

quality_assurance: Jun Tao Lei, Grace Mao, Sophie Nichol

P#00

SoftDev1 pd9

2019-10-17

The objective of this project is to design a basic blog that fulfills the following conditions:

- Ability to sign up by providing a unique username and a user generated password
- Ability to login with a username and a password
- Displaying the entries by the logged in user
- Adding new entries authored by the logged in user that feature the following:
 - Entry title
 - Author name
 - Content
 - Timestamp
- Modifying entries authored by the logged in user with the following functionalities:
 - User functionality:
 - Edit the content in the selected entry
 - System functionality:
 - Timestamp updates when the entry is updated
- Ability to view entries generated by other users:
 - Assumptions:
 - User must be logged in to view other users' contents
 - User can not modify other users' contents
 - Procedure:
 - Navigate to the homepage that displays a list of usernames
 - Each username is a link to a page that displays a list of the selected user's entries
 - The list of the selected user's entries can display links to the entries.
- Ability to logout

Since each page in a blog should have a similar design, there should be a base template with two broadly scoped blocks: header and footer. The body block should also be included in the base template, albeit remaining empty for the child(ren) to modify.

The following core forms should be included:

- Login form that features the following:
 - Username input field
 - Password input field
 - Login button to submit the form
 - Signup button to redirect to the signup page
- Signup form that features the following:
 - Username input field
 - Password input field
- Edit form (capable of adding or editing an entry) that features the following:
 - Title text field
 - Content text area
 - Submit button to add or update the form

The database should contain a table that contains a list of users and corresponding entry tables for each user. The corresponding entry tables should be named the same as the user's username or another applicable naming convention.

The users table should contain the following:

- Primary key that auto-increments to be used as id for the user
- Username

The username tables should contain the following:

- Primary key that auto-increments to be used as the entry id for the user's blog
- Title
- Content
- Timestamp

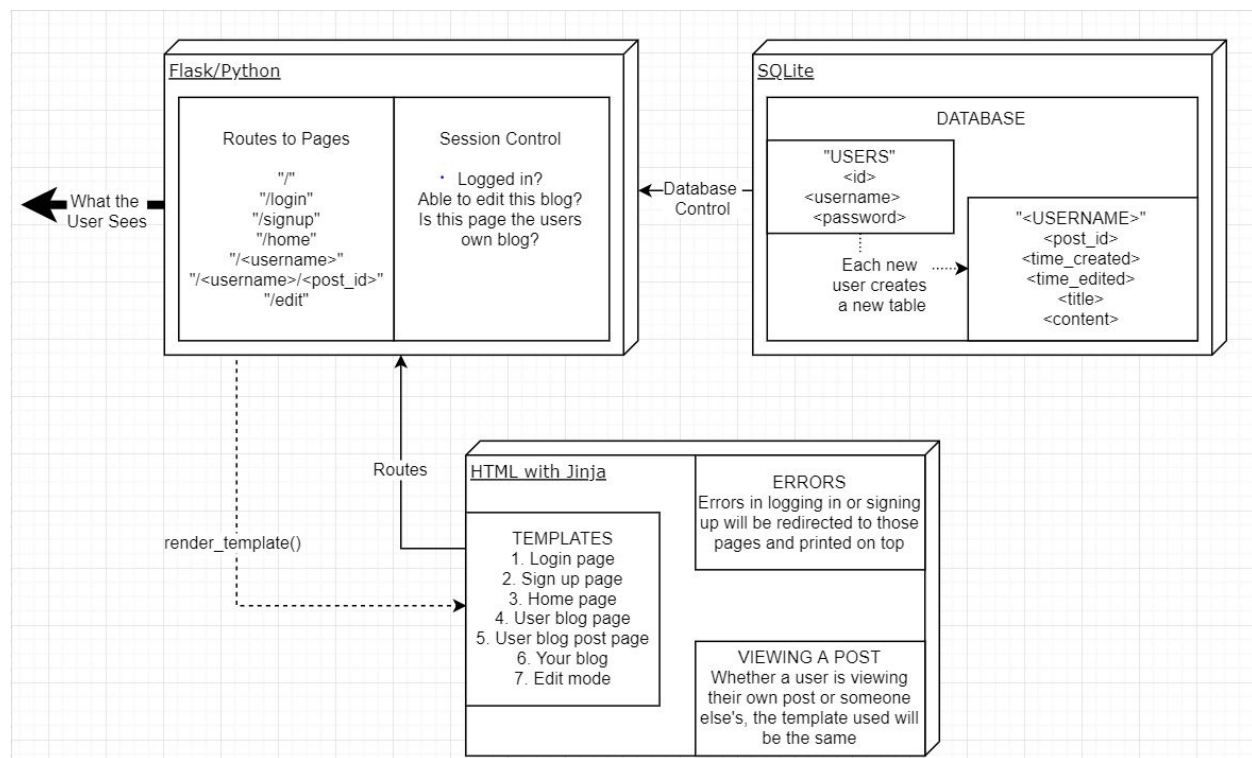
Assuming that a user visits this site and access the root route and that the user does not have a session, the user is prompted to the login page.

