**Database Application Programming**

**Project Status and Design Report**

|  |  |  |
| --- | --- | --- |
| **Topic:** | Milestone Final Blog Application: Revise, improve, and complete the final project | |
| **Date:** | *06/16/2019* | |
| **Revision:** | *1.0* | |
| **Team:** | 1. *Daniel Cutrara* | |
|  | |
|  | |
|  | |
| **Team Status:** | |  |  |  |  | | --- | --- | --- | --- | | **Task** | **Team**  **Member** | **Hours**  **Worked** | **Hours Remaining** | | *Create a user registration page for your blog.* | *Daniel* | *1* | *0* | | *In MySQL, build the necessary tables to store the information* | *Daniel* | *1* | *0* | | *In PHP, write the functions that will enable the capturing of user input and store in the database* | *Daniel* | *1* | *0* | | *Create a login page for your blog.* | *Daniel* | *2* | *0* | | *Create a new blog post* | *Daniel* | *2* | *0* | | *Capture the new post and store in the database* | *Daniel* | *2* | *0* | | *Implement a simple language filter* | *Daniel* | *2* | *0* | | *Create an Azure account and upload all files to your account* | *Daniel* | *2* | *0* | | *Interface for the blog administrator to categorize (tag) and remove posts.* | *Daniel* | *4* | *0* | | *Create the page that allows an authorized user to search for posts using multiple criteria.* | *Daniel* | *4* | *0* | | *Create the mechanism that allows an authorized user to reply (comment) on a post and rate it.* | *Daniel* | *5* | *0* | | *Revise, improve, and complete the final project* | *Daniel* | *8* | *0* | |  |  |  |  | | |
| **GIT URL:** | <https://github.com/Dcutrara1/CST-126> | |
| **Hosting URL:** | <https://dan-blog.herokuapp.com/> | |
| **Peer Review:** | *Y/N* | We acknowledge that our team has reviewed this Report and we agree to the approach we are all taking. |

**Supporting Design Documentation**

**Install Instructions:**

*I have moved my application back to Heroku due to Azure trial ending. This project is developed in PHP and MySQL using Eclipse. Included with the report is the application source code, DDL, ERD, and sitemap.*

**General Technical Approach:**

*Admin Key to create admin user is "blog".*

*An example Admin User to test system is username:* [*cutrarads@gmail.com*](mailto:cutrarads@gmail.com) *password: test3*

*Testing Phase completed utilizing Activity 7 Part 2. Application was able to pass all but 2 tests due to functionality development not allowing the requested function.*

**Key Technical Design Decisions:**

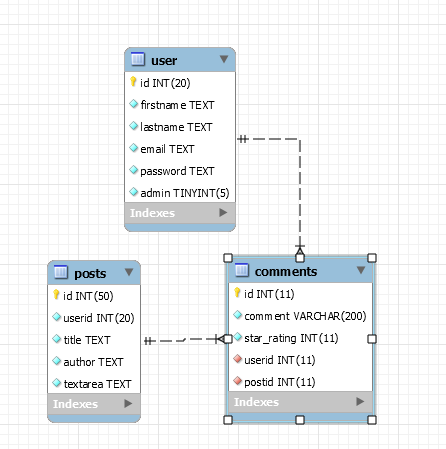
*Any final technical design decisions, such as framework decisions and so forth, should be documented here. This should list the technology/framework, its purpose in the design, and why it was chosen.*

*This project was developed with 2 approaches to the blog site.*

1. *The first is the view of the user, who is required to login to access the site, and has access to read blog posts, add blog posts, search, and add comments to existing posts. I did include any functionality for the user to be able to edit or delete their own posts. This is something that would be considered for future releases of the site.*
2. *The second developed approach is the view of an administrative user, which not only requires registration access, but also have the correct admin key to register as an admin or have an admin update their privileges to admin. Admin’s have all the same abilities as users, but also can edit and delete blog posts. The system is designed that if a post is deleted, all existing comments are automatically deleted as well.*

