UMKC

Advanced Software Engineering

Course # CSEE 5551 0001

Semester: Spring 2017

Project Increment #1 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 #1 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 and Objectives

➢ *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, re-doing 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.

> ➢ *__System Features:__* Web Application

What is a Web Application?

A Web Application is a barrier between the client and server. The client runs the application in a web browser like Google chrome, safari, Mozilla Firefox etc. The application can be run on a ordinary window or for security purposes some people use incognito window which doesn't show the history of the browser.

The figure below shows the three stage web application procedure in a detailed manner.
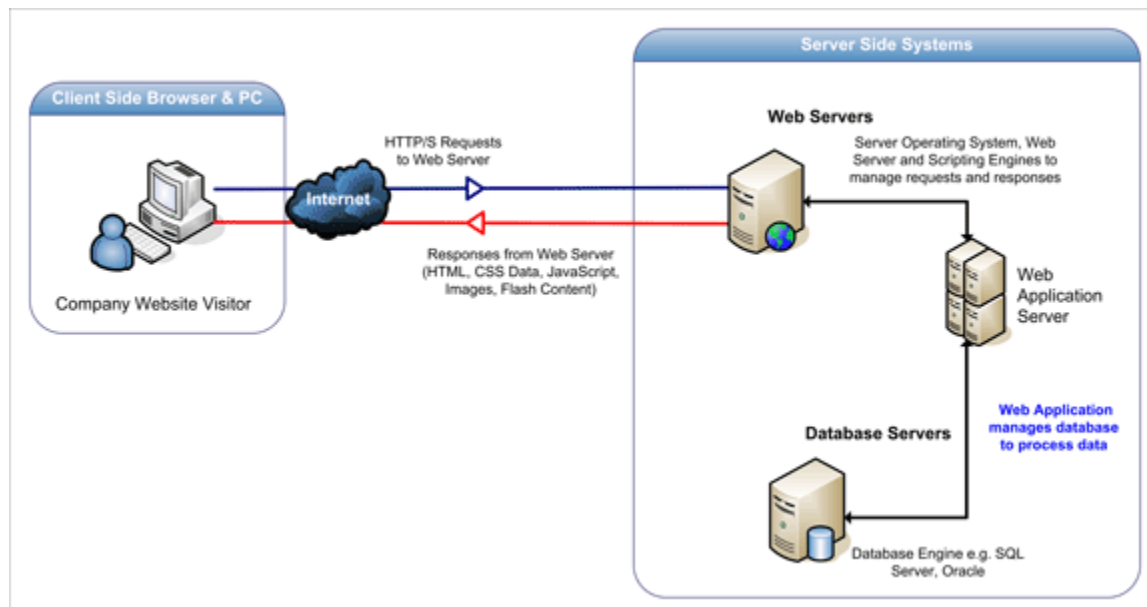
- The first layer is the normal web browser or the user interface which runs the application created.
- The second layer is the Web server where the java script code is written to run the application, this mainly acts as the controller for the application to run.
- The third layer is the database server which takes in the data and stores in the local storage of the web application. The accounts created are stores basically in the database servers for an web applications.

For our project, we have selected a web browser which has faster speed of execution and widely used around the world in day to day life

Web Browser: Google Chrome Internet connection

III. Project Plan
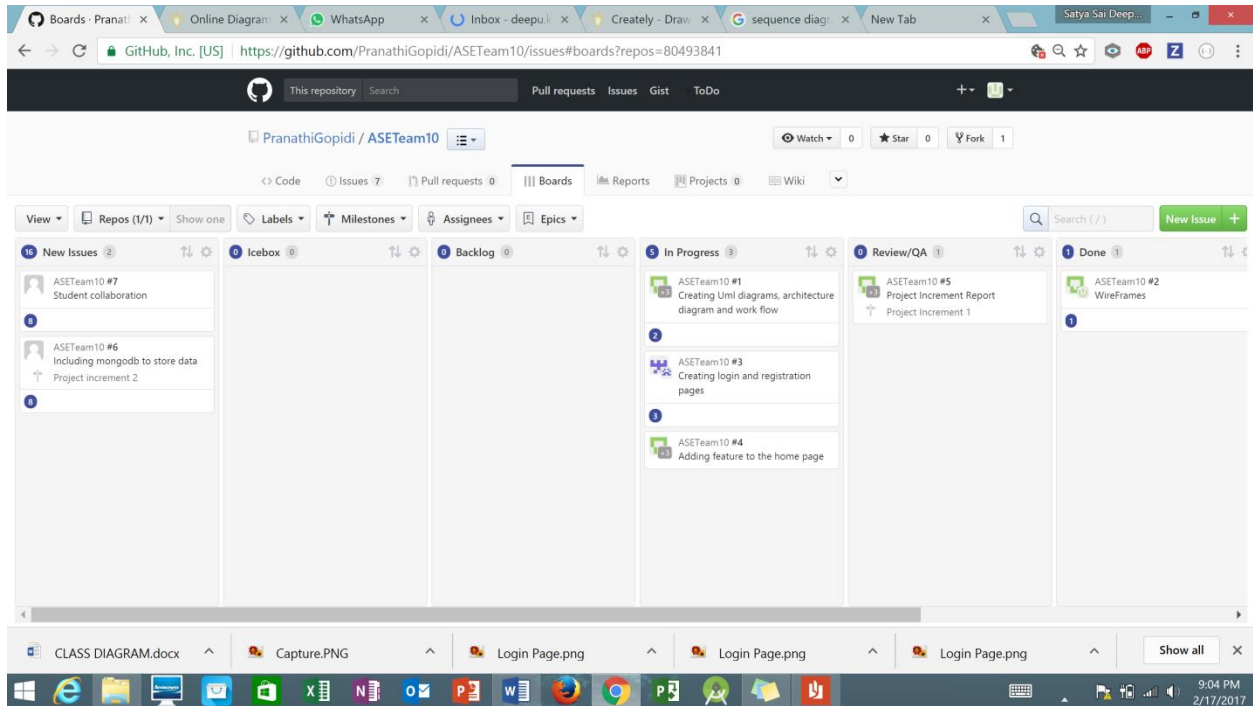
> ## *Schedule for Four different increments*

The schedule for the four different increments is designed and allocated to the members of the team. The detailed schedule is given below.

1. Issues faced in the Project

The issues faced in the project are created in Github whose screenshot is provided below:
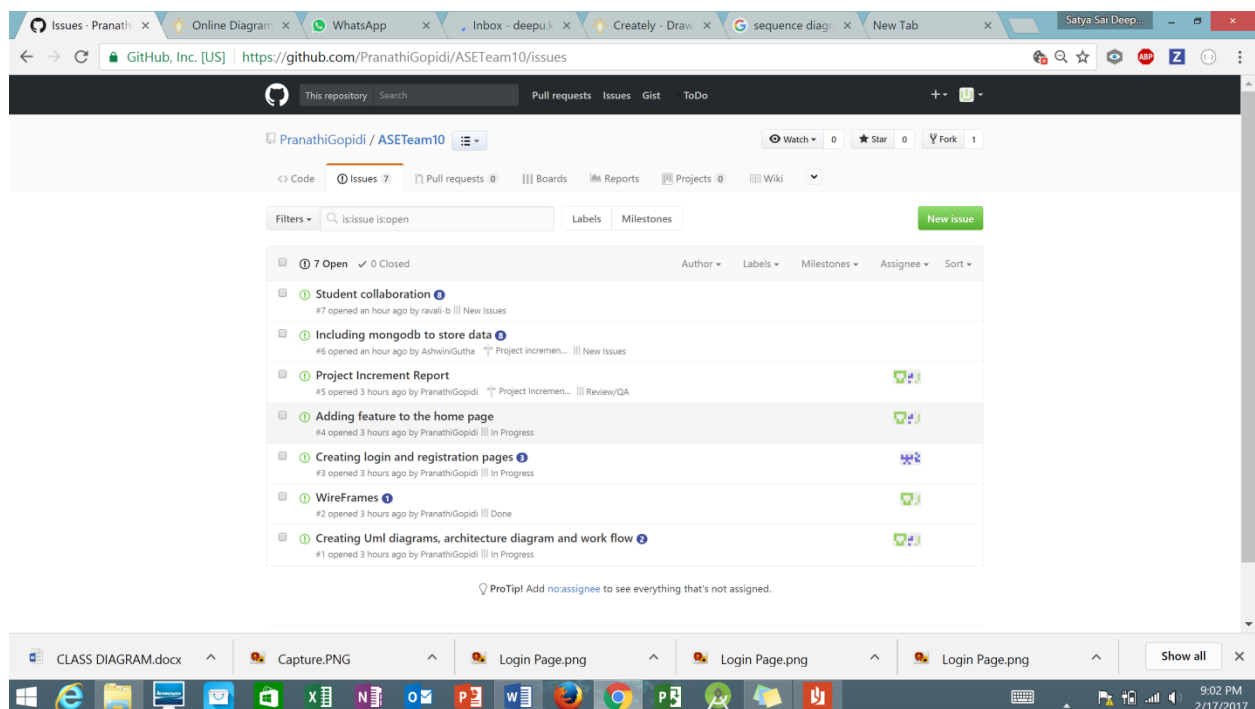
The board results are displayed in the screenshot below:

2. **Detail Service Design**

3. **Service Implementation and Testing**

These two aspects are covered in the First increment report given below.

