CEN 4010: Principles of Software Engineering

Milestone 3: More Detailed Requirements, Architecture and a Vertical Software Prototype

Group Number: 4; Team Name: “Cheesy“

Group Members:

* Anthony Pancione – apancione2018@fau.edu
* Jonah Livingston - livingstonj2015@fau.edu
* Nicholas Child - nchild2019@fau.edu
* Tyler Do - tdo2018@fau.edu

Due Date: July 14, 2020

Vertical Prototype: <https://lamp.cse.fau.edu/~cen4010s2020_g04/server/index.php>

Video Demonstration: <https://youtu.be/uxS5WedS710>

Revision Table:

|  |  |
| --- | --- |
| Date: | Revision: |
| 6/15/2020 | Added Revision Table, Executive Summary, Competitive Analysis, Data Definitions, Overview & Scenarios, High Level functions, Non-Functional requirements, High Level System Architecture, Team Roles, added documentation onto GitHub |
| 7/10/2020 | Added High Level UML Diagrams, High-level System Architecture, and Key Risks sections alongside a link to our vertical prototype DB search demonstration and website. The website’s basic functionality also has been implemented. |

Executive Summary

For our project, we chose to create a communication platform that allows people who are stuck in quarantine to interact and share ideas. Due to the COVID-19 outbreak, students at FAU have been unable to communicate in a healthy sociable manner. To fix this issue, our team has come up with a website that will allow students at Florida Atlantic to create, comment and like posts made by other students. This project, named “OwlTalk”, is useful because socialization at FAU has been almost entirely erased since quarantine began. College is a critical time to socialize and create new connections. We believe there is no available platform where FAU students can freely socialize in an unmoderated environment. It will be a site of the students, by the students and for the students. The setup will be a reddit style posting board where verified users can make new posts and like/comment on old posts.

Competitive Analysis

Competitor Features vs. Ours:

|  |  |  |
| --- | --- | --- |
| Product | Similar Features | Why ours is superior |
| FAU Facebook Group | Allows students to post and communicate in a popular social media, and is relatively unmoderated | Ours is solely dedicated to the communication of students, (rather than being just a page of a major social media platform). The Facebook group is difficult to get to, fairly formal, and not widely used. Ours will better facilitate socialization through the idea of an unmoderated |
| Canvas | Simple, aesthetically pleasing site for teachers to communicate with students and occasionally used for student to student communication | Canvas is too formal of a space to allow students to really create friendships and get to know each other. Our site will be orientated to being informal and fun |
| Discord Groups | Many students will create discord groups to interact between clubs, classes and friend groups but are hard for outsiders to access | Our site will be welcoming to everyone if and only if they have a verified FAU email address, allowing all students to be welcome to the site |

Data Definition

* Moderators- the users who have privileges to remove posts or ban users
* Post Block- each post will be in a post block which will contain data on the posts time of post, content, likes, posting user and comments
* Main Page - the main discussion page that holds all the most recent/popular post blocks
* User Page- the page of an individual user that shows all their posts and public information

Overview, scenarios and use cases

The application will begin with a login window. Depending on who logs in, a user or admin, the application performs different functions. Users can also choose to make a new account or recover password.

A user logs in with a username and password and is redirected towards the main page. The main page displays the main discussion blocks that holds the most recent/popular posts. Above sits a navigation bar with different subjects to browse through. From the main page the user can post, comment, like, and vote on blocks. Users may select a block to display the main thread and view it more in detail.

The user can select profile to view user page and edit information. On the user page the system displays all their posts and public information. The information consists of name, DOB, grad year, major, interests, hobbies, photos, and email. The user can choose to edit their existing blocks or delete blocks.

An admin logs in with a username and password and is redirected towards the main page. The admin has the same functionality as a user but can delete any block and ban users.

Use Cases:

1. Admin Logs in
   * 1. Admin logs into application
     2. Admin moderates discussions
2. User Logs in
   * 1. User logs into application
     2. User can enter username and password or create a new account
     3. System redirects to main discussion

Variation #1. Invalid Login

* + 1. Start at step A. or B.
    2. The application determines there is no account associated with user information
    3. The application signals the user that the login has failed

1. User Browses Main Discussion
   * 1. User browses main page
     2. System displays most recent/popular discussion blocks
2. User Views Block
   * 1. User selects a block
     2. System updates window and redirects to main thread
3. User Post Block
   * 1. User selects post new block
     2. System waits for user to fill out information
     3. System updates discussion board with new block
4. User Edits Block
   * 1. User selects edit block
     2. System waits for user changes
     3. System updates discussion board with revised block