*Design framework consists of a combination of PHP, HTML, and CSS pages. During the final review of the blog site, I created the index page with the user login on the page. This seemed to be a normal approach for most blog’s reviewed and I was able to incorporate this feature in the final design. I also removed several pages previously built and used since they no longer seemed necessary including addPostView.php, and addPostController.php which were incorporated into other milestone updates, as well as adminManager.php by redirecting initially to the Blog Admin view page. I also removed the need to click “Retry” or other buttons upon error messaging or input errors. I added a timed refresh and returned to the previous page. Although I created a CSS file, I ended up having to code additional CSS to individual pages to obtain the desired results in the look and feel of the site.*

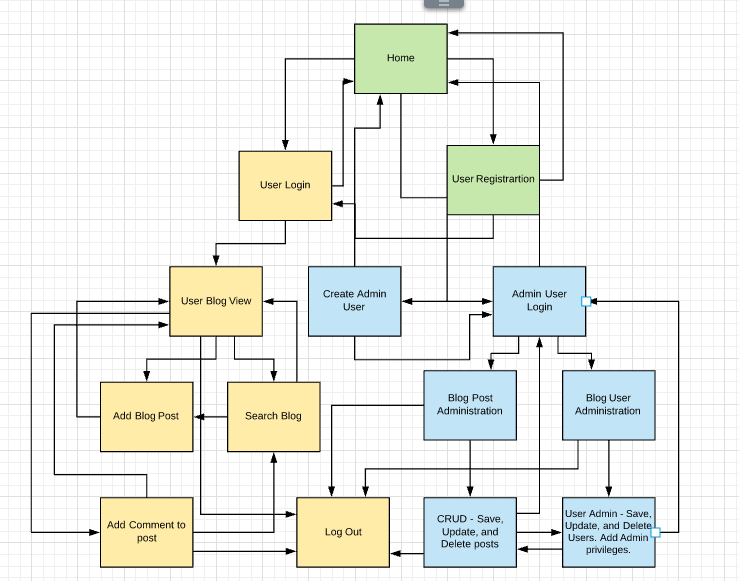
**ER Diagram:**

**

**DDL Scripts:**

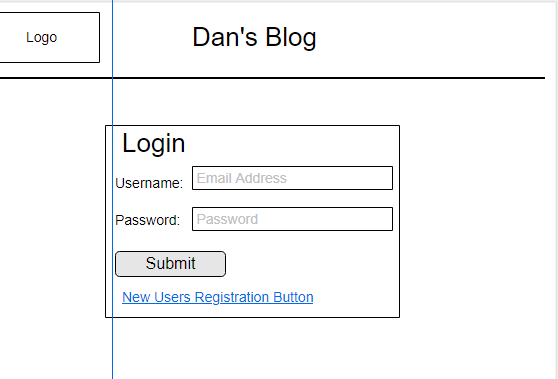
<https://github.com/Dcutrara1/CST-126>

**Sitemap Diagram:**

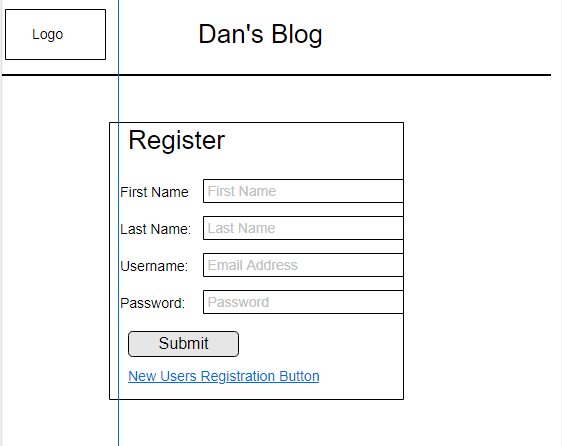
**

**User Interface Diagrams:**

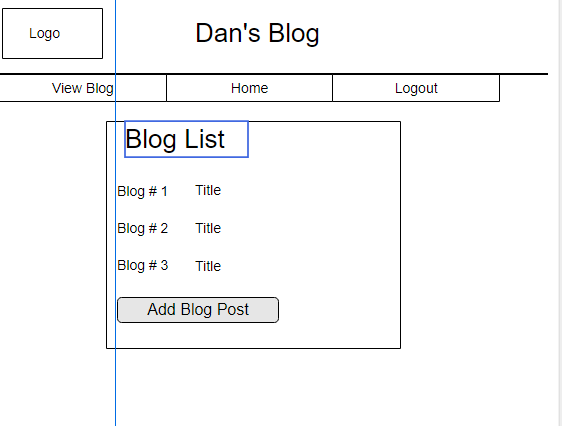
Login Wireframe



Blog User Registration Wireframe



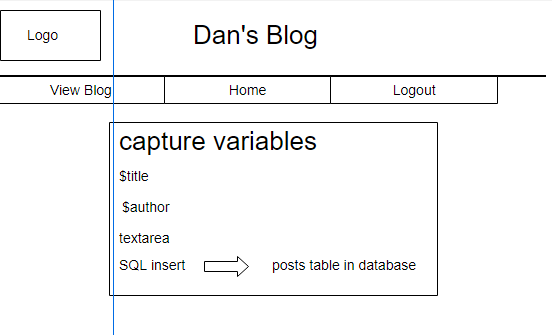
Blog List Wireframe



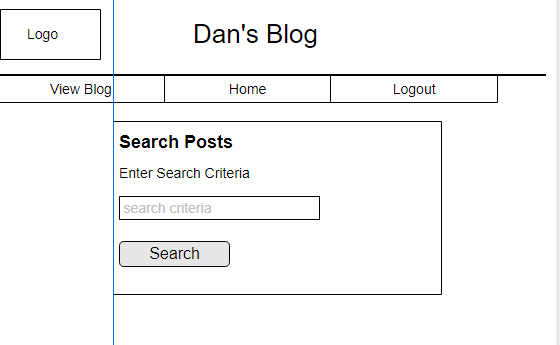
Add Post Wireframe



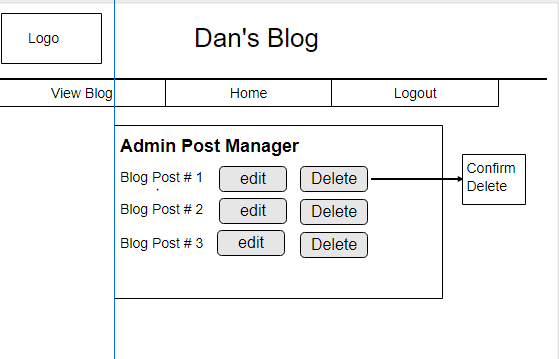
Capture Post to DB Wireframe



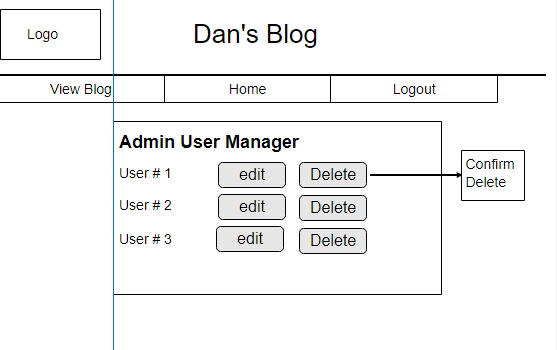
Search Wireframe



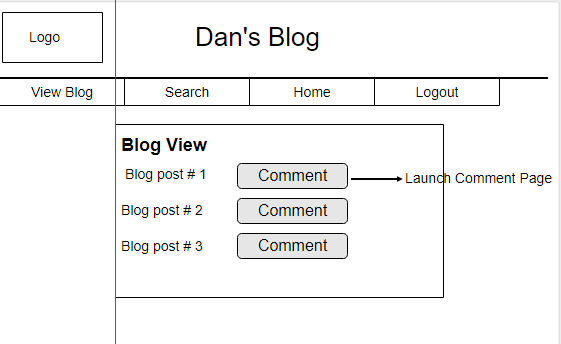
Admin Post Manger Wireframe



Admin User Manger Wireframe



Comment Select Post Wireframe



Add Comment Wireframe