➢ *__Project timeline, Members and Task Responsibility__*

The scheduled tasks are given the timelines based on the deadlines given the google docs. Based on the capability of the members, the task responsibilities are assigned.

➢ *__Burndown Charts__*

The Burndown charts are taken from the Github and are displayed below.

IV.   **First Increment Report**

➢ *Existing Services*

The services and the API's used are given below with the details description of each service.

Google Calendar API:

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.

➢ *Detailed Design of Features*

The best way to understand the features of the project is to look for their diagrammatic representations. UML diagraming is the Unified Modeling Language which portraits 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

- **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:
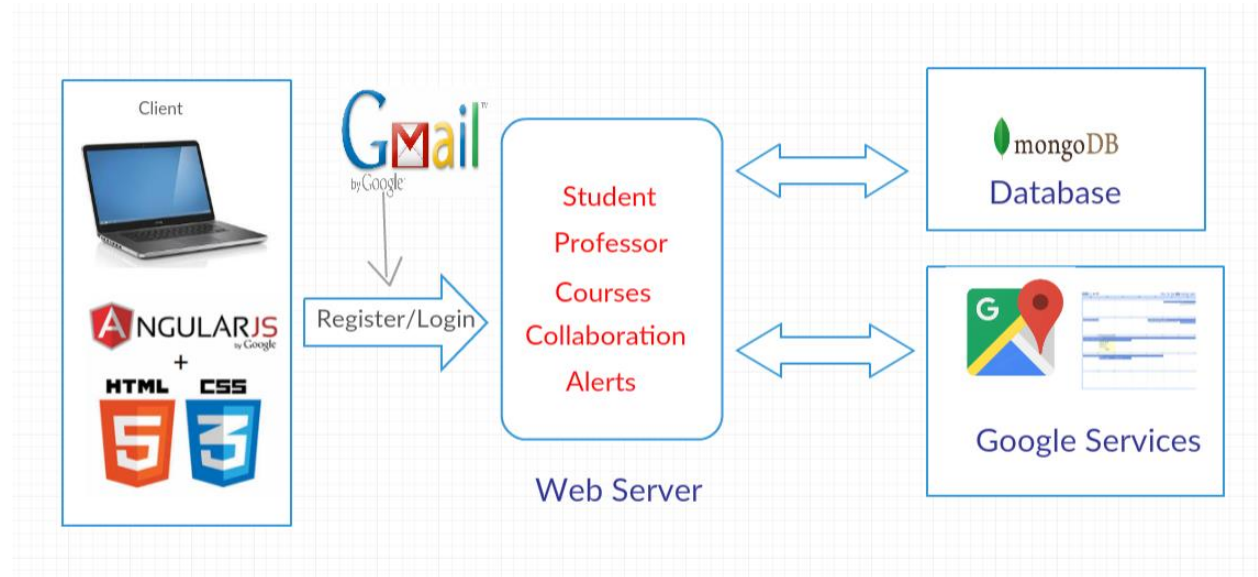
- **Class Diagram:**

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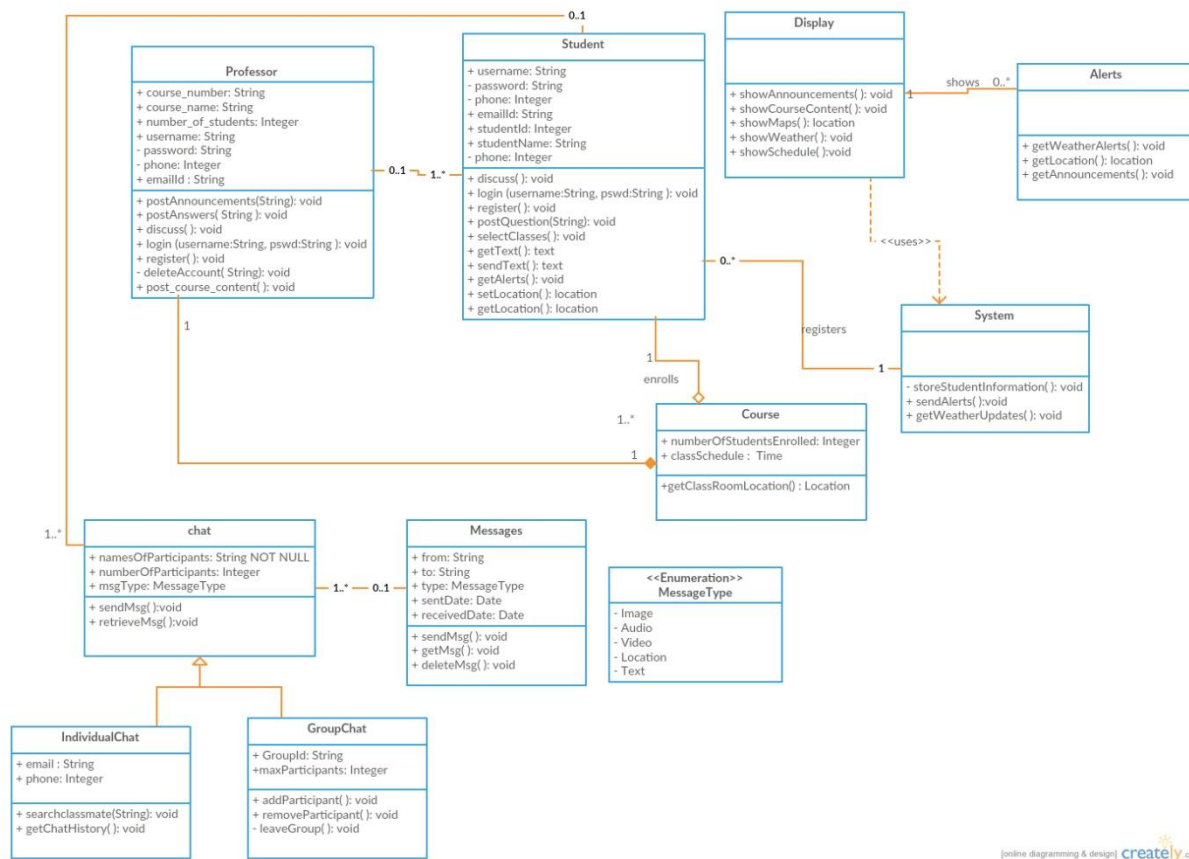
In the class diagram, the classes named professor, students, display, system, alerts, class and chat are created.

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.

- **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 are the possible ways for the given use case.

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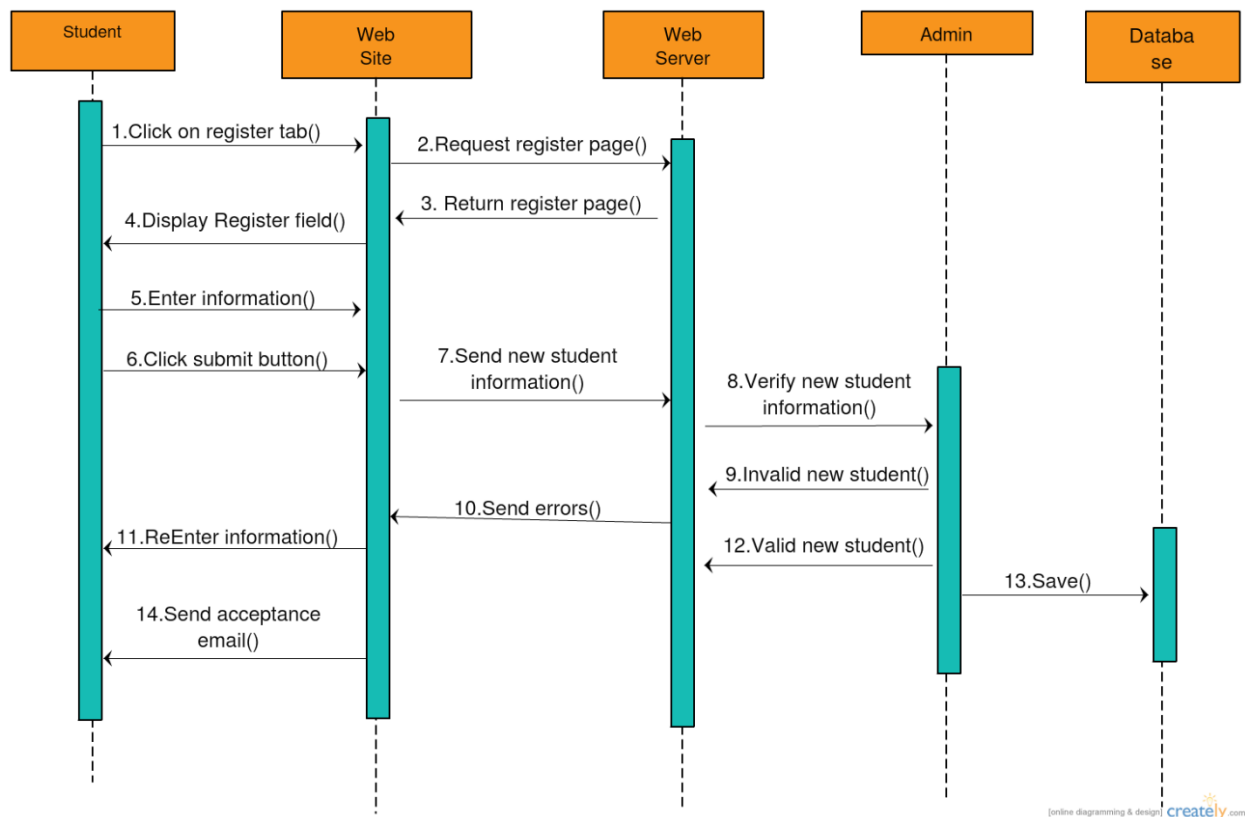
TESTING:

