



# **Advanced Software Engineering**

**Course # CSEE 5551 0001**

**Semester: Spring 2017**

**Project Increment #2 Report**

**Submission Date: 2/17/2017**

## **TEAM 10**

Ravali Bolem (UMKC ID: 16231463; Class ID: 7)

Ashwini Gutha (UMKC ID: 16233065; Class ID: 33)

Pranathi Gopidi (UMKC ID: 16233917; Class ID: 29)

Satya Sai Deepthi Katta (UMKC ID: 16231371; Class ID: 41)

**Instructor: Dr. Yugyung Lee**

**Email: LeeYu@umkc.edu**

# *Welcome to Project Increment #2 Report!*

PROJECT TITLE:

**PLANOVAC** (MEANS "SCHEDULER" IN CZECH LANGUAGE)

## **I. INTRODUCTION:**

### ➤ **Problem Statement:**

It is scientifically proved that work tensions make people stressful, which directly effects their functioning of the mind. This ineffective functioning mainly causes people to loss some parts of their memory. This leads to lack of proper work, not able to meet their deadlines, couldn't manage time and work, no work life balance.

As we all are aware of many remainder apps in the market. But the questions raised are mentioned below:

QUESTIONS?

How effective are the apps in the market?

What is the percentage utility of the apps?

Are these for all kinds of people like business people, students, professionals etc.?

By using the app effectively, what is the significant increase in their managing of time?

How many App's mainly focus on students?

## II. PROJECT GOALS

### ➤ Overall Goal:

In the scenario of vast set of people, students are the main focused study group who needs to meet the deadlines without fail. Many people have the backup plans if they miss the scheduled dates, what if students skip their deadlines, their grades go down, re-registering the course, redoing the work, need to spend lots of money etc. etc. In order to deal with such issues, our application makes on making the application which portraits the problems faced by students and gives the simple solution called the PLANOVAC which means Scheduler in Czech Language.

### ➤ Specific Features:

- Students can register all their class schedules in the application and set the time for alerts (For eg. 10 mins before the class starts).
- Then the application notifies the user regarding important tasks.
- It reads all the contacts from the user's phone and allows collaboration with those who possess an account in this application.
- The students can then initiate individual or group chats.
- Even the instructors can also communicate with the subjects through discussion forums.
- This application also displays the location of various classrooms according to our class schedule.

### ➤ Significance/ Uniqueness:

This application can be particularly very useful for students. Students will be able to register all their important tasks in the application and the application sends periodic alerts to users regarding the status of these tasks.

### III. DETAILED DESIGN OF FEATURES

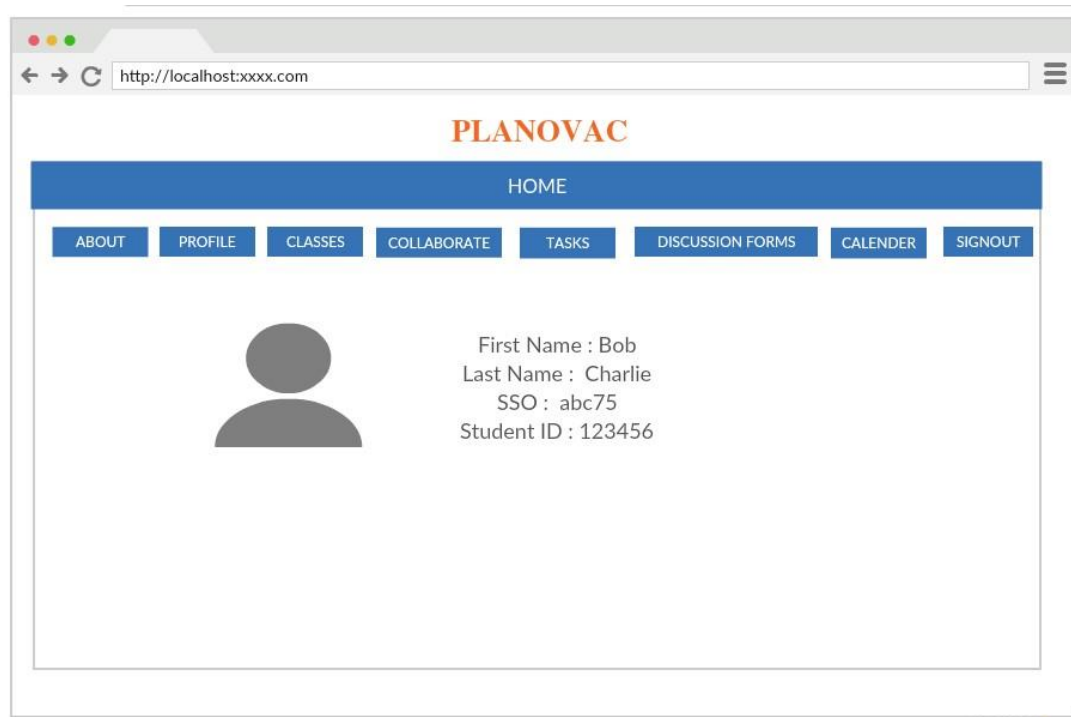
The features in our Web Application 'Planovac' are discussed giving every detail through Wireframes, Architecture diagrams etc.

#### (a) WIREFRAMES AND MOCKUPS:

Wireframes are created through a web service for designing called CREATELY.COM.

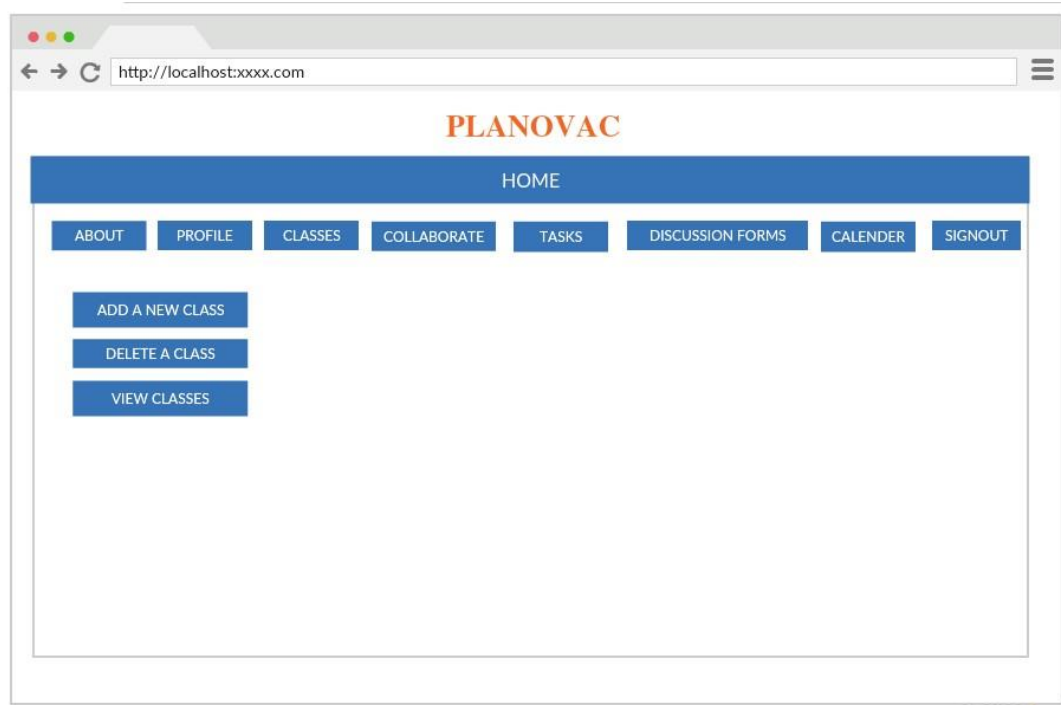
##### *Feature 1 - Profile*

Profile is the basic feature of any application which showcases the user account details of the login user. This provides the information of the user when logged in from gmail. The profile informations are stored in Mongo Database.



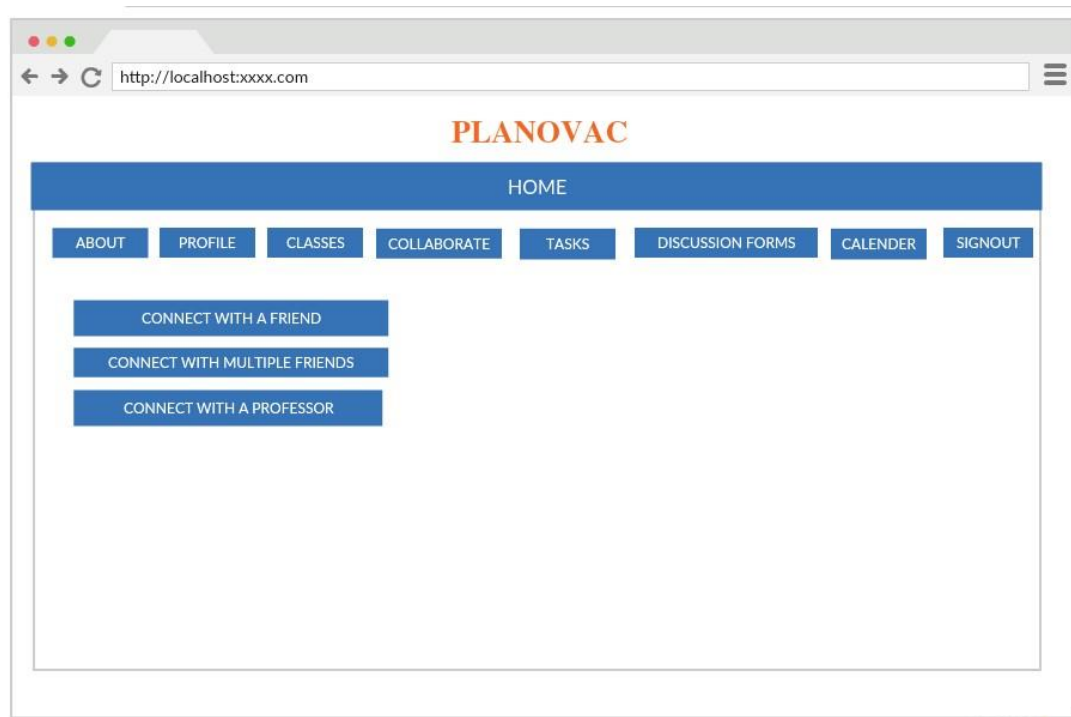
##### *Feature 2 - Classes*

Planovac gives the student an opportunity to store their class schedule, so that they get notifications time to time to maintain their schedules well. The schedules are given stored in Mongo Database which is interlinked with Google calendar to access the events.



