CEN 4010

Milestone 3 

**GROUP INFORMATION**

**Project name:**

Campus Snapshots

**Team Name:**

Group 3

**Team Members:**

Alina Tutuianu [atutuianu2017@fau.edu](mailto:atutuianu2017@fau.edu)

Christian Coronel [ccoronel2016@fau.edu](mailto:ccoronel2016@fau.edu)

Naiara Foster [nfoster2016@fau.edu](mailto:nfoster2016@fau.edu)

Prince Abraham [pabraham2015@fau.edu](mailto:pabraham20152@fau.edu)

**Trello account link:**

<https://trello.com/b/Z9kx4ifI/cen-4010-milestone-1>

Vertical Software Prototype:

<http://lamp.cse.fau.edu/~nfoster2016/4010>

**Course:**

CEN 4010 Principles of Software Engineering, Summer 2018

**Date**:

July 23, 2018

**Revision History:**

|  |  |
| --- | --- |
| Reason for Changing | Date |
| Edit Competitive Analysis | 07/19/2018 |
|  |  |
|  |  |

1. **Executive summary**

This is the proposal for the development of the Campus Snapshots, a web system capable of displaying and alerting students’ real-time information about events happening on campus. In addition, the system will allow registered users to upload information to the website notifying issues or events happening in real time. An administrative team will be in charge of maintaining the webpage, reviewing reports and checking the uploaded news on the page.

The goal is to create a powerful and user-friendly tool for students, a one-stop experience, where they can find useful and reliable information about the school. In addition, Campus Snapshots will help to quickly and more efficiently solve problems by reporting in real time.

Campus Snapshots will be developed in the summer term, 2018, for the class Principles of Software Engineering by group 3. Its creation will significantly improve the way today’s students get information about events in their schools. Although the school offers an alert email system that notifies us of certain events and emergencies, Campus Snapshots will improve upon the concept and allow students to report events or issues themselves in a website environment, this feature will allow information to be exchanged quicker, making problem solving and information sharing significantly more dynamic.

The Campus Snapshots will be a website, so it won’t have a need for particular hardware or local software other than a web browser. The project will use a database to store events and user information. A user friendly website will be developed that showcases all the features available.

The project will be developed throughout the semester and it will be divided in teams and stages. There will be a web development team, a database management team and a testing and quality control team. In the first stage the group will discuss and decide the main features and visibilities for the website. Next, we will develop the website and implement the desired features, later, we will integrate with the database. The final stage will be to implement and test the website as users.

**2. Competitive analysis**

After deciding about the project’s theme and its main features, we analyzed other competitive products available today on the market. Following is a table comparing the key features of competitors versus our product. In comparing the products we focused mainly on the high-level functionalities.

Campus Snapshots vs  Competitors

|  |  |
| --- | --- |
| Report instantaneously issues | **Infinite Campus, Inc**  User reviews/feedback |
| Large Database | **Campus Explorer, Inc**  User friendly website |
| Search for School Activities | **Data Snapshot**  Easy issue upload/report |
| 24/7 Service/emergency | **Infinite Campus**  More variety of report tools |
| Directions on safety procedures | **Campus Life**  Data on previous reports |

There are many online school systems, where university campuses around the country can keep up to date and track on environment issues and school activities. However, some competitors do not offer a 24/7 events/issues happening in real time. Competitor’s products are mainly a news page, rather than a tool for detecting and solving problems on campus. Campus Snapshots will offer a better quality of service for students to feel safe, a good living and good study environment keeping the campus clean and safe.

**3. Data definition**

Following are the definitions of the major data items we are going to use in this project:

|  |  |
| --- | --- |
| Campus Snapshots | A dynamic web system that publishes information about FAU’s on campus events and problems, in addition to allowing students to contribute by submitting information about issues happening in real time. |
| User accounts and privileges | The term ***general user*** will be used to define everyone that can interact with the web system. There are two types of general users: ***user*** and ***admin***. The user can be a student, faculty, or staff, that utilizes the web system to create a post, comment, or to look-up for upcoming events. The admin is a member of FAU’s facility management team, and utilizes the system to manage the problems reported through Campus Snapshots. Only the admin can change the status of a problem. |
| Registration info | To create an account with Campus Snapshots the general user needs a ***username***, a ***password,*** and a FAU email. The username consists of up to 30 characters, and the password should have between 8 and 15 characters, containing at least one special character. |
| Post | We will use the term ***post*** to refer to the content published by a registered user on the Campus Snapshot’s page. The post can be regarding either an ***event***, or a ***problem***. The event can be a concert, a game, or other types of meetings important to student’s life. A problem is an issue that requires the attention of the campus’ management. Examples of problems are leaking in classrooms, broken AC, overflowing dumpsters etc. |
| Media | Following image format will be supported: jpeg and png, with a maximum size of 5 MB. The accepted formats for videos are mp4 and mov at a maximum of 10 MB.s |
| Database | A collection of information, that is organized so that it can be easily accessed, managed and updated. In our project we will use the Oracle database. |

**4. Overview, scenarios and use cases**

Campus Snapshots is a web system that provides real time information about the life on FAU’s campus. The system allows users to share events and activities that take place on campus, but also to report problems that need to be fixed. The administrators of the university, called in the document admins, can use the system to monitor the well functioning of the campus, taking action when necessary to solve the reported issues.

