1 Title page

CEN 4010 Principles of Software Engineering

MILESTONE 3 PROJECT PROPOSAL AND HIGH-LEVEL DESCRIPTION

Team Name: THE DREAM TECHS
Team/Group Number: 9

Group Members:

Samuel Parker - Sparke25@fau.edu

Joshua Vayntrub - jvayntrub2015@fau.edu

Abdullah Shafique - ashafique2014@fau.edu

Donny Stephens II - dsteph18@fau.edu

Date: November 4, 2019

History Table:

1. 10-28-2019 Database setup.
2. 10-30-2019 Updated GUI.
3. 11-1-2019 Web Pages Created.
4. 11-2-2019 Website Connected to Database.

2 Executive Summary

In this project, we will create a web system called Campus Snapshots. By using this system, people (students, teachers & staff) on campus can report issues that need to be addressed as well as campus events. We are developing this project to help students and faculty stay in the know of issues, events, and news. Also, to ensure that the upkeep, the appearance, and maintenance of the campus remains properly maintained.

3 Competitive analysis

Campus Snapshot	**Competition (FAU)**
1. Report instantaneous issues/events	Only can report concerns NOT events instantly.
2. Real-time Snapshots of campus status to University administration (teachers, faulty)	If need immediate attention, call 9-1-1.
3. Motivate users to upload and report problems and events	Doesn't motivate users to upload and report problems and events
4. Ability to upload pictures along with status update.	Can't upload pictures at all.

The planned advantages for this project page are the instantaneous reports of issues and events, University administration is notified immediately for issues, the motivation the page gives users to upload and report problems and events, along with the ability to upload pictures with statuses or issues.

4 Data definition

Table: User
Username - varchar (30) - Primary - Unique
Password - varchar (100)
Email - varchar(30) - Unique
Account_Type - varchar (30)

Table: Post
ID - int (11) - Primary - Unique
Text - text
IMGpath - varchar (30) - Unique
Post_Type - varchar (10)
Date - date
Username - varchar (30)

5 Overview, scenarios and use cases

The web system called Campus Snapshots will allow students, teachers, staff, and all campus goers to report all news, events, technical breakdowns, sanitary issues, and any other issues that take place on campus instantaneously. A student can use this system to post a need for tutor, or even offer tutoring services. A teacher can report a toilet overflowing, or an activity organizer can announce a charity concert. Campus statuses are also given to campus staff when they periodically check the system to monitor campus activity. The system can also be used for security reasons. A student can post suspicious activity, and a campus officer can check the system to irregularities reported by other campus goers.

Sam is a student at FAU. He wishes to organize a fundraising concert to raise money to fund an initiative to bring awareness to social injustice and poverty in inner cities. The concert will consist of local talent that includes students, and any faculty and staff. The concert headers will be determined by having a talent show, and the finalist will be the entertainment for the concert. Sam needs to be able to spread the word for both the talent show and the concert to all campus goers.

Sam sends a message through Canvas to his fellow students in his calculus class on how he can best get the word out to all fellow campus goers of his upcoming talent show and concert. A student messages back to him that he can use a system called Campus Snapshots, which is a site that allows all campus goers to update, in real time any news and reports of things happening on campus. Campus goers can post pictures associated with the news that the are sharing, that will be accessible to all other campus goers. Sam uses Campus

Snapshots to upload content related to his upcoming events to notify campus goers on the details and demographics to get the word out.

A user or a group of users have a photo(s) to be posted on the system. These can be photos taken on a smartphone, or text typed on a post directly to the system. The user must create an account to the system.

The user logs into the system giving their username and password. Once logged in, the campus goer will be redirected to the members page that will list a feed of user upload submissions, and the page will also give them the option to type a description of the event in a text box, along with any associated media. Uploads are listed under the description of the event, along with the poster's username. Once an event is posted, the system sends an email to the poster confirming the post was successful, and a message is displayed to the poster that their post and event has been uploaded and is now visible to all users.