### ***Feature 3 - Collaborate***

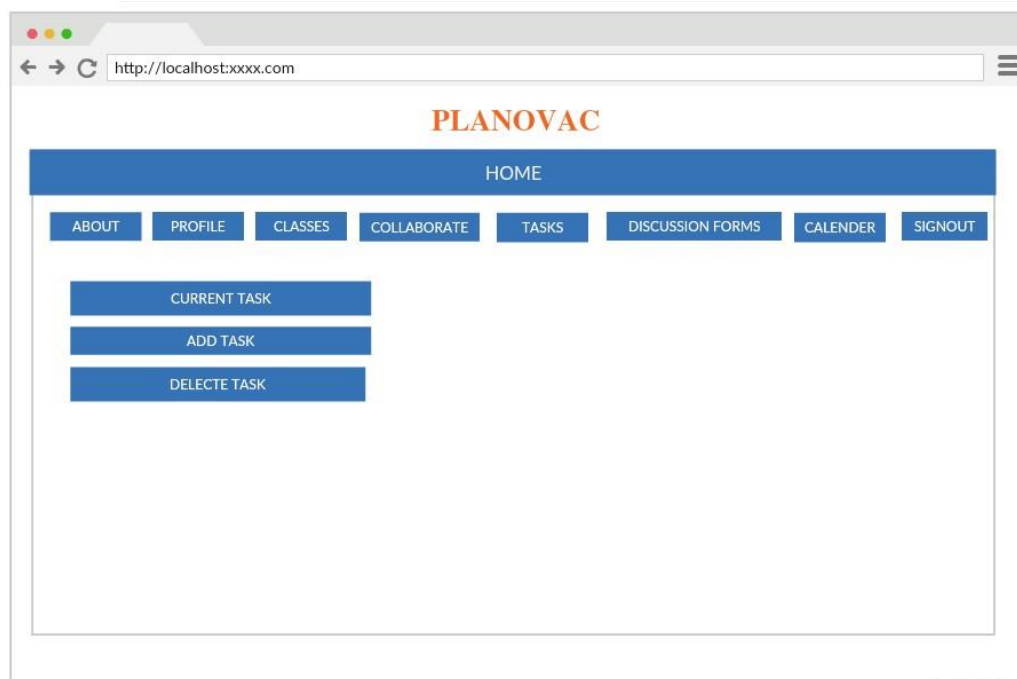
Collaborate is similar to the Chat Application which are in the market like Watsapp, Facebook Messenger. But Planavoc gives the flexibility only for the students to interact with each other or with couple of other friends to chat privately.



#### ***Feature 4 - Tasks***

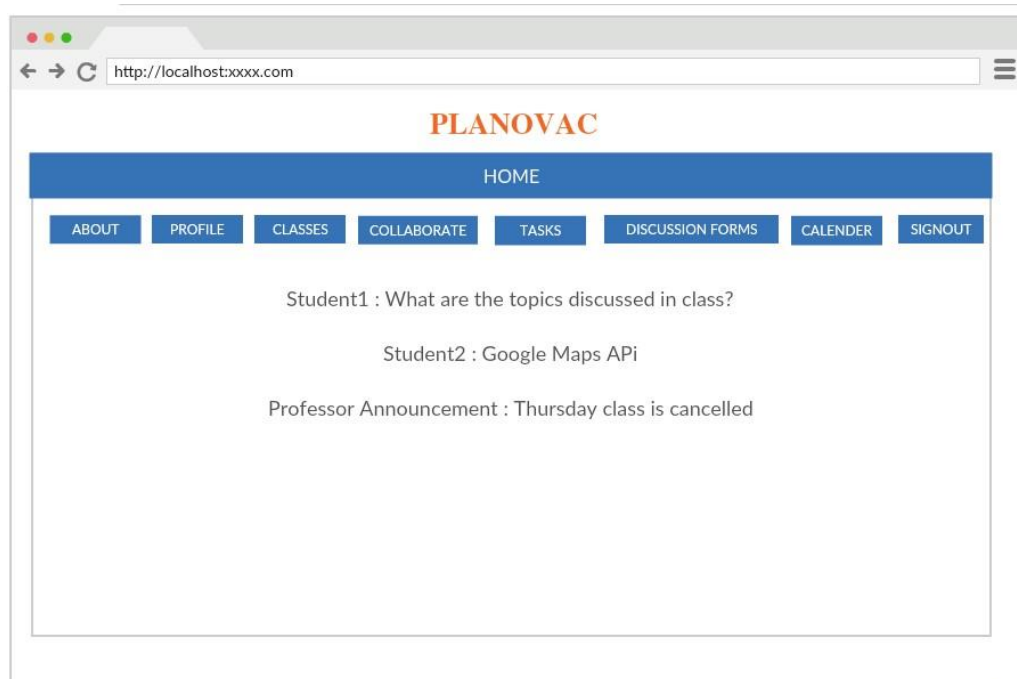
Students need to plan their work to complete their assignments on time, so they can assign tasks in Planovac. These tasks remind them of their work yet to complete. They show the colour notifications indicating the days left to complete the task.

For example, Red indicates dates near to the deadline while Green indicates the safer days yet to start the work.



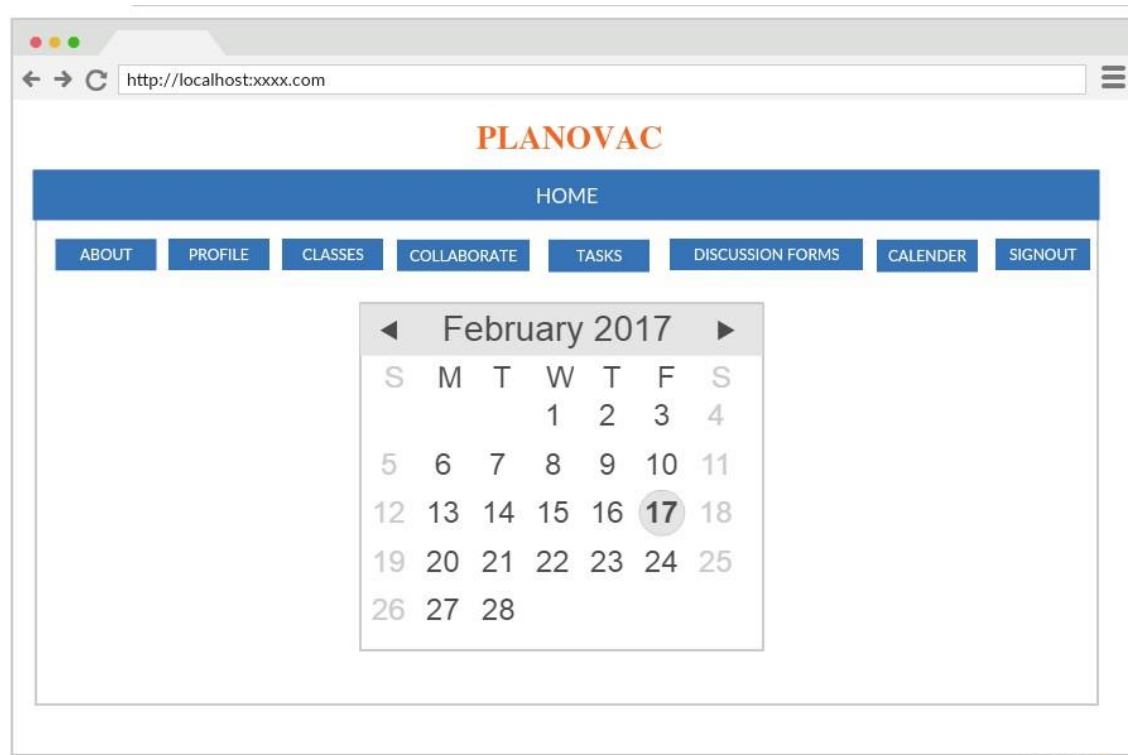
#### ***Feature 5 - Discussion Forum***

This is different to the collaborate of Planovac. The discussion forum heps the students to discuss the knowledge and clarify the doubts with friends and even with professor. Professor can post important information about test syllabus, cancellation of classes etc in Discussion Forum . This Forum helps students to expand their knowledge by having healthy discussion with their classmates.



### ***Feature 6 – Calendar***

Planavoc calendar includes the dates of important events and shows added by the students. This runs with the Google Calendar API. Notifications also send to students based on the events stored in the Mango database.



