Scenario 1:

SJS: Shinji, Jeffery, Sebastian
SoftDev
P00 -- Scenario 1
2022-10-28
Target ship date: {2022-11-15}

Collaborative Story Website Design Doc

Program components:

-SQLite3 database to store information such as: user id's, passwords, current story, new additions, etc and can be accessed for various uses.
-Python file to run the flask server and manage collecting information from the site making changes to it (GET/POST).
-Another Python file for SQLite3 commands to manage the database and retrieve information from it.
-HTML file to store text and non-css structure. Ran by the Python file using Flask and also connects to them using app routes.
-CSS file to store design and/or additions that cannot be made on the HTML file. Will coincide with the HTML file.

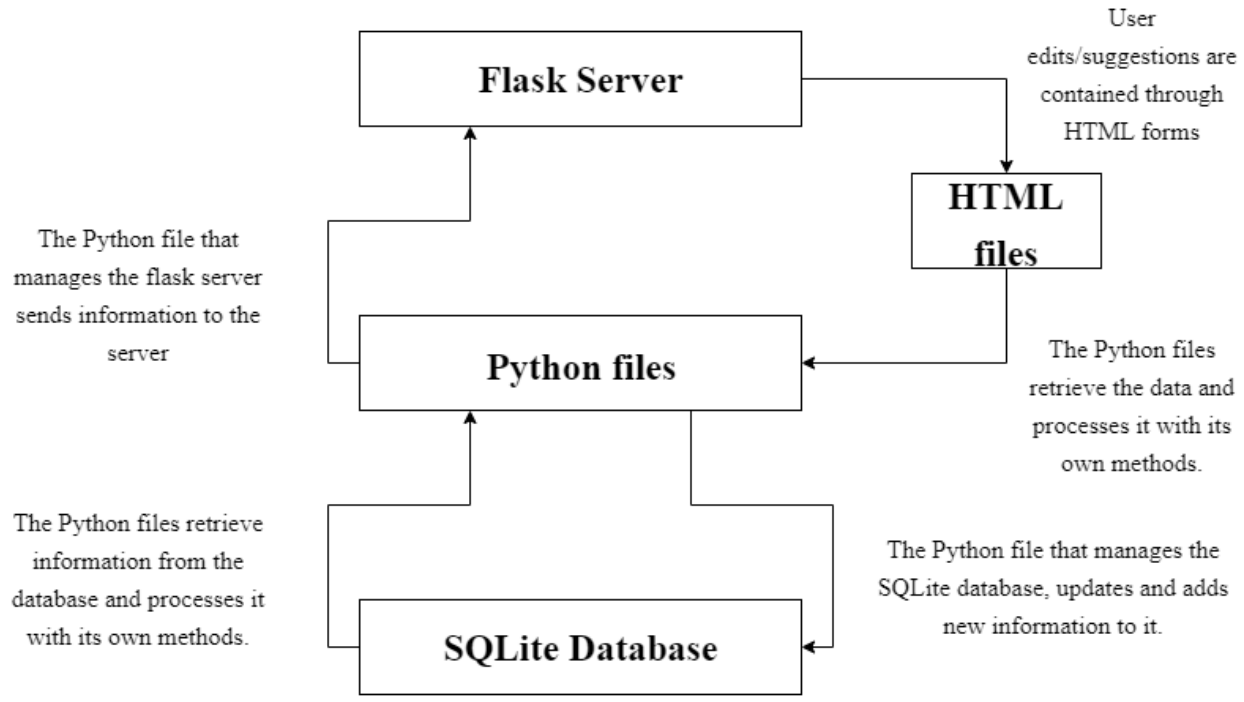
Database Structure:

-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 and contributors columns are UNIQUE.

Component Map:

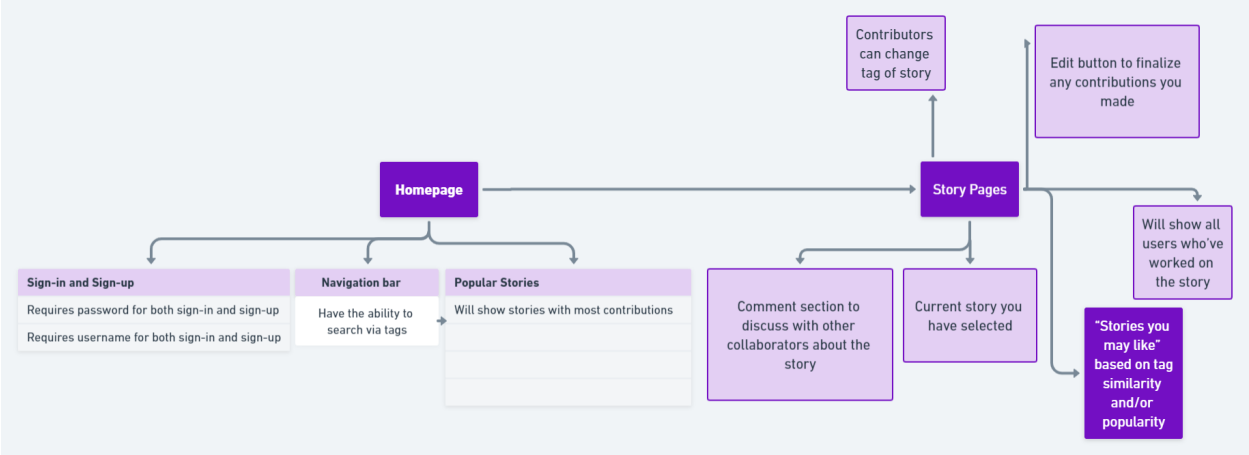
Summary: Python is the middle man. It takes information from the database and processes and/or sends it to the flask server and also retrieves information from the flask server and processes and/or sends it to the database.



Site Map:

- Homepage
    - Shows popular stories on the site and their tags, as well as a nice picture showing the most recent addition to the story.
    - Sign in button
    - Will have navigation bar for different story genres
        - Stories will be given tags when made to categorize them for users to find specific types of stories.
    - Search bar to find stories to read or contribute to
- Sign in page
    - Create an account button for users who don't have an account yet.
    - When the user inputs login information, it is compared to the account information database to see if they gave the correct information.
- Create account page
    - User inputs a username and password in a form/textbox
        - Username and password both have to be unique. The password created by the user will have to be 8 characters or greater, and include at least one digit. The submitted password and username will be sent from the flask server back to the Python file and reviewed there and compared to the account information table in the database to check if it meets the requirements.

        - If the username or password do not fit the requirements, the user will be prompted again.
    - Upon successfully creating an account, redirects to the sign in page.
- Story pages
    - Displays the current story, will try to operate in real time, allowing changes made to appear on another users screen/or will alert user that there have been changes made and will recommend a refresh
    - Edit button to contribute to the story
        - When a user tries to edit a 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.
        - Once a user submits their addition to the story, they can see the full story.
    - Indicates the original poster and all users who have worked on the story.
    - "Stories You May Like" section on the bottom that includes a selection of similar stories(by tag).
    - OPTIONAL: Comments section on the bottom for users to talk about the story and connect



Tasks:

1) Create Python file with Flask app to run project HTML file. - Jeffery
2) Obtain .csv data (user inputted data) and put the new info into databases - Sebastian
3) Create Python file to utilize SQLite3 to convert .csv files to .db files - Shinji

4) Structure site/style through CSS and write info/ necessary structuring in HTML file- Sebastian
5) Integrate HTML and Flask app for site functionality using app routes - Jeffery
6) Run site using Flask - Shinji