Possible issues with the system include the following:

The system is self-sustaining, and because 24-hour monitoring cannot be done by the moderator, uploaded content cannot be filtered in real-time. This poses a problem given the real-time nature of the system. Moderation will require other users report content that is either offensive, abusive, or unsafe for any user of the system.

The system is user driven, and being so makes it difficult to validate the trustworthiness post that are uploaded by users. Moderators will have to frequently check posts for user violations and compliance issues.

The system administrator may be logged in, flagging posted content that goes against the user agreement.

System state on completion:

The campus goer is signed into the system. The event and associated media have been posted and is currently visible to all users that are signed in. Posted information is visible to the system administrator, who will do nothing if the content of the posted event is compliant to the user agreement.

6 ~~Initial list of~~ High-level functional requirements

1. Create Account: First-time users will be able to create an account and choose their account type based on their status with the school. Different account types include student, faculty, and staff. Each account is associated with a unique username and password.

    1.1.1 To create an account, users will be prompted to enter a username, password, email, and account type.

    1.2.1 The username and email will be compared to those in the db, and if there are no matches, the submitted info above will be saved to the database, and the user will be redirected to the members page.

2. Sign In with Current Account: Each existing user will be asked to sign into their accounts at the start of the session.

    2.1.1 To sign in, user will be prompted to enter their username and password.

    2.2.1 The username and password will be compared to those in the db, and if found, user will be given a welcome message, and redirected to the members page.

    2.3.1 If not found in the db, the user will be given a message saying the username and password was not found as well as saying if they don't have an account to sign up , and will be redirected to the sign in page.

3. Report Issue: Once signed in users will be able to report issues based on what they are experiencing. Issue types include (for example) suspicious activity and technical breakdowns.

3.1.1 Once signed in, user will be able to view all updates submitted by previous users that are saved on the database.

3.2.1 On the same member page, a user will be able to upload a picture, and associated text.

3.3.2 Once uploaded, the user will be shown a message confirming submission and redirected to the member page where their submission will be displayed along with all other user submissions saved on the database.

4. Post Status: All users can make status posts.

4.1.2 All users will be able to upload statuses. It is up to the discretion of the user what the status is about based on relevancy. So a student account would post a status about an event, whereas a staff member would post a status about an update to an issue that someone reported

7 List of non-functional requirements

1. The application should run on two latest versions of all major web browsers: Mozilla, Safari, Chrome, IE, Firefox, etc.
2. Data shall be stored in the database on the server
3. Privacy of users shall be protected, all privacy policies will be appropriately communicated to the users.
4. The language used shall be English.
5. The site shall be very easy to use, intuitive (no prior training required).
6. Site shall be attractive and media-rich in appearance.
7. Security of the site shall require users to register and log in (students to decide what functions require registration and login).
8. Effort shall be made to make the site easily searchable by internet search.

8 High-level system architecture and database organization

1) High level Architecture of the code must be consistent with UML class diagram (see below).

User will be able to register, log in, and log out. They will consist of a username, password, email, and account type. Posts will be able to be created or deleted (by their respective user) and will contain an ID, text, image path, post type, date, and username.

2) DB organization: Describe the main database schema/organization (high level), e.g. list main DB tables and items in each DB table

The tables in the DB are User and Post. The Register_Info table is what will comprise of the users of the website, each with a username (primary and unique), password, email (unique), and account type. The password is hashed using a bcrypt hash function. The account type is what determines whether a user is a student, teacher, or staff. The username and email are unique in that they cannot be used more than once per account.

The Post table will hold the information each user posts. It will have an ID (primary and unique), text, image, post type, date, and username. The ID will be a unique marker to the post so that each post can be identified properly in the DB, especially incase of deletion of the post, the text will be the text body, the image will denote the posted image (if there is one), the post type denotes if it is an issue or status, the date will signify when the post was made, and the username is there to show who is the one who posted the issue/status by connecting it to the Register_Info table.

3) Media storage: Decide if images and video/audio will be kept in file systems or in DB. Describe any other special data format requirements like for video/audio/GPS etc.