## (b) UML MODELING

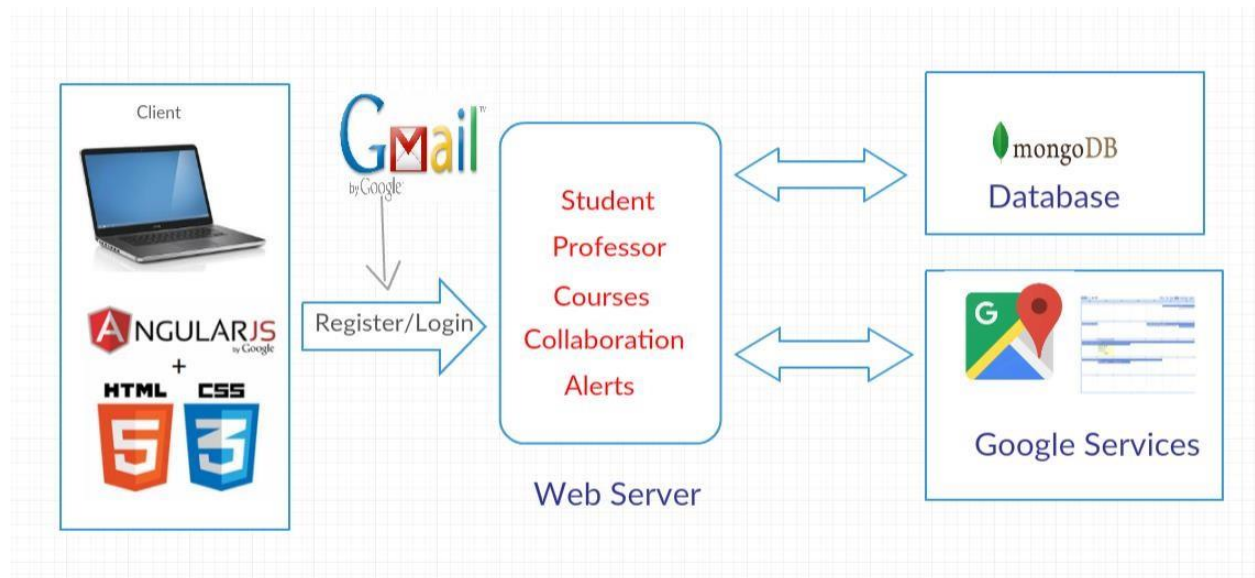
The best way to understand the features of the project is to look for their diagrammatic representations. UML diagramming is the Unified Modeling Language which portrays the project particulars through pictures and helps the user to have complete idea of the diagram. Three of such representations are used in our project to show case the features of the application.

- Architecture Diagram
- Class Diagram
- Sequence Diagram

### 1. Architecture Diagram:

Architecture Diagram is the pictorial representation of the internal functioning of the software. The components like system, database etc are indicated as the blocks in the picture. The connected lines heading with arrows picturizes the direction of the connection.

The architecture diagram for our project is presented below:

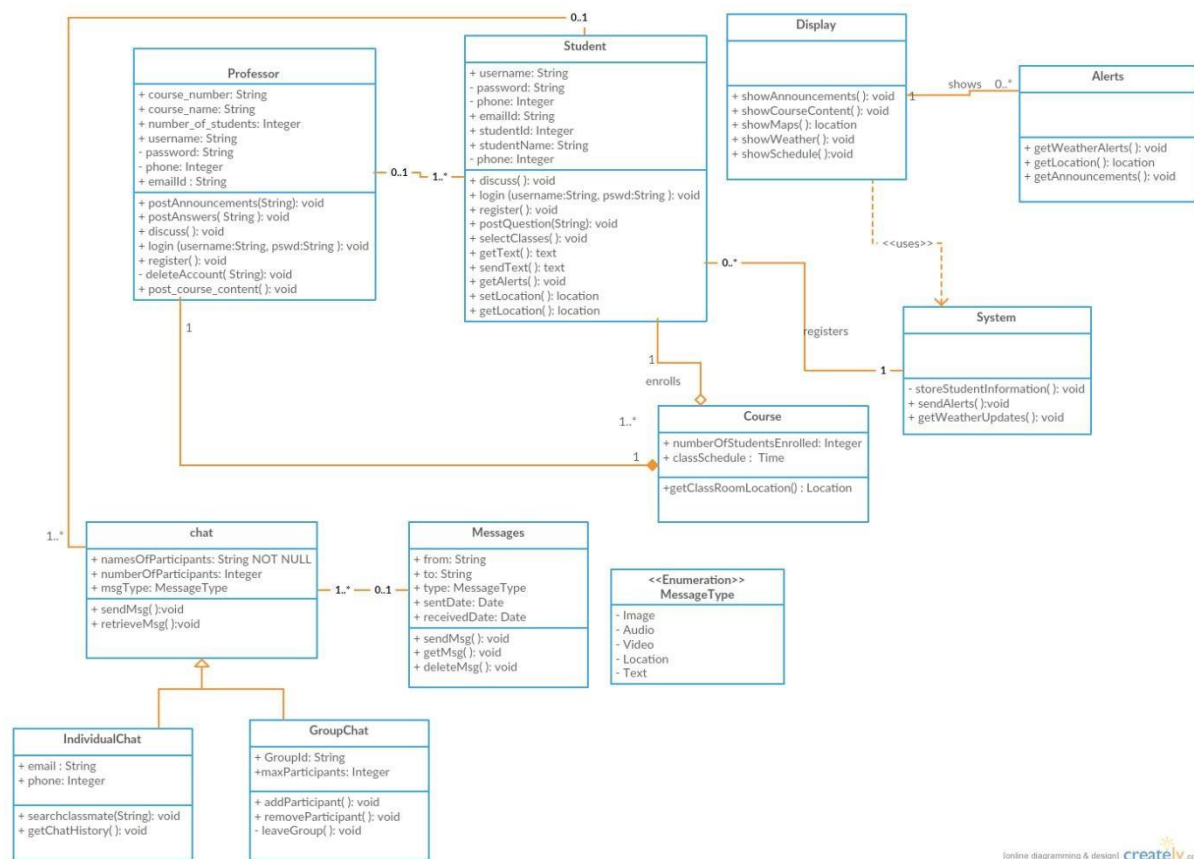




## 2. Class Diagram:

For any software, the tasks are modeled and detailed descriptions are present in the form of classes. The Multiplicity, navigability etc are presented with the arrows showing aggregation, composition and generalization between the classes.

The class diagram for the project is presented below:



In the class diagram, the classes named professor, students, display, system, alerts, class and chat are created.

The relation between student and class describes the Aggregation (whole-part relationship) where a student is a part of a class.

Chat contains individual and group chat where a person can even share multimedia messages like audio, video, location and images.

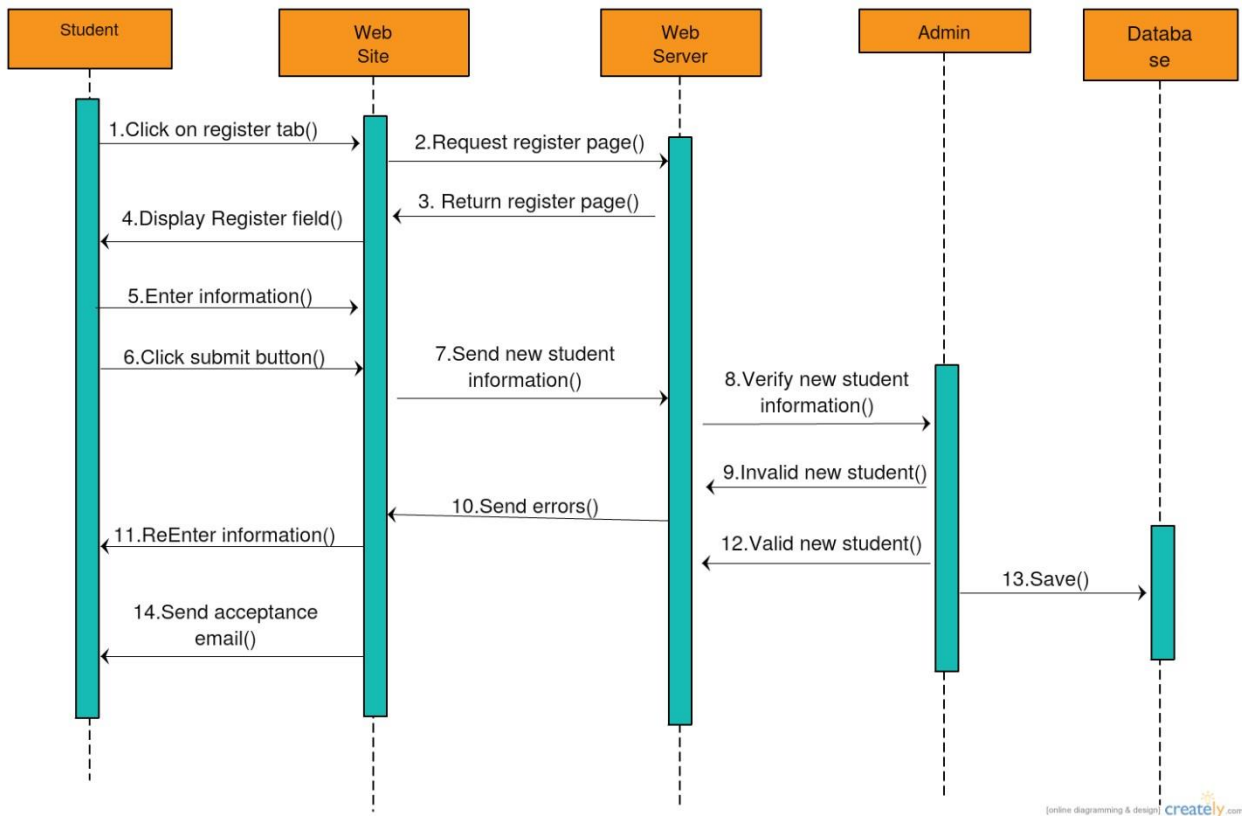
Professor can post announcements, course contents, provide answers for the questions posted by the students and involve in the discussion with the students.