If the user does not have an account, the user can navigate to the signup page through a link in the login page. Otherwise, the user completes the required fields and submits the login form. If the operation is successful, the user will be redirected to the logged in homepage. Else, the user is redirected to root route that will show an error message. Likewise, the user will also be redirected to the signup page with an error message if the operation is not successful. Otherwise, the user will be redirected to the logged in homepage.

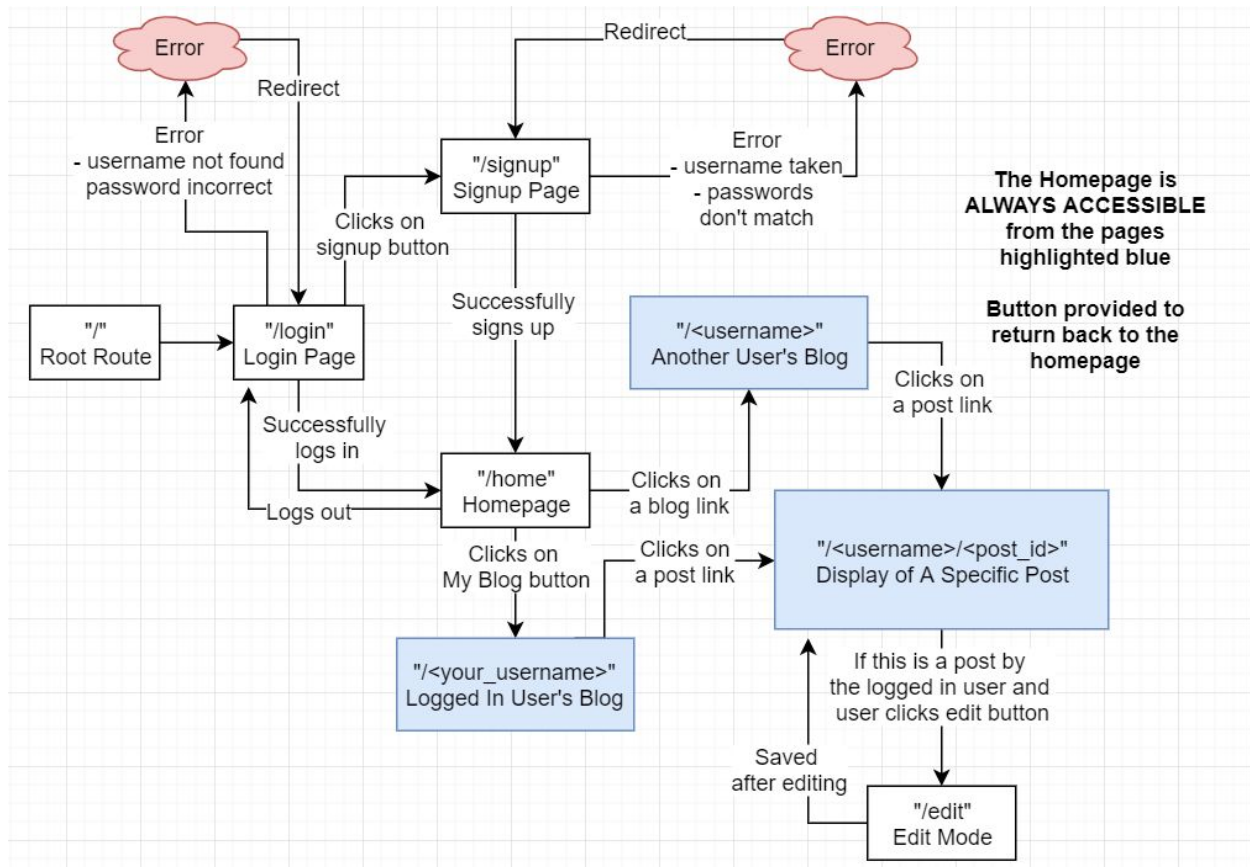
On the homepage, the user is able to navigate to a user's blog page as aforementioned. If the user has clicked on a user link, the user will be redirected to a page with all the entries made by the selected user. If the user clicks on an entry, the user will be redirected to that specific entry.

