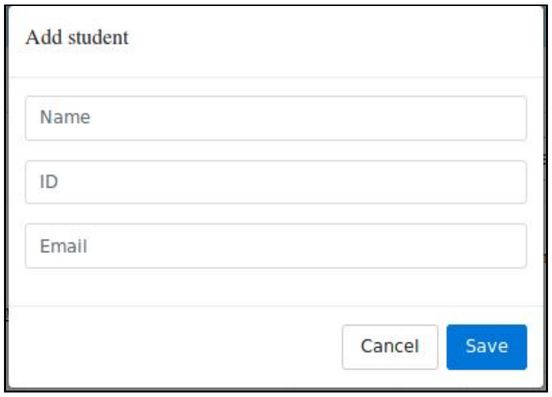

Please upload a file with extension csv

Figure-22: Dialog box to upload a CSV file

Figure-23: Error for not uploading CSV file

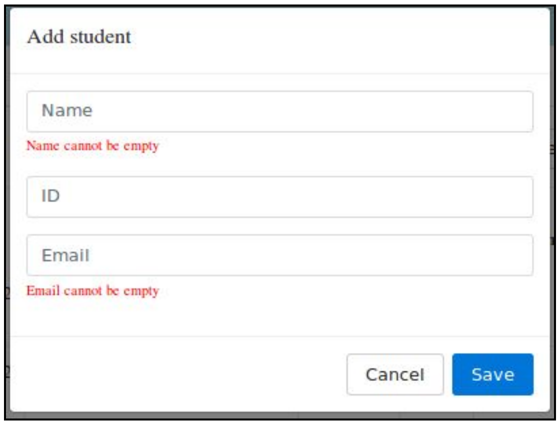
Name student button also causes a dialog box to appear, which asks user to provide student's name, ID and email. Figure-24 shows the dialog box. The ID and email are the required field, empty input in those fields shows error on the browser as shown in figure-25.


Add student

Name

ID

Email


Add student

Name

Name cannot be empty

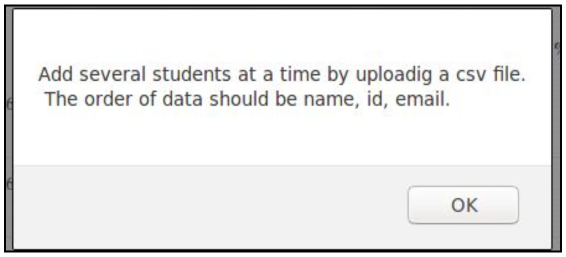
ID

Email

Email cannot be empty

Figure-24: Add student dialog box Figure-25: Add student dialog box with error.

User might get confuse between upload and add student. To clear the confusion clarification is attached beside each button. On clicking the clarification icon, clarification message pop up. Figure-26 shows the messages.


Add several students at a time by uploadig a csv file.
The order of data should be name, id, email.

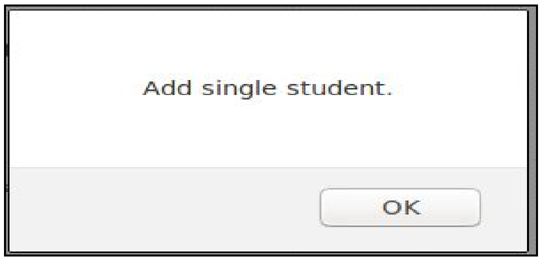

Add single student.

Figure-26: Clarification messages

Delete Student

Faculty can delete any student by clicking the trash button in the same row. The server removes the student from the course and the student no longer appears in the marksheet.

Add Assessment

Faculty can add assignment, field work, final, mid, quiz, presentation and project by selecting corresponding option from the drop down menu as shown in figure-27, which appears on clicking the add assessment button. This action creates a new column for the assessment. Application provides the name for the assessment which is assessment type hyphen assessment number. If any type of assessment is added for the first time, along with the assessment column a total column is also created.

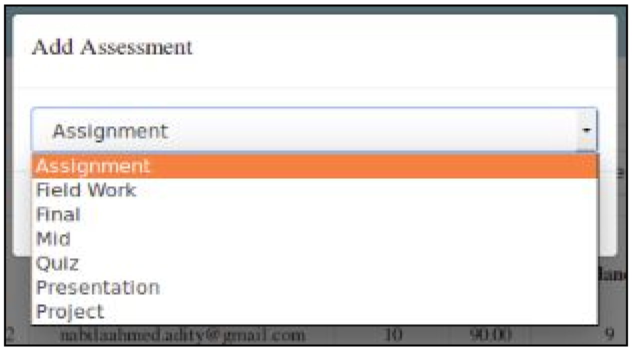
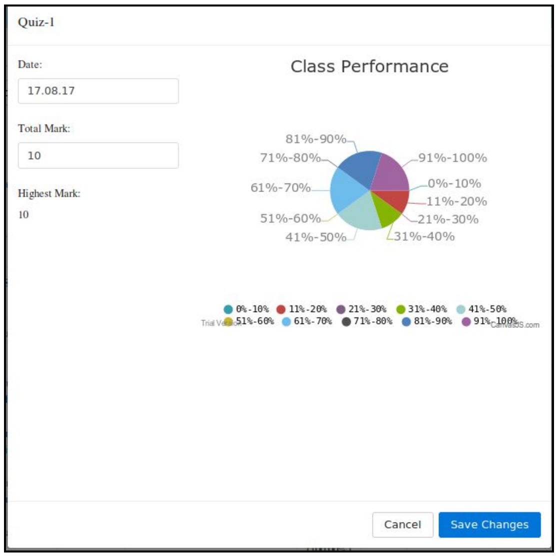


Figure-27: Add assessment dialog box

Assessment Details

On clicking on the assessment name, a dialog prompt slides down and fade in from the top of the page. If the assessment is a total, user can set the best count and the weight of the assessment. But if it's not a total, user can set the date and the total mark of the assessment. Details also contain the highest mark of the class and the overall class performance in a pie chart. These highest mark and class performance are calculated by the application on server side. Each kind of dialog boxes is shown in figure-28.


Quiz-1

Date:
17.08.17

Total Mark:
10

Highest Mark:
10

Figure-28: Assessment details dialog box.

Assessment details can be viewed by both the faculty and students of the course.

Edit Marksheet

On clicking the edit marksheet button, it takes user to a new page. Here faculty can give marks and grade to each student.

Calculation

Calculation part is performed by the application on the server side.

Below the marks of assessment, a percentage is shown. This percentage is calculated by the (given marks / total marks) * 100. If the total mark is not set by the faculty then by default the total mark is 100.

Total assessment marks are also in percentage. First of all the average mark of best k assessments is calculated by the following formula. By default, k is the total number of assessment.

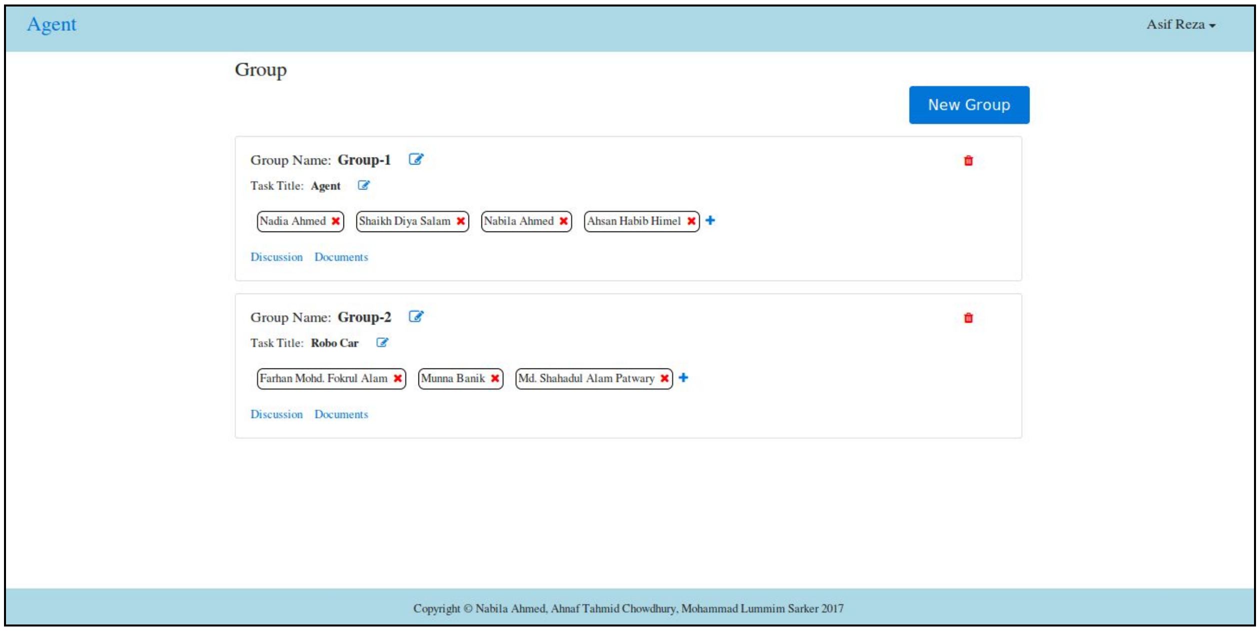
$$\text{Average mark} = \left(\sum_{i=1}^{k} \text{percentage mark of assessment } \mathbf{i}\right) / k$$

Then converts the average marks according to the weight of the assessment by the (100 / average mark) * weight. If the weight is not set by the faculty then by default the weight is 100.

By adding the values from each assessment total and from the attendance the total column is calculated.

Group

On clicking the group link of any course brings user to the group page of that course. Figure-29 shows the group page.This page contains a list of all the groups. Each group has a group name, task title, group members name, group related discussion and documents. Only the group member and the faculty have access to the group. Although both the student and the faculty can edit group name and task title, add and remove group members can only be performed by the faculty. Even the faculty can create a new group and can delete a whole group.