Images that are posted in the issues/status posts will be kept in the database

4) Search/filter architecture and implementation: what will be the algorithm for search; what DB terms will be searched, how it will be coded and organized in the DB. Similarly, say what DB items will be filtered/sorted

When creating a user, the code will search to see if the username that is being registered is already as well as the email. If either of those are in use, it won't allow the user to register until they pick a different username/email. If deleting a post, the ID for it needs to be searched for as the unique identifier of which post is being removed.

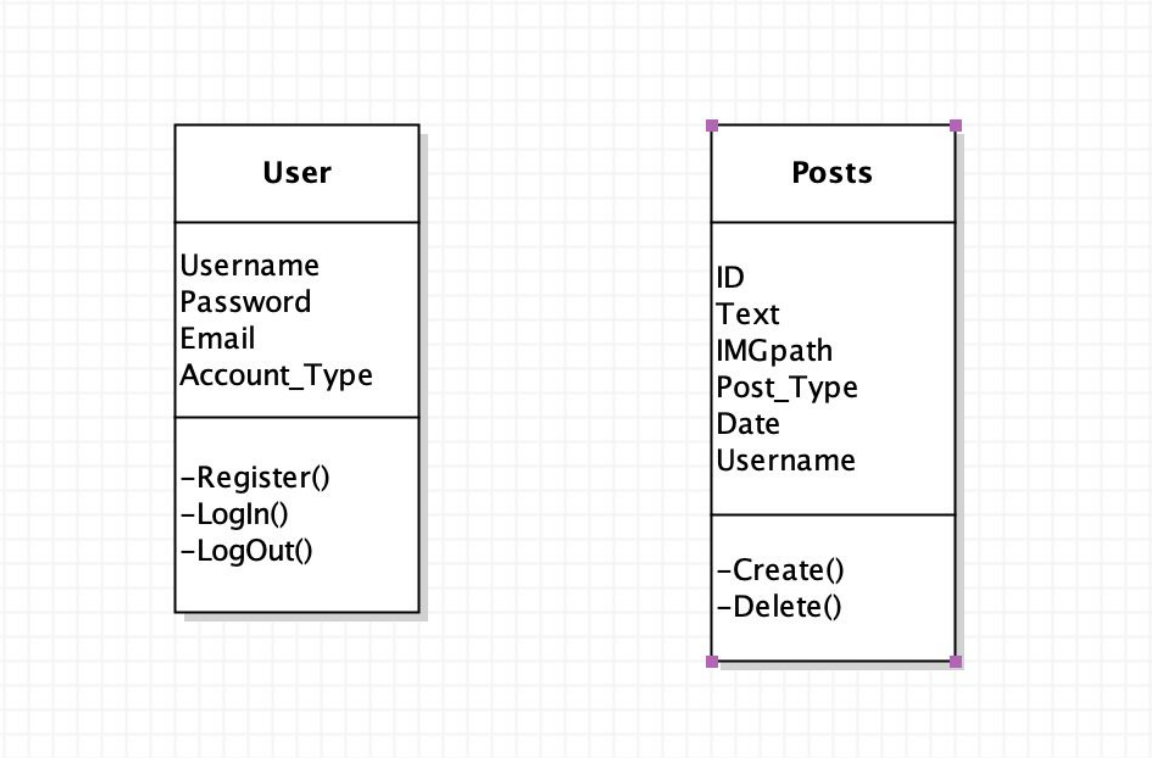
5) Your own APIs: Describe and define at high level any major APIs that you will create.

We will not be creating our own APIs for this project.

6) Describe any significant non-trivial algorithm or process (like rating, ranking, automatic prioritizing of items etc.)

The newest posts by users will appear at the top of the screen, like a stack, based on the date. For passwords, passwords will be hashed using the bcrypt hash method.

9 High-Level UML diagrams



10 Identify actual key risks for your project at this time

1. Technical risks.
   1.1. We will have to figure out some unknowns for certain high-level requirements for the project.
      a) While we have the needed skill sets to code in the needed languages, we will need to figure out how this would need to be done technically. We plan to research the web for an algorithm needed to accomplish this functionally, and rewrite the code using this information.

Team member participation form:

Team Members: A, B, C, D all participated evenly within this project Milestone.