Students can login using their credentials, post questions, get the location to the class, get class schedule and get alerts regarding weather updates as well.

### 3. Sequence Diagram:

A Sequence diagram is the interactive diagram displaying how the objects operate or communicate with one other. This diagram is usually drawn for a particular use case. This handles the possible ways for the given use case.

The Sequence diagram depicted below has the use case for New Student Registration into Web Application.



### **(c) USER STORIES**

The User Story is the helpful operational tool which gives the information of what user requires from the application developed. The user stories can be developed by taking feedback, conducting surveys etc. The user story helps to improve the application which makes it circulate to more and more people. The significance of the user story is to give sufficient information for a developer to develop a application effectively and accurately.

#### Feedback:

The feedback obtained from the technical people who viewed our application while conducting survey for the user stories for Project Increment 2.

Q1. How useful is “Planovac” ?

It is easy to operate as it is Web Application. Its main focus is students which makes it unique from market applications today. Students find to easy to make schedules and tasks which helps them to meet their deadlines.

Q2. What are suggested changes for Increment 2 by TA?

Need to change user interface created from templates which doesn't look good.

Q3. What is the good feature implemented in the application ?

It is challenging to Web Application which contains multiple applications like chat feature, including tasks, storing in larger database. The best part of the application is to implement discussion board which helps students to not only interact with students but also professors without any prior appointment.

Q4. What do you dislike about the app?

This is a web application which doesn't give the flexibility to convert to mobile application.

## **IV. EXISTING API**

An API is the application programming interface which have the set of protocols, rules and design components which specifies the software components how to interact among themselves. Many leading companies like Google, Facebook, Twitter give the developer their programmable API's which give the developers to access their unique features in their own applications.

### **Google Calendar API:**

In PLANOVAC, we have used Google Calendar API for Calendar feature. This web application uses the Google calendar API, where students can add their events and seminars on the desired dates. This also provides the students with remainder mails and notifications before the class timings or any important seminars. This constantly updates itself based on the settings provided by the user like the remainder set for every hour before class starts or alarm beeps before seminar starts. The updates are posted through emails.

### **Mongo Database:**

Mongo DB is the larger database which stores the values of created in the applications. In PLANOVAC, we have used Mongo DB to store the values created in the Web Application. As the application have lots of values to be stored like the profile synced through google amount, discussion forums created while discussing an issue etc. Whenever the application needs any information to be displayed on the Webpage, it retrieves the information stored from the database.

## **V. TESTING**

Testing is the basic operation which gives the information about how good the application is running. This is created in many ways. Basically the required information about the status of the application is given by doing Unit Testing.

### **Unit Testing:**

This is the software development process. In this, a small testable parts are used called Units. Various units of the application like the login, registration, etc are to know the operation status of the application being developed.

Among various ways to do Unit Testing both manually and through software tools. As manual testing results are shown in Increment. We have used three renowned tools to unit test our web application in Increment 2.

**(1) *Selenium IDE* :**

Selenium IDE is integrated development environment suitable mainly for Mozilla Firefox web browser. This testing records each click on the application and gives the unit test results and their status in a tabular format. This also allows to have suite cases which involves testing for more than two web applications at a time.

**(2) *YSlow***

Yslow is an open source website performance testing plugin. This plugin is compatible with Google Chrome web browser. This gives grading for the websites which helps to understand the standards of the website by looking at the grades. The overall grade is based on the average of all the grades given to each unit test done on the website.

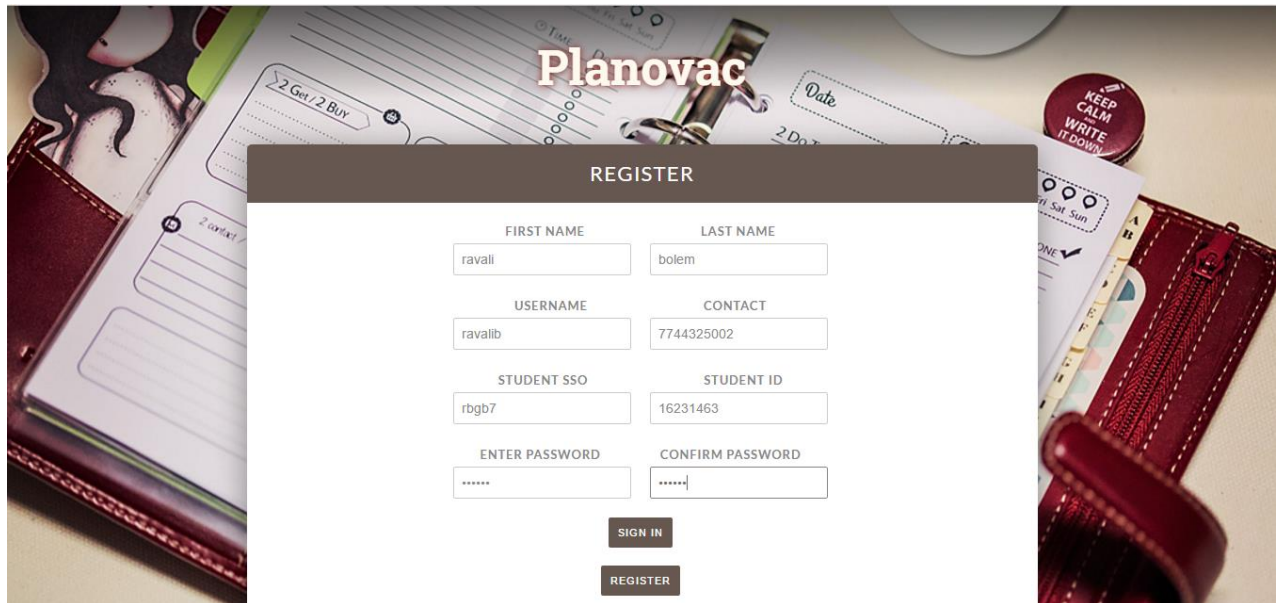
**(3) *Firebug***

Firebug is the web development tool which is flexible to Mozilla Firefox web browser. This takes into account of all CSS, Javascript and HTML code involved in developing the application. Firebug analyses all these code structures and presents errors and debug comments which helps to improve the level of coding in Webpage.

## VI. IMPLEMENTATION

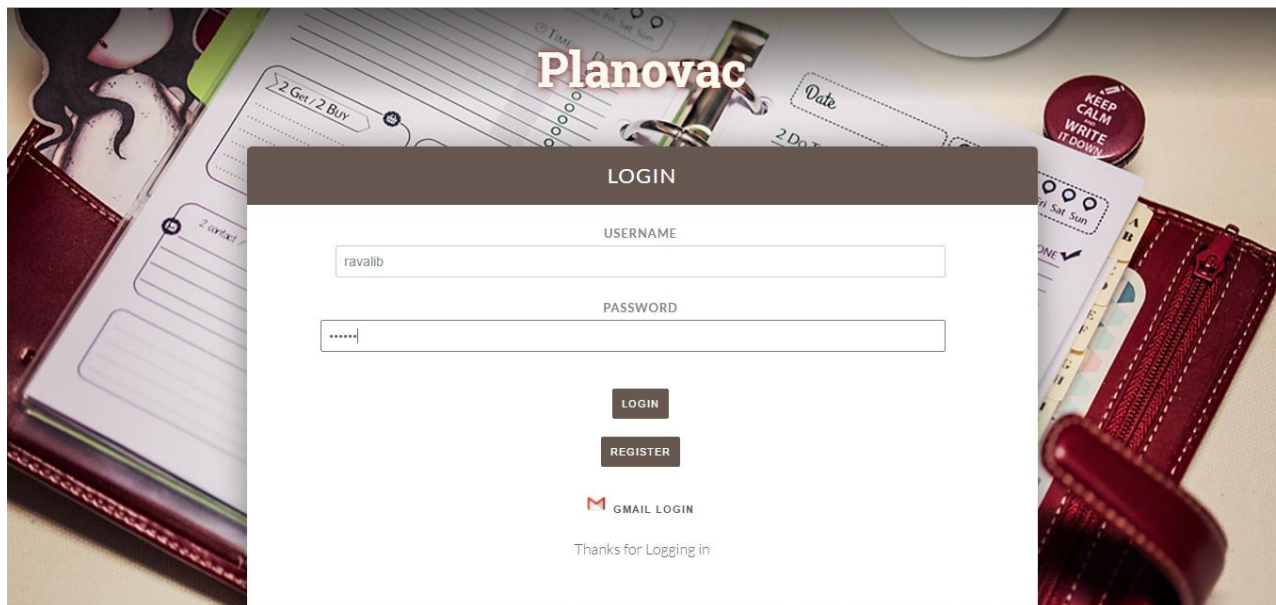
The suggested changes are modified in the Project increment 2. Along with it, new additions to the application like creation of Mongo DB and a new feature Discussion forum are implemented for Project Increment 2.

A new user is created in the application by signing into Register Page of Planovac.