> ## *Unit Testing*

Serial No	Test Case Title	Description	Expected Result	Actual Result
1.	Successful Login	User logs in using the correct login credentials. User should be redirected to home page	Successful login	Success
2.	Gmail Validation	User needs to login using his Gmail credentials and on clicking sign in, home page has to be displayed	Login Successful	Success
3.	Invalid credentials	User provides wrong credentials to login	Invalid Username/password. Enter correct credentials	Success
4.	No credentials provided	User clicks login without providing any credentials	Please enter the details in the requires fields	Success
5.	User Registration	Clicking on register button in login page must redirect to register page	Register page should be displayed	Success
6.	Registration validation	User needs to fill all the mandatory details and click on register	Registered successfully if all the details are provided else alert gets displayed asking to provide the valid details	Success
7.	Logout validation	On clicking on logout button in the home page,	Login page has to displayed	Success

		it should be redirected to login page		
8.	About button validation	On clicking about button in the home page, the page needs to display all the information about the application	Information regarding the application must be displayed	Success
9.	Calendar button validation	User when clicks calendar button, calendar is displayed	Calendar has to be displayed	Success
10.	Task button validation	On clicking on task button in the home page, menu is displayed which has current tasks, add and delete tasks	Menu has to displayed	Success

➢ ***Implementation***
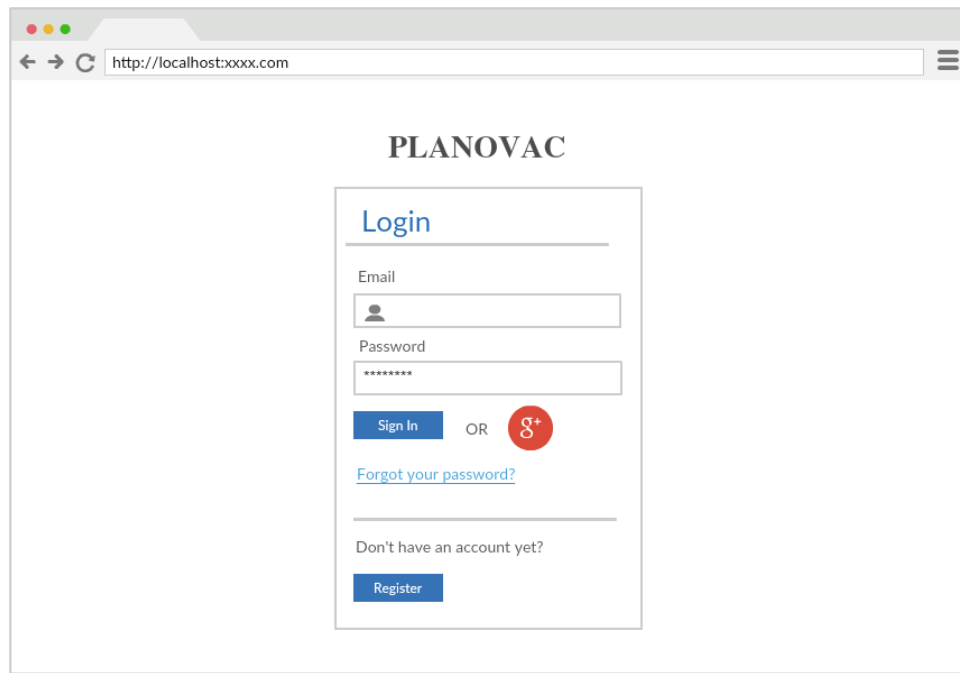
Technologies Used:

- HTML5
- Java Script
- CSS
- Angular JS

The implementation of the project is shown through the Wireframes provided below.

On opening the browser, the login page pops up with the user name and password details. If the user doesn't have any account, they can create an account using Registration Page.
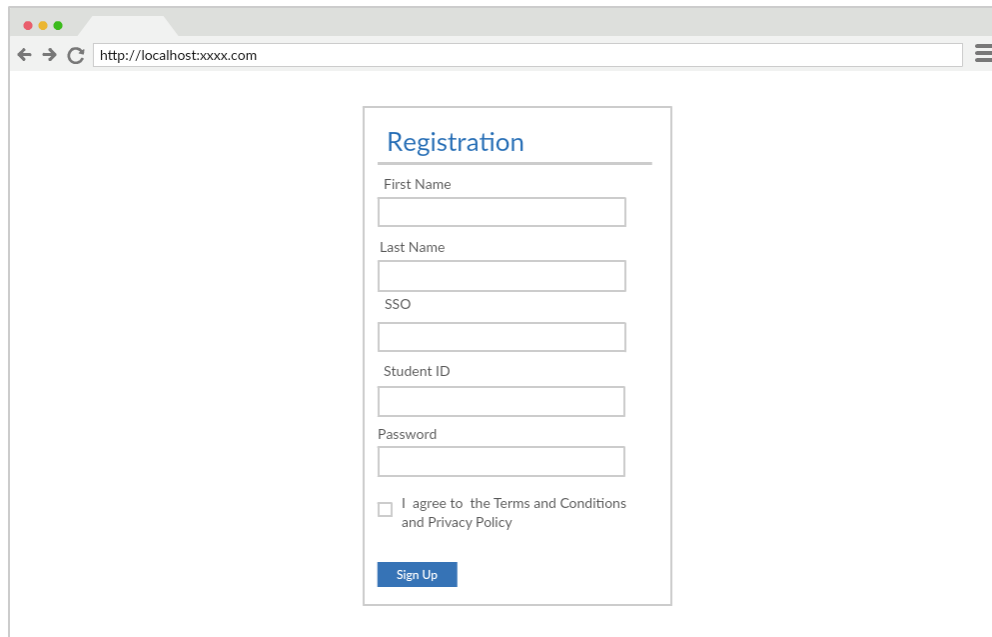
The Login page wireframe for the web application.

For a new user to create an account in Planovac, need some user information like the Default First and last names, creating a password. In addition to these, as the application is collaborating with UMKC Blackboard, it needs UMKC student information of the user like Student ID and SSO login.

The Registration page for Web Application.

