Gastric Bypass Train - Sam Cowan, Anna Fang, Sadi Nirloy

P00: Move Slowly and Fix Things

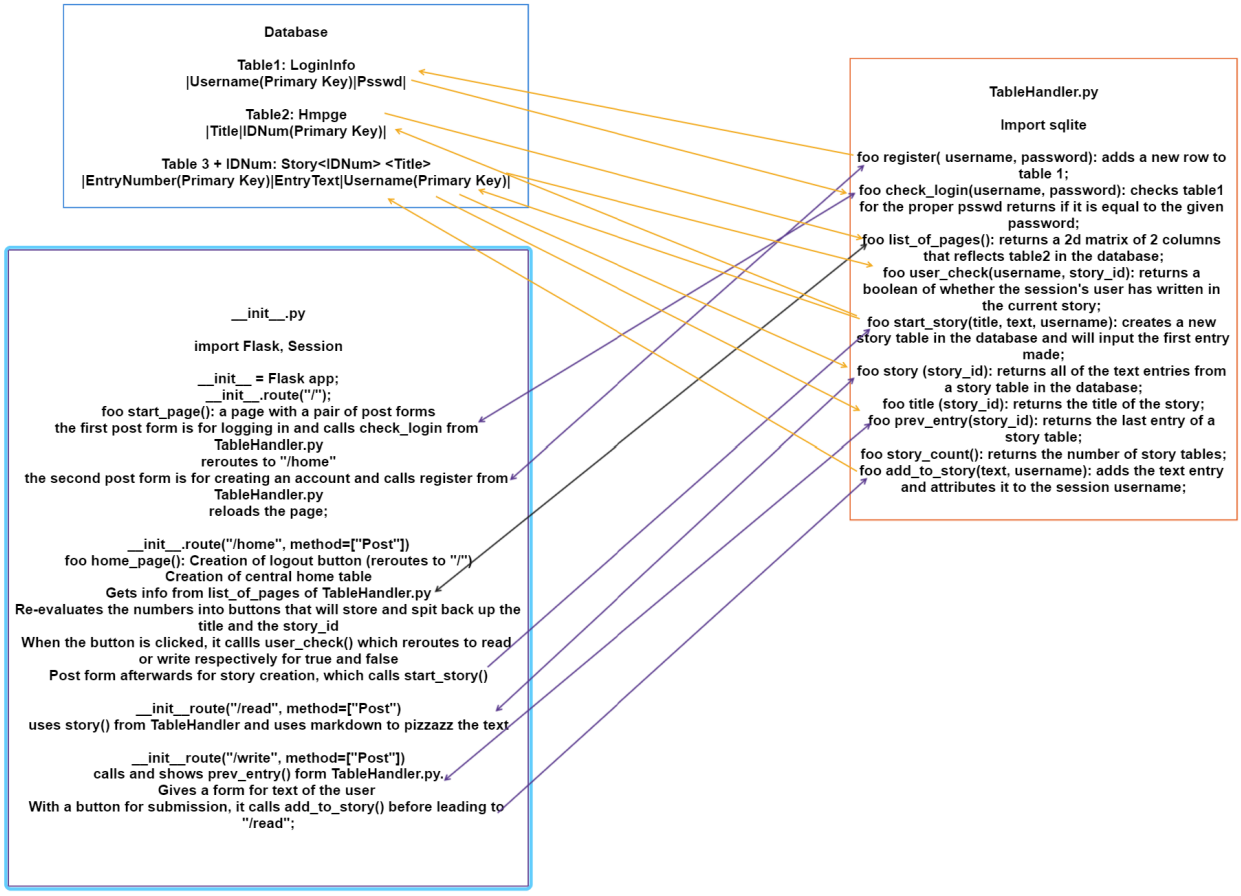
SoftDev

11-2-2022

Time Spent: 2 hrs

Target ship date: 2022-11-8

Design Plan



Login page:

- HTML form(s)
    - The register form stores and adds new user login information (username and password as text) into a table

    - The login form references existing entries in the user-login table and allows the user to continue their *__session__*

Homepage:

- HTML Table with all of the different pages
    - Each entry in the table consists of a row ID and story title
- Story creation form
    - Each entry consists of a pair of textboxes for the title and inputted text
    - Adds the story to the main homepage table (storing a unique row ID and the inputted title)

Loaded Pages:

