

# Project 0: Design Document

Ely Sandine, Celine Yan, Kevin Zhang, and Yikai Wang

## I. Directory Structure

```
~/Github/storyTellingTale/  
- app.py  
- /utils/  
  - authenticate.py  
  - dbEditor.py  
- /templates/  
  - login.html  
  - homepage.html  
  - add.html  
  - create.html  
- /data/  
  - database.db  
- /static/  
  - Style.css
```

## II. Functionality of Each File:

### **app.py**

Main file that controls everything. This is where Flask will run.

### **authenticate.py**

Contains functions for login and register. It will access *database.db* for the *users* table and edit it for logging in and registering. Inputted usernames and passwords will be checked against the database for accuracy and redundancy.

### **dbEditor.py**

Contains functions for modifying the *stories* table in *database.db*. Whenever a user adds to a story, the *last post* will be modified to the latest post. The full story string will be appended the most recent post. This file will also have functions to return information regarding *database.db*.

### **login.html**

Root page. Users are required to login to access the site. If user does not have an account, he may register, as long as his username is not already taken. After he successfully logs in, user will be redirected to the homepage. The username will also be stored in a session.

### **homepage.html**

Features a navigation bar along the top. The page's logo can be seen in the middle, and there will be two buttons on the right and Logout button on the left. The first button will lead you to creating a new story; the second button will lead user to

adding to an existing story user has not added to yet. The rest of the page is devoted to displaying the full stories of those stories that user has contributed to. These stories will be organized by latest -> earliest user contributed to.

### **add.html**

For when user is going to add to a story. A user will be assigned to three random story that he has not added to before. This page displays the ***last contributions*** to each story. Underneath each contribution is a textbox/form for user's contribution. There is

a minimum number of words required for submission (2 words). If the user wishes for more

Options, there will be a "More stories" button below, displaying three more new stories every time the user clicks it. If there are no more stories to display, nothing will happen. There is a submit button at the bottom, and pressing it will commit user's changes. It will then redirect user to the homepage.

### **create.html**

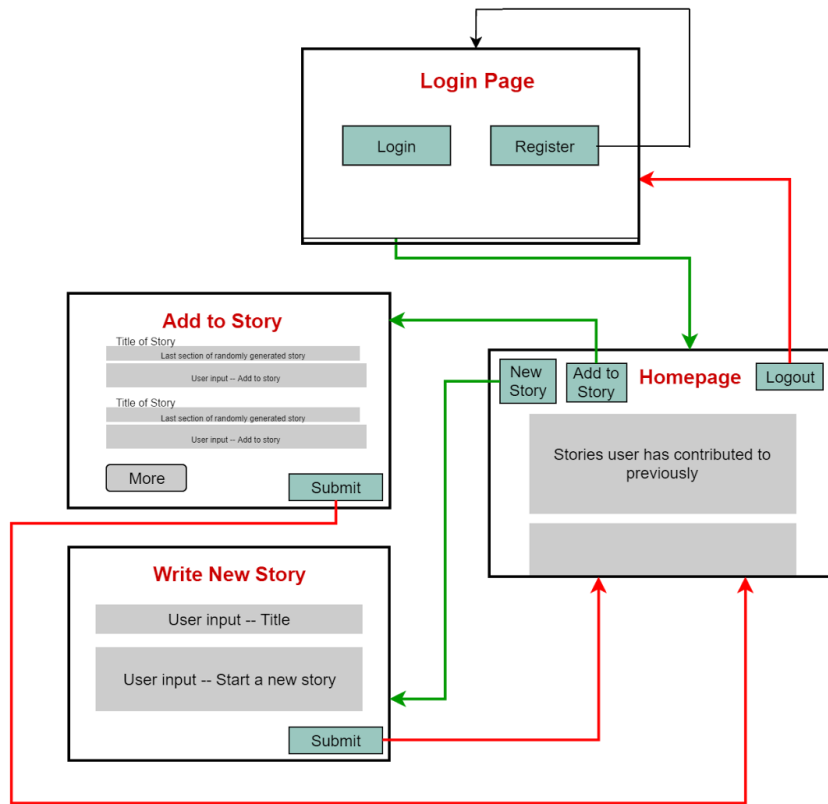
For when a user is creating a new story. There will be a form for the title and a form for the

story. The user is allowed to submit as many words as he would like; there is a minimum word count of 2 words. A title is required. There is a submit button at the bottom, and pressing it will commit user's changes. It will then redirect user to the homepage.

