CleMenTIne Storytellers - Masha, Tiffany, Clive, Iris

Component Map:

Routes

- **/**
    - Form to login or link to /CreateAccount
        - Login form sends data to /Authorize
    - Redirects to /Welcome if User is already logged in
    - Renders login.html if User is not logged in
- **/Authorize**
    - Checks form login info against database
    - Stores a session if data is correct, and redirects to /Welcome
    - If data is incorrect, flashes a message, and redirects to /
- **/Register**
    - Allows a user to create an account, modifies usersInfo table
    - Redirects to /Login after account is created
- **/CreateAccount**
    - Creates account in SQL
    - Redirects user to Login
- **/Welcome**
    - Renders welcome.html template
    - Displays a two columned list with links to stories:
        - Stories to edit (accessed from usersInfo table)
        - Stories to view (all stories not in usersInfo table)
    - Link to /Create
    - Contains logout button
- **/EditStory?title=<TITLE>**
    - Renders editStory.html template
    - Access data about story from the log table
    - Contains logout button
- **/EditStoryAction**
    - Updates database (modify log table)
    - Redirects to view story
- **/ViewStory?title-<TITLE>**
    - Renders viewStory.html template
    - Accesses data about story from the log table
    - Contains logout button
- **/CreateStory**
    - Renders createStory.html
    - Allows a user to give a story a title and its first line
- **/CreateStoryAction**
    - Updates database (modifies log table)
    - Redirects to the /ViewStory page after the story is written

        - Contains logout button

Python files:

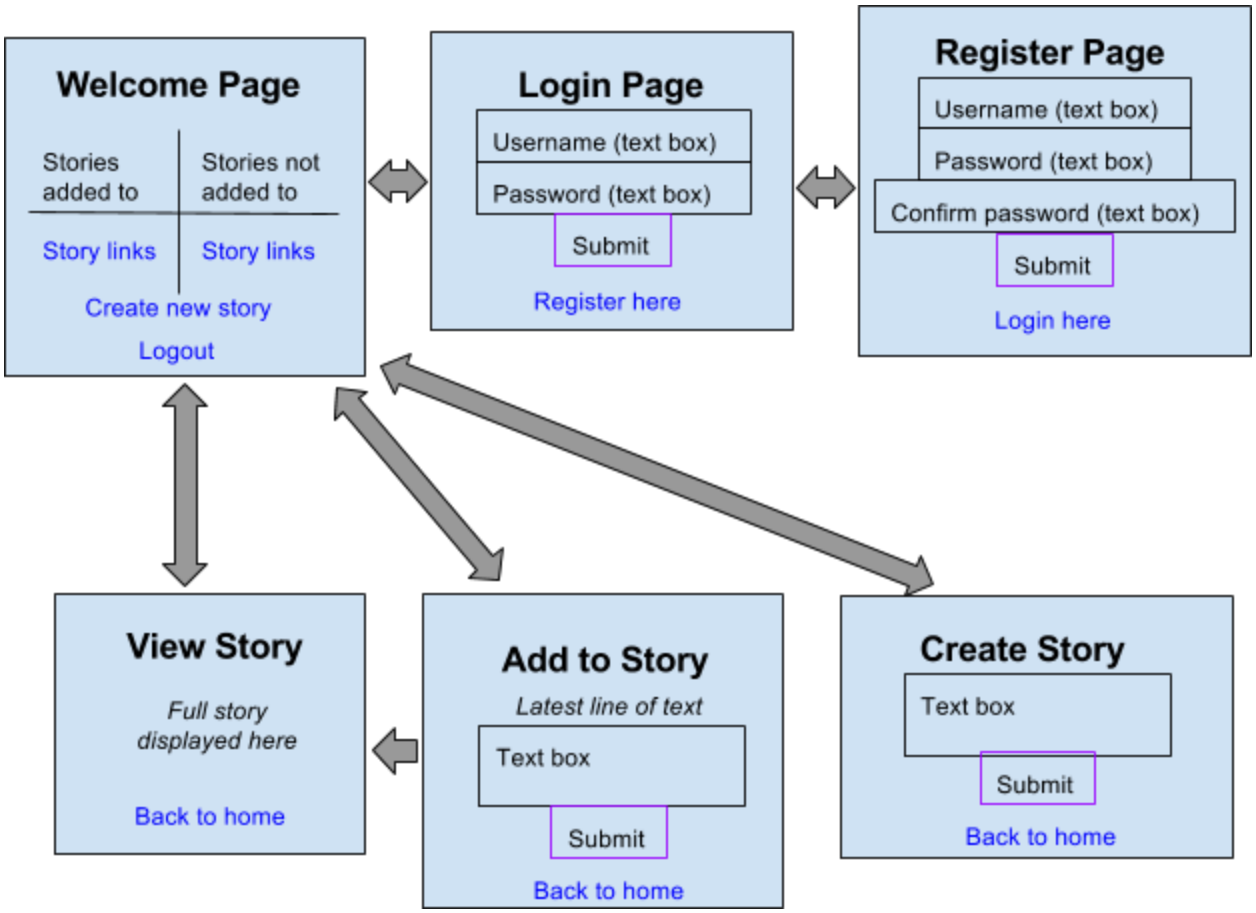
- app.py
    - Contains all routes methods
- database.py
    - checkUsernames(username)
        - Checks if username is taken
        - Returns True if username is taken, returns False if username is not taken
    - checkTitles(title)
        - Checks if title of story is taken
        - Returns true if title is taken, returns False if title is not taken
    - authorize(username, password)
        - Returns true, or false if username matches password
    - updateUsers(username, password)
        - Inputs: username and password
        - Outputs: updates database with user
    - addLine(line, title, userid)
        - Outputs: updates database with edit
    - newStory(title, text, userid)
        - Creates story id and calls addLine()
    - checkEdited(userid)
        - Outputs: array with titles that user edited
    - checkNotEdited(userid)
        - Does opposite of checkEdited()
    - getUserID(username), getStoryID(username)
        - Gets ids
    - getStory(storyid)
        - Gets story text

HTML files:

- Base.html
- Login.html
    - Form variables: username, password
    - action=”/Welcome”
- Register.html
    - Form variables: username, password
    - Action = “/Login”
- welcome.html
    - Variables needed: username, titles_edited, titles_not_edited
- viewStory.html
    - Has query string w/ story title
    - Variables needed: text, title
- editStory.html
    - Has query string w/ story title

    - Form for editing story
        - Form variables: text
    - Variables needed: title
- createStory.html
    - Form for creating story
        - Form variables: title, text

Site Map:



Database Schema:

- **UsersAndTales.db**
    - table userInfo (user TEXT, password TEXT, id INTEGER PRIMARY KEY)
    - table edited (id INTEGER, storyId INTEGER)
    - table log(storyId INTEGER PRIMARY KEY, title TEXT, body TEXT, lastLine TEXT)

Order of stuff to completed:

1. Make login and register page
2. Make dummy database
3. Make welcome page
4. Make view story page
5. Make edit story page
6. Make create story page
7. Add extra features if time permits

Roles:

- Project Manager (and miscellaneous tasks - some HTML): Masha
- Front end (some HTML and CSS): Iris
- Back end database stuff (SQLite and SQL): Tiffany
- Back end flask stuff (Flask): Clive