Agent   Asif Reza

Group

New Group

Group Name: **Group-1**
Task Title: **Agent**
Nadia Ahmed ✖ Shaikh Diya Salam ✖ Nabila Ahmed ✖ Ahsan Habib Himel ✖ +
Discussion Documents

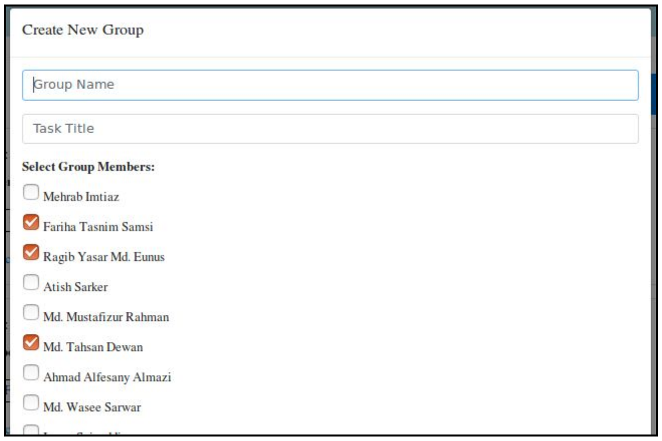
Group Name: **Group-2**
Task Title: **Robo Car**
Farhan Mohd. Fokrul Alam ✖ Munna Banik ✖ Md. Shahadul Alam Patwary ✖ +
Discussion Documents

Copyright © Nabila Ahmed, Ahnaf Tahmid Chowdhury, Mohammad Lummim Sarker 2017

Figure-29: Group page

New Group

To create a new group user needs to click the new group button. A dialog prompt slides down and fades in from the top of the page. Figure-30 shows the dialog. The dialog contains a form, the form shows a list of students, those have not enrolled into any group yet. User needs to fill up the form and submit to the server. Server then stores the group information into the database and a new group appears in the list.


Create New Group

Group Name

Task Title

Select Group Members:

- [ ] Mehrab Imtiaz
- [x] Fariha Tasnim Samsi
- [x] Ragib Yasar Md. Eunus
- [ ] Atish Sarker
- [ ] Md. Mustafizur Rahman
- [x] Md. Tahsan Dewan
- [ ] Ahmad Alfesany Almazi
- [ ] Md. Wasee Sarwar

Figure-30: Dialog box for new group

Change Group Name and Task Title

To change group name or task title user just needs to click the edit button. A prompt window pops up and waits for user input. Once user provides input and press ok, the server records the changes and that change appears on the browser. Figure-31 shows the window.

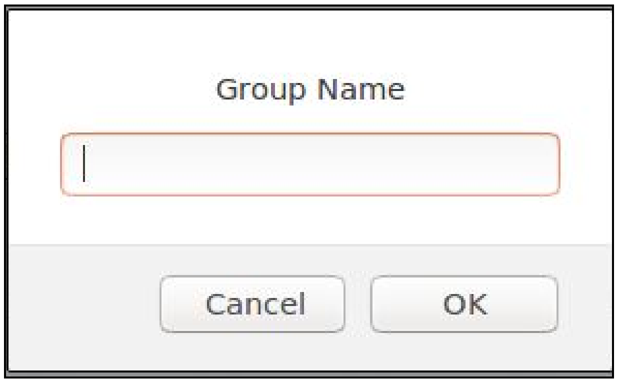
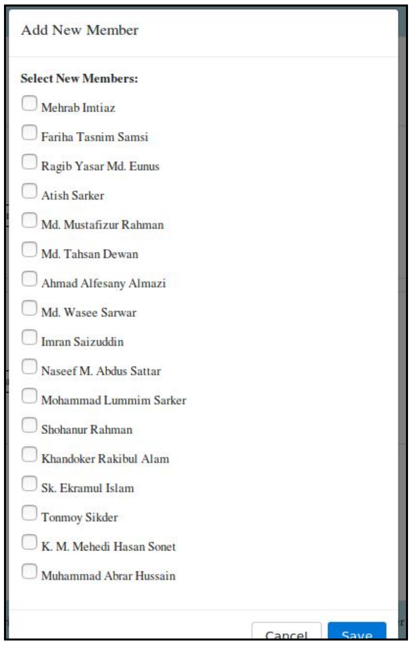


Figure-31: Prompt window during changing group name.

Add New Member

To add a new member to a group, user needs to click on the '+' button of the group. A dialog prompt slides down and fades in from the top of the page. Figure-32 shows the dialog. The dialog contains a form, the form shows a list of students, those have not enrolled into any group yet. User can select multiple members and on press save button, the new members are added to the group.


Add New Member

Select New Members:

- Mehrab Imtiaz
- Fariha Tasnim Samsi
- Ragib Yasar Md. Eunus
- Atish Sarker
- Md. Mustafizur Rahman
- Md. Tahsan Dewan
- Ahmad Alfesany Almazi
- Md. Wasee Sarwar
- Imran Saizuddin
- Naseef M. Abdus Sattar
- Mohammad Lummim Sarker
- Shohanur Rahman
- Khandoker Rakibul Alam
- Sk. Ekramul Islam
- Tonmoy Sikder
- K. M. Mehedi Hasan Sonet
- Muhammad Abrar Hussain

Figure-32: Dialog box to add new member

Remove Member

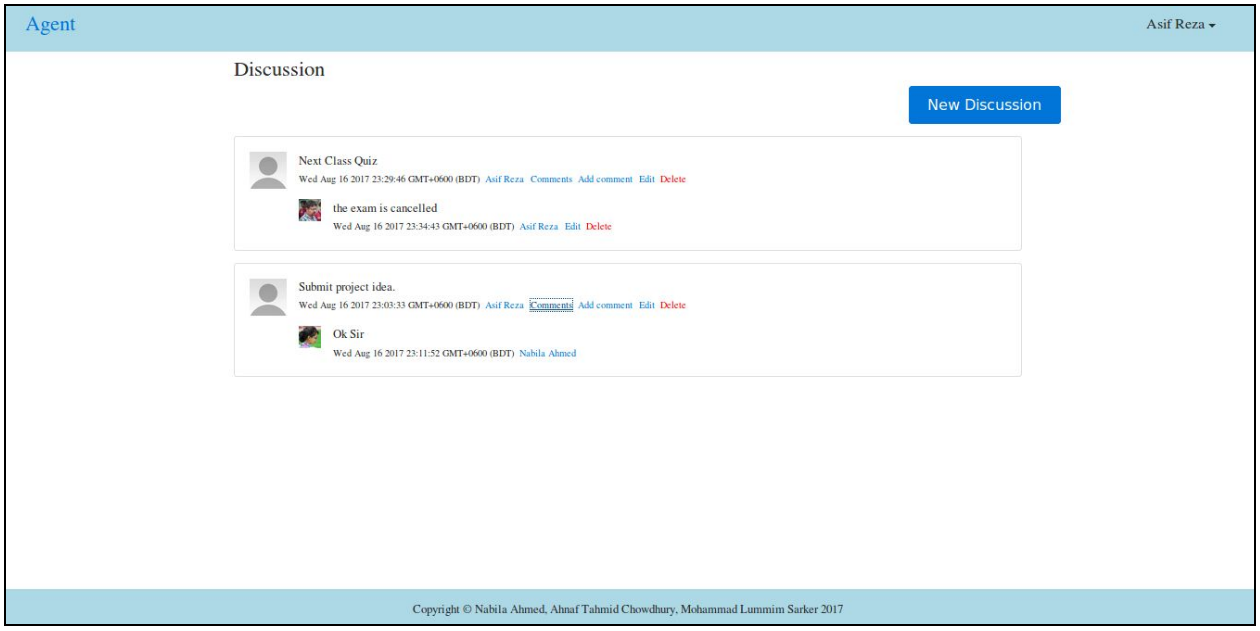
To remove a member from a group user just needs to click the ‘x’ button beside the member’s name. The server removes the member from the group and the member no longer appears in the group.

Delete Group

To delete a group user needs to click the trash button of the group. The server delete the group from the database and the group no longer appears in the list.

Discussion

On clicking the discussion of any group, brings user to a discussion page of that group. Figure-33 shows the discussion page. Here group members and faculty can view all the discussions with the poster’s name and the post date, can start a new discussion, can comment on any existing discussion and can edit or delete own posts. For better user experience the comments are initially hidden. On toggling the comment button user can view and hide comments. Not only this if user clicks on the poster's name, it takes user to the poster's profile page.



Discussion

Next Class Quiz
Wed Aug 16 2017 23:29:46 GMT+0600 (BDT) Asif Reza Comments Add comment Edit Delete

the exam is cancelled
Wed Aug 16 2017 23:34:43 GMT+0600 (BDT) Asif Reza Edit Delete

Submit project idea.
Wed Aug 16 2017 23:03:33 GMT+0600 (BDT) Asif Reza Comments Add comment Edit Delete

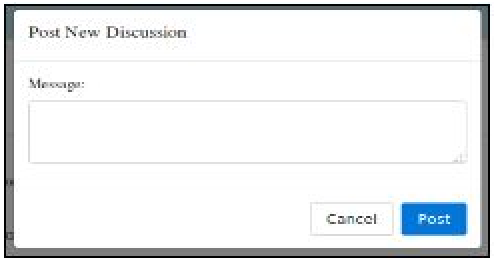
Ok Sir
Wed Aug 16 2017 23:11:52 GMT+0600 (BDT) Nabila Ahmed

Copyright © Nabila Ahmed, Ahnaf Tahmid Chowdhury, Mohammad Lummim Sarker 2017

Figure-33: Discussion Page

New Discussion and Comment

On clicking new discussion button or add comment button a dialog prompt slides down and fades in from the top of the page. Figure-34 shows the dialog boxes. The dialogs contain a form. In the form, user writes down message or comment and then presses the post button. Server record the post into the database and the web browser shows the changes. The new post appears at the top of all the posts and the new comment on a discussion appears at bottom of all the comments of that discussion.



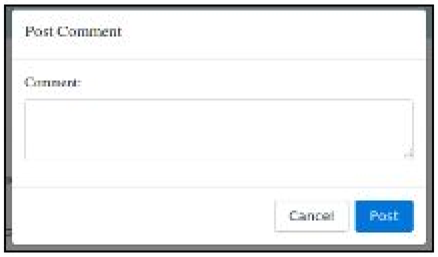


Figure-34: Dialog boxes for new discussion and new comment

Edit Discussion and Comment

User can only edit if the user is the owner of the discussion or comment. On clicking edit button a dialog prompt slides down and fades in from the top of the page as shown in figure-35. The dialog contains a form but this time the form is not empty, rather it contains the original message. User edits the message and presses save button.