For successful logging into Web application, the home page is displayed with the ABOUT button. This gives the description about the project and the motivation behind it.
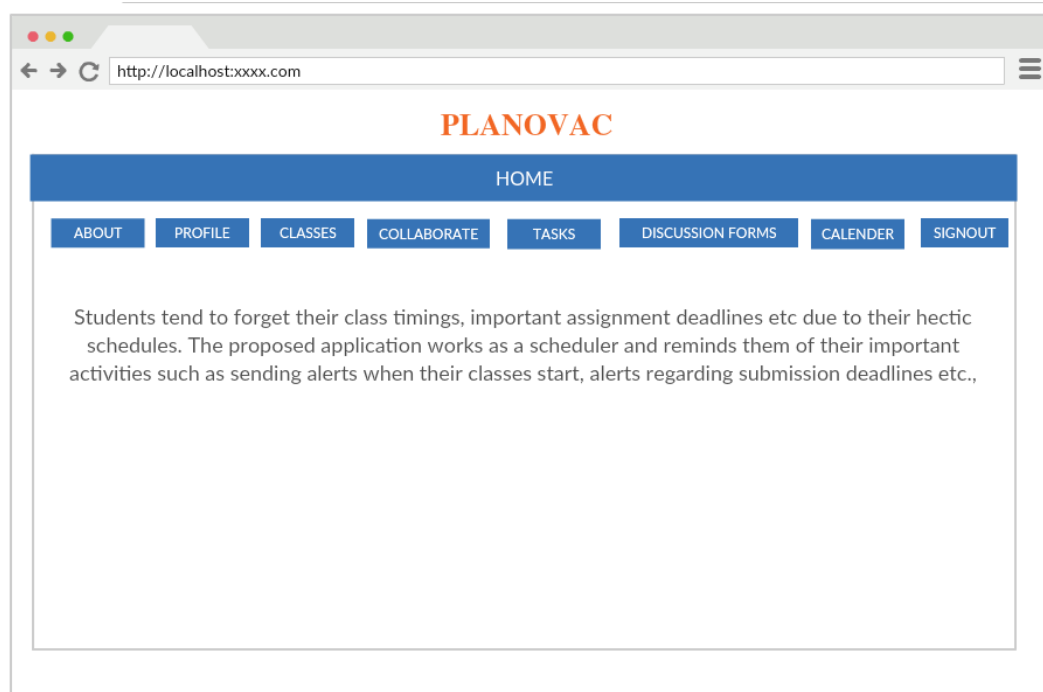
By selecting the buttons on the Home page gives the various features of Web application. The PROFILE shows the account details of the user.
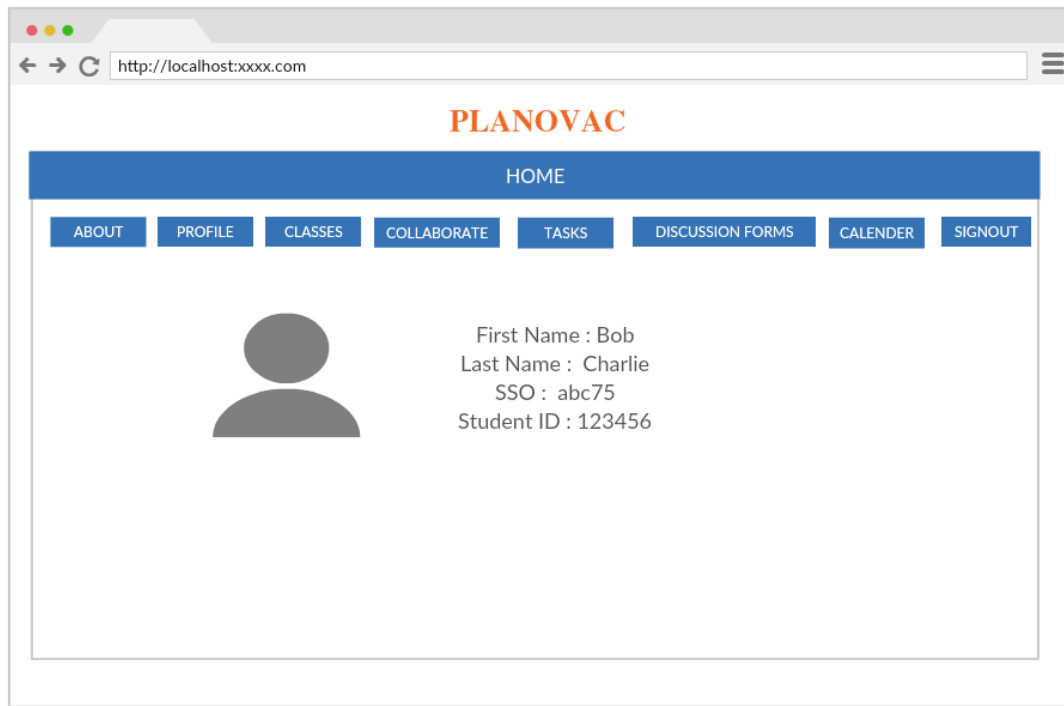
For Adding or Deleting the Classes to the Application can be done through the CLASS button. This also view the classes registered for the semester.
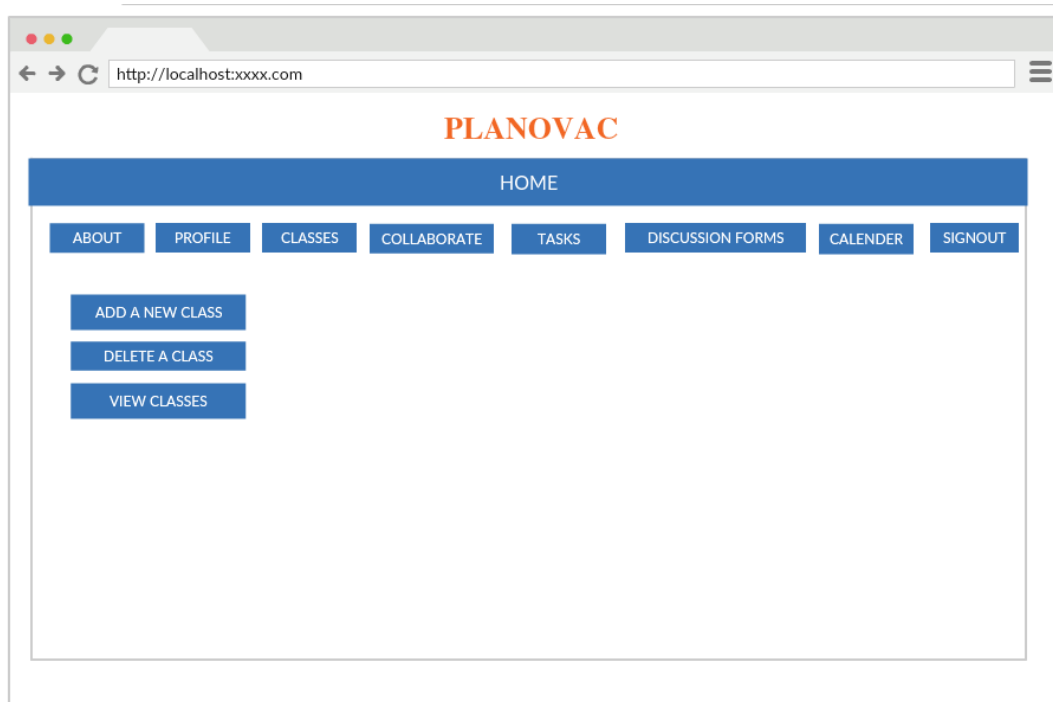
To connect with the students on the application, Planovac provides an option to COLLABORATE with single friend or multiple friends. The significant feature is student can directly interact with Professor without any appointments.
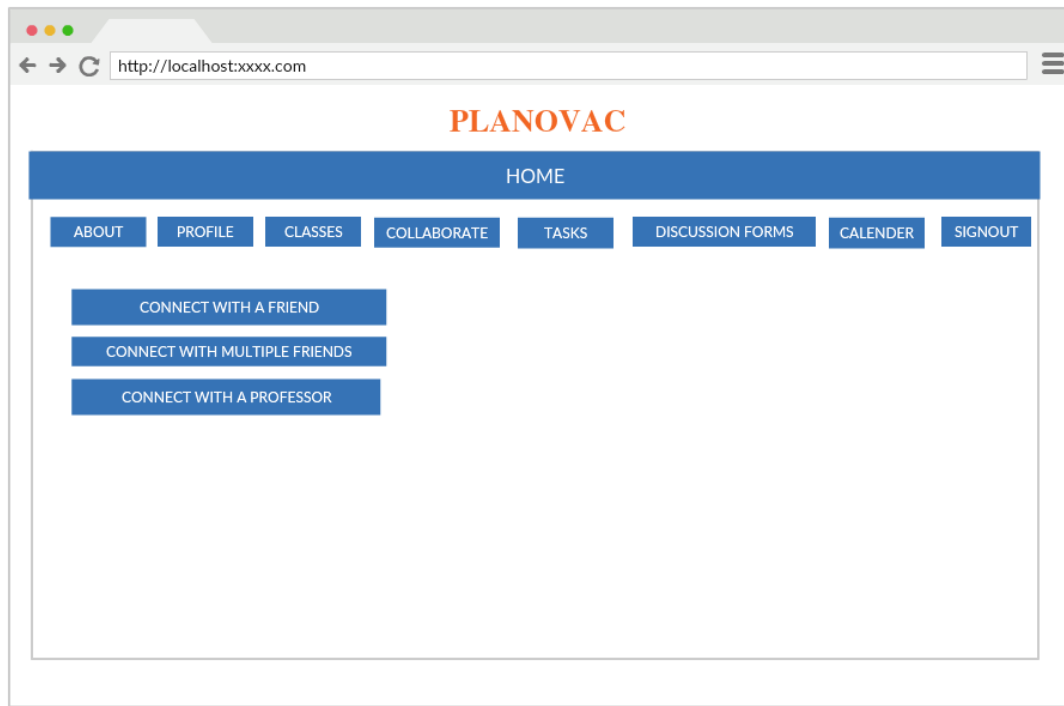
Students can assign tasks to themselves, planning to complete a project, schedule to prepare for an exam. Planovac provides the TASK option to add new task, Delete completed tasks and also to show current tasks.
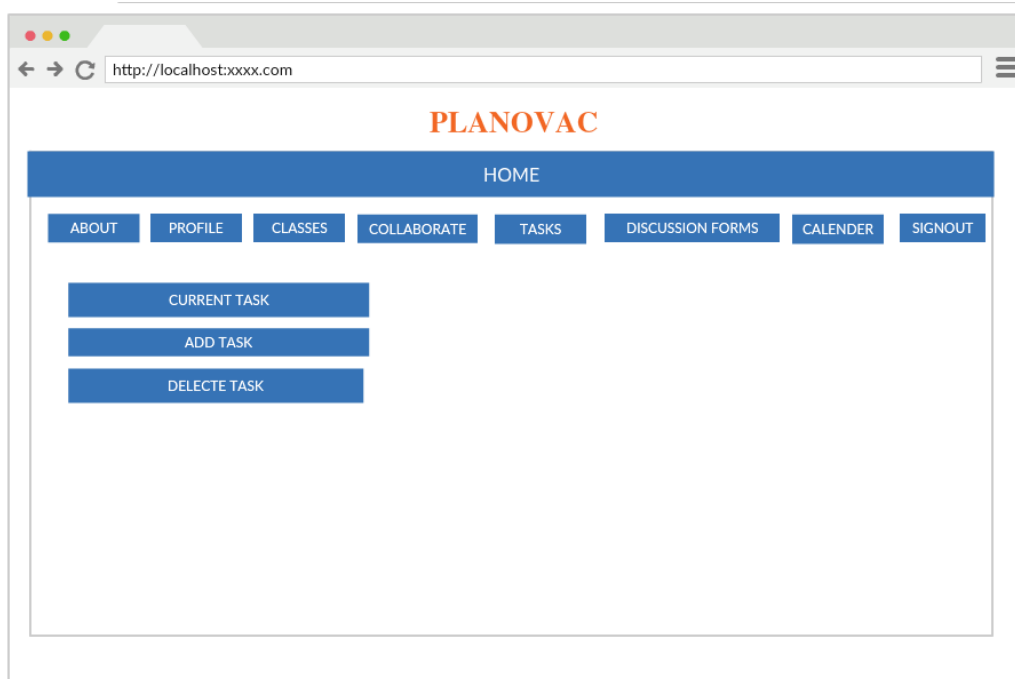
The web application gives a unique DISCUSSION FORM where students can put questions and interact with each there. Professors can post their announcements like Exam timings, Class hours, Quizzes, Class cancelled Notices etc.
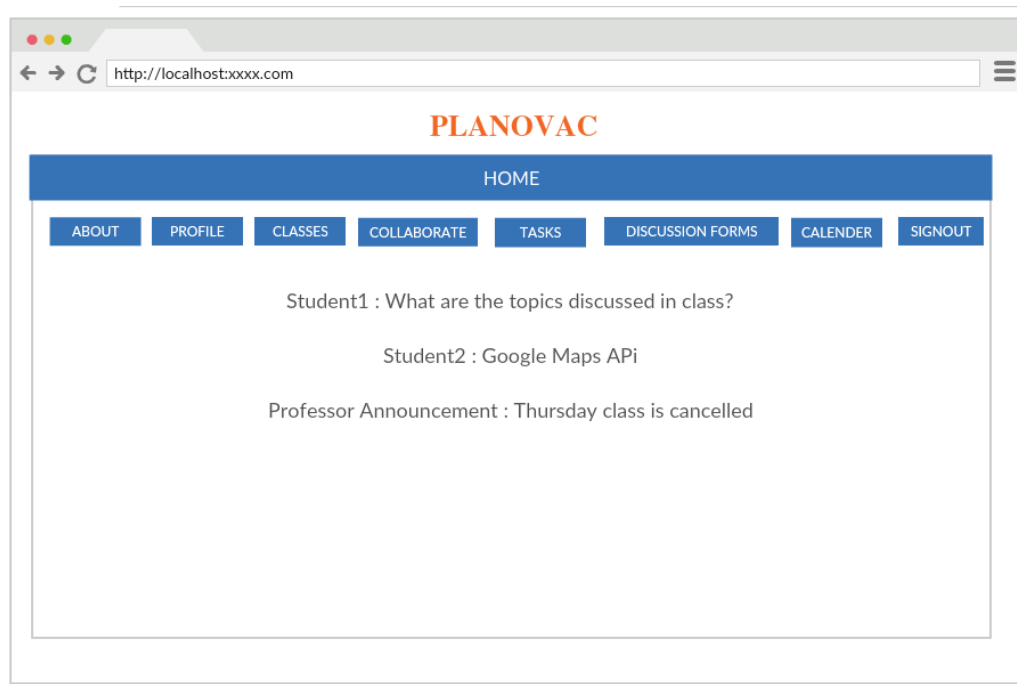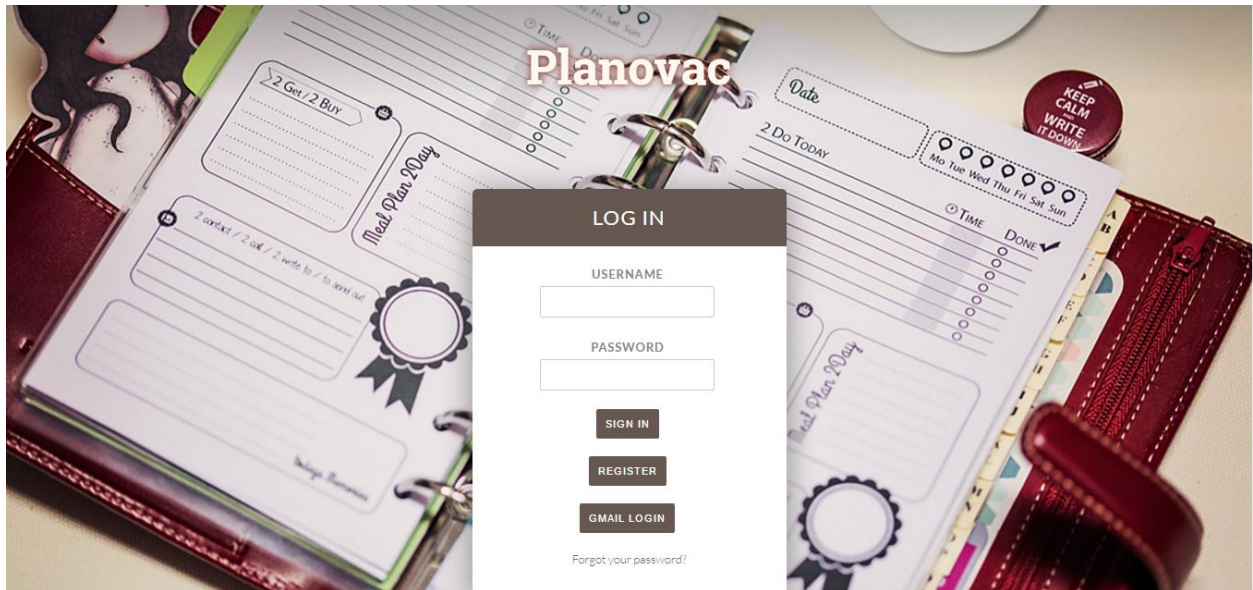