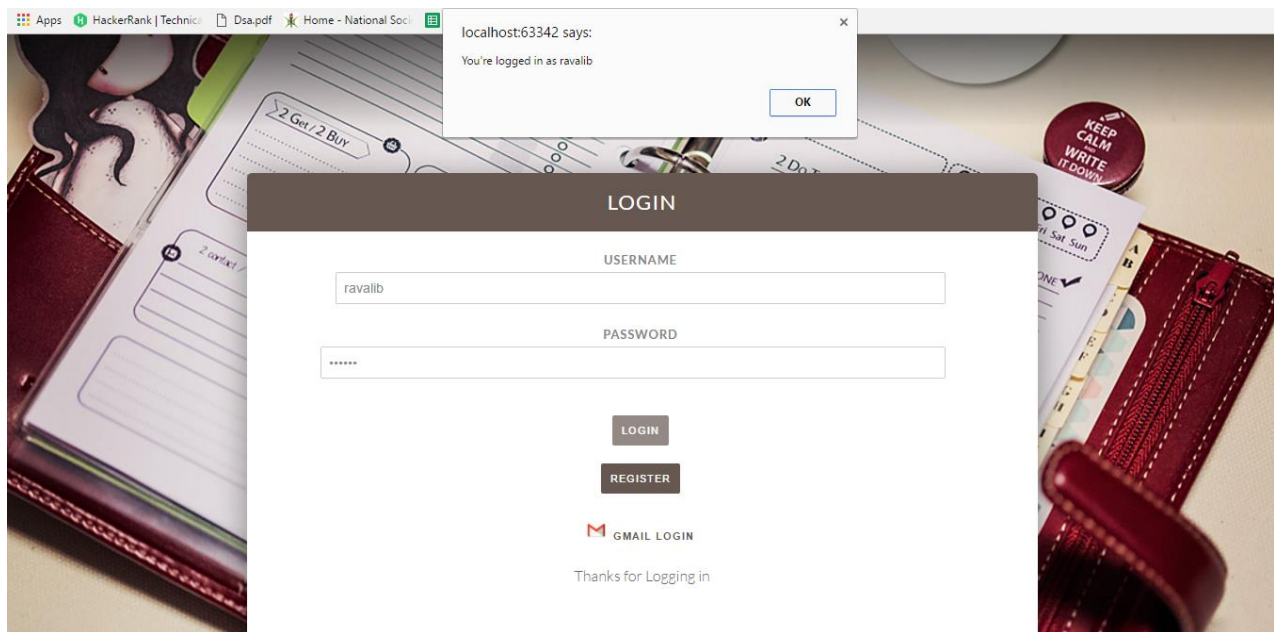
The screenshot shows the 'REGISTER' page of the Planovac application. The page is overlaid on a background image of a desk with a notebook, a pen, and a red leather bag. The 'REGISTER' form has a dark header with the word 'REGISTER' in white. Below the header, there are two columns of input fields. The first column contains fields for 'FIRST NAME' (with the value 'ravali'), 'USERNAME' (with the value 'ravallib'), 'STUDENT SSO' (with the value 'rbgb7'), and 'ENTER PASSWORD' (with masked characters). The second column contains fields for 'LAST NAME' (with the value 'bolem'), 'CONTACT' (with the value '7744325002'), 'STUDENT ID' (with the value '16231463'), and 'CONFIRM PASSWORD' (with masked characters). At the bottom of the form, there are two buttons: 'SIGN IN' and 'REGISTER'.

After successful registration, the user is directed to the login page where the user credentials need to be entered to login into the Web application.

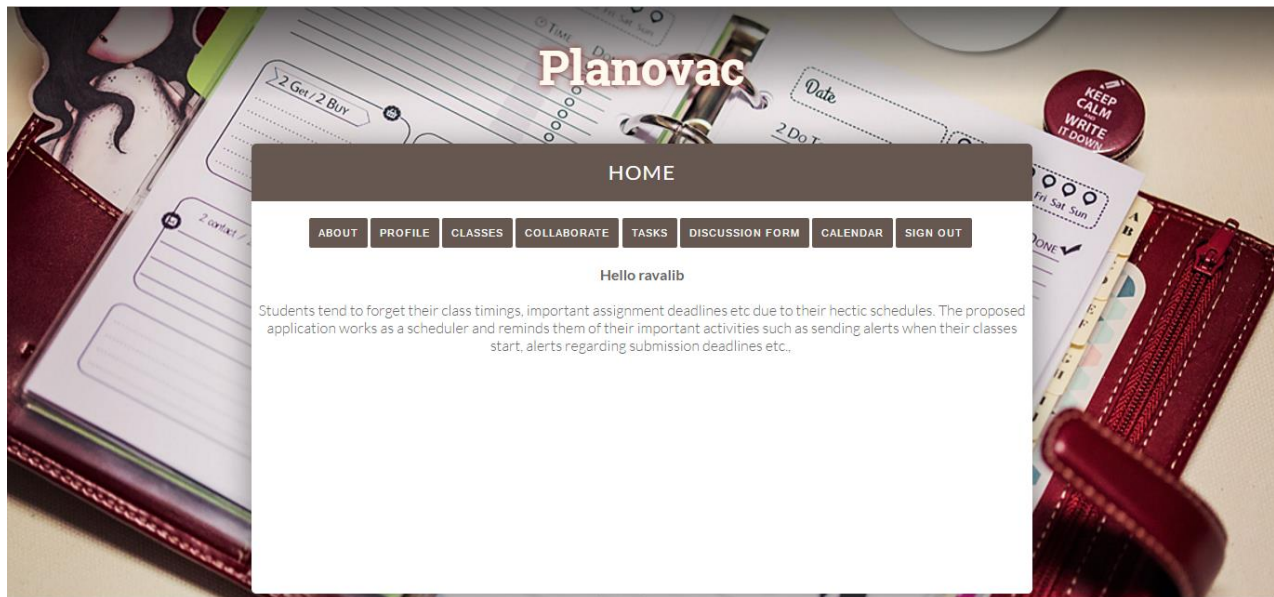


The screenshot shows the 'LOGIN' page of the Planovac application. The page is overlaid on the same background image as the register page. The 'LOGIN' form has a dark header with the word 'LOGIN' in white. Below the header, there are two input fields: 'USERNAME' (with the value 'ravallib') and 'PASSWORD' (with masked characters). At the bottom of the form, there are two buttons: 'LOGIN' and 'REGISTER'. Below the buttons, there is a link for 'GMAIL LOGIN' with a Gmail icon. At the very bottom, there is a message that says 'Thanks for Logging in'.

After successful login, a pop up message is shown on the top on the webpage.

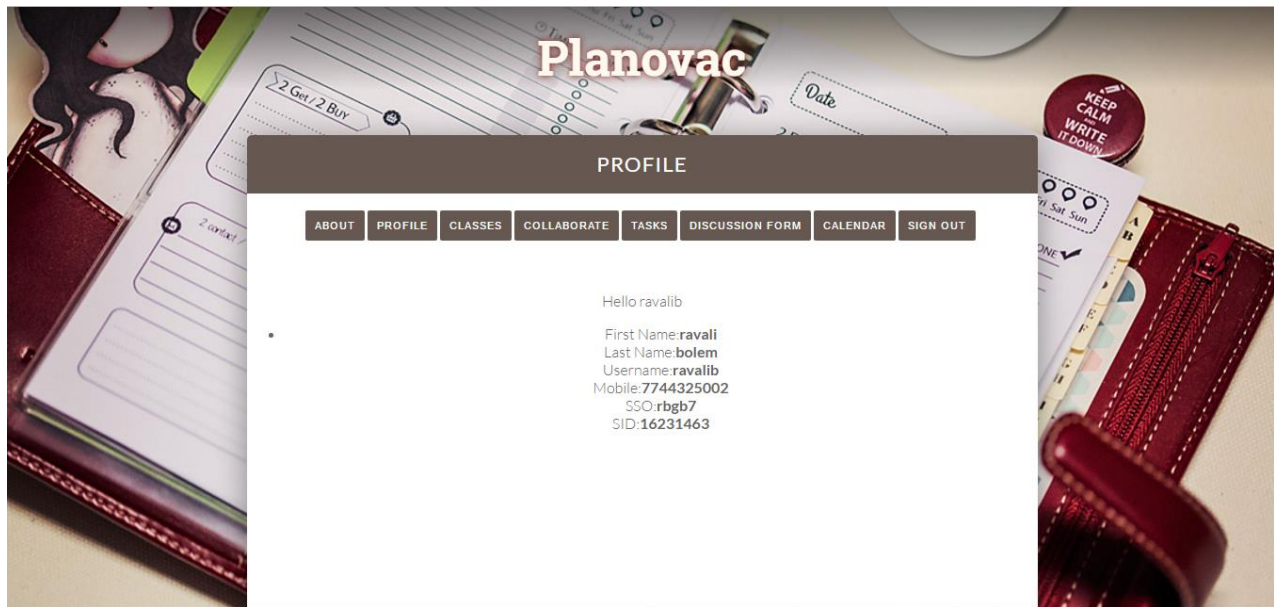


The about page is shown with the user created name showing “hello user”.