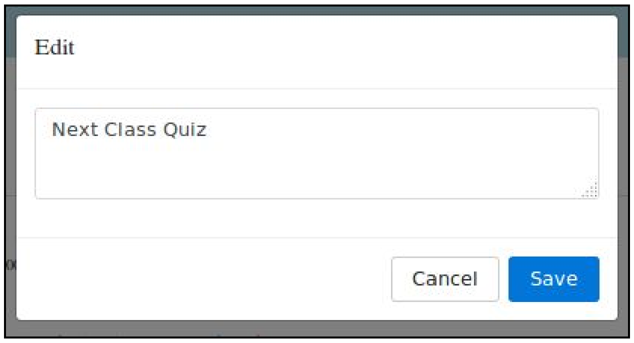


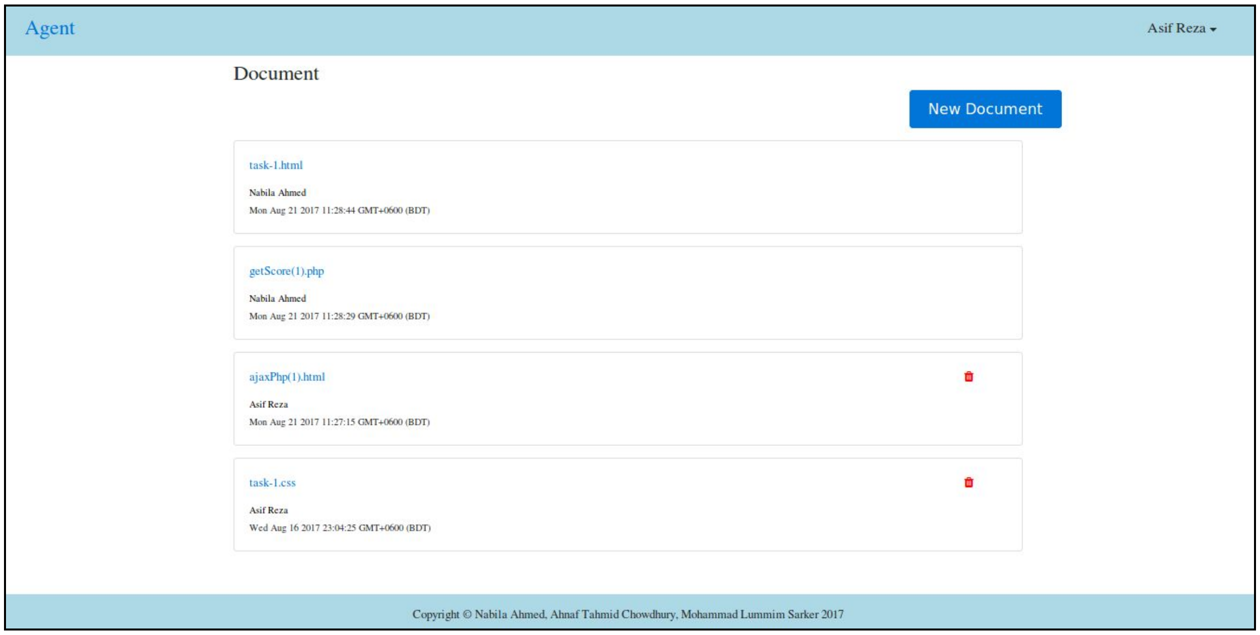
Figure-35: Edit discussion or comment dialog box

Delete Discussion and Comment

User can only delete if the user is the owner of the discussion or comment. On clicking delete button, server deletes the post from the database and the post is no longer appears on the browser.

Documents

On clicking the documents of any group, brings user to a documents page of that group. Figure-36 shows the documents page. Each document has the file name, uploader's name and the upload date. Here group members and faculty can download any document and can upload a new file. User can only delete own uploaded files.



Document

task-1.html
Nabila Ahmed
Mon Aug 21 2017 11:28:44 GMT+0600 (BDT)

getScore(1).php
Nabila Ahmed
Mon Aug 21 2017 11:28:29 GMT+0600 (BDT)

ajaxPhp(1).html
Asif Reza
Mon Aug 21 2017 11:27:15 GMT+0600 (BDT)

task-1.css
Asif Reza
Wed Aug 16 2017 23:04:25 GMT+0600 (BDT)

Copyright © Nabila Ahmed, Ahnaf Tahmid Chowdhury, Mohammad Lummim Sarker 2017

Figure-36: Document page

New document

To upload a new document, user needs to click the new document button. It then opens a window asks user to choose the file from user's local directory as shown in figure-37. After choosing the file, user has to upload it by clicking the upload button. Server then stores the file into server's local directory, makes the required update in the database and the new files appears at the top of the document list.

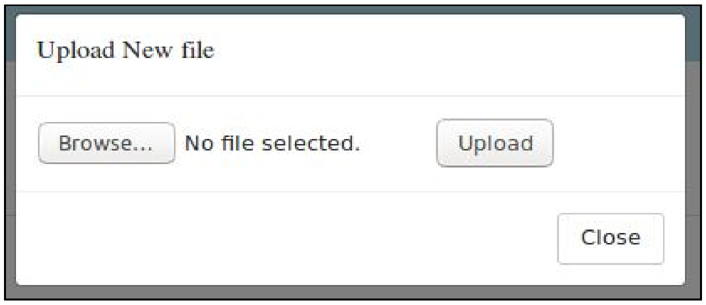


Figure-37: Document upload window

Delete Document

User can only delete if the user is the owner of the document. On clicking trash button, server deletes the document from the server's local directory and from the database and the document is no longer appears on the browser.

Resource

Resource of any course is same as the documents of any group. They both have the same functionalities except that in resource, file can only upload by the faculty of the course and instead of group members and faculty, students and faculty of the course have the access to the resource.

Post

Post of any course is same as the discussion of any group. They both have the same functionalities except that instead of group members and faculty, students and faculty of the course have the access to the post.

Submission

Clicking on the submission of any course takes user to a submission page as shown in figure-38. The page contains a list of all the submission windows of that course. Each window has a end time and a status attached to it. The status is either closed or open. Here faculty can open a new submission window and can view all the submission in any window. Student can submit in an open window and can only view own submissions.

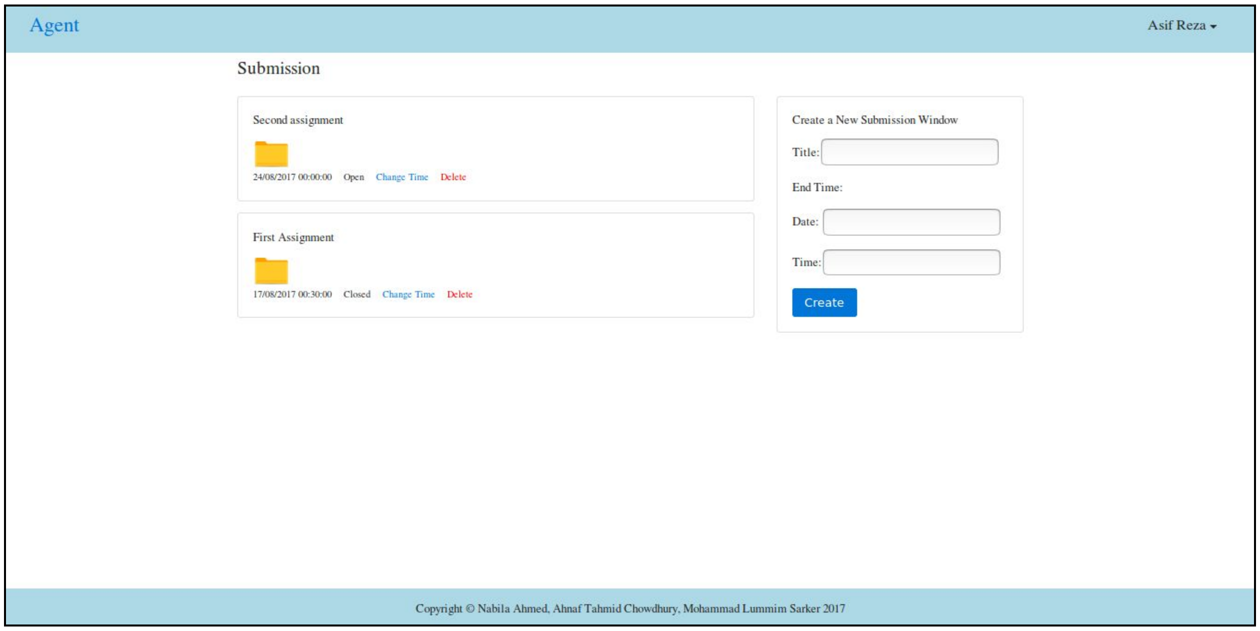

Agent
Asif Reza
Submission
Second assignment
24/08/2017 00:00:00 Open Change Time Delete
First Assignment
17/08/2017 00:30:00 Closed Change Time Delete
Create a New Submission Window
Title:
End Time:
Date:
Time:
Create
Copyright © Nabila Ahmed, Ahnaf Tahmid Chowdhury, Mohammad Lummim Sarker 2017

Figure-38: Submission page

Create New Submission Window

Create New Submission Window is only visible in faculty account. Here the faculty can give the submission title and can fix the end time of the window. After the end time, the window closes and does not accept any more submission. User can select end date and time by datepicker and timepicker as shown in figure-39. On clicking the create button server creates a new submission window in the database and the window appear on the browser at the top of all the windows.