This application remind students their class timings, events, seminars through Google Calendar API. Students can add their events and set remainders by constant updating.
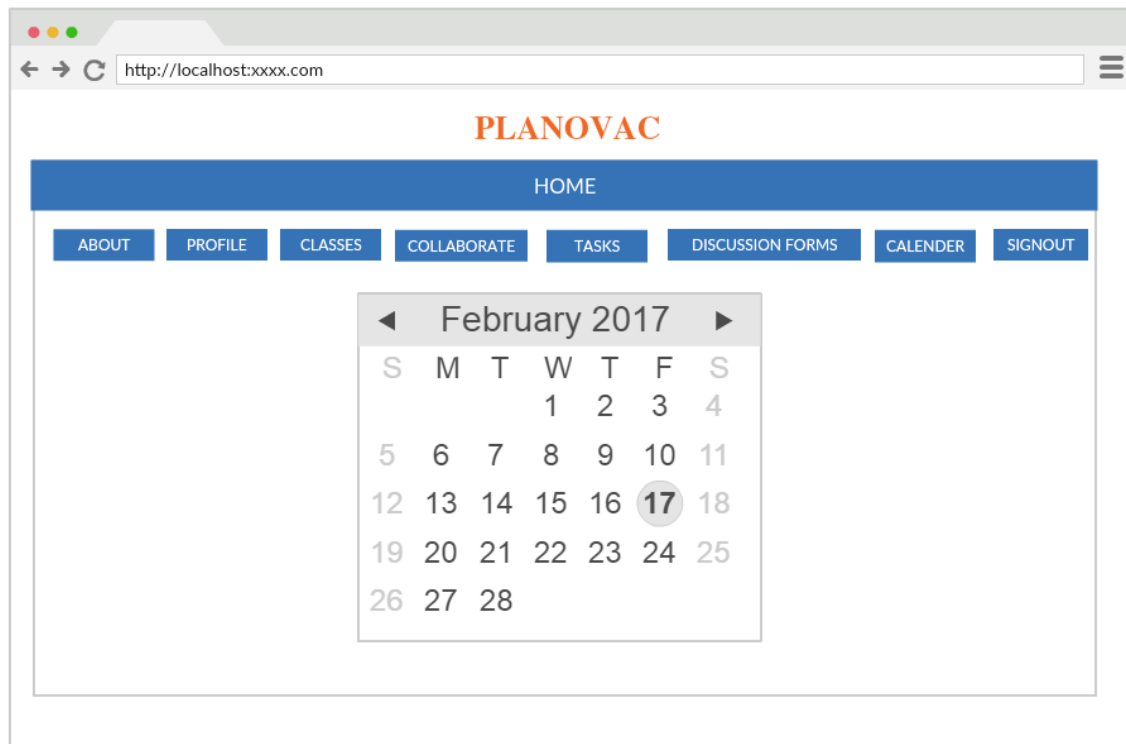
> ### *Deployment*

