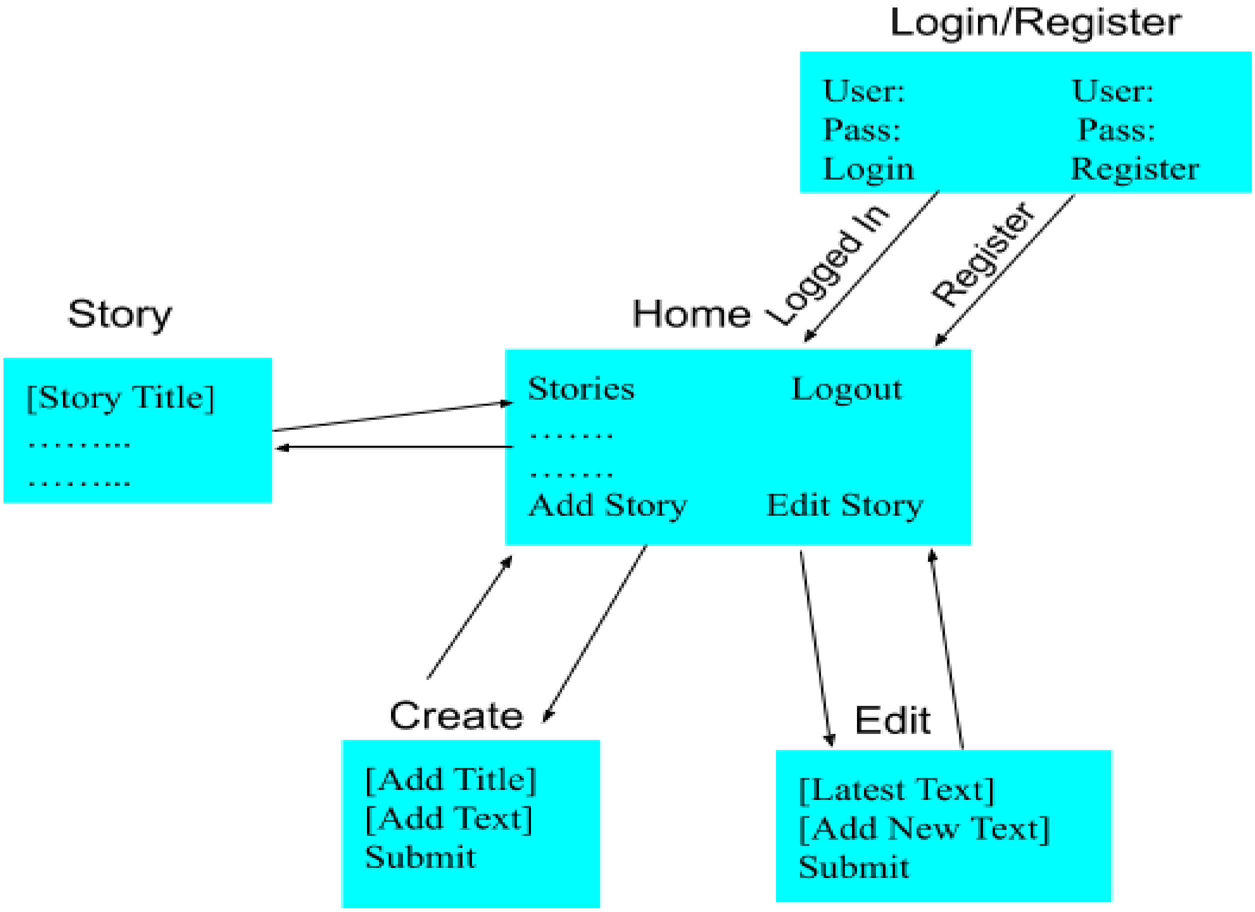
Team: Fine Pizza

Benjamin Avraham, William Lin, Tyler Huang, David Wang

TEAM ROLES:

- Benjamin Avrahami - Create all tables, facilitate adding new data to tables, make sure data from tables are visible on website, other odds and ends
- William Lin - Create python page to take in login information when account is created, enable users to log in, save session until browser window is closed
- Tyler Huang - Make basic outline of project, connect to pages, ensure flexibility of connecting to new pages once created
- David Wang - Create page for editing and creating new articles, display previous edit on page, make sure new story typed by user is saved and added to the appropriate table

