

Project Review

Current State

In its current state, the project implements the following:

- User Authentication
 - Users are able to register using a username, email address, and password
 - User registration data is stored for future use
 - Users are able to log in using the username and password they provided in the registration process
- Topics and Comments
 - Users are able to create topics
 - Users are able to edit topics they have created
 - Users are checked to ensure they have permission to edit desired topics (ie. only the topic author can edit the post)
 - Users are able to comment on topics they or other people have created
 - Users are able to edit comments they have created
 - Users are checked to ensure they have permission to edit desired topics (ie. only the topic author can edit the post)
 - Users able to view topics
 - Topics are shown on the home page for authenticated users
- Subscriptions and Notifications
 - Users are able to subscribe to topics
 - Users are able to unsubscribe from topics
 - User are automatically subscribed to topics they create
 - Users are notified of comments on topics they are subscribed to
 - Notifications are removed when a user clicks on them to be redirected to the new post they are notified of
- User Profiles
 - User have their own profiles
 - User's own profiles (My Profile) show their subscriptions, notifications, groups, topics, and comments
 - Users are able to view other users' profiles
 - Other users' profiles show their posts (comments and topics)
 - Users are able to search as user by username to get to their profile
 - Users are able to click on usernames of post authors to get to the user's profile
- Groups
 - Users are able to create and name groups
 - Users are able to add other users to groups
 - Users are able to add topics to groups, making them only available to groups members
 - Users are able to leave groups, removing themselves as members

Design and Implementation Concerns

In its current state, the project has the following concerns:

- DatabaseManager
 - Code was written by multiple members creating non-standardized code throughout the file. This would have to be fixed for easier editing and reading of the code for the following programmers.
 - Utilization of a class based structure for organization issues.
- Test Suites
 - The creation of more thorough testing processes as the amount of testes that would be needed to test every process in the program would be extremely high. The number of tests needed is unrealistic to create in the time frame of the project
- Templates
 - The templates have lots of unneeded HTML code as they were rewrote to implement bootstrap after the functionality of the site was created.
- Administrator Function
 - The lack of an administrator profile that would have access to editing/deletion of threads and comments.
 - Lack of a deletion possibility for threads and comments.

Next Steps

Some prospective and refactoring designs ideas for the project are as follows:

- Extending the design and functionality of the user profile page and allowing the user to have more control over what is displayed on his/her page would allow for a more unique experience. For example, controlling what personal information other users see or the possibility to use a profile picture or avatar are useful features that would be added in a future build.
- Creating a system that records specific statistical data about user postings. Adding a comment upvoting system to allow users to inform a commenter when their comment was helpful, and also as incentive for commenters to leave amiable and helpful comments. These votes would be used with other collected user statistics (i.e number of topics/comments created, how active a user is) to determine a user scoring system that would be displayed on each users profile. This system would not only create some friendly competition, but also allow users to determine how useful a comment is.
- Developing an admin user type to help control posted content on the website. Admins would perform tasks that require human judgement, like reviewing topics and comments for appropriate content, removing unwanted content, responding to user troubleshooting concerns, and handling trouble with user accounts could all be helpful features only possible by creating admins. Development of a simple terms of service would help users and admins determine what type of actions are acceptable while using the application.

- Implementing bootstrap into the UI design later in the project resulted in the developers reconstructing html templates, causing unnecessary loss of time. Considering bootstrap earlier in the design process would have avoided this problem and would of helped cement the final look and feel of the application.
- As the application grows it would be useful to help break the larger modules into smaller, more precise modules to help organization and make debugging easier. For example, breaking routes.py into separate modules that all handle unrelated functionality. Instead of having a view for login, registration, creating topics, etc in one large views file, this information could be separated into different python files.
- Completely decouple DatabaseManager.py from any Flask library. Right now DatabaseManager is using Flasks built in alert system Flash to display messages to the users. In the future, if we decided to switch from flask to another library we would have to rewrite some aspects of DatabaseManager. Removing this dependency would help make our program more modular.