Team Bord | Austin Ngan (Gerald), Mark Zhu (Bob the 3rd Jr.), Thomas Yu (Perry), Roshani Shrestha (Pete)
SoftDev
P00: Move Slowly and Fix Things - Design for P00
2021-10-27
Time spent: 2 hours

Project Components:

Python:

- Will create the database and tables as well as update and obtain data from the database.
- Will update the database according to the interactions of the user with blog posts (adding a new one or editing an old one)

Sqlite3

- Used to create and interact with the database.
- Will have be used to store information for account logins to be used by Flask in order to create a session
- Will also store information about each blog post so Flask can display it to the user.
- Will be updated every time a user creates or changes a blog post

Flask:

- Uses Jinja2 and HTML templates to create the web pages that the user will see and interact with
    - Data is pulled from the database in order to populate each webpage
- Will get username and passwords from the front end when a user logs in or registers
- Will create sessions for each user when they are logged in

Jinja2 / Templates:

- Used to create a layout the web pages the user will interact with

Login/Main Page:

- If there is a session, the user will directly be sent to the main personal blog page.
- There will be inputs for username and password.
- There will be a register button if the user doesn't have an existing username and password.
- Submitting correct login credentials will direct the user to the user's main personal blog page.

Register Page:

- Will request for username and password
  - If the user submits a username already in use, the register page will return an error and notify the user what is wrong
- If the registration is successful, the username and password will be added to a database table
  - After successful registration, the user will be redirected back to the login page.

User's Personal Blog Page:

- This is the page that will be rendered after the user submits correct login credentials.
- The existing blogs will be displayed. The user will be able to edit their past entries or create a new entry.
- There will be a button so that the user can create a new blog. This will send the user to a different page so that the blog can be created.
- There will be a list of other users and links to their blogs.

Blog Creation Page:

- A form will be created so that the user can type in the information for their blog.
- Once the user submits, the information will be saved in a database and they will be sent back to their personal blog page.

Other Users' Blog Pages:

- This will show up after the user clicks on another user's blog page link.
- These pages will all have the same overall template.
- The user can only view these pages.

Database Organization:

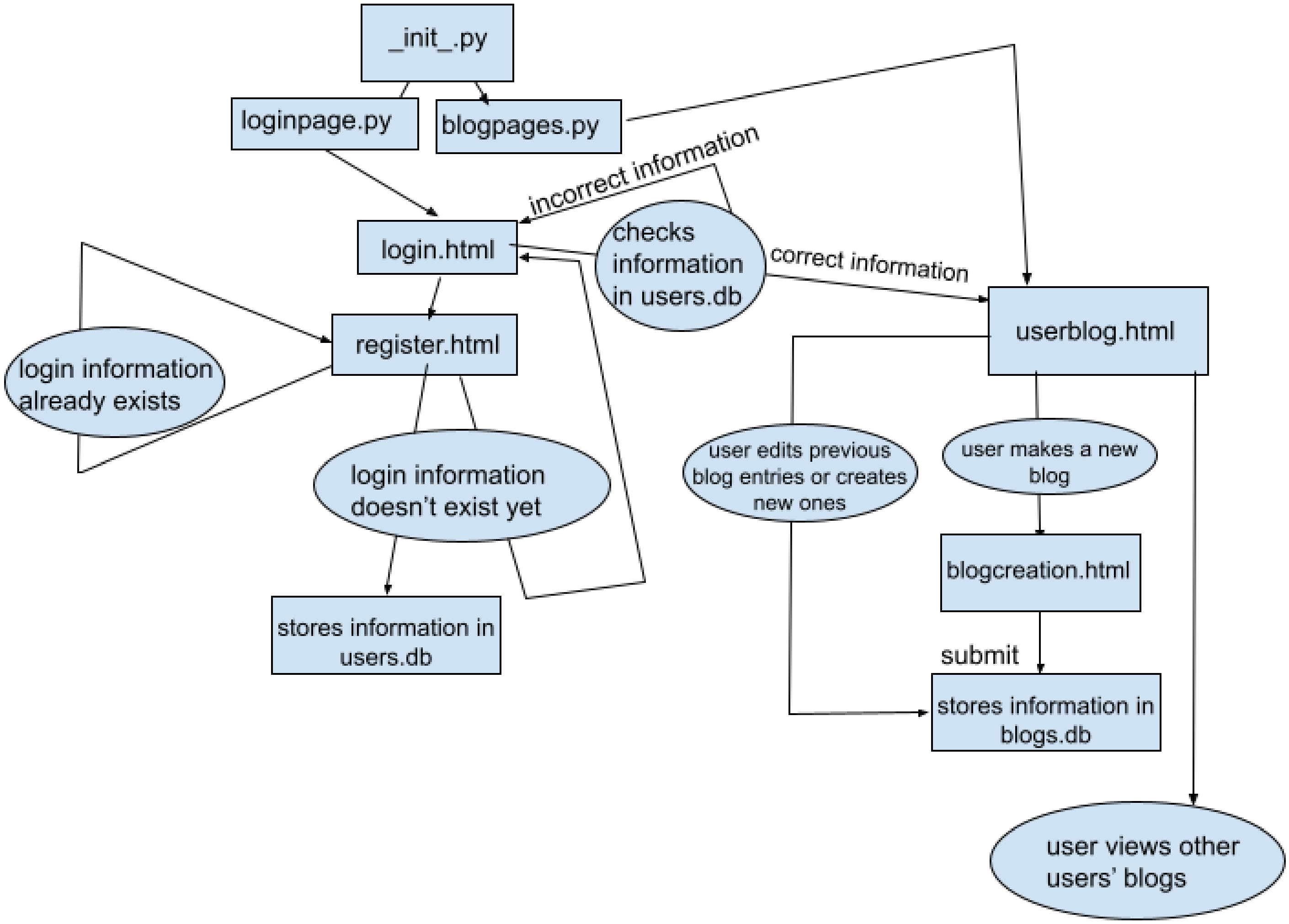
- Tables will be used to organize the usernames and passwords of different users. This will be used to check if the login credentials inputted by the user are correct and if the registration information inputted by the user already exists.
  - Will have to check usernames to avoid duplicates