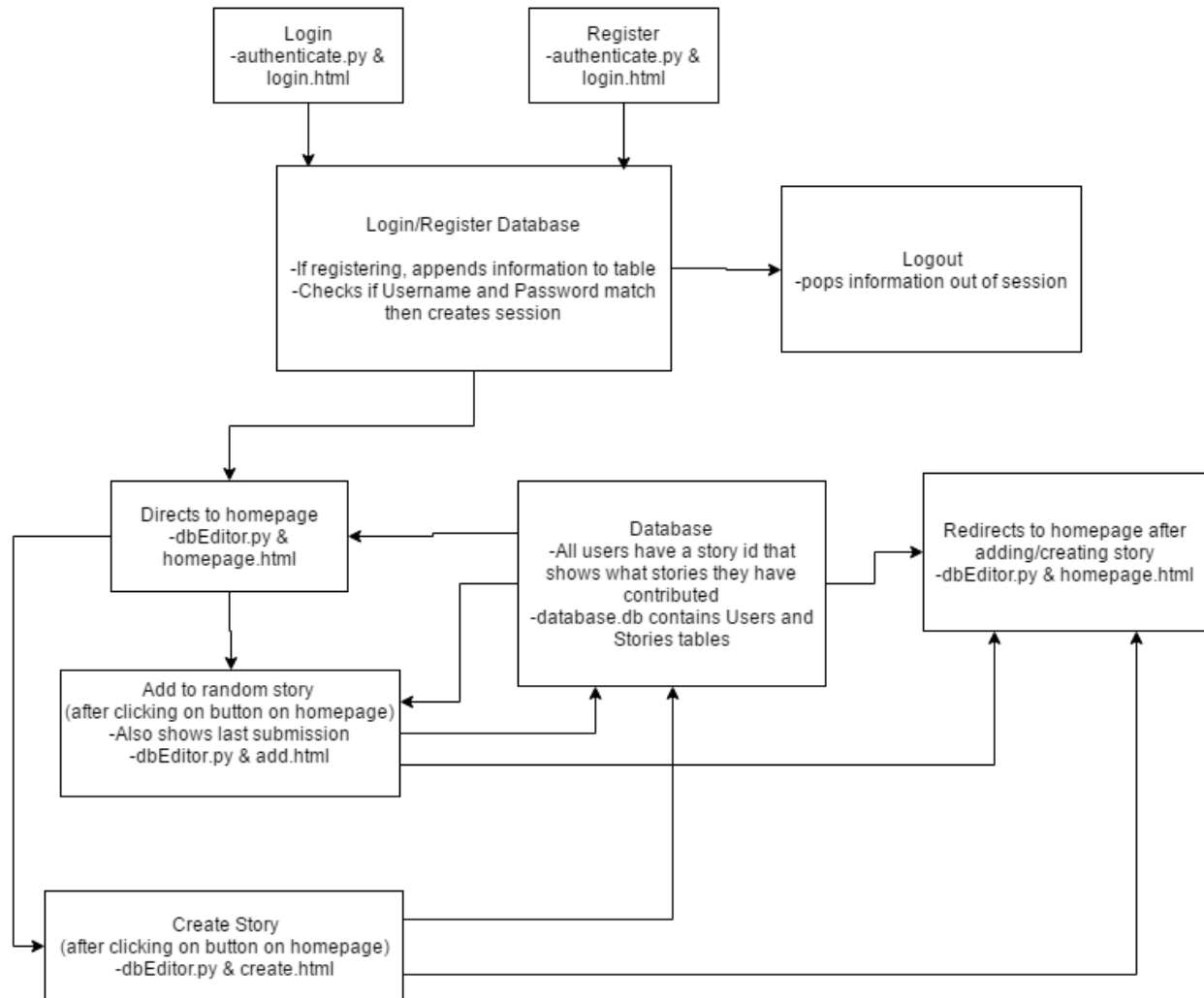
### **database.db**

Stores the data. See *V. Database Schema* for more information.

### III. Site Map



### IV. Component Map



## V. Database Schema

### database.db

- **users** - table for all user data. Each record is a user.
  - username - string - The username of the user.
  - password - string - The hashed password of the user.
  - story\_ids - string - A list of story ids that the user has already submitted to separated by commas.
- **stories** - table for all story data. Each record is a story.
  - id - int - Every story has a unique id.
  - title - string - The title of the story.
  - time - float - The time the story was last edited. This is used to render stories in the user's homepage.
  - last\_submission - string - The last submission, rendered when the user edits a story. When the user adds to the story they replace the last\_submission.
  - story - string - The complete text of the story.

### Users

Username	Password (hashed)	Story ids (Text)

### Stories

Story id	Title	Date	Last Submission	Story

## VI. Work Distribution

**Celine** (Project Manager): login.html, homepage.html

**Yikai**: add.html, create.html, style.css

**Kevin**: authenticate.py, app. Py

**Ely**: dbeditor.py, database.db