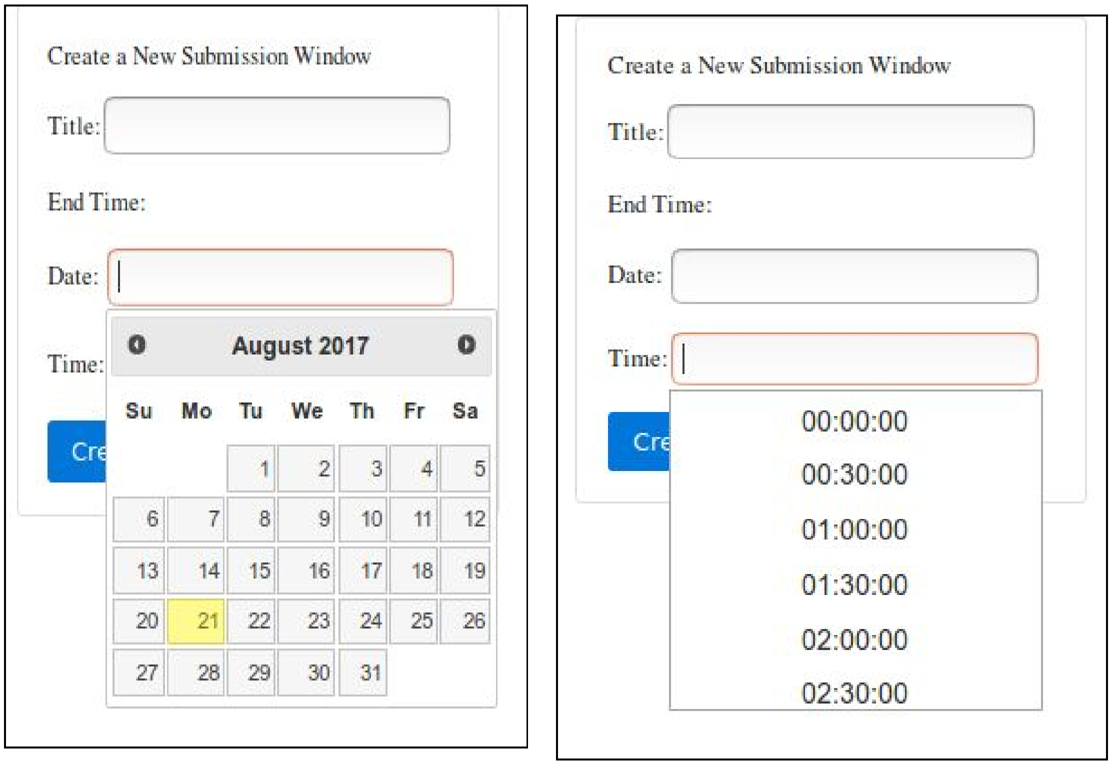
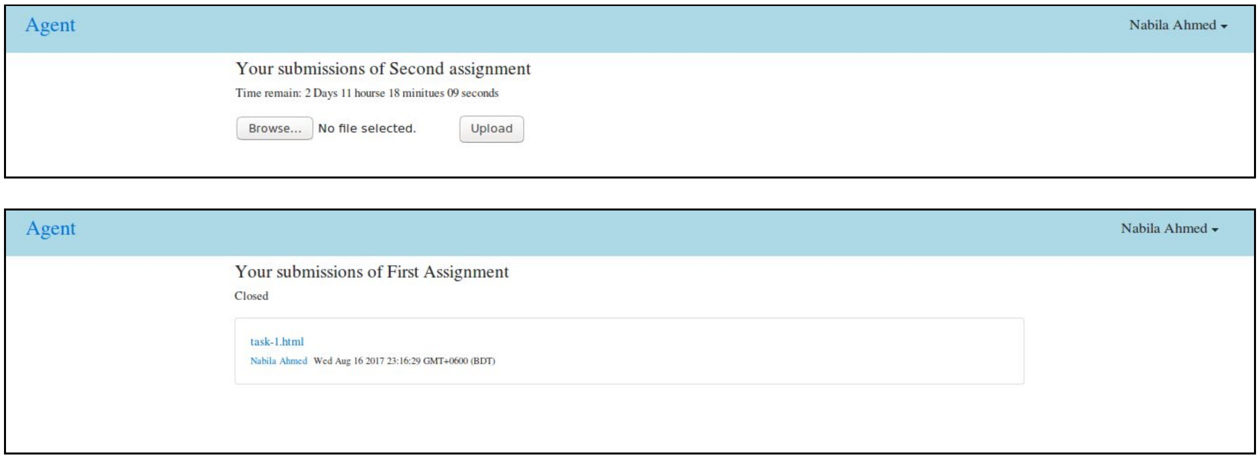


Figure-39: Datepicker and timepicker

The end time can be change. Only faculty can click the change time button and choose the new time.

Submit in Submission Window

When student of the course clicks any submission window it takes the student to a new page. Now the page can be two types. Firstly if there is still time for submission, at the top of the page remaining time appears and an option to upload file is there. But if the end time is passed, instead of remaining time the 'Closed' text appears and the student can view all own submissions. Both types of pages are shown in figure-40.



Your submissions of Second assignment

Time remain: 2 Days 11 hourse 18 minitues 09 seconds

Your submissions of First Assignment

Closed

task-1.html

Nabila Ahmed Wed Aug 16 2017 23:16:29 GMT+0600 (BDT)

Figure-40: Submission windows

Delete Submission

Only faculty can delete any submission window. On clicking trash button, server deletes the window from the database and the window is no longer appears on the browser.

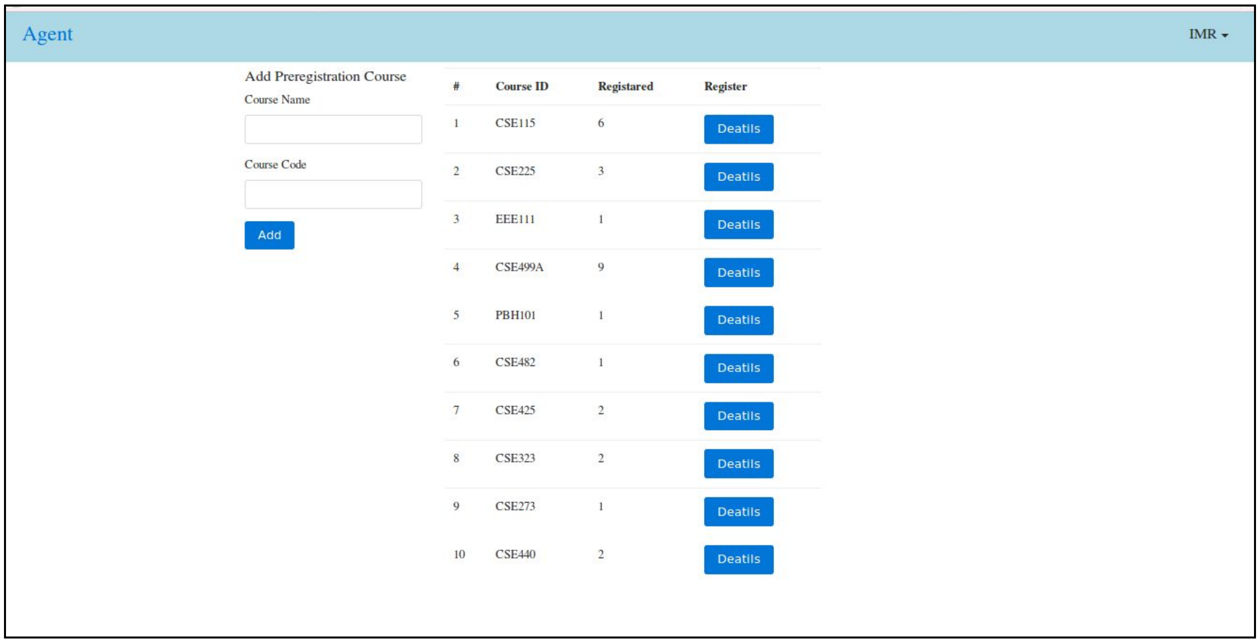
Logout

On clicking the logout button from the drop down menu as shown in figure-11, user logged out from the system.

Chairman Portal

So far the faculty and student portals are described. Now let's see what's new in chairman portal. There can be only one chairman and the chairman can only be fixed by the application. Though the chairman portal is not different from the faculty portal still it consumes some extra power.

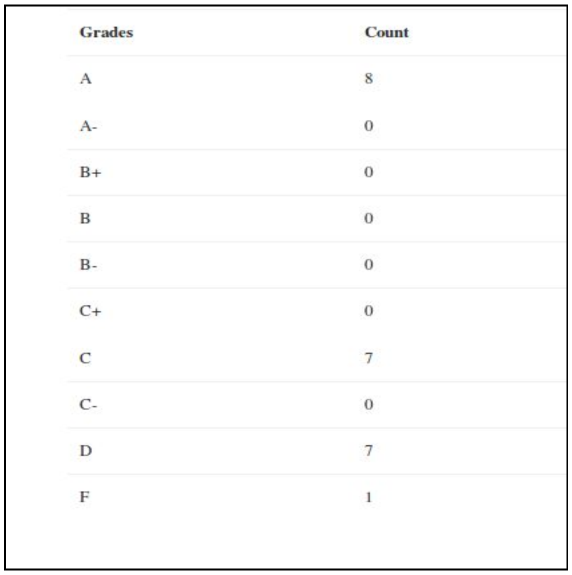
In the homepage of charman, beside the new course button there is a preregistration button. On clicking the preregistration button it takes charman to a new page as shown in figure-41. Here is a list of all the courses, those are open for preregistration. It is the only charman who can add course in pre registration.



#	Course ID	Registared	Register
1	CSE115	6	Deatils
2	CSE225	3	Deatils
3	EEE111	1	Deatils
4	CSE499A	9	Deatils
5	PBH101	1	Deatils
6	CSE482	1	Deatils
7	CSE425	2	Deatils
8	CSE323	2	Deatils
9	CSE273	1	Deatils
10	CSE440	2	Deatils

Figure-41: Preregistration page

Number of students registered for a course is shown by registered number. In details, chaiman can view past result of the course as shown in figure-41. By the combination of registered number and the past result, chairman can fix how many sections should open for this course in next semester.



Grades	Count
A	8
A-	0
B+	0
B	0
B-	0
C+	0
C	7
C-	0
D	7
F	1

Figure-41: Previous result of a course

4.3 Error Handling:

In this part, different types of error that might have occurred in use cases are described. Also, the solutions that are implemented in the program are mentioned.

Error can occur during database query. For example, during find, update, insert and delete, database can lose its connection or some internal error may occur. These errors are handled by the callback functions of Node. Error during database query means the action that the user has asked the server to perform cannot be completed. So as the response server has to inform user about the error. Server does this by rendering a page with message “ReferenceError: error is not defined.”

By mistake or by intention user may wants to go to a URL, which is not defined by the application. Since the application cannot match any route with this URL, it has to inform user about the bad request. Server does this by rendering a page with message “Page not found.”

In profile, user can update profile picture. To change profile picture user has to upload an image file from his/her local directory. But by mistake user may upload any other file rather than an image file. In that case, server has to inform user about the mistake. Server does this by showing a warning as an alert on the user browser. Figure-43 shows the error.