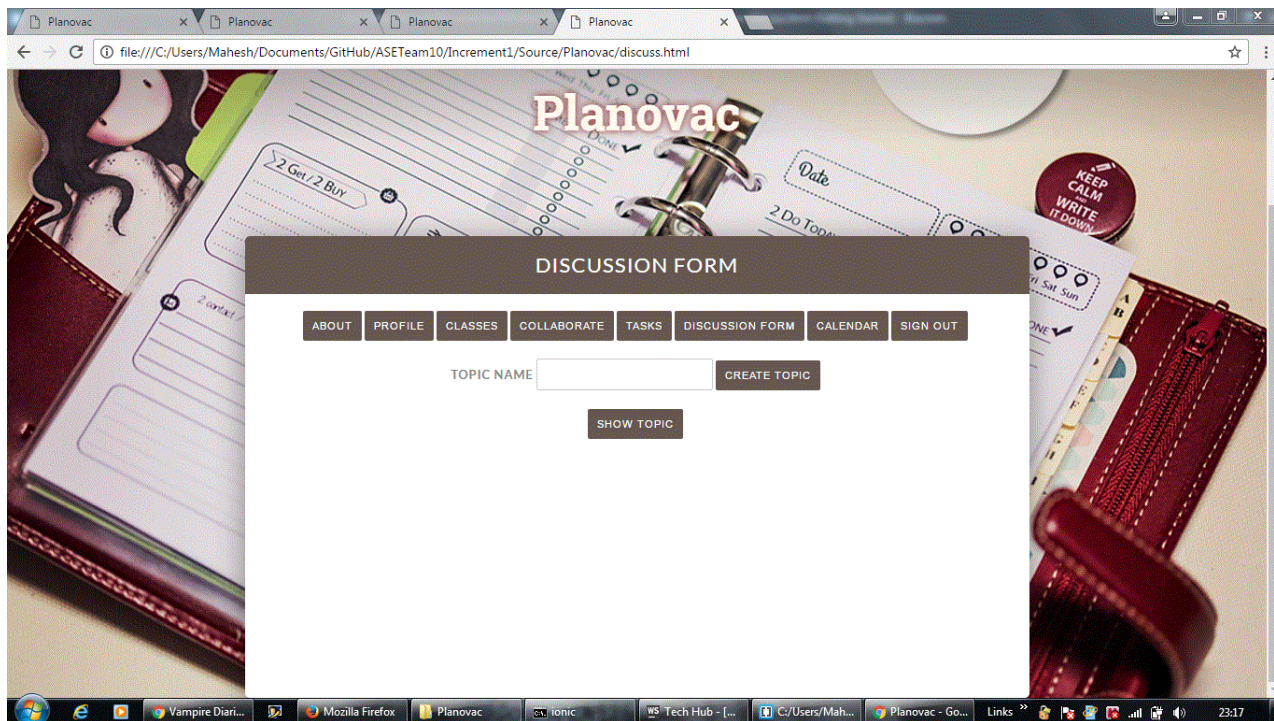




In the profile the user details are shown by which the user created the account.

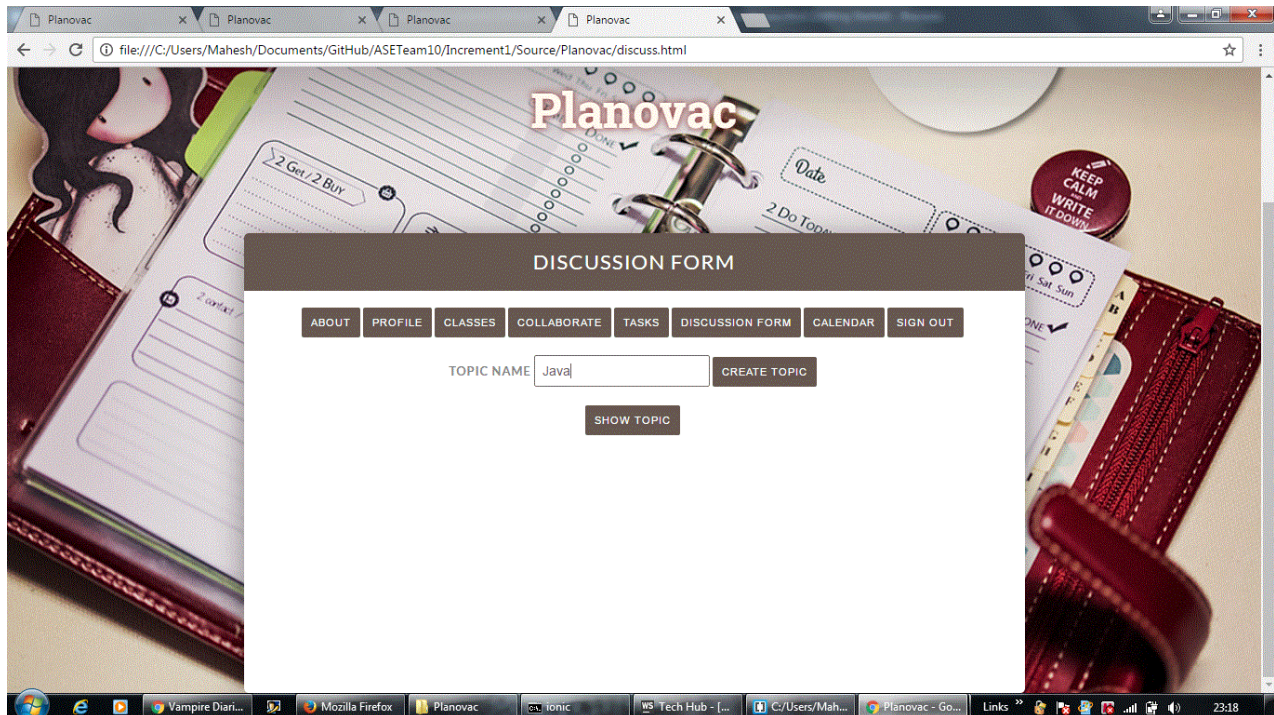


The discussion forum is created where new topics are created and the last topics created are shown by clicking in 'Show Topics' button.