The "My Blog" button will redirect to the logged in user's own blog page. The logged in user's own blog will display a list of entries made by the user. If the user clicks on an entry, the user will be redirected to that specific entry. Under that page, the user has the ability to edit that entry. Otherwise, there should be an option to add a blog on the logged in user's main blog page.

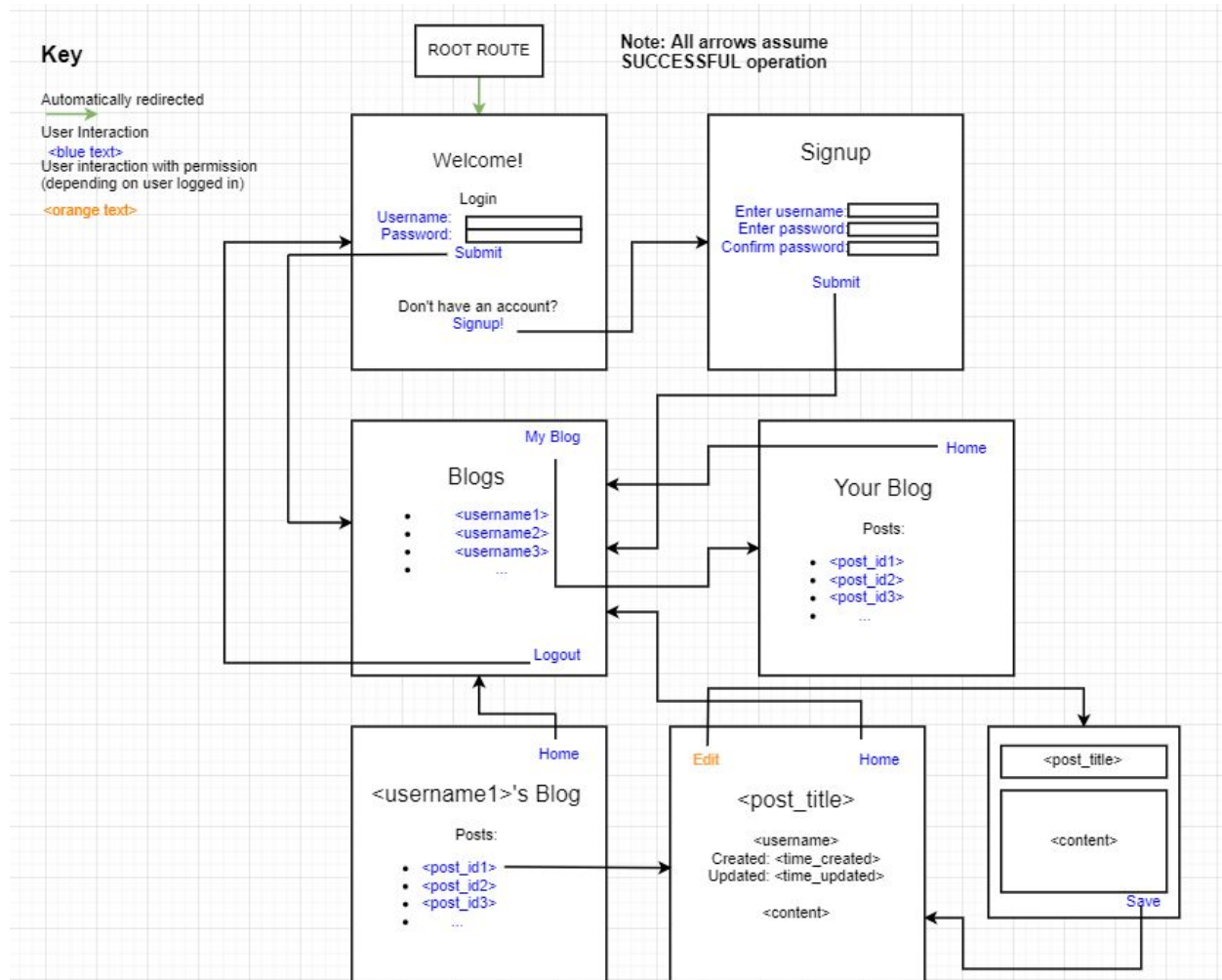
Component Map



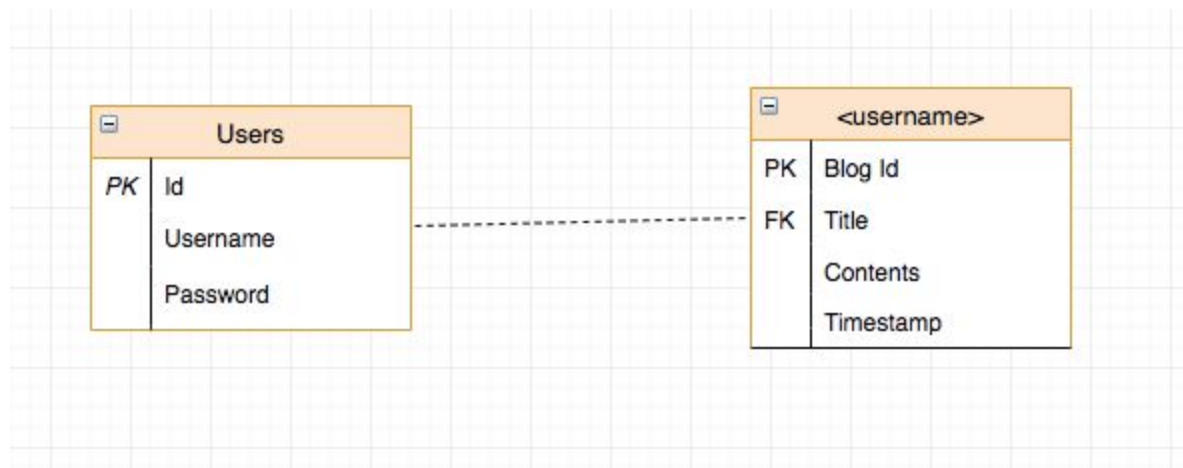
Route Map



Site Map



Database Layout Diagram



Roles:

- Grace Mao:
 - Templates
 - Make a template that will store and format the title, headers, content, and footers for each page of the blog to be displayed
 - Building Forms
 - Make the login and signup forms, as well as facilitate submission once a user is done editing a new post
 - Handle buttons in the templates (Home button on each page, submit buttons etc)
 - Facilitating Entry Addition/Update Using Built Database Operations
 - On a user's My Blog page, make a button "New Post", which will redirect them to a blank edit page (identified by the next available id) where they can put in their title and content
 - Make the "Edit" button on any existing post's page, provided that it is the logged in user's own post, and allow the user to update their table of posts in the database so that, once changes are submitted, the template will display the updated post
- Sophie Nichol:
 - Initialize Database
 - Make the Users table containing the username, id, and password for every user. For each user, make a <username> table containing a row for each post they make and storing the post id, title, contents, and timestamps
 - Connect Flask and SQLite3
 - Establish a database connection with Flask
 - Keep track of executions of SQLite commands and Flask actions
 - Database Operations Module (Insert to Table, Edit a Row, Create a Table)
 - Facilitate any changes that the user will request to the table, such as editing the table once a user updates a blog post, creating a new row when a user makes a new post, or creating a new table when a new user joins.
- Jun Tao Lei:
 - Project Manager
 - Route Functions
 - Renders the template for each page requested with routes pertaining to unique users generated using certain user attributes
 - Implement any other necessary functions required for the route

- Login/Signup Module
 - Initialize a new session when the user logs in or signs up and remove it once they log out
 - Authenticates the user in a sensible manner