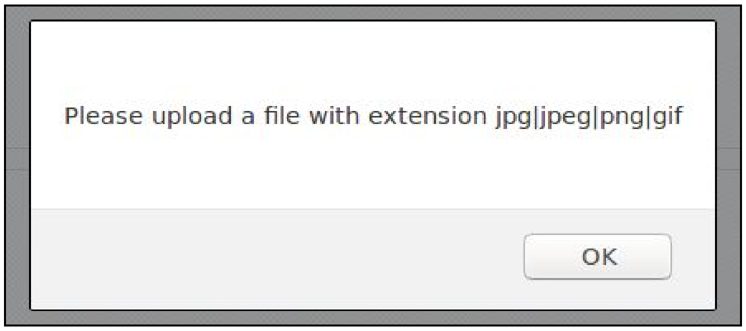

Please upload a file with extension jpg|jpeg|png|gif

Figure-43: Error for not uploading image file

In marksheet, faculty can upload multiple students at a time by uploading a CSV file. But by mistake user may upload any other file rather than a CSV file. In that case, server has to inform user about the mistake. Server does this by showing a warning as an alert on the user browser. Figure-44 shows the error.

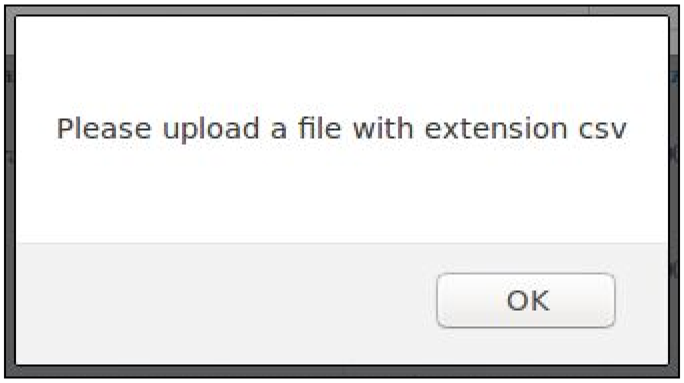
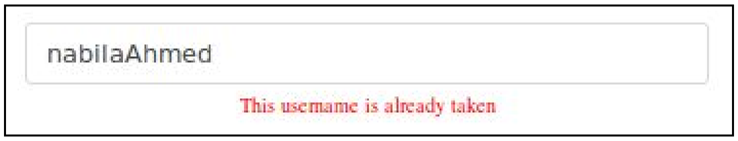

Please upload a file with extension csv

Figure-44: Error for not uploading CSV file

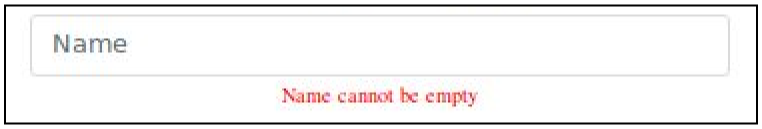
In database, some fields are unique identifiers. That means all the values of this field should be unique. For example in our application username and email must be unique. No user can register with the username or the email address which has been used previously to create any account. If anybody tries to insert the same value twice in a unique field, database generates an error. To avoid such error server first checks whether the value already exists in the database or not. If the value exists, server informs user about the error. Figure-45 shows how server presents unique username error on the browser.


nabilaAhmed

This username is already taken

Figure-45: Unique username error

In database, some fields are required. That means if anyone tries to save a document which has a required field and the field is empty, database generates an error. For example in our application name field during registration is required. If anybody tries to register without a name, database generates an error. To avoid such error server first checks whether the required field is empty or not. If the field is empty, server informs user about the error. Figure-46 shows how server presents required name error on the browser.



Name cannot be empty

Figure-46: Required name error

CHAPTER 5

Security Requirements

5.1 Access Requirements:

Since the app is a webapp, we need to take care of the security more than anything. Our user base is very restricted, without a proper identity it can’t be revealed. So, before registering a user needs to go through some tough steps to confirm that s/he is eligible to that portal. The access requirement is restricted thoroughly. We have checked if a user is student or faculty or chairman. If a user is a student, then s/he will be restricted to faculty and chairman portals. Again, if a user is a faculty, then s/he will be restricted to student and chairman portals. Again, if a user is a chairman, then s/he will be restricted to student portal. Since a chairman him/herself is a faculty, s/he has access to his very own faculty portal. Finally, each user is restricted to have access to the account of other users.

5.2 Integrity Requirements:

We are pretty clear about what data we are giving to whom. Now, in some cases we need permissions. For example, we tried to share students grade with a faculty. But, later surveyed and found that most of the students don’t want it to share their CGPA. So, later we sticked to not revealing student’s grade until we know that if it’s legal or not to share a student’s CGPA to his/her faculty.

Since the system is not dependent for student grades to admin, it will access the student data revealing only to the student. The faculty who is making the marksheet of the students is going to have the full access of the class data. But, the chairman can visit the result of the previous semester courses as a statistics to take the decision for course allotment.

5.3 Privacy Requirements:

Privacy is one of our best interest when it comes to security requirements. For example, in faculty portal we restricted it to the faculties only. Every faculty has a faculty ID. If someone is trying to access a faculty account, s/he must need to be a faculty. Again for student account, only a student can access his/her account. Now, In case of an admin account, an admin can add or delete an student from the system. Admin account is highly restricted. For ensuring more security, we hashed passwords with a random salt, so that it can prevent attacks from outside of our user base. The password hash is random hexadecimal value which unknown to the system itself. The system only stores the random salt value for a user. When a user logs in, the user given passphrase is hashed with the stored salt. So, the salt is retrieved from the database, then it hashes the whole password for making sure the and finding the correct user log in.

5.4 Audit Requirements:

For auditing any bug on any portal, we are going to have a developer mode which will be used to help out our users in case of buggy features. In our codebase we followed the ***gitflow*** system. In gitflow system, we maintain a developer branch and a hotfix branch. If we are issued with any bugs or any feature, we dealt with it in our developer branch. Then we tested it with unit test. After that, we make sure the whole system is not affected for the changes. For this reason, we ran integrated test to the system making sure the decoupling is working properly. Then we pushed the whole system to master and in production. Now, if we found something horrendous, we jump to fix it in as soon as possible. We open the issue under hotfix brunch and solve it, test it, integrate it and deploy it master and in production as soon as possible.

5.5 Immunity Requirements:

For giving immunity, we want to host our server under a SSL(Secure Socket Layer) certified domain. By doing this, we can protect our user data from their ISP's. Since SSL will enable https, user are at least secured until the server hits the NAT wall. Another important thing is database security, now this is implemented in backend that a document of a struct can only be retrieved if it belongs to the certain struct. For the security, the whole system is divided into four parts where the database is completely modularized into a separate module. Now, for achieving data latency we integrated the whole system under data module and tested it under unit test. Finally, we tested the integrity test after integrating the database module with the whole system.

CHAPTER 6

Test Plans

6.1 Features to be tested:

Test plan is a document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks.

Features those are tested:

- A user can not access an account’s data without login into that account.
- A student can not act as a faculty and vice versa.
- Each course can only have exactly one faculty.
- Whenever a faculty adds a student into a course, the student gets access to the course. Even if the student registers later, he/she can view the courses in which they are already enrolled. That keeps faculty’s activity completely independent from student’s activity. That means a faculty doesn’t have to wait for a student to register before adding him/her into a course.
- All types of error in the error handling part.
- Valid email address.
- Password’s length is between six and thirty characters.
- Password and retype password are same.
- During password change, the current password is correct.
- During editing marksheet, if any faculty writes something which is not a number, the calculation part in the server ignores that and consider it as zero.
- Parsing data in a CSV file.
- Group privacy matters. No one except the group member and the faculty has access to the group.
- Nominees for group members are the students those have not enrolled in any group yet.
- Files in documents and resource can download.
- When a user updates profile picture, the photos in his/her previous discussion, comments and post also change.

- A faculty can change the time of a submission window. Depending on the change the closed and open status updates.
- One cannot submit after a submission window is closed.
  - Depending on the grades in the marksheet, the course details in preregistration changes.

6.2 Approach:

There are mainly four levels of testing

- Unit testing
- Integration testing
- System testing
- Acceptance testing

In Agent, we have only performed Unit testing and Integration testing.

Unit testing

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. The primary goal of unit testing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as expected. Each unit is tested separately before integrating them into modules to test the interfaces between modules.

Each module that is developed by designers needs to be tested individually to verify proper operation so that any faulty module can be fixed immediately rather than let it exist and then cause some major issue in the integration phase. Once all of the units in a program have been found to be working efficiently and without any bugs, larger components of the program can be evaluated by means of integration testing. Unit testing may be time consuming, tedious and requires thoroughness on the part of the development team, but in the long run it can avoid major pitfalls in the software.

Integration testing

integration is the process of assembling unit-tested modules. Developers need to test the aspects, which have not been addressed previously while independently testing the modules.

First accept is to ensure "interface integrity," the transfer of data between modules is tested. When data is passed to another module, by way of a call, there should not be any loss or corruption of data. The loss or corruption of data can happen due to mis-match or differences in the number or order of calling and receiving parameters.

Second accept is that module combinations may produce a different behavior due to combinations.

6.3 Outcomes:

In unit testing, we gave inputs and according to the code, the application generated output. If any output was unexpected, we debugged our code, fixed the problem and then ran again. This process continued until we get our preferred output from all the functionalities.

In integration testing, we checked data transfer between one module to another. For example between browser and server, between server and database. We stopped checking when all the data transfer seems correct.

CHAPTER 7

Project Summary

7.1 Result and Discussion:

We finished the project successfully.We have accomplished what we are supposed to complete. We tested the whole system thoroughly. First, we divided the whole thing based on the module. Then for each module we ran the unit test. After integrating the whole system, we ran integration test to the whole system. Since every cases passed based on the given test suite, we can assume that we are okay with most of the cases.
Security is another most important thing to consider. We took care of the security in different section. From Database modularization to password, from routing to model integration every is field is covered. We also took care of data immutability. Most of the front end data are immutable.

