

Team Pina Colada (Max Millar, Tim Marder, Bo Lu, Claire Liu)

SoftDev pd6

P00 -- Da Art of Storytelling

T 2018-10-16

Scenario Two: Your team has been contracted to create a **web log hosting site**.

- Users will **have** to register to use the site.
- Logged-in users will be able to:
  - Create a new blog
  - Update their blog by adding a new entry
  - View and edit their own past entries
  - View the blogs of other users

#### Component Map:

Authorization:

- Login system with a table that contains user/password columns for storage of accounts
- User can register their account on the home page
- User will be prompted to login when they want to enter the blog site

Searching and Viewing Blogs:

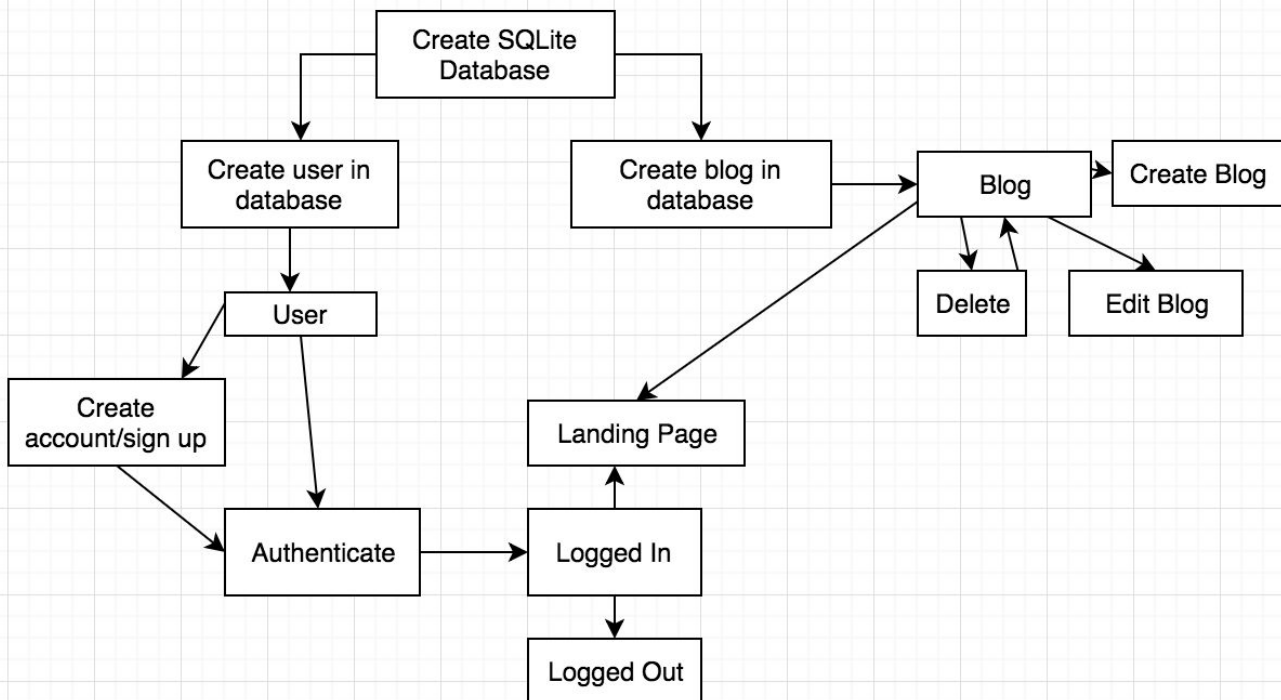
- There will be a text input for searching blogs associated with the inputted user
- The table with all the blogs will have user as one of the columns to facilitate easier search. Since the user is in a column for both tables, it helps us with connecting the two of them with the use of a simple sqlite command.
- Displays all posts written by the user inputted in the search bar.
- The logged in user can view his/her own posts as well - will have a button on home page leading to this
- Next to the title of the blog will be the category that the user selected during creation

Posting and editing blogs:

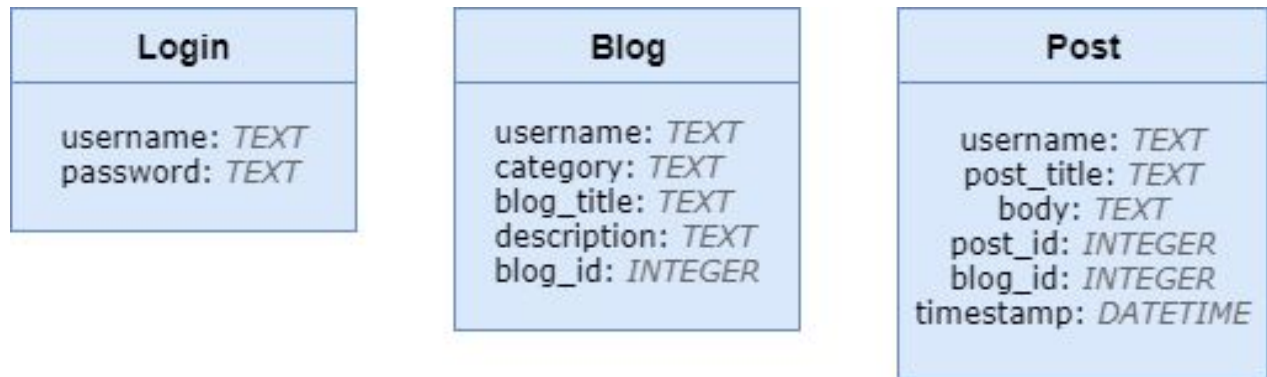
- Every edit that is saved by the user will replace the old blog in the table containing all the blogs and its associated users
- If a blog post doesn't exist already, a new row will be created in the table mentioned above. When posting a blog, there will be several things for the user to fill out, such as the title of the blog, a category, and a description of the blog.