The Web Application is deployed to the Google Chrome Web Server and the screenshots are provided below.
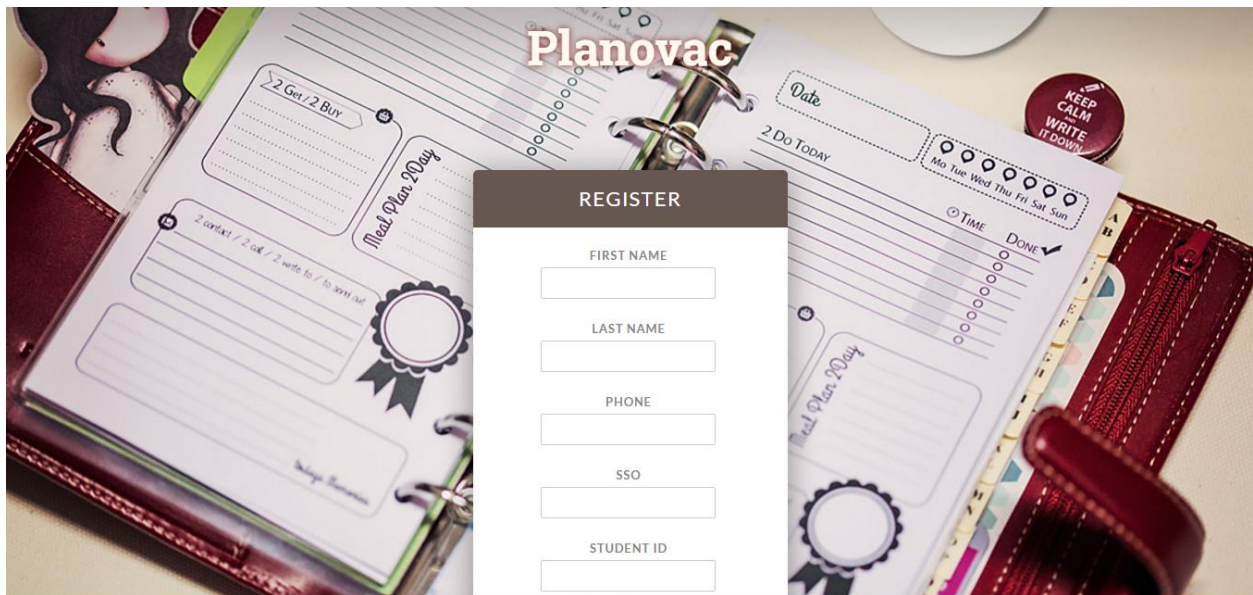
Login Page:

Successful Login:
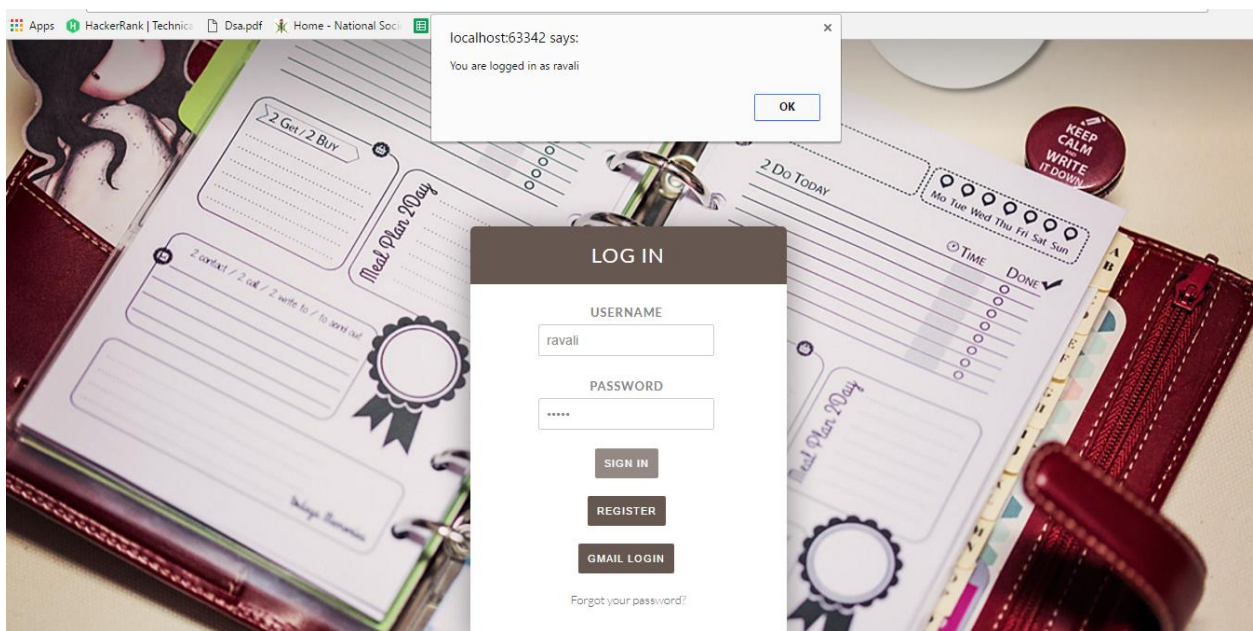
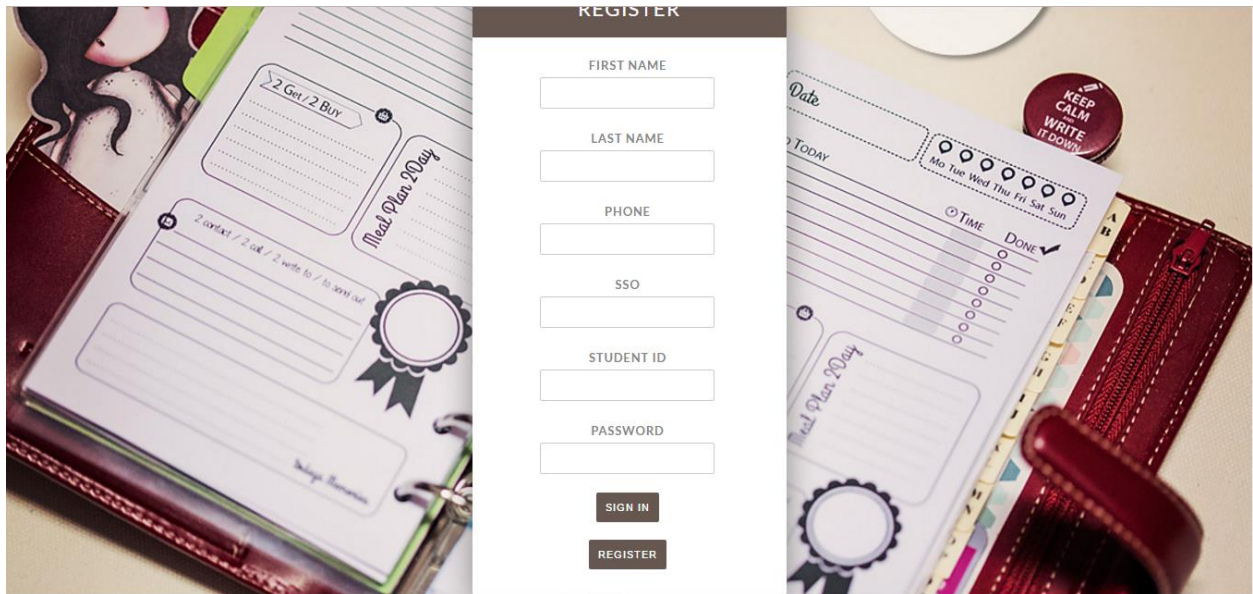Unsuccessful Login:
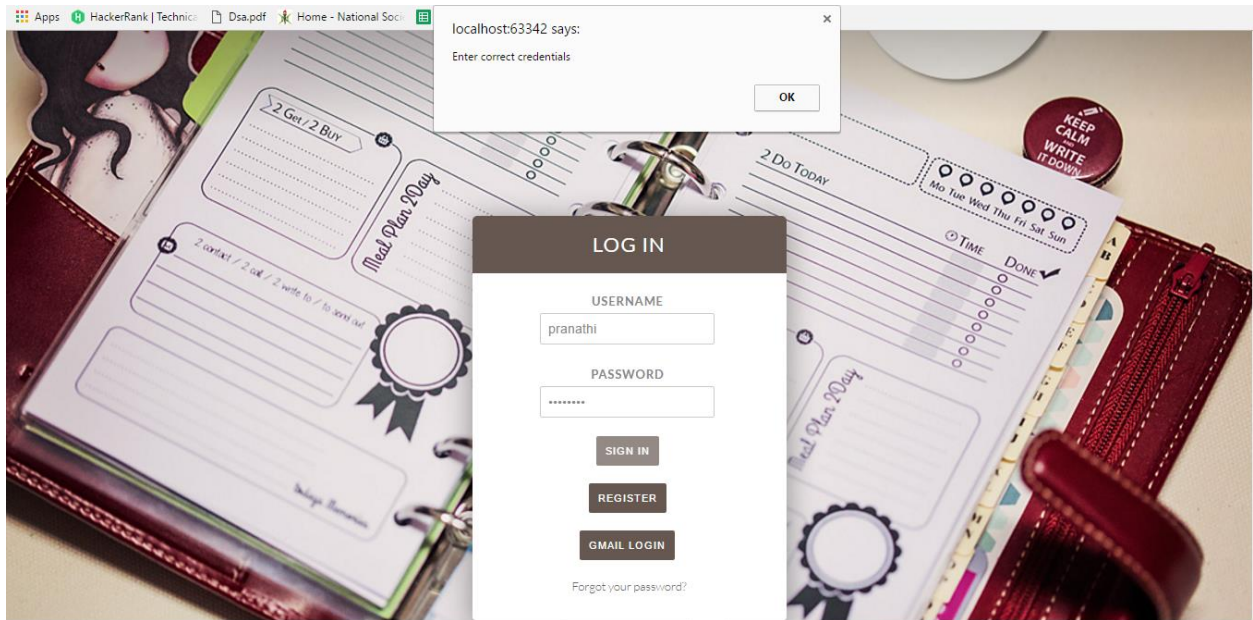
Register Page:

REGISTER

FIRST NAME

LAST NAME

PHONE

SSO
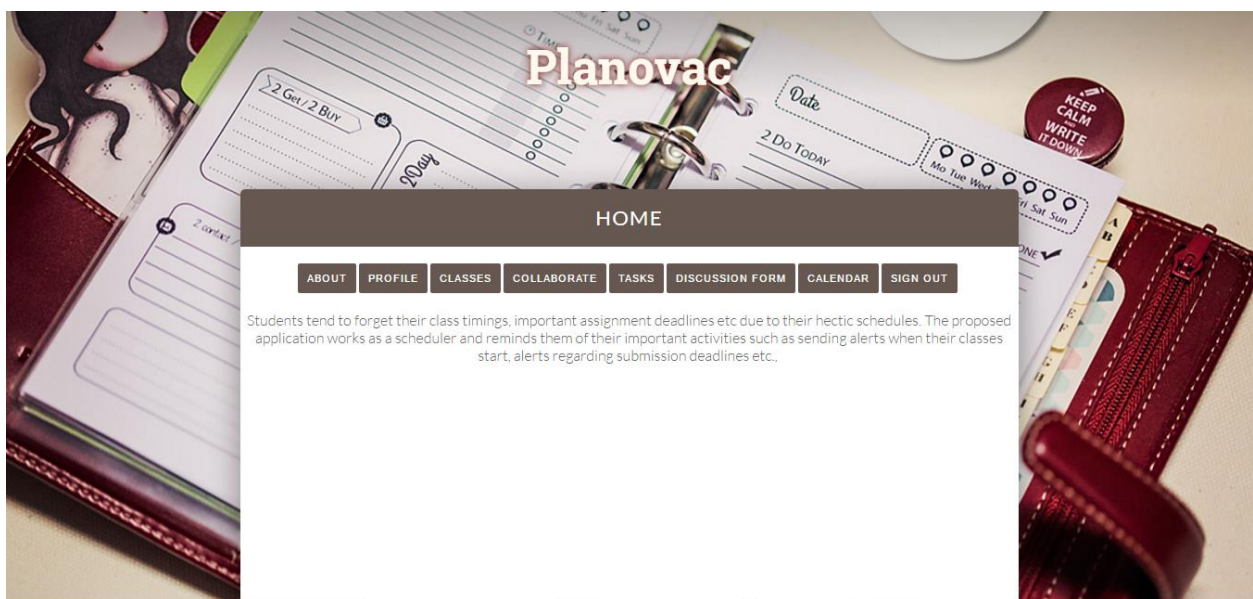
STUDENT ID

PASSWORD

SIGN IN

REGISTER

Home Page:

Planovac

HOME

ABOUT | PROFILE | CLASSES | COLLABORATE | TASKS | DISCUSSION FORM | CALENDAR | SIGN OUT

Students tend to forget their class timings, important assignment deadlines etc due to their hectic schedules. The proposed application works as a scheduler and reminds them of their important activities such as sending alerts when their classes start, alerts regarding submission deadlines etc.,
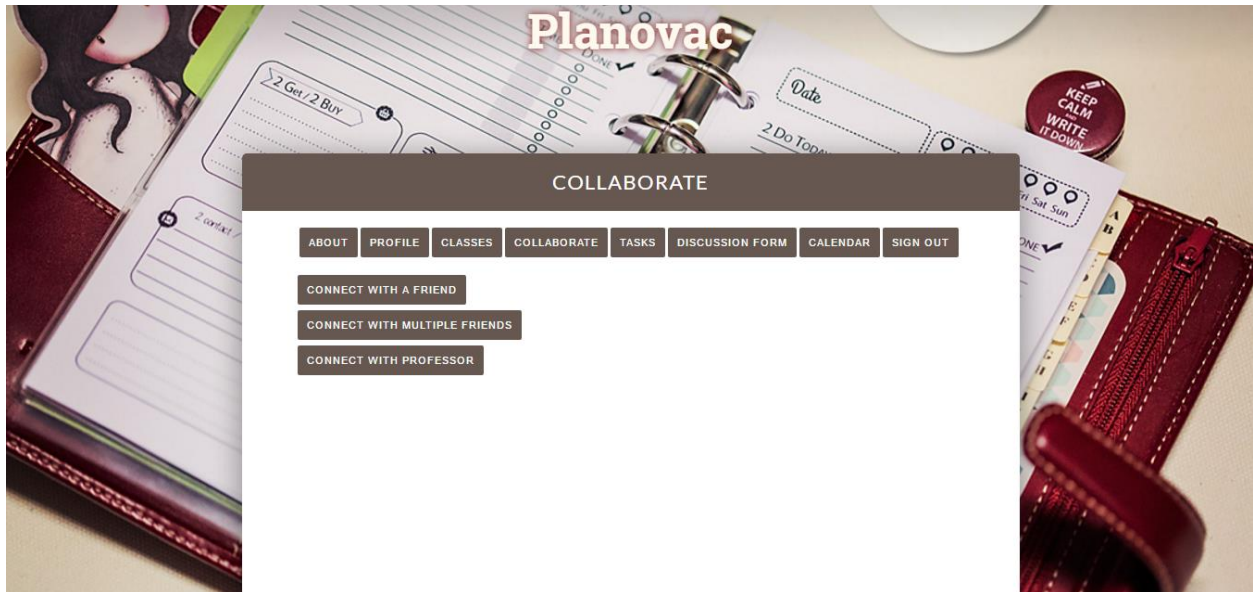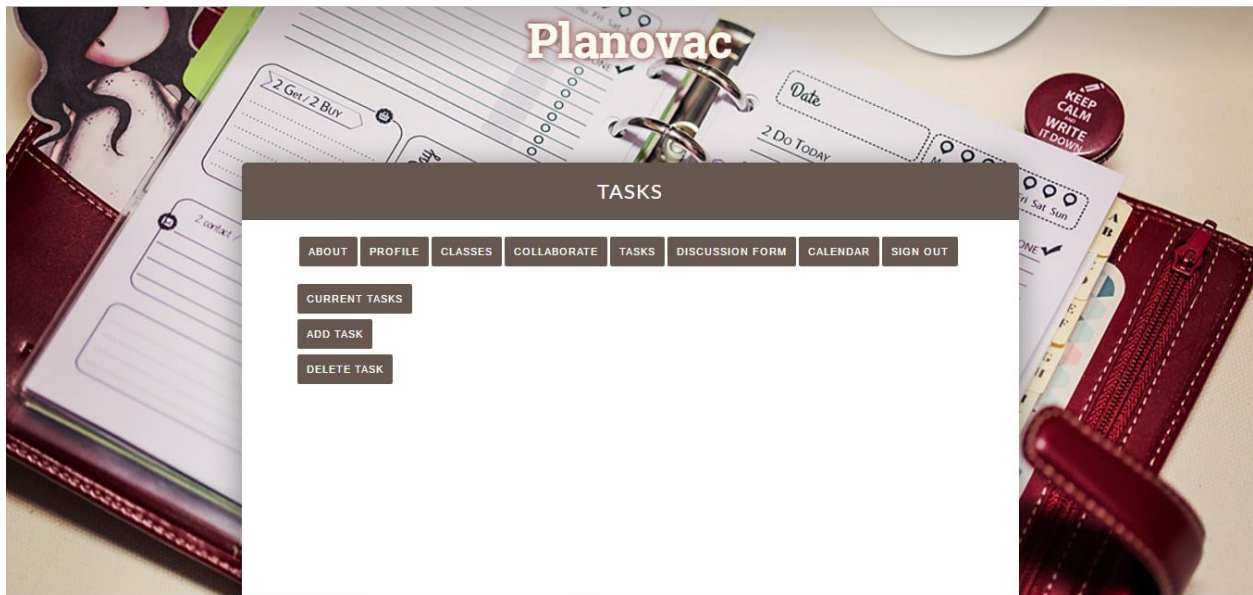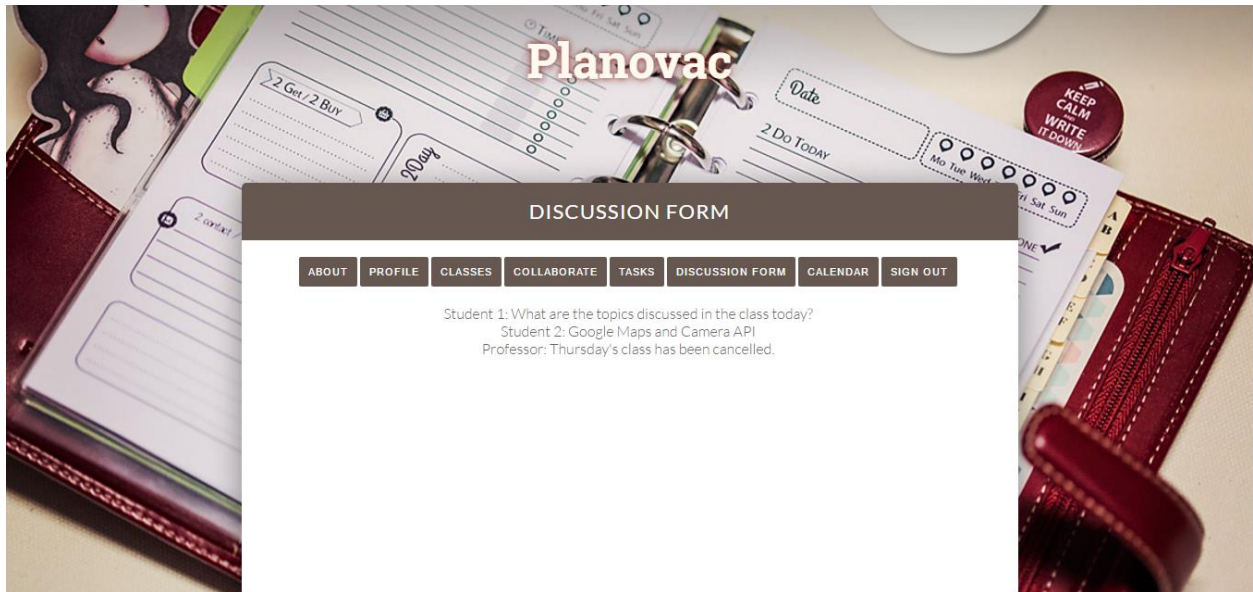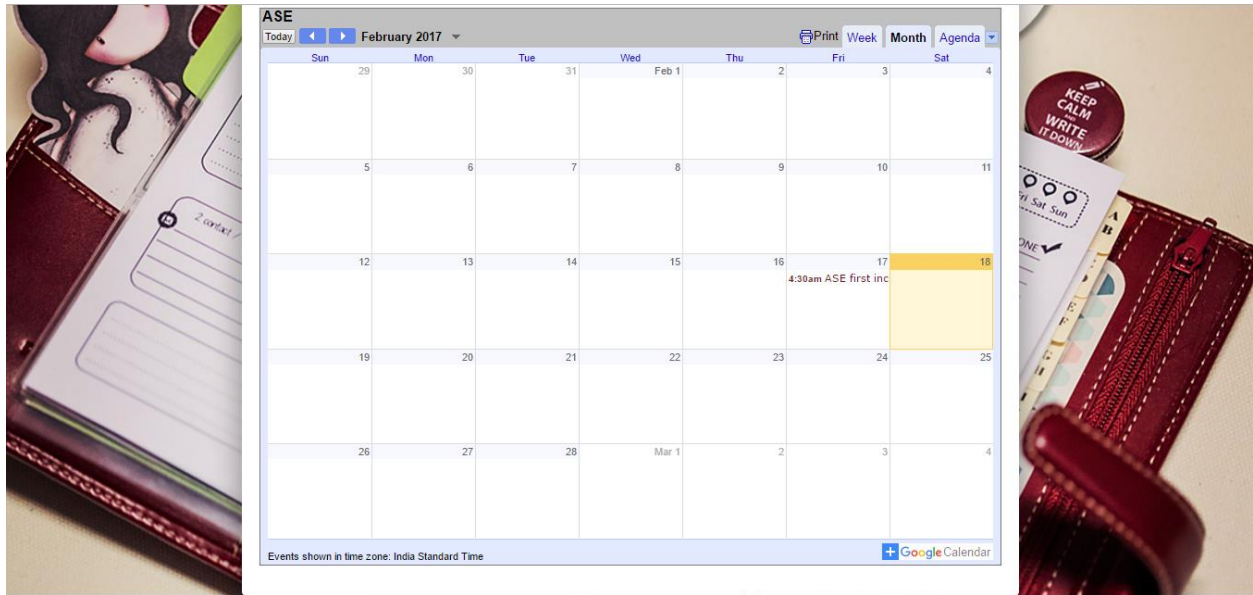
Profile:

Classes:
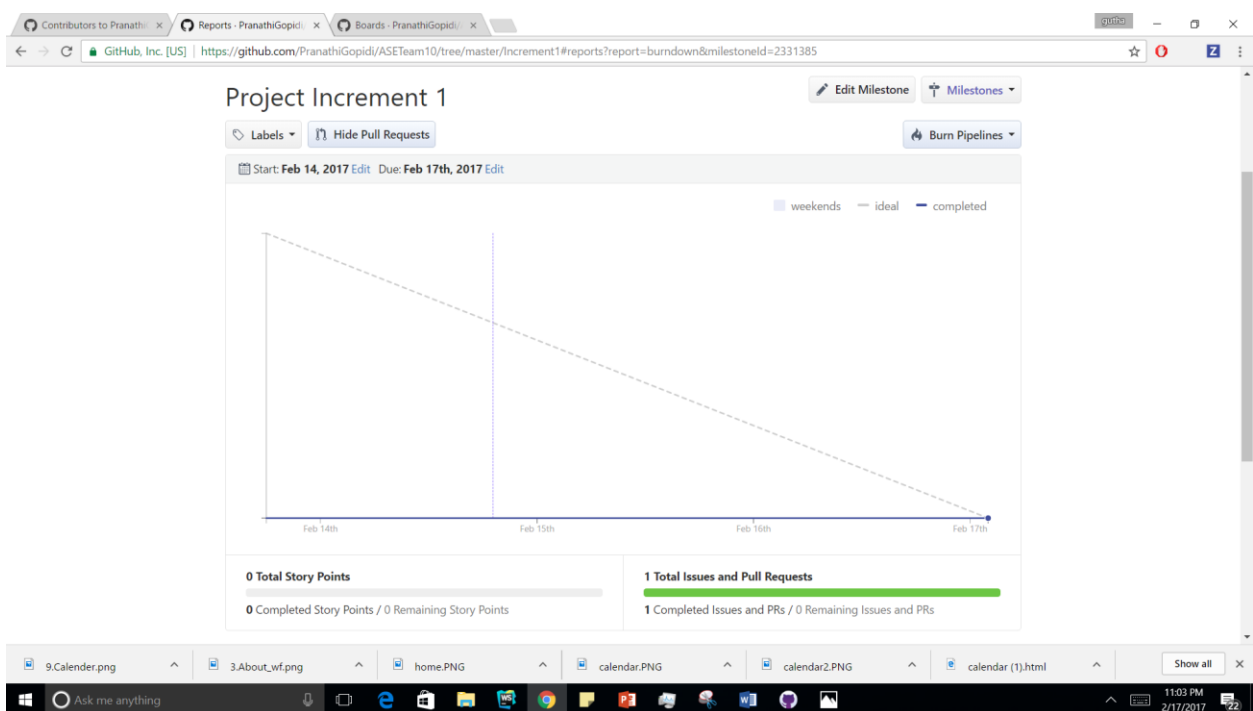
Collaborate:

Tasks:

Discussion Form:

Calendar:

Burndown Chart:

-

Project Management:

Work Completed:

Description:

- ✓ Planning for the project
- ✓ Creation of Login, Registration and home page
- ✓ Used Gmail OAuth login
- ✓ Used Calendar api to display calendar in the home page

Responsibility and time taken:

- Login and register page, Ravali 3hr
- Home Page and Calendar API, Ashwini 2hr
- UML Creation, Ashwini-30 min, Deepthi-30 min
- Social login OAuth, Pranathi-1hr
- Project Increment Report, Deepthi-2hr,Pranathi-2hr,Ashwini-30 min, Ravali-30 min
- User Stories, Pranathi 30min
- Wireframes, Deepthi 1hr
- Wiki, Ashwini 30min

Work To Be Done:

- ✓ Use MongoDB to store the information
- ✓ Use maps api to locate the class rooms
- ✓ Include chat feature

Contributions:

Everyone had equal contribution in the project increment.

Related Work:

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

<u>**Bibliography**</u>

UMKC Blackboard

https://shibidp.umsystem.edu/idp/profile/SAML2/POST/SSO?execution=e1s1

When I Work

http://wheniwork.com/l/aw-schedule-maker

Team Snap

https://go.teamsnap.com/promotions/fourmonths?utm_source=bing&utm_medium=cpc&u%20tm_campaign=General%2520Terms&utm_term=free%2520sports%2520scheduling%2520so%20ftware&utm_content=Scheduling