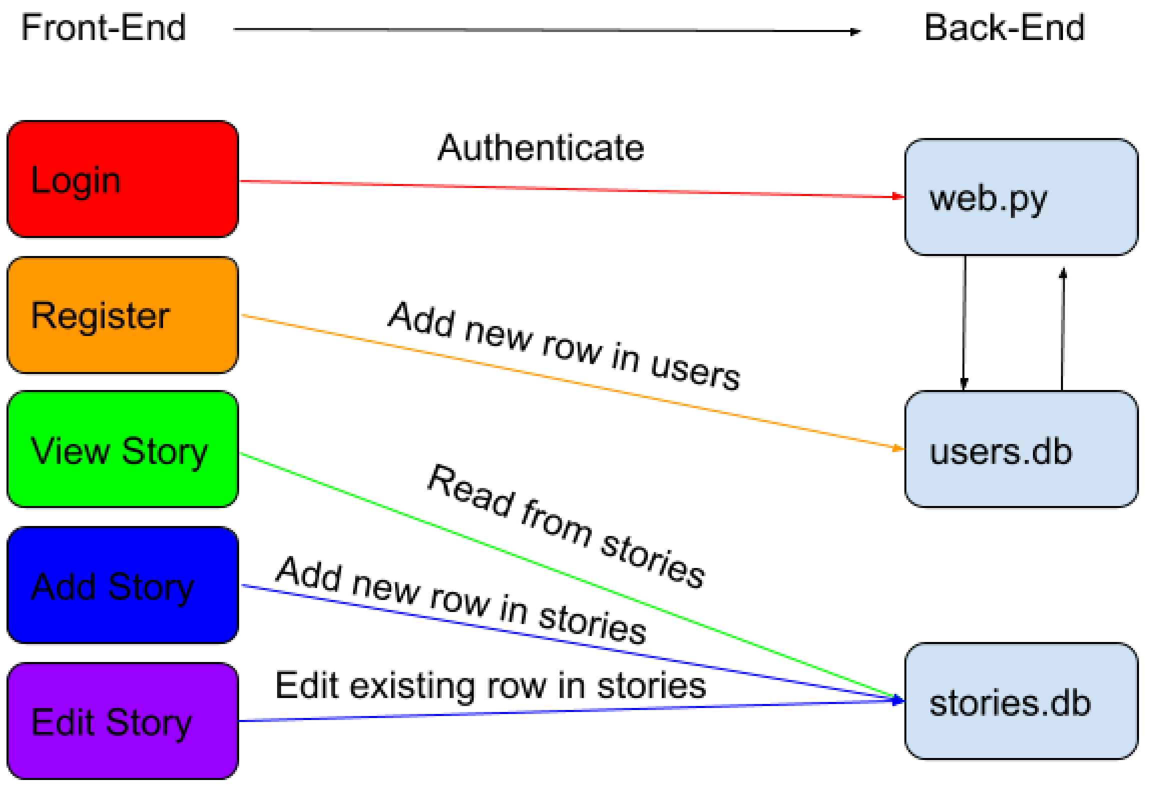
SITE MAP



DESIGN

- Use flask to create the pages of the website
- The landing page should be a login page that allows the user to type in their username and password.

COMPONENT MAP



- After they click login, their info is checked with the data base to make sure that they are a registered member. If they aren't, then it redirects to the home page and flashes a message regarding the error.
- Users also have the option to register for an account, which will redirect them to another page for signing up

- Users will need to enter a secure username and password of at least 6 characters long
  - If the username already exists, then flash an error message
  - If the passwords already exists, don’t do anything because that could give away someone else’s password
- After registering, users will be redirected back to the home page to log in with their new account
- Registered accounts will be stored in a table using SQLITE3

- Once login is successful, users are redirected to a page with a list of stories. There will be an option to log out, add a story, or edit an existing story. The log out option should be on every page hereafter.
  - If the user clicks a link to a story, they will be redirected to a page that only allows them to read the most recent entry. There will be an option to edit the story within that page
  - If the user clicks “add story”, then they will be redirected to a template page that prompts the user to enter a title story and the story itself
  - If the user clicks “edit story”, then they will be prompted to select a story to edit
    - Once they click a story, they will be redirected to an editable text file of the most recent version of the story and there will be an option at the end to save changes
  - All changes will be recorded in a table that shows the username of the user who made the changes, what changes they made, and the time that they did it
  - All users are only allowed to edit a story once, so within the users table there is also a column that stores the names of the stories that the user has contributed to.
    - Every time a user tries to edit a story, check the database to see if they have edited it before.
      - If yes, flash an error and redirect back to home page.
      - If not, add the story name to the database and grant access to edit

- No need to check for anything if the user is adding a story, but still add the story name to the database as a story that the user has edited

DATABASE TABLES

All table names are lowercase

Users

user_id	username	password
INTEGER	TEXT	TEXT

Stories

story_id	story_name	story_text	last_edit
INTEGER	TEXT	TEXT	TEXT

Edits

user_id	story_id
INTEGER	INTEGER

Variables:

user_id - id which corresponds to a user/pass combo, starts at 0 and increments by 1 for every new user

username - a string input from user

password - a string input from user

stories_edited - list of story_id’s a user has edited

date_created - date of creation of

story_id - id of a story, starts at 0 and increments by 1 for every new story

story_name - name of a story, string input by the creator of the story

story_text - full story stored as a string

last_edit - the latest edition to the story stored as a string

date_created - date of creation of story

story_add - the addition to the story made by the user

Web.py Functions:

- “/”
- “/register”
    - register(user, pass)
        - Takes the user inputs and stores them as a username and password in the database
- “/home”
- “/home/viewStory”
    - readStory(storyName)
        - Takes in the story that is clicked as an input, and prints the most recent entry of the story
- “/home/addStory”
    - addStory(storyName, text)
        - Takes in what the user inputs and adds another row to the database of stories
- “/home/editStory”
    - editStory(storyName, text)
        - Finds the storyName in the database and updates the text