- Blog posts are in a new table, and a new row is added to that each time a post is made. Those posts will have a post ID that we will use to identify who's blog the post belongs to

Component Visualisation:



### Database Schema:



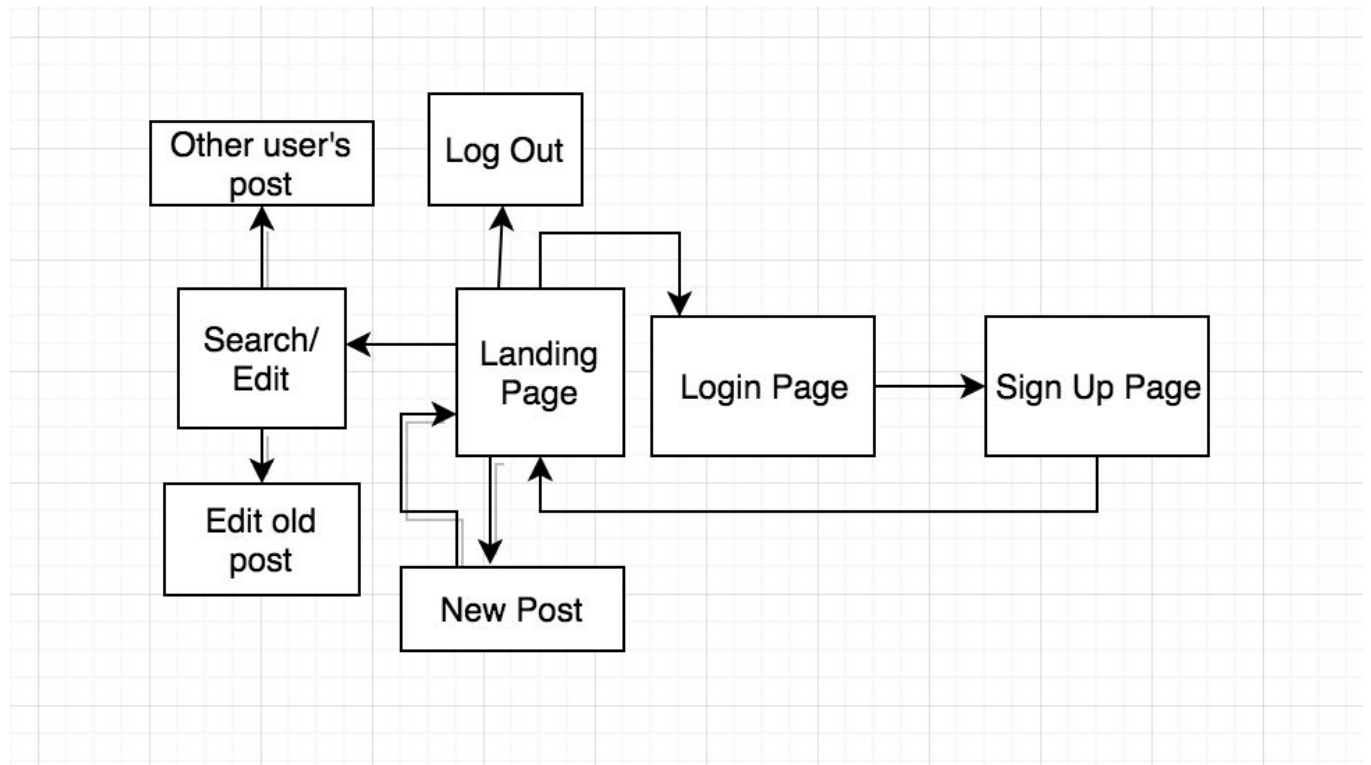
**Login Table:** Used for storing the credentials of users of the site. Stores username and password both as TEXT types and table is accessed and checked for data when user attempts to login. If the text entered by the user matches the username and password in a row of the table, the user is granted access to the site.

**Blog Table:** Used for storing information for each blog. The username column is used again to keep track of what blogs belong to specific users. Category(TEXT) is used to place the blog into the section that the user specifies, but mostly serves as a helpful distinguishing element when viewing blogs. Blog\_title(TEXT) is used to hold the title of the blog and description(TEXT) holds a brief explanation about what the blog is about. Blog\_id is used to keep track of the order of the blogs posted and make searching for them easier.

**Post Table:** Username(TEXT) is used once again to keep track of which posts belong to specific users. Post\_title(TEXT) holds the title of the post and body(TEXT) is the main part of the post where the user types the contents of the post. Post\_id(INTEGER) is used to keep track of the order of posts and make searching for them easier. Blog\_id(INTEGER) is used to correctly tie the posts to their respective blogs. The timestamp(DATETIME) column is used to keep track of when the user made the edit.

Site Map:  
(Starts at the landing page)

(Starts at the landing page)



### Breakdown of Tasks:

1. Basics: Create a repo, add a flask starter kit, add a landing page (Claire)
2. Database for users, account creation (Tim)
3. Account authentication and verification (Bo)
4. Database for blogs -- creating new ones (Claire)
5. Search/view/edit existing stories that the user has contributed to (Max)
6. HTML templating for blogs (Tim)
7. Project Manager(Bo)