Scenario for reporting a problem in Campus Snapshots

The user that wants to report the problem has successfully logged on to Campus Snapshots. The user has one or more digital photographs documenting the problem saved on a personal computer.

After the user chooses to report the problem, the system will ask the user to complete a report form. The user is asked to choose from a list of possible problems, to choose the location, and fill in the date of the observation. Based on the specifications given by the user, the system will search for similar reports. It will then generate a list of posts containing similar problems, and the user will have to check that he/she isn’t reporting the same problem as other users. If the problem is unreported, a new post is being created. The user is required to write a brief description and to upload photos. On completion of the post, the system automatically sends a notification to the admins, and generates an on-screen message to the user that the problem has been successfully reported to the administrators.

The user will get notifications whenever the status of the problem was changed.

Scenario for monitoring problems in Campus Snapshots

The admin is successfully logged on to Campus Snapshots. The admin can view the problems classified in the following categories: reported, in process, solved. If the admin chooses to see the reported problems, the system will display a list of the new posts. Each post has a status, which the admin can change. If the admin chooses to change the status of a problem to in process, the system will generate a form asking the admin the name and contact information of the person in charge of solving the issue. If the admin chooses to change the status to solved, the system automatically sends a notification to the user posting the problem, and to the other users that subscribed to the thread.

Use Cases

I, in the role of a user, need the functionality of creating a new post, to achieve the goal of communicating an event or a problem from the university campus.

I, in the role of a user, need to be able to upload photos, to achieve the goal of documenting a post.

I, in the role of a user, need the functionality of reading and adding comments on existing post, to achieve the goal of creating a thread.

I, in the role of an admin, need the functionality of notifications, to achieve the goal of being informed about the current problems reported.

I, in the role of a user, need the functionality of notifications, to achieve the goal of being informed about the status of a problem.

I, in the role of an admin, need the functionality of a problem status, to achieve the goal of updating the status of a reported problem.

**5. High-level functional requirements**

Following is a list of the functional requirements of the web system and their priorities.

|  |  |
| --- | --- |
| Priority Level | Functional requirement |
|  | 1. The application shall offer information about events and problems |
| 1  1  1  1  1  1  1  3  1  2  2 | * 1. The general user shall be able to create a new account   2. The general user shall be able to change his/her password   3. The general user shall be able to reset a forgotten password   4. The general user shall be able to login to his/her account   5. The general user shall be able to logout from his/her account   6. The user shall be able to create a new post   7. The user shall be able to upload photos to their post   8. The user shall be able to upload a video to their post   9. The user shall be able to read and add comments to a post   10. The user shall be able to search for an existing post   11. The user shall be able to view the latest posts on his main page |
|  | 1. The application shall offer a notification system |
| 1  2  3  3  3 | 2.1 The admin shall receive an in-app alert and an email when a new problem is reported  2.2 The user shall receive an in-app alert when the status of a reported problem changes  2.3 The user shall be able to opt for receiving notifications via SMS and/or email  2.4 The user shall be able to choose what kind of notifications he/she wants to receive (reported problem updates / upcoming events)  2.5 The upcoming events shall be organized into a calender |
|  | 1. The admin shall be able to manage the reported problems |
| 1  1 | 3.1 Each reported problem should have a status report (reported, in process, solved)  3.2 The admin shall be able to change the status of a problem |

**6. Non-functional requirements**

1. The application should run on the latest versions of the following web browsers: Chrome, Internet Explorer
2. Data shall be stored in the database on the server
3. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users
4. Security of the site shall require users to register and login in order to make a post
5. The language used shall be English
6. There should be no prior training required for using the application
7. Site shall be attractive and media rich in appearance
8. The site shall be easily searchable by the major search engines

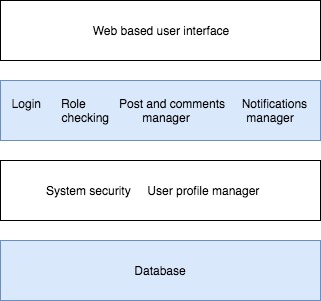
**7. High-level system architecture and database organization**

Following is a list of the main software products, tools, languages and systems that we will use in the development of Campus Snapshots:

Back end: MySQL and PHP  
Front end: HTML5, CSS, JQuery , JavaScript  
Other: Internet explorer, Chrome, Mozilla, Microsoft edge, GitHub, Canvas, MEETS for WebEx, WhatsApp, Atom and Brackets.