Variation #2. Admin Edits Block

* + 1. Admin selects edit block
    2. System waits for admin
    3. System updates discussion board with revised block

1. User Views Profile
   * 1. User selects view profile
     2. System updates window and redirects to user profile
     3. User can view their posts and edit personal information (name, DOB, grad year, major, interests, hobbies, and email)
2. Admin Deletes Block
   * 1. Admin selects delete block
     2. System updates discussion board and removes block
3. Admin Bans User
   * 1. Admin selects a user
     2. Admin selects ban user
     3. System gathers user information and disables account
     4. System removes all blocks associated with account

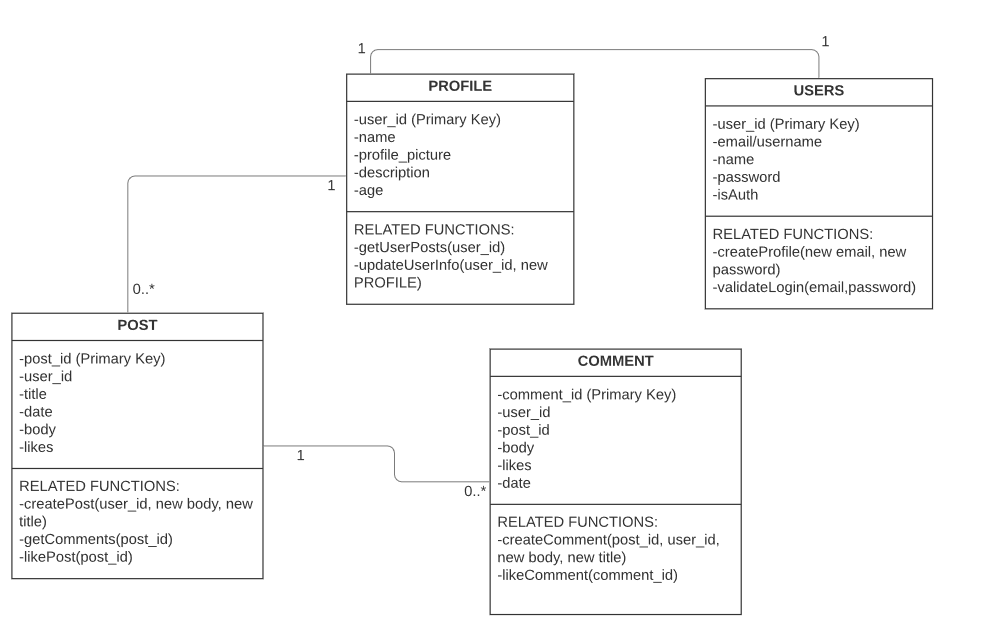
Initial list of high-level functional requirements

The website will be available to registered students of FAU and have the following functionality:

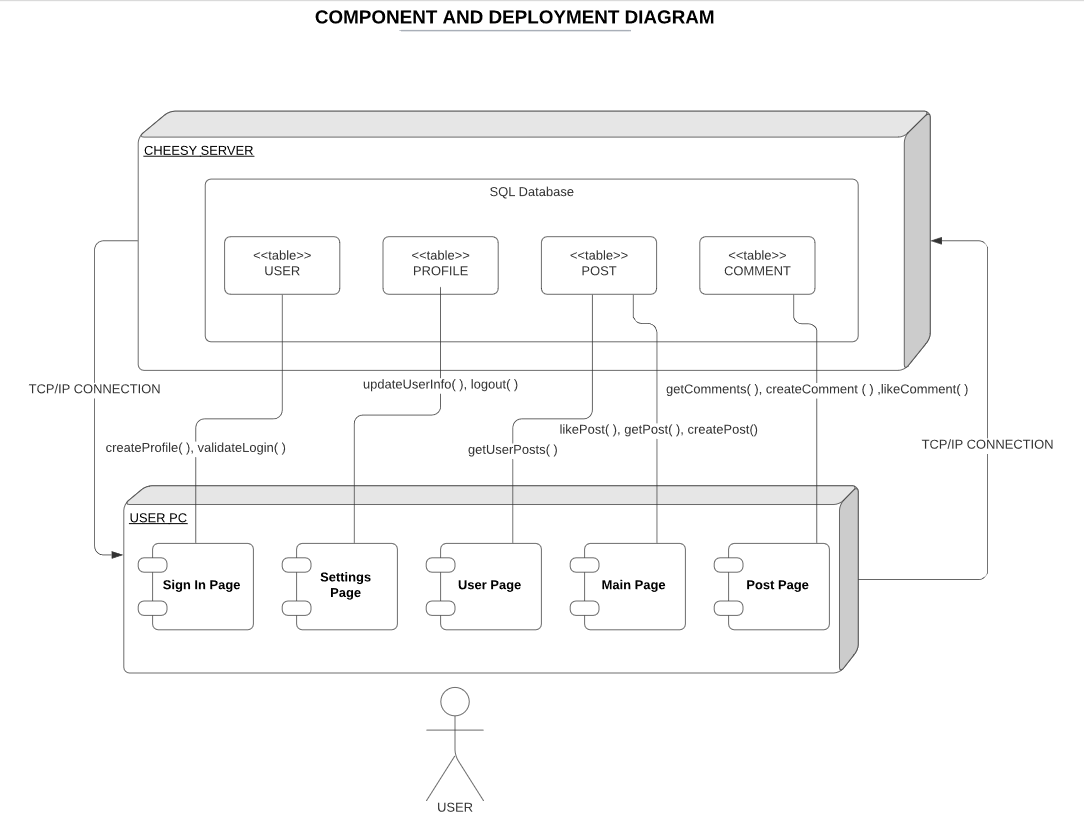
* Create Account: First-time users will be able to create an account if they have a valid FAU email. Each account is associated with a unique username and password **(Priority 1)**
* Sign in with Current Account: Each existing user will be asked to sign into their accounts at the start of the session **(Priority 1)**
* Edit Profile: Users can customize the profile image that will be associated with everything they post as well as various other account details **(Priority 3)**
* Create Post: Users will be able to create posts that can feature descriptions and images **(Priority 1)**
* Reply Comments: Users will be able to add comments and see previous comments on existing posts **(Priority 2)**