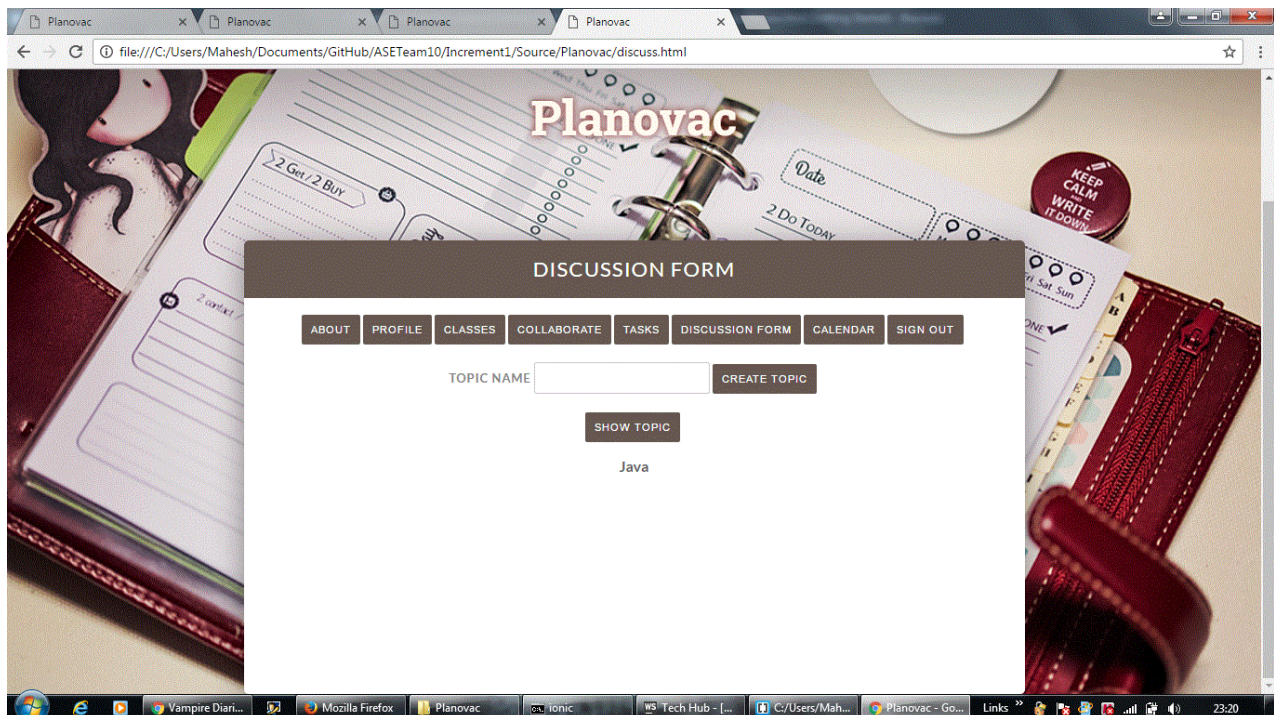




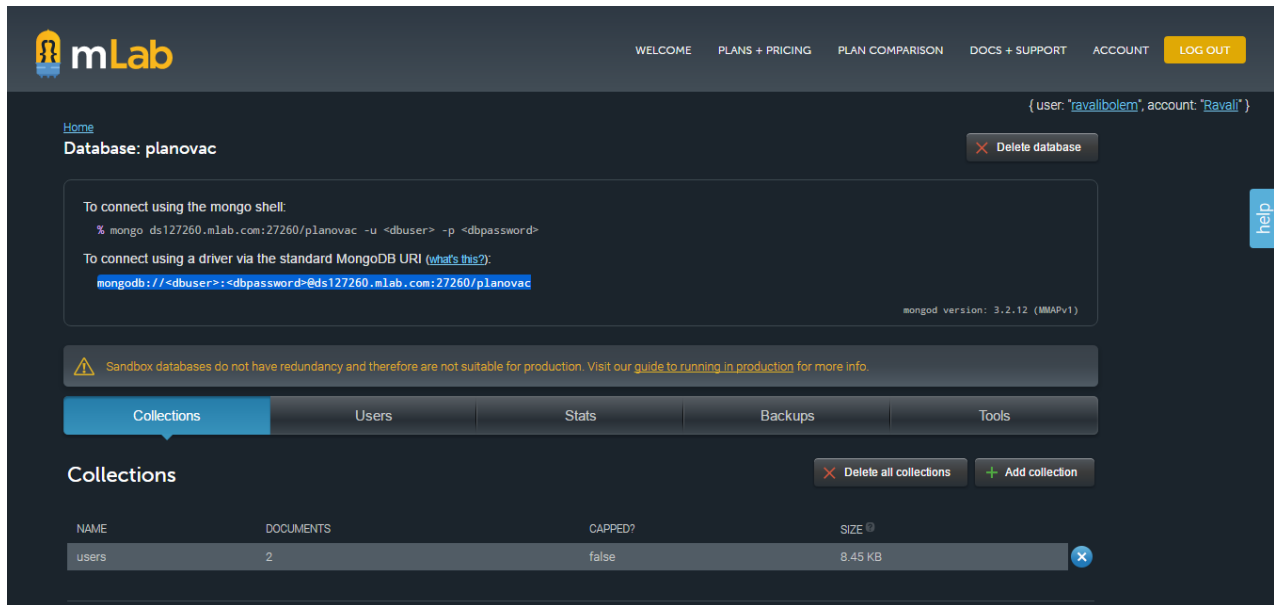
A new topic 'Java' is created in Discussion forum.



Created topic 'Java' is shown in list of all topics stored in the database.



The database shows the creations of the values from the application.



The screenshot shows the mLab web interface for a database named 'planovac'. The top navigation bar includes links for 'WELCOME', 'PLANS + PRICING', 'PLAN COMPARISON', 'DOCS + SUPPORT', 'ACCOUNT', and a 'LOG OUT' button. A user session is active for 'user: "ravalibolem", account: "Raval"'. The main content area displays connection instructions for the MongoDB shell and a standard MongoDB URI. Below this, a warning message states that sandbox databases are not suitable for production. A horizontal menu contains tabs for 'Collections', 'Users', 'Stats', 'Backups', and 'Tools'. The 'Collections' tab is selected, showing a table with one collection named 'users' containing 2 documents. The table headers are 'NAME', 'DOCUMENTS', 'CAPPED?', and 'SIZE'. A 'Delete database' button is located in the top right, and 'Delete all collections' and 'Add collection' buttons are in the top right of the collections list.

Database: planovac

To connect using the mongo shell:

```
% mongo ds127260.mlab.com:27260/planovac -u <dbuser> -p <dbpassword>
```

To connect using a driver via the standard MongoDB URI (what's this?):

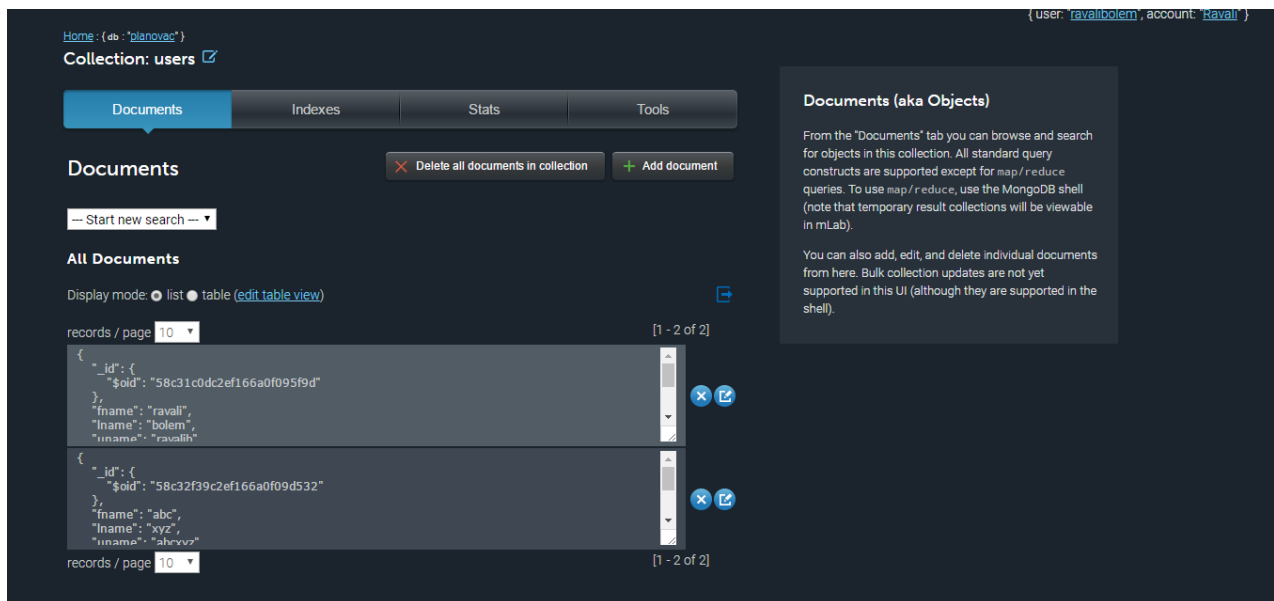
```
mongodb://<dbuser>:<dbpassword>@ds127260.mlab.com:27260/planovac
```

mongod version: 3.2.12 (MMAPv1)

Sandbox databases do not have redundancy and therefore are not suitable for production. Visit our [guide to running in production](#) for more info.

Collections

NAME	DOCUMENTS	CAPPED?	SIZE
users	2	false	8.45 KB



The screenshot shows the mLab web interface for the 'users' collection. The top navigation bar is the same as the previous screenshot. The main content area displays the 'Collection: users' with tabs for 'Documents', 'Indexes', 'Stats', and 'Tools'. The 'Documents' tab is selected, showing a list of documents. A search bar is present with the text 'Start new search'. The 'All Documents' section shows the display mode set to 'list' and a table view. The table contains two documents, each with fields for '\_id', '\$oid', 'fname', 'lname', and 'uname'. The first document has 'fname': 'raval', 'lname': 'bolem', and 'uname': 'ravalib'. The second document has 'fname': 'abc', 'lname': 'xyz', and 'uname': 'abxvuvz'. A 'Delete all documents in collection' button and an 'Add document' button are located in the top right of the documents list. A sidebar on the right provides information about the 'Documents (aka Objects)' tab, explaining that standard query constructs are supported except for map/reduce queries, and that bulk collection updates are not yet supported in this UI.

Collection: users

Documents

Documents (aka Objects)

From the "Documents" tab you can browse and search for objects in this collection. All standard query constructs are supported except for map/reduce queries. To use map/reduce, use the MongoDB shell (note that temporary result collections will be viewable in mLab).

You can also add, edit, and delete individual documents from here. Bulk collection updates are not yet supported in this UI (although they are supported in the shell).

Documents

Display mode: list table (edit table view)

records / page 10 [1 - 2 of 2]

{ "_id": { "\$oid": "58c31c0dc2ef166a0f095f9d" }, "fname": "raval", "lname": "bolem", "uname": "ravalib" }	x	e
{ "_id": { "\$oid": "58c32f39c2ef166a0f09d532" }, "fname": "abc", "lname": "xyz", "uname": "abxvuvz" }	x	e

records / page 10 [1 - 2 of 2]