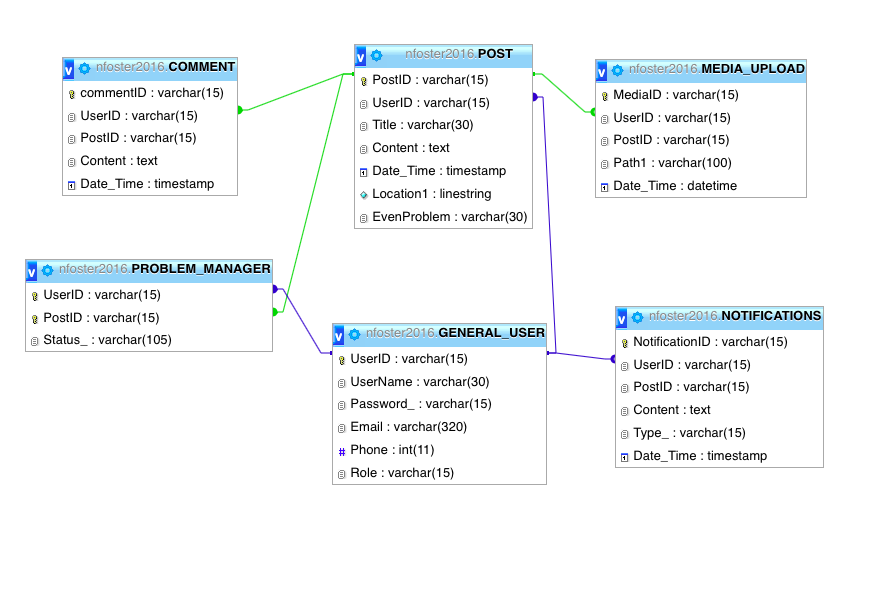
High-level Architecture

The high-level architecture of Campus Snapshots is modeled using a layered approach. The top layer is a browser-based user interface. The second layer provides the user interface functionality that is delivered through the web browser. It includes components to allow users to log in to the system, and checking components that ensure that the operations they use are allowed by their role. This layer also includes a post and comments manager that allows users to create a post, upload media, add and read comments, and change the status of a problem. The last component at this layer is a notification manager. The bottom layer is the system database.



Database organization

Following is an entity-relationship diagram showing the main entities and their attributes that will be included in the database of Campus Snapshots.



Media storage and search/filter architecture

Media will be stored in the database using the BLOB data type.

The Problem Manager table shall be sorted based on the status: reported, in progress, solved. A binary search algorithm will be used when searching within this table. For the General User table a clustered B+ tree file shall be kept with search key <UserID, UserName>. The algorithm for B+ tree search is given below, as per Ramakrishnan (2003):

func tree\_search (nodepointer, serch key value K) returns nodepointer

if \*nodepointer is a leaf, return nodepointer;

else

if K < K1 then return tree\_search(P0, K);

else

if K>=Km then return tree\_search(Pm, K);

else

find i such that Ki <=K <Ki+1

return tree\_search(Pi, K);