High Level UML Diagrams

Interfaces:



UML Component and Deployment Diagrams:



List of non-functional requirements

1. The website should be functional on at least two of the latest versions of major web browsers such as: Mozilla Firefox, Google Chrome, etc.

2. The only language planned for the site to be displayed in is English.

3. Data will be stored in the database on our server.

4. All privacy policies will be communicated to the users upon their registration so that they are aware their privacy shall be protected.

5. The interface of the site should be intuitive and easy to use by regular people; no prior training should be required to use properly.

6. Security of the site will require users to register and log in at launch.

High-level System Architecture

Database:

* MySQL

Server:

* LAMP (Linux, Apache, MySQL, PHP/Perl/Python)

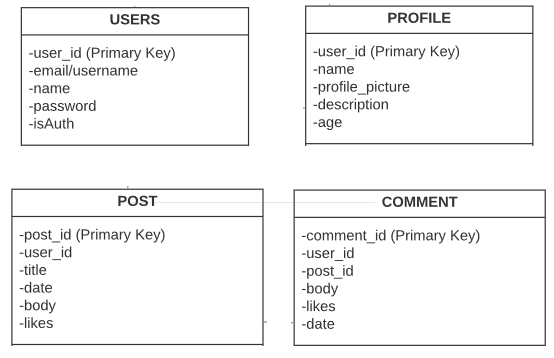
Languages:

* HTML, CSS, PHP

Browsers:

* All major browsers

Database Organization:



Media Storage:

We will be using a file system as the method of storage for the majority of our site’s media files and content. However, some exceptions will need to be stored on an online database such as each user’s uniquely uploaded image they will have associated with their account. Supported media formats for the website will include: JPG, and PNG.

Search/Filter Architecture:

The search architecture for our project will be a fairly standard model that requires an input from the user which is then used to reveal all of the existing results stored in the database with Names or Usernames that contain those inputted words.

API’s:

+getPost();

+likePost(postId);

+createAccount(email,username,password);

+validateLogin();

Key Risks

* Schedule Risks

The difference in each of our team member’s time preference for working on the project could be seen as a potential issue, as we tend to work on each other’s assigned goals at somewhat disparate times. We are in constant communication via a group chat we created on the “Discord” platform, so it is not like we are without methods of easily assigning objectives to each other or collaborating ideas. Still it is something of note if eventually this slack approach to design led to scheduling complications in a more serious task environment.

* Skill Risks

Our team’s varying skillsets in themselves could become drawbacks if we for example, assigned someone to complete a task they weren’t optimal at performing in comparison to the efforts another team member doing the same task could achieve. Therefore, we must become aware of each of our team member’s abilities to properly task ourselves in an efficient manner.

Team Roles

Scrum Master – Anthony Pancione

Product Owner – Nicholas Child

Front-End Lead – Tyler Do

Back-End Lead – Jonah Livingston