Username (TEXT)	Password (TEXT)

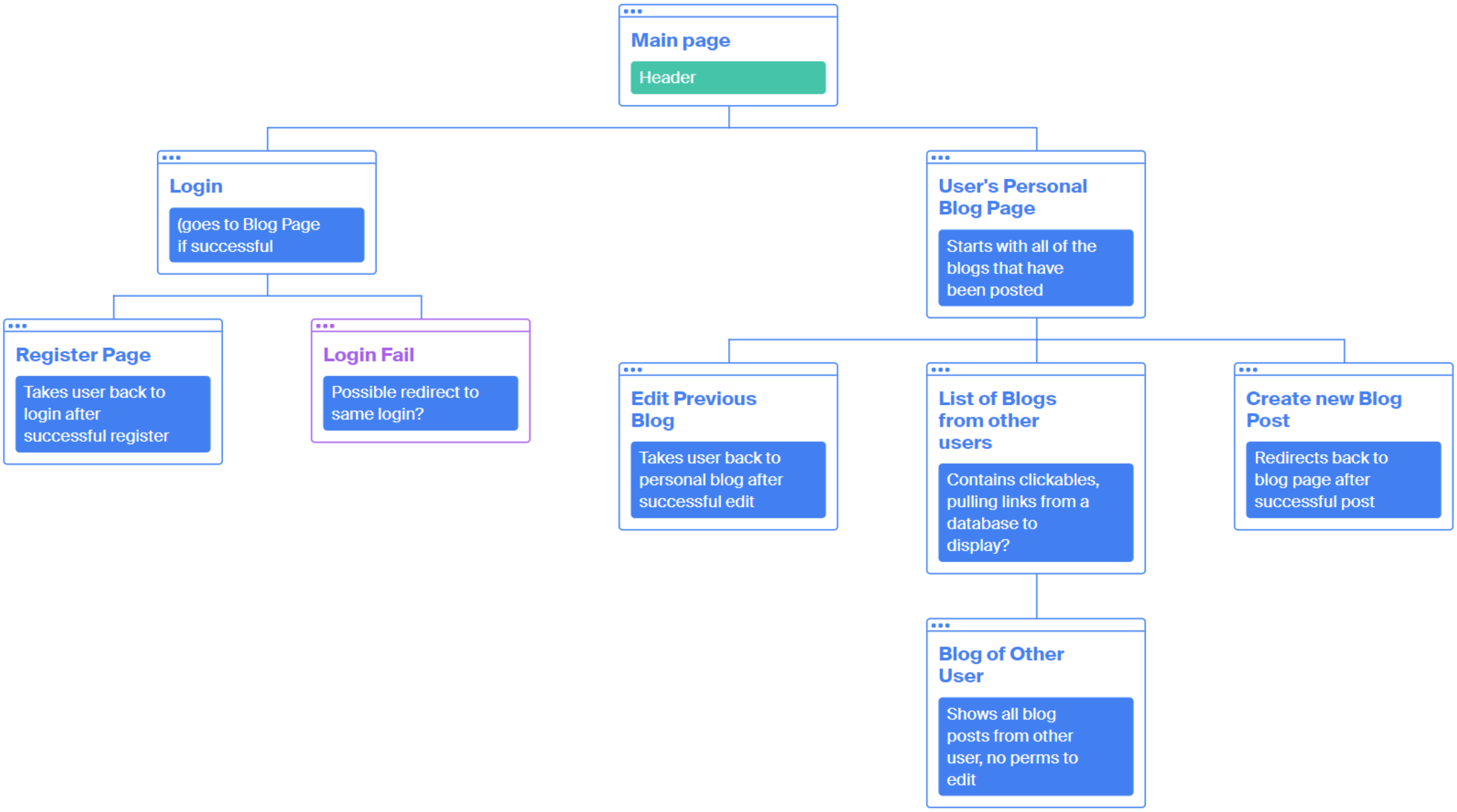
- Tables will also be used to organize information for different blog posts.

Username Associated With Blog (TEXT)	Entry Number of the Blog (INTEGER)	Blog Title (TEXT)	Text for Blog (TEXT)

Component Map:


init.py
loginpage.py
blogpages.py
incorrect information
login.html
checks information in users.db
correct information
userblog.html
register.html
login information already exists
login information doesn't exist yet
stores information in users.db
user edits previous blog entries or creates new ones
user makes a new blog
blogcreation.html
submit
stores information in blogs.db
user views other users' blogs

Site Map: (https://octopus.do/)


Main page
Header
Login
(goes to Blog Page if successful
User's Personal Blog Page
Starts with all of the blogs that have been posted
Register Page
Takes user back to login after successful register
Login Fail
Possible redirect to same login?
Edit Previous Blog
Takes user back to personal blog after successful edit
List of Blogs from other users
Contains clickables, pulling links from a database to display?
Create new Blog Post
Redirects back to blog page after successful post
Blog of Other User
Shows all blog posts from other user, no perms to edit

Tasks:

Austin Ngan: Creation/editing blogs (Python and Database)

Mark Zhu: Personal blog page and blog pages of others (HTML templates and Flask to connect the database to the webpage)

Roshani Shrestha: Login/Register Page (HTML templates and Flask sessions)

Thomas Yu (Project Manager): Assisting with HTML templates and database work (user accounts and blogs)