endfunc

Our own APIs

<config>

<api>

<resources>

<user translate=“title” module=“user”>

<title>User Resource</tittle>

<methods>

<list translate="title" module="user"> <title>Retrieve User</title> </list>

<create translate="title" module="user">

<title>Create User</title> </create>

<info translate="title" module="user">

<title>Retrieve User data</title>

</info>

<update translate="title" module="user">

<title>Update User data</title> </update>

<delete>

<title>Delete User</title>

</delete>

</methods>

<faults module="user"> </faults>

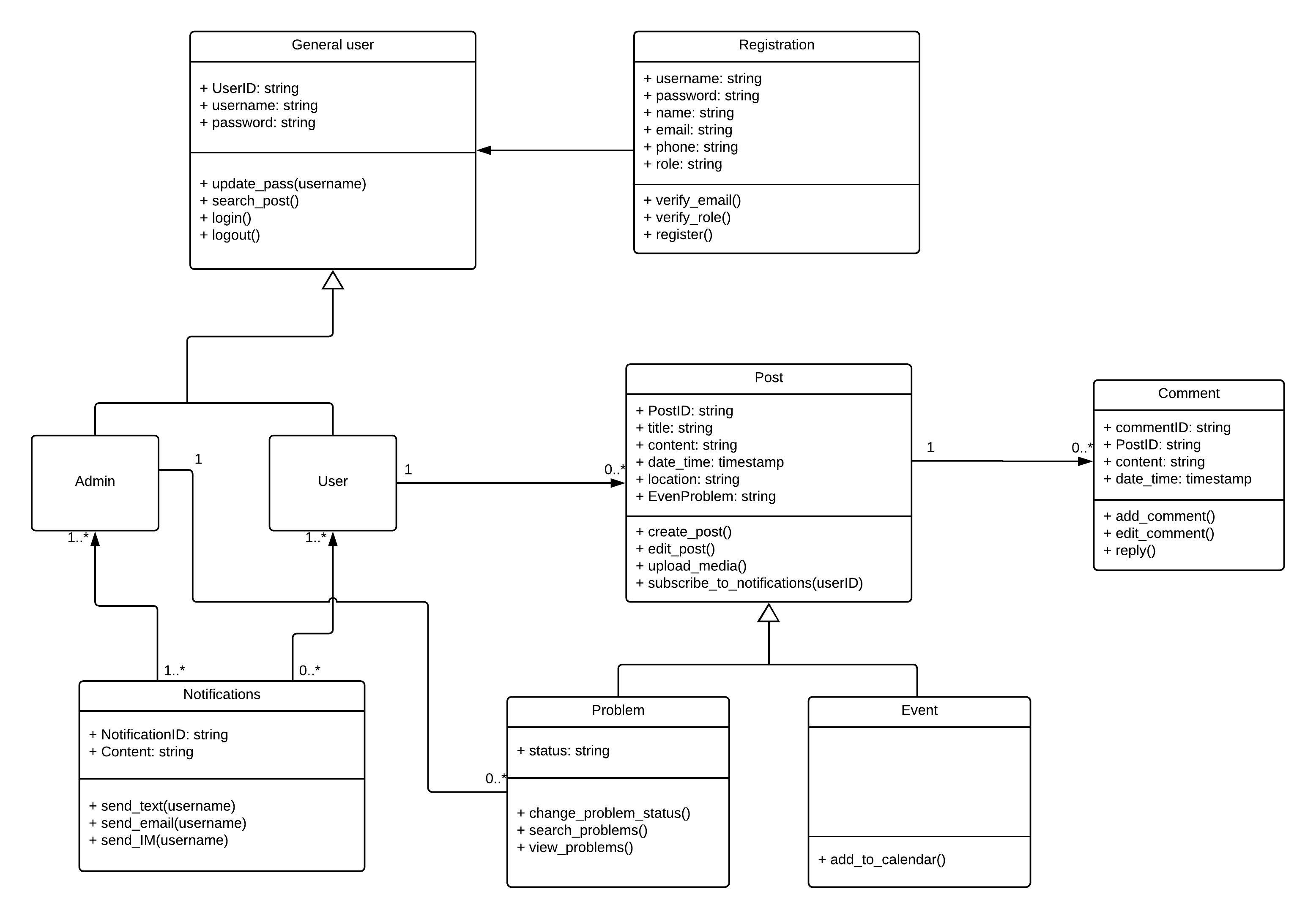
</customer>

</resources>

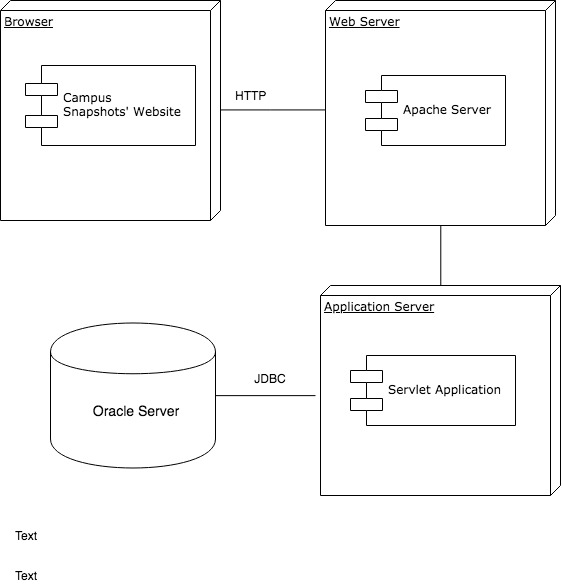
</api>

**8. High-level UML diagram**

Campus Snapshots’ class diagram



Component and deployment diagram



**9. Key risks**

The only risks that we are facing during the development of the project are schedule related. Due to most of the team members having multiple classes and working at the same time, we had trouble finding a time to meet up. There are also only two weeks left until the final project delivery.

To solve this risk we used WhatsApp to communicate and distribute roles. Trello also helped us managing the milestones and committing to our tasks.