7.2 Feasibility Study:

While making the app, we were concerned with the outcome of the app.Because, whatever happens in the technology side, the app must need to be user friendly.
If a user needs to find what we offer, then we have made wrong decisions in our design system. So, for this reason, we tested it by giving the software to the multiple people to use it as their daily software for class management. Most of the students and faculty were very satisfied and it was very easy to operate.
Few faculties requested for some extra features. And, we hope to add it in our future releases. Students were also happy to see everything in one place. They have also suggested to add few things, so that it make their life more easier.
Based on those test, we can say, our app is feasible to the user at all aspects.

7.3 Problem Faced and Solution:

The very first and the big problem that we faced was the language that we chose during our 499A.The language was Go. Go is comparatively a new language and it has limited resource on the internet. It has no framework which can provide boilerplate for MVC architecture. So all together it’s difficult for a beginner to land on web developing world with this language. We solved this problem by shifting into a new language in our 499B. The language is JavaScript and the runtime environment to execute the JavaScript code in server-side is Node.js. Node has many frameworks but Express framework is one of the best. It has great support and a bunch of helpful features.

Next problem that we faced was the authentication problem. Once user logged in, he/she got the power to enter into other accounts. How? Let’s say your homepage URL is localhost:3000/user/asif, if you changed the URL to localhost:3000/user/imr it could have taken you to the homepage of the user with username imr. So you could have got the access to the another person’s homepage without logged into his account. So it was a big problem. We fixed it by attaching middlewares with every possible route. Before proceeding any user request, these middlewares check the user authentication.

There are the tremendous number of other problems we faced throughout the developing process. But since JavaScript has become one of the most popular languages in recent years, due to the humongous demand for web applications and so as the framework Express. It was not very difficult to find a solution to any problem from the internet.

7.4 Future Development:

- In future, we will focus more on the student portal. We will try to predict student grade from the current marks by the help of AI.
- Will try to resolve RACE conditions, those may discover by users during using the app.
- Will change our database schema to more simple and effective one.
- We will allow faculties to make customize assessment according to their wish.
- Now faculties have to provide grade but in future, the grade will be automatically generated by our system.
- Will implement a better user interface for marksheet.

- Will add the functionality where faculty can give mark from the submission window.

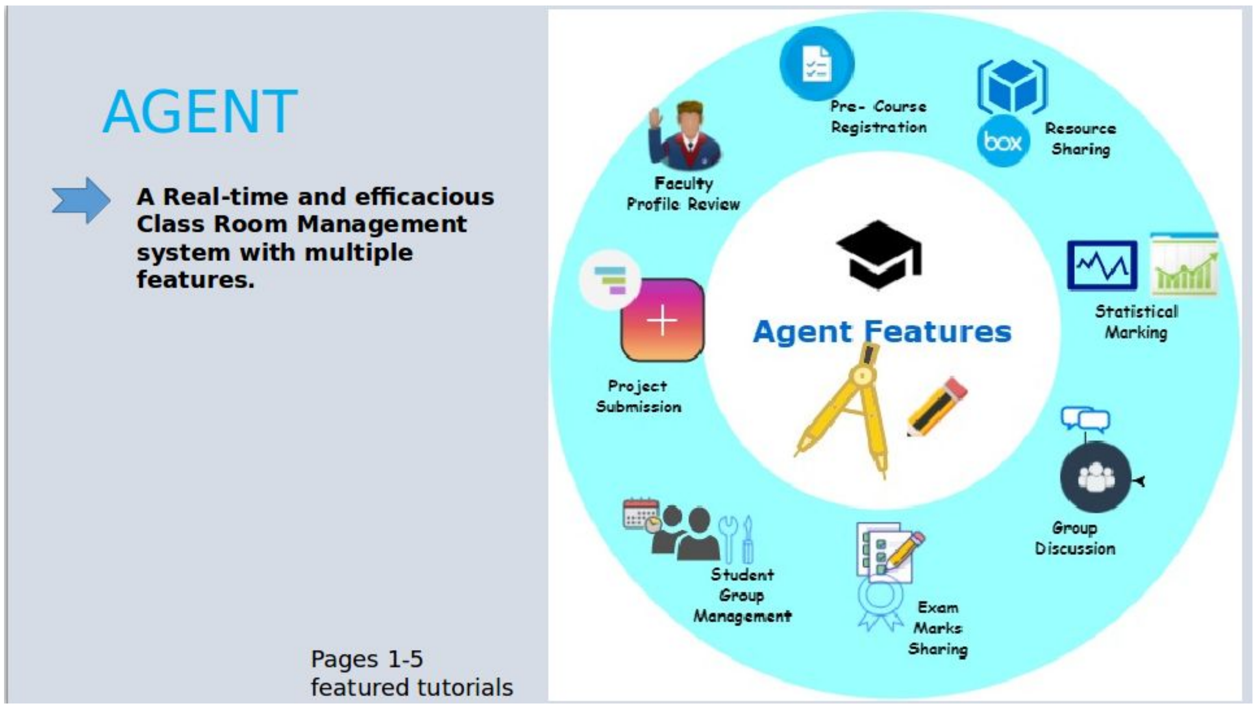
7.5 Conclusion:

This application, Agent, helps in managing the class and grading the students as well as deciding to open a new section for a course over a pre-registration feature. After collecting all the information regarding classroom management and grading as well as seat allocation problem and many others, we came up to the solution building this software with desired outcome. This management system finally can give an idea of whether to open a new course for the next semester or not by seeing overall class performance and the "pre-registration" feature. In the future it will be also possible for students to solve problems and also for admins to decide the quality of students in real time.

7.6 Project Demonstration Review:

After testing and demonstrating, the outcome was outstanding. Demonstration held in the University arena and reviews from both of the massive collection of students and faculties of the various departments are gathered. Students found the "ease-of-use" and more featured than the existing other solutions. And the statistical part of the software attracts the faculties more bye titling "organized grading and student quality". The assignment publishing and submission features also very innovative in terms of time and date picking and showing exact due time for the project and assignment. The group management feature in the software was very idealistic and useful, reviewed from the software engineering faculties. Overall the reviews were on behalf of the pros, less number of cons and all reviews shows the future of this software is a big picture in the world wide web era.

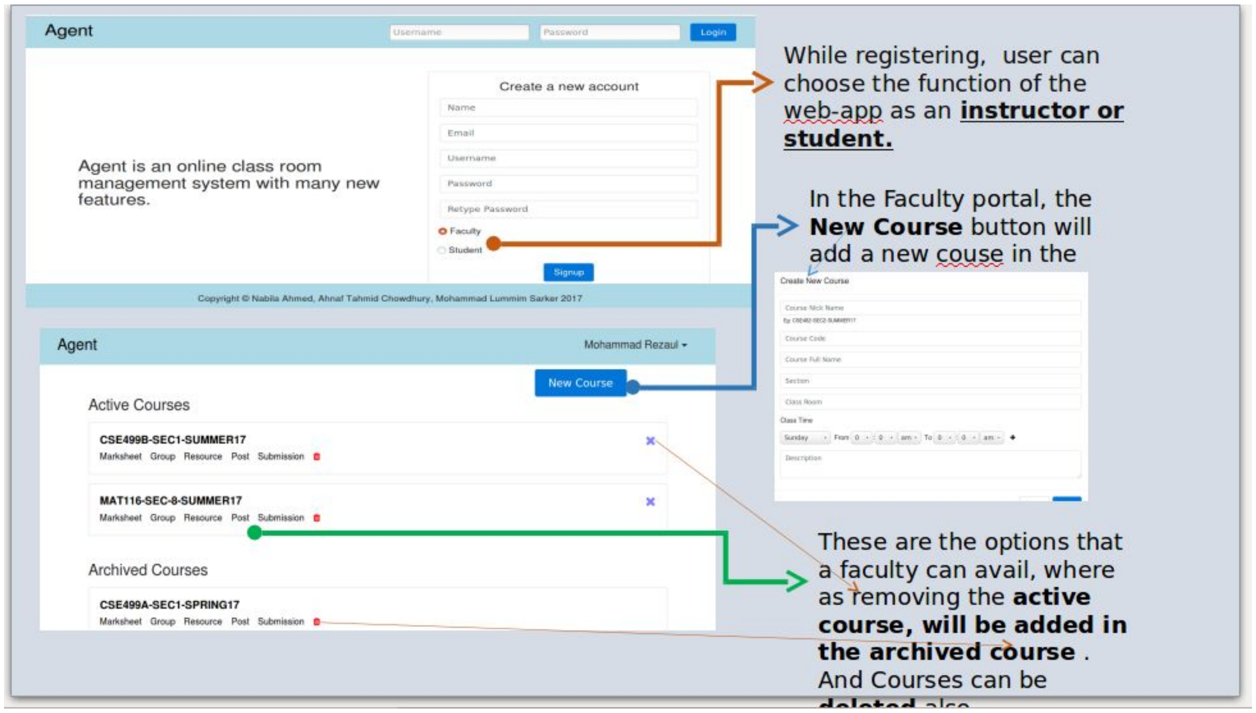
7.7 Features Book:



AGENT

A Real-time and efficacious Class Room Management system with multiple features.

Pages 1-5
featured tutorials



Agent

Agent is an online class room management system with many new features.