- A way to return to the home page (button)
- Will either be a editing page or a viewing page based on the existence (or lack of existence) of user edits (reference Database Organization and Site map for more details)
    - The editing page will contain the story title, the most recent entry, and a textbox for the user to enter new text
    - The viewing page will display the full story (text)
        - Pulls from individual story tables

[Files]

Database:

- Table 1: stores user login information
    - Username (text) and password (text)
- Individual tables for each story
    - Stores all information
    - Reads username from the session to check if the user has already added to *this* specific table
        - Entry Number
        - Entry text
        - Username of the person who entered the text

Python:

- Flask app with four routes:
    - / (login)
    - /home (homepage)
    - /write (additions to story)
    - /read (displays full story)
- Adds to database based on corresponding post requests from login, write
- Stores cookie to check whether one should be redirected to /read or /write when they navigate to a specific story page on homepage
- Uses render_templates to display pages properly

HTML:

- Templates for the display pages
    - Like a form

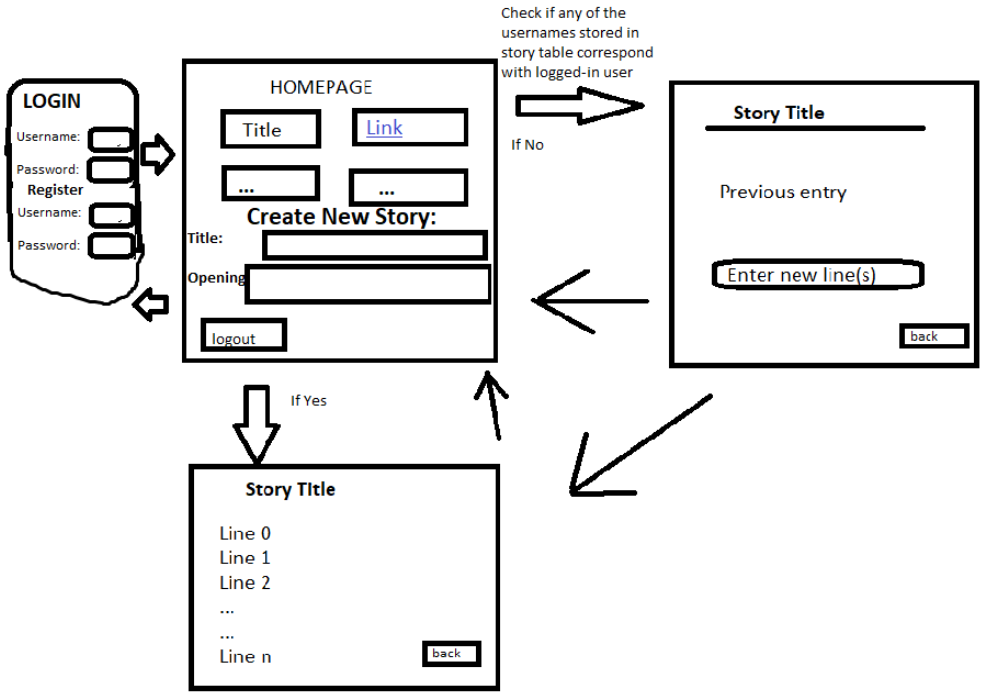
Sqlite database (which is dynamically added to by flask app):

- Contains all tables with user inputted information

Database Organization

One table for homepage which stores a row ID (integer), story title (text)

- A table to store user's login information: username (text) and password (text)
- Each story is housed in a table (of its own) which has three columns - row ID (integer), story text (text), and the user who wrote the story (text). The story can be seen by looking at the text row one by one



Site map:

MVP:

- Login/register capability using sessions (with no practical use)
- Homepage which links to a single story that can be added to by user without limit

Expectation:

- Login/register capability using sessions
- Homepage with a table linking to every story

- Boxes on homepage to seed a new story
- When the user goes to a story page, their username is compared with the username column of that story's table. If there is a match, the user has already edited the story and is taken to a page showing the full table. If not, the user has not edited and is taken to an edit page where they can add a line to the story

Stretch Goal:

- N/A

Task Assignments:

MVP/Expectation:

- Flask app - **Sadi**
- HTML templates and CSS - **Anna**
- Sqlite-managing Python files - **Sam**

Stretch:

- [Functional] Logins - **Sadi**
- Help Page with instructions (?) - **TBD**