Code for the database creation are presented below:

```
app.controller("LoginController", function ($scope, $http, $httpParamSerializerJQLike, $window) {

    $scope.pageClass = 'login';
    $scope.login = function(username, pword) {
        sessionStorage.setItem('username1', username);
        $scope.uname = username;
        alert("You're logged in as " + $scope.uname);
        $http({
            method: 'GET',
            url : 'https://api.mongolab.com/api/1/databases/planovac/collections/users?apiKey=1fB-Vh6r9XKXu-n0eW_4OeXv1AEViZ13',
            /*contentType: "application/json"*/
        }).then(function success(data) {
            //alert(data);
            $scope.data1 = angular.fromJson(data);
            $window.location.href = "home.html";
        })
    }

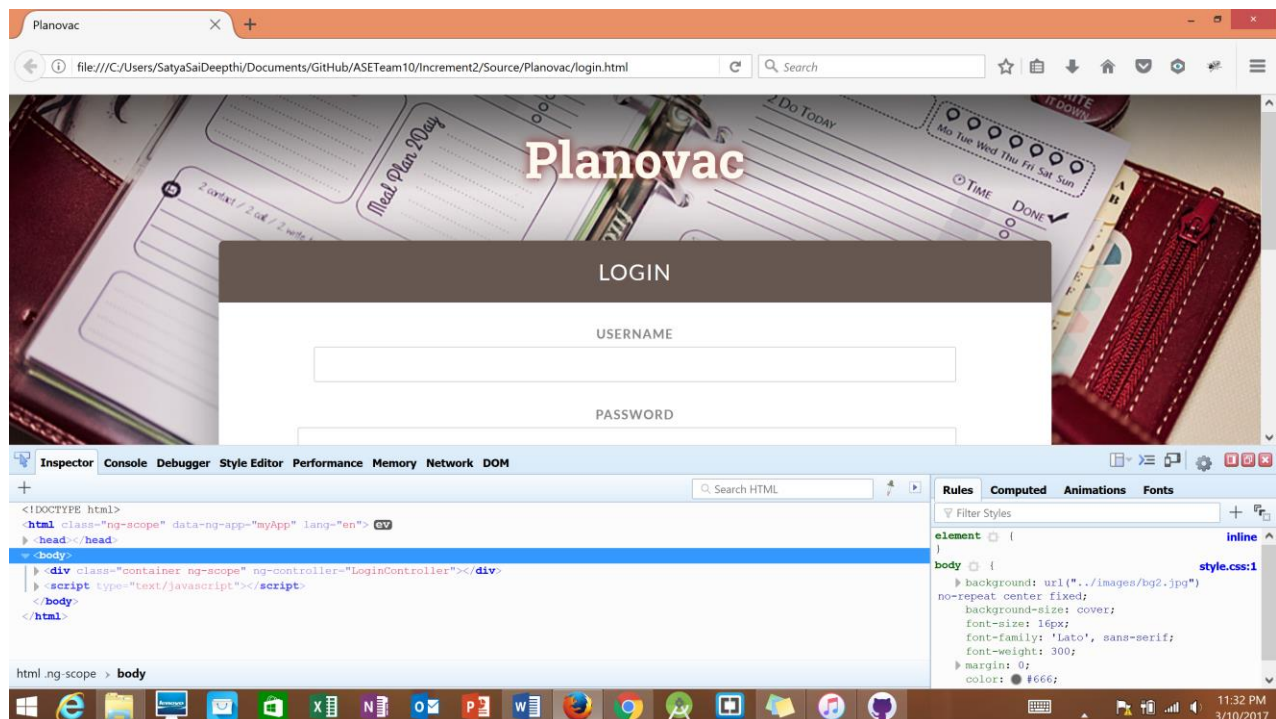
});

app.controller("ProfileController", function ($scope, $http, $httpParamSerializerJQLike, $window) {

    $scope.pageClass = 'profile';

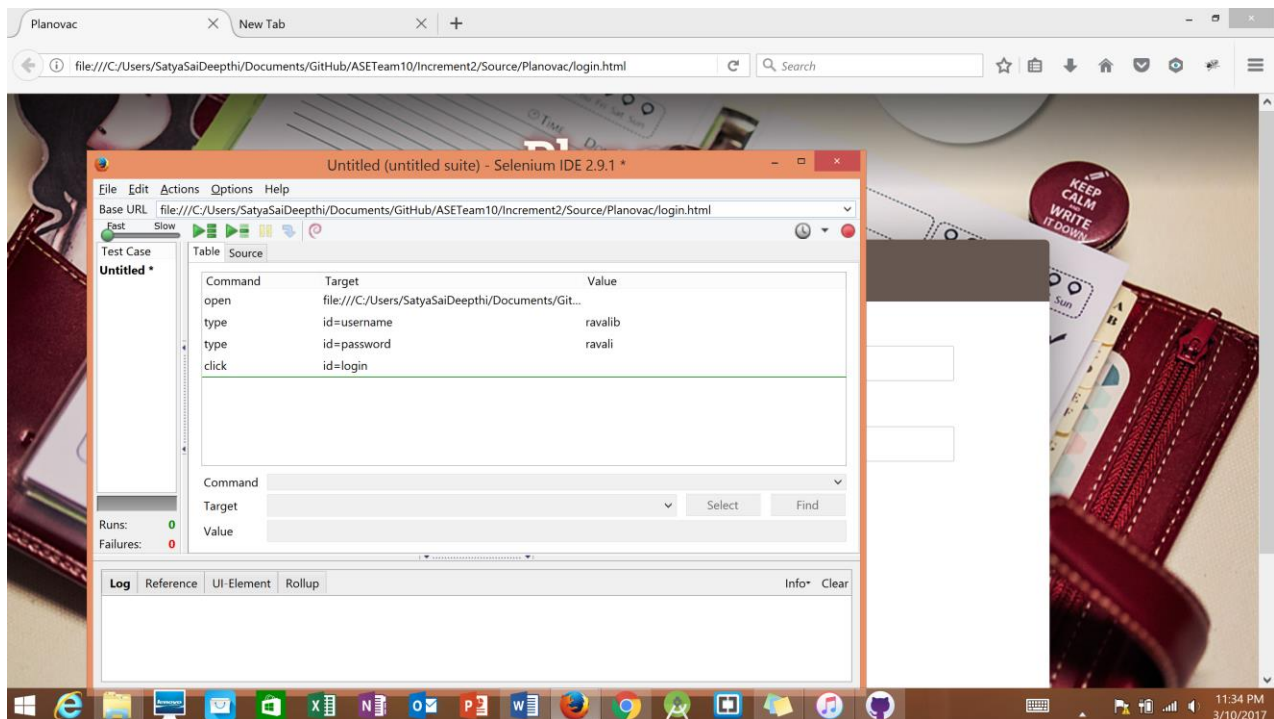
    $scope.username = sessionStorage.getItem("username1");
    alert($scope.username);
    $http({
        method: 'GET',
        url : 'https://api.mongolab.com/api/1/databases/planovac/collections/users?apiKey=1fB-Vh6r9XKXu-n0eW_4OeXv1AEViZ13',
        /*contentType: "application/json"*/
    }).then(function success(data) {
        //alert(data);
    })
});
```

Testing done through Firebug, which shows the inspector for errors in Html page.



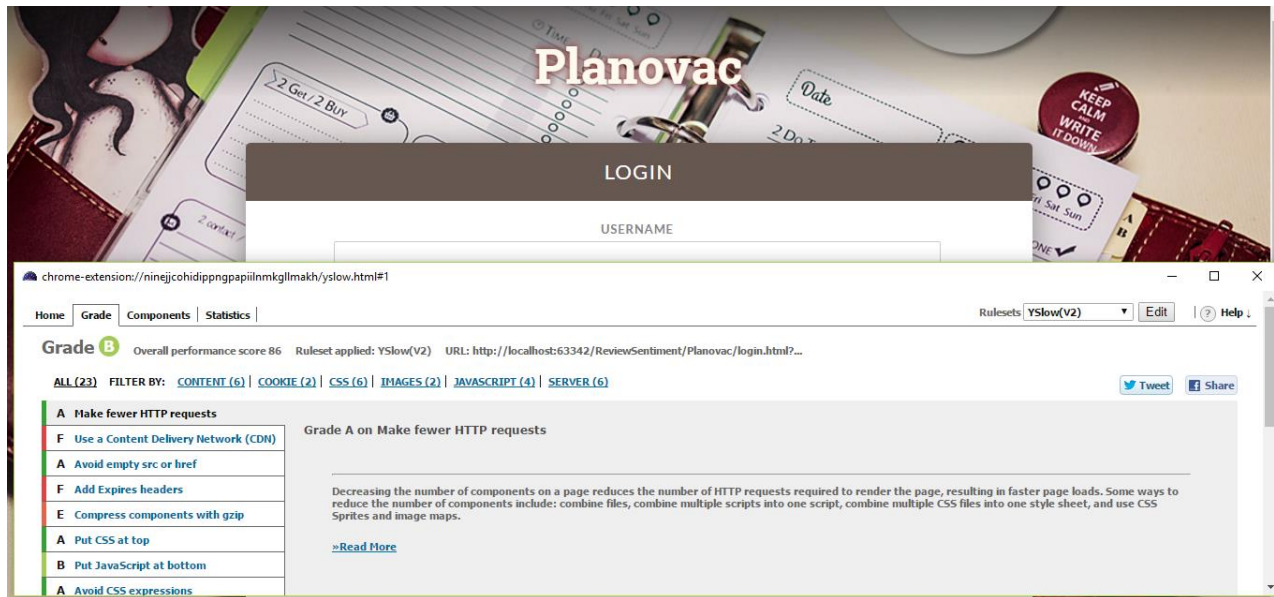


Testing done through Selenium IDE which shows the



test		
open	/ReviewSentiment/Planovac/login.html?_ijt=a6cvuc1u8eu6cocsfainqls7kc	
type	id=username	ravalib
type	id=password	ravali
click	id=login	
assertAlert	You're logged in as ravalib	
clickAndWait	//a[2]/button	

Testing done through Yslow plugin which gives Grade B for Planovac Web Application.



## VII. PROJECT MANAGEMENT

### Work Completed:

- Modifications to Project Increment 1.
- Creating Mongo Database to store values from Web Application.
- Creating Discussion Forum in Planovac.

### Work To Be Done:

- ✓ Use Google Maps API to locate the class rooms.
- ✓ Adding Task Feature.
- ✓ Include Chat feature.

### TEAM CONTRIBUTION:

Name	Contribution
Ravali Bolem (7)	<code> MongoDB creation <code> Increment 1 modifications <design> Web Page designing
Aswini Gutha (33)	<code> Discussion Forum creation <code> Increment 1 modifications <design> Google calendar design and modification <report> UML designing (architecture diagram)
Pranathi Gopidi (29)	<code> Increment 1 modifications <code> displaying student details in profile. <report> UML Designing (class diagram) <testing> Selenium IDE
Satya Sai Deepthi Katta (41)	<code> Increment 1 modifications <design> Wireframes Creation, User stories <documentation> Overall data collection required for Report. <report> UML designing (sequence diagram) <documentation> WIKI Page Creation <testing> YSlow & Firebug

## **Related Work:**

- ✓ UMKC Blackboard
- ✓ When I Work Schedule Planner
- ✓ Team Snap

## **Bibliography**

- (i) When I Work  
<http://wheniwork.com/l/aw-schedule-maker>
- (ii) Team Snap  
<http://developer.teamsnap.com/>
- (iii) MongoDB Documentation  
<https://docs.mongodb.com/>
- (iv) Google Calendar API  
<https://developers.google.com/google-apps/calendar/>
- (v) Creately Web Service  
<https://creately.com/Draw-UML-and-Class-Diagrams-Online>