Create a new account

Copyright © Nabila Ahmed, Ahnaf Tahmid Chowdhury, Mohammad Lummim Sarker 2017

Agent

Mohammad Rezaul

Active Courses

CSE499B-SEC1-SUMMER17

MAT116-SEC-8-SUMMER17

Archived Courses

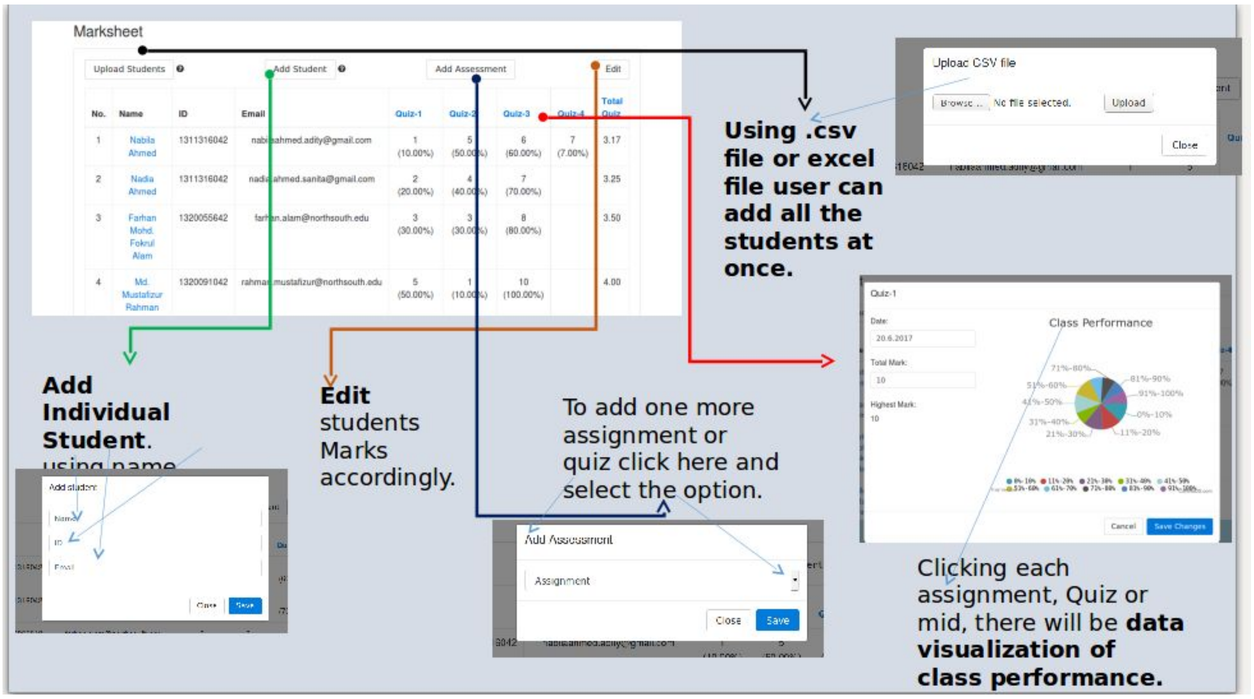
CSE499A-SEC1-SPRING17

While registering, user can choose the function of the web-app as an **instructor or student.**

In the Faculty portal, the **New Course** button will add a new couse in the

Create New Course

These are the options that a faculty can avail, where as removing the **active course, will be added in the archived course** .
And Courses can be



Marksheet

No.	Name	ID	Email	Quiz-1	Quiz-2	Quiz-3	Quiz-4	Total Quiz
1	Nabila Ahmed	1311316042	nabilaahmed.adity@gmail.com	1 (10.00%)	5 (50.00%)	6 (60.00%)	7 (7.00%)	3.17
2	Nadia Ahmed	1311316042	nadia.ahmed.sanita@gmail.com	2 (20.00%)	4 (40.00%)	7 (70.00%)		3.25
3	Farhan Mohd. Fokrul Alam	1320055642	farhan.alam@northsouth.edu	3 (30.00%)	3 (30.00%)	8 (80.00%)		3.50
4	Md. Mustafizur Rahman	1320091042	rahman.mustafizur@northsouth.edu	5 (50.00%)	1 (10.00%)	10 (100.00%)		4.00

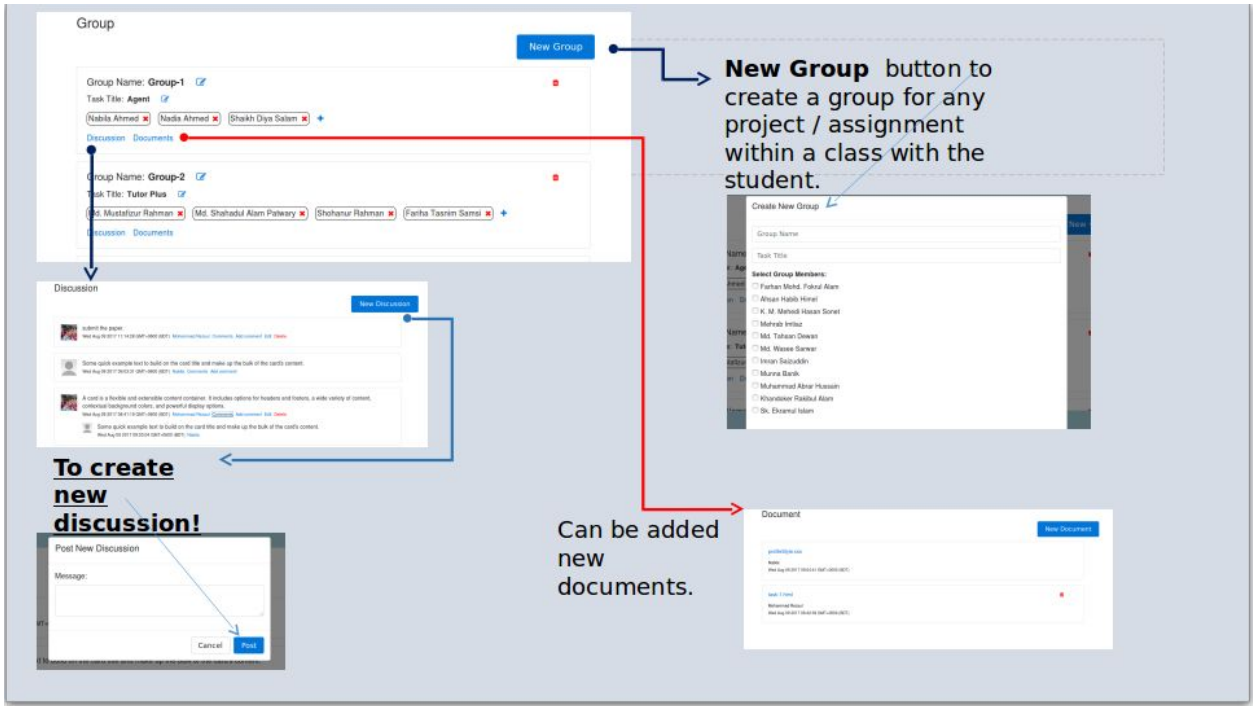
Using .csv file or excel file user can add all the students at once.

Add Individual Student. using name

Edit students Marks accordingly.

To add one more assignment or quiz click here and select the option.

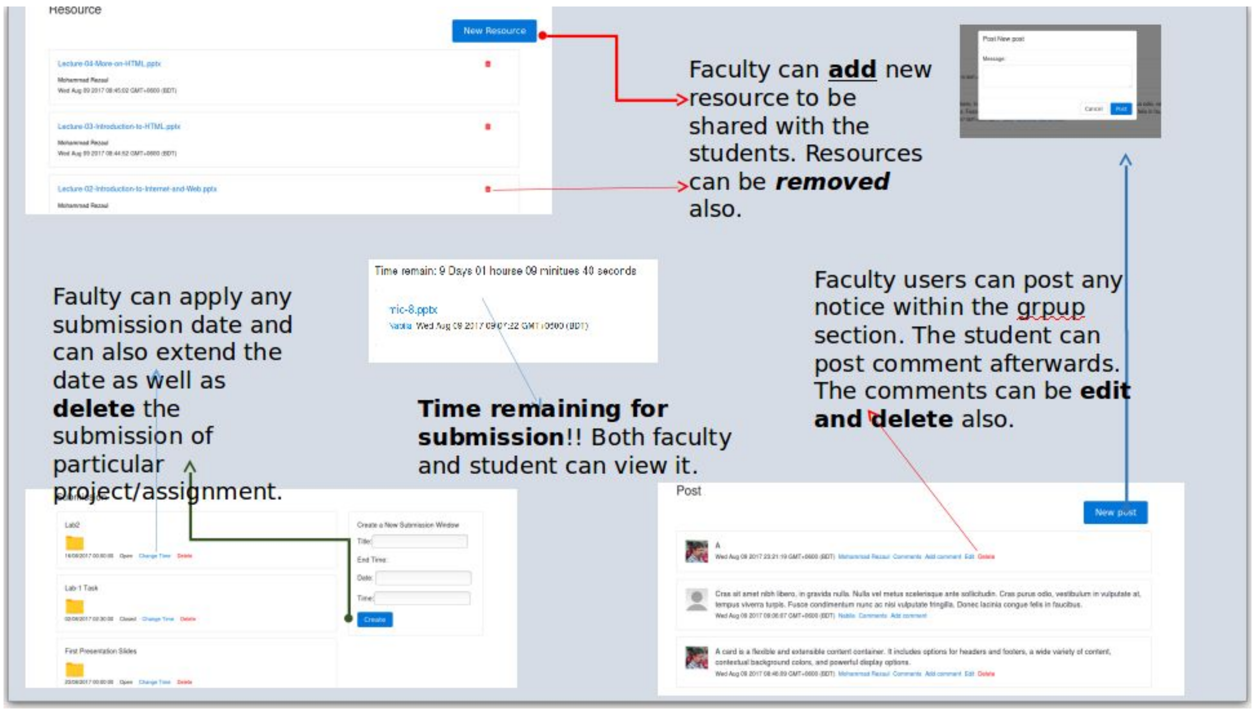
Clicking each assignment, Quiz or mid, there will be **data visualization of class performance.**



New Group button to create a group for any project / assignment within a class with the student.

To create new discussion!

Can be added new documents.

