Scenario 1:

SJS: Shinji, Jeffery, Sebastian
SoftDev
P00 -- Scenario 1
2022-10-28

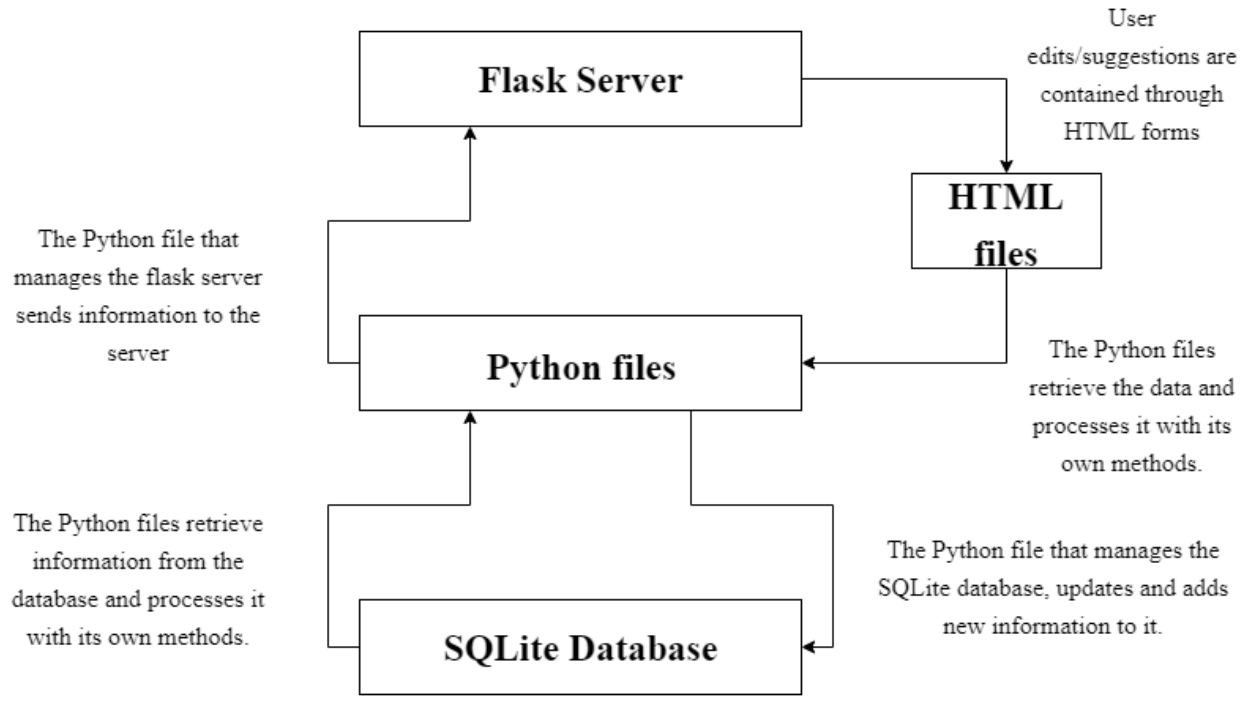
Collaborative Story Website Design Doc

Program components:

-SQLite3 database to store information 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.

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, with detailed explanations about the story, as well as a nice picture showcasing a snippet of the story.
    - Sign in and Sign up buttons on the top left corner
    - Will have navigation bar for different story genres
    - Search bar to find stories to read or contribute to
- 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
    - Indicates the original poster and all users who have worked on the story
- Headers and/or Side Bars
    - Links
    - More info
    - Pictures
- Moderation Page
    - Requires password to get in
      -There will be a check for certain requirements when making a password. The password created by the user will have to be 8 characters or greater, include at least one digit and be unique. The submitted password will be sent from the flask

server back to the Python file and reviewed there to check if it meets the requirements.
    - See suggested changes on each page and section and has the ability to approve or reject them

Tasks:

1) Create Python file with Flask app to run project HTML file. - Jeffery
2) Obtain .csv data to be put 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