Scenario 1:

SJS: Shinji, Jeffery, Sebastian
SoftDev
P00 -- Scenario 1
2022-11-14
Target ship date: {2022-11-15}

Collaborative Story Website Design Doc

Program components:

-SQLite3 database to store information such as: usernames, passwords
-Another SQLite3 database to store story titles, authors, genres, the full story, the latest update, and amount of times updated
-Python file to run the flask server and manage collecting information from the site making changes to it (GET/POST).
-HTML template files to store text and non-css structure. Ran by the Python file using Flask and also connects to them using app routes and GET/POST methods.

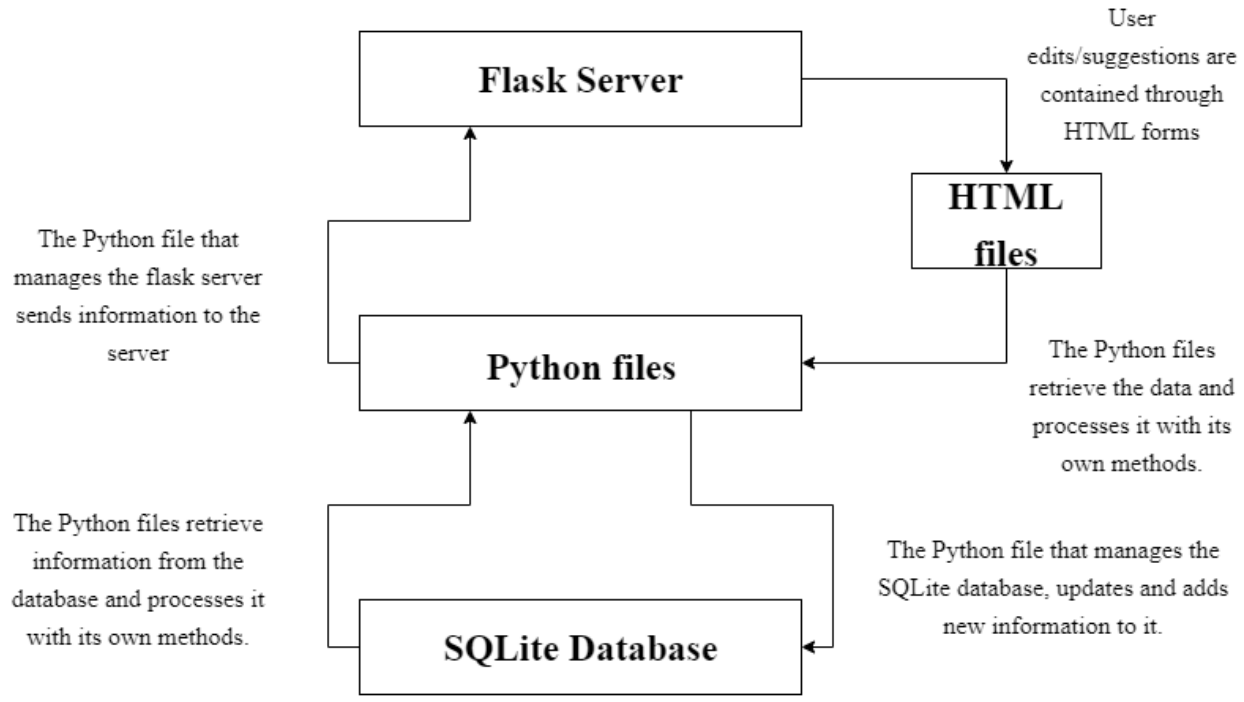
Database Structure:

-Account information table. includes username and password columns, both are UNIQUE.

-Story list table: includes story-name, tag, contributors, full-story, total-updates, latest-update, latest-contributor and latest-update-time columns. Story-name column is UNIQUE.

Component Map:

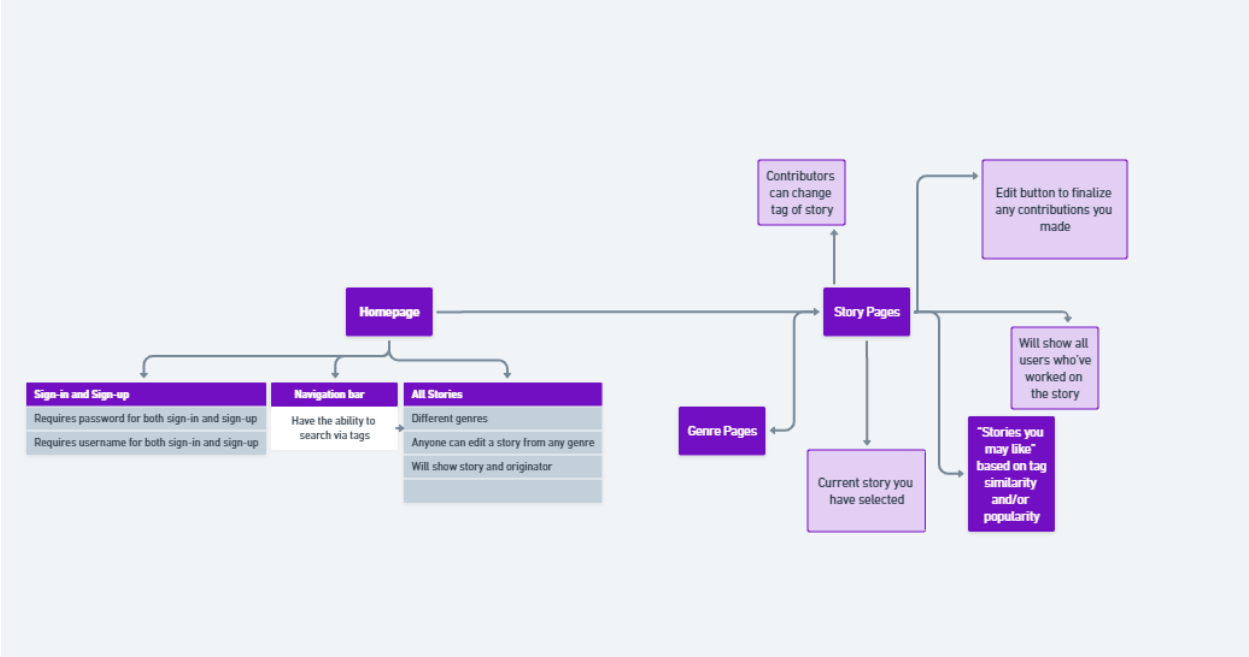
Summary: Python is the middle man. It takes information from the database and processes and/or sends it to the flask server and also retrieves information from the flask server and processes and/or sends it to the database.



Site Map:

- Home page
    - Users will be led here after signing in
    - Text to welcome users and give brief information about the site
    - Logout button to log the user out and redirect them to the login page
    - List of different story genres
        - Stories will be given a genre tag when made to categorize them for users to find specific types of stories.
        - Users can click on a genre to go to the page for that genre
    - Create A New Story hyperlink to lead user to create page
    - Shows a list of all stories on the site and their contributors
- Login page
    - Users will be led here if they logged out or if it is their first time using the app
    - Field for username and password and a button for submit
    - Register button for users who don't have an account yet that goes to the homepage
    - When the user inputs login information, it is compared to the account information database to see if they gave the correct information.
    - Reloads the page with an error message saying either username or password is incorrect if that is the case
    - Directs user to home page if information is correct
- Register page

    - Field for username and password and a button for submit
    - Back button to return to login page
    - Upon successfully creating an account, redirects to the login page.
- Create page
    - Textbox to type in the desired title
    - Selection of genres:
        - Non-fiction
        - Horror
        - Romance
        - Fantasy
        - Educational
    - Textbox to write the story
    - Submit button to publish the story to the site
        - Submission will lead back to home page
    - back hyperlink to return to home page
- Genre pages
    - A picture for that genre
    - A list of storys on the site tagged with that genre; displays title, contributors, and story contents
        - Clicking on the title of a story redirects to its story page
    - Go back to home hyperlink to return to home page
- Story pages
    - Displays the title of the story, version of the story, and its contents
    - Add to this story hyperlink to contribute to the story
        - the Python file will check if their username is in the contributor column of the story collection table and will allow them to make edits if their username is not in that column.
        - Hyperlink will not display if user is unable to edit the story
        - Once a user submits their addition to the story, they can see the full story.
    - Go back to home hyperlink to return to home page
- Add page
    - Displays the current contents of the story
    - Displays a textbox to add to the story
    - Submit button to append contributions
    - Go back to home hyperlink to return to home page



Tasks:

1) Have a repository structure in place as requested. - Jeffery
2) Create both SQLite3 databases login, register, and logout functionality - Shinji
3) Login, register, and logout functionality - Shinji
4) Create home and genre pages - Sebastian
5) Make create page functional - Shinji
6) Add relevant text and quality of life functions to pages - Jeffery & Sebastian
7) Use for loops to display stories on home and genre pages - Jeffery
8) Style pages by adjusting font sizes and adding images - Sebastian
9) Create story page template and make editing function - Shinji
10) Debug and final tweaks - Shinji & Sebastian & Jeffery
11) Finalize requirements.txt, README.md, and design doc text - Jeffery
12) Finalize design doc visuals - Sebastian