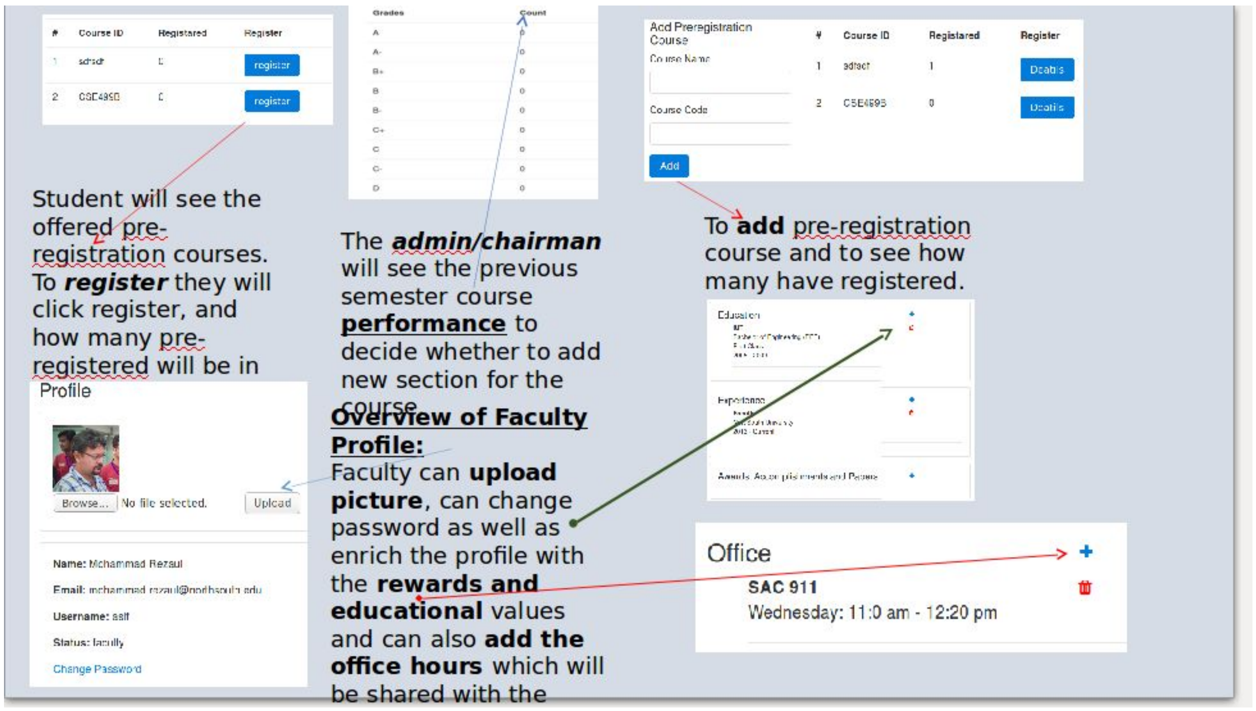

Resource
New Resource
Lecture-04-More-on-HTML.pptx
Lecture-03-Introduction-to-HTML.pptx
Lecture-02-Introduction-to-Internet-and-Web.pptx
Post New post
Message:
Cancel
Post
Time remain: 9 Days 01 hourse 09 minitues 40 seconds
Lab2
Lab 1 Task
First Presentation Slides
Open
Closed
Change Time
Delete
Create a New Submission Window
Title:
End Time:
Date:
Time:
Create
Post
New post
Comments
Add comment
Edit
Delete

Faculty can **add** new resource to be shared with the students. Resources can be ***removed*** also.

Faulty can apply any submission date and can also extend the date as well as **delete** the submission of particular project/assignment.

Time remaining for submission!! Both faculty and student can view it.

Faculty users can post any notice within the grpup section. The student can post comment afterwards. The comments can be **edit and delete** also.



#	Course ID	Registared	Register
1	sdfsdf	0	register
2	CSE499B	0	register

Grades	Count
A	0
A-	0
B+	0
B	0
B-	0
C+	0
C	0
C-	0
D	0

#	Course ID	Registared	Register
1	sdfsdf	1	Deatils
2	CSE499B	0	Deatils

Name: Mohammad Rezaul
Email: mohammad.rezaul@northsouth.edu
Username: asif
Status: faculty

SAC 911
Wednesday: 11:0 am - 12:20 pm

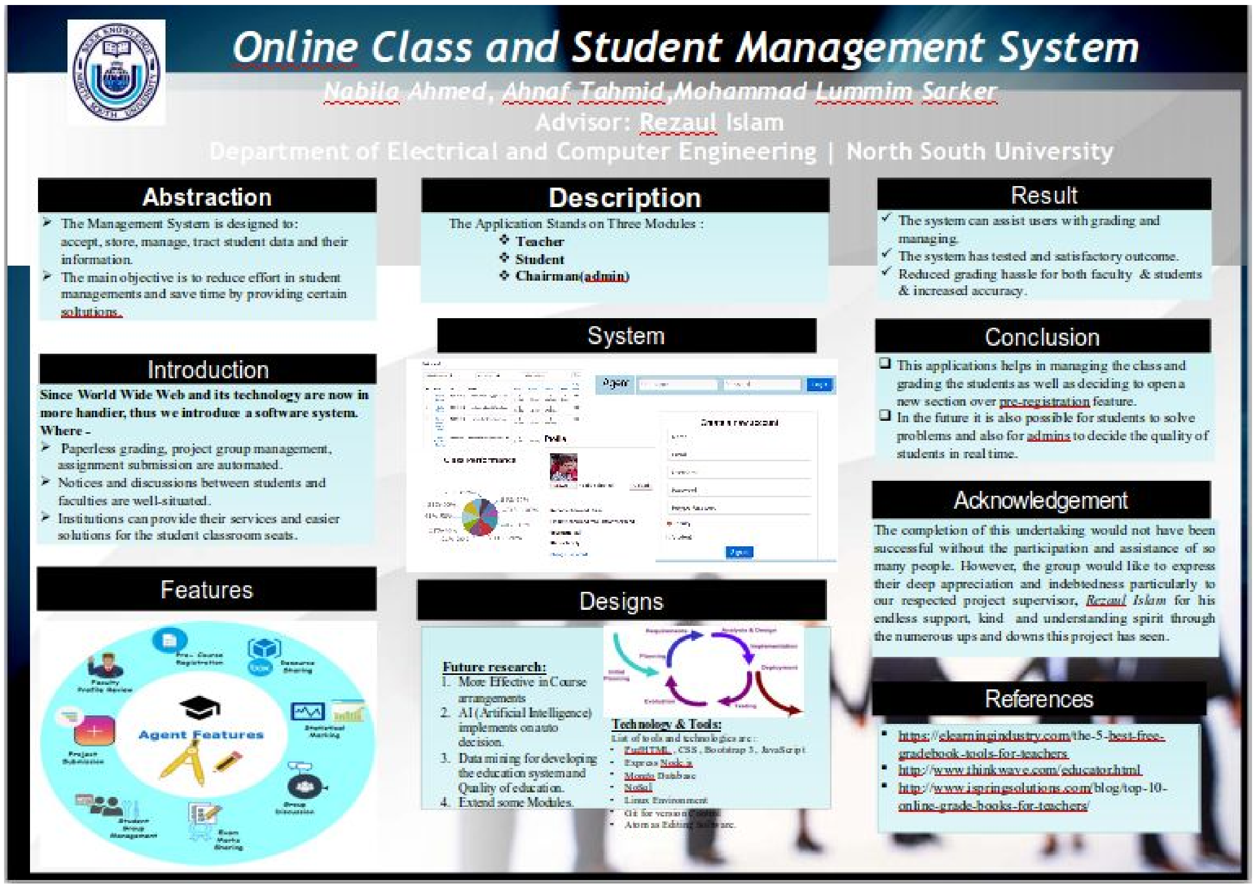
Student will see the offered pre-registration courses. To ***register*** they will click register, and how many pre-registered will be in

The ***admin/chairman*** will see the previous semester course **performance** to decide whether to add new section for the course.

To **add** pre-registration course and to see how many have registered.

Overview of Faculty Profile:
Faculty can **upload picture**, can change password as well as enrich the profile with the **rewards and educational** values and can also **add the office hours** which will be shared with the

7.8 Poster:


Agent Features
Pre- Course Registration
Resource Sharing
Statistical Marking
Group Discussion
Exam Marks Sharing
Student Group Management
Project Submission
Faculty Profile Review
Requirements
Analysis & Design
Implementation
Deployment
Testing
Evaluation
Planning
Initial Planning

Online Class and Student Management System

Nabila Ahmed, Ahnaf Tahmid,Mohammad Lummim Sarker
Advisor: Rezaul Islam
Department of Electrical and Computer Engineering | North South University

Abstraction

- ➢ The Management System is designed to: accept, store, manage, tract student data and their information.
- ➢ The main objective is to reduce effort in student managements and save time by providing certain solutions.

Introduction

Since World Wide Web and its technology are now in more handier, thus we introduce a software system. Where -

- ➢ Paperless grading, project group management, assignment submission are automated.
- ➢ Notices and discussions between students and faculties are well-situated.
- ➢ Institutions can provide their services and easier solutions for the student classroom seats.

Features

Description

The Application Stands on Three Modules :

- ❖ **Teacher**
- ❖ **Student**
- ❖ **Chairman(admin)**

System

Designs

Future research:

1. More Effective in Course arrangements
2. AI (Artificial Intelligence) implements on auto decision.
3. Data mining for developing the education system and Quality of education.
4. Extend some Modules.

Technology & Tools:

List of tools and technologies are :

- EurlHTML , CSS , Bootstrap 3 , JavaScript
- Express Node.js
- Mongo Database
- NoSql
- Linux Environment
- Git for version Control
- Atom as Editing Software.

Result

- ✓ The system can assist users with grading and managing.
- ✓ The system has tested and satisfactory outcome.
- ✓ Reduced grading hassle for both faculty & students & increased accuracy.

Conclusion

- ❑ This applications helps in managing the class and grading the students as well as deciding to open a new section over pre-registration feature.
- ❑ In the future it is also possible for students to solve problems and also for admins to decide the quality of students in real time.

Acknowledgement

The completion of this undertaking would not have been successful without the participation and assistance of so many people. However, the group would like to express their deep appreciation and indebtedness particularly to our respected project supervisor, *Rezaul Islam* for his endless support, kind and understanding spirit through the numerous ups and downs this project has seen.

References

- https://elearningindustry.com/the-5-best-free-gradebook-tools-for-teachers
- http://www.thinkwave.com/educator.html
- http://www.ispringsolutions.com/blog/top-10-online-grade-books-for-teachers/

REFERENCES

APPENDIX A

[1] *Getting Started - Pug*. Retrieved from website: https://pugjs.org/api/getting-started.html

[2] *Interpolation - Pug*. Retrieved from website: https://pugjs.org/language/interpolation.html

[3] *Bootstrap*. Retrieved from website: https://v4-alpha.getbootstrap.com/

[4] *Font Awesome*. Retrieved from website: http://fontawesome.io/

[5] *W3Schools Online Web Tutoria*l. Retrieved from https://www.w3schools.com/

[6] *CanvasJS Pie Chart*. Retrieved from https://canvasjs.com/docs/charts/chart-types/html5-pie-chart/

[7] *Express*. Retrieved from https://expressjs.com/

[8] *Express Routing*. Retrieved from https://expressjs.com/en/guide/routing.html

[9] *Mongoose*. Retrieved from http://mongoosejs.com/docs/documents.html

[10] *Stack Overflow*. Retrieved from https://stackoverflow.com/

Source Code APPENDIX B

Source Code: https://github.com/NabilaAhmedAdity/agent.v.2