

# P00: Design Doc

Presented by AEIOU

Brian Chen, Vansh Saboo, Weichen Liu

Target ship date : {2022-11-20}

## Flask

### Templates:

- homepage.html
- login.html
- register.html
- story.html
- edit.html

### Inside app.py

- **@app.route("/")**
  - Homepage
  - If logged in, will list stories user has edited
    - Will use edited column from user table, see below
- **@app.route("/login")**
  - Will only be available if not logged in
  - Option to login, link to register
- **@app.route("/register")**
  - Will only be available if not logged in
  - Redirects to /login
- **@app.route("/story/<x>")**
  - Returns story.html with specific entry information
  - Button to edit if logged in
- **@app.route("/edit")**
  - Will only be available if not logged in
  - Used for both creating and editing
    - Will differentiate between creating and editing by having an input field with type="hidden", and value=title
    - When clicked from a story page, there will be a value for the title
  - Opens edit.html page

- If editing, displays title and last update, textarea for new content
- If creating, textarea for title and content
- Editing and creating both replace into the story table
  - When editing, an already existing primary key id is used(so it replaces original)
  - When creating, creates a new primary key id
- Redirects to /story
- **@app.route("/logout")**
  - Pop username from session,
  - Redirect to /index

## SQLite3

### USER

<b>id</b>	int primary key
<b>username</b>	text
<b>password</b>	text
<b>edited</b>	text

**\*primary key is used because it must be unique, allowing for users to be changed by replacing into the table with the same id but different values in other columns**

**\*edited will be a list of story ids the user has edited**

- Ex. edited="1,3,4,"

### STORY

<b>id</b>	int primary key
<b>user_id</b>	text
<b>title</b>	text
<b>content</b>	text
<b>recentuser_id</b>	text
<b>recent_content</b>	text
<b>user_date</b>	text
<b>recent_date</b>	text

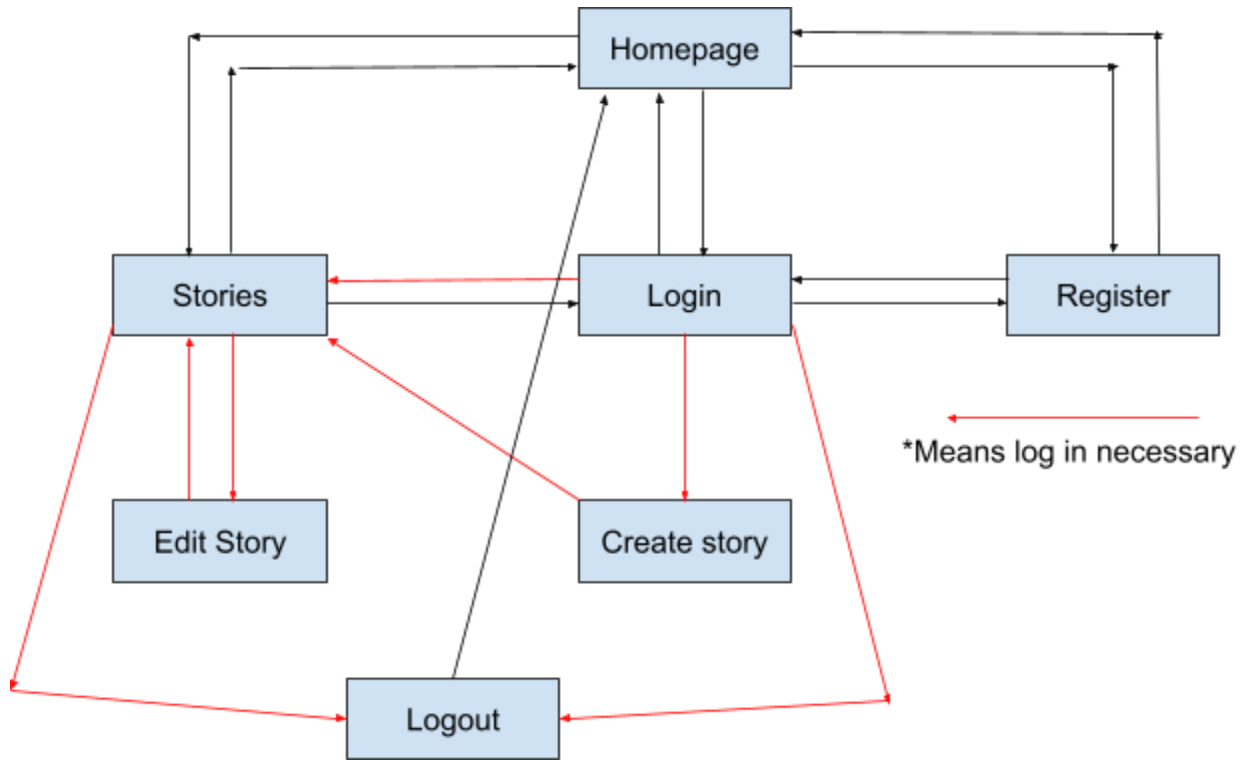
## Helper Variables

- **currentusers** = {}
  - Dictionary for users, format user:password
- **stories** = set()
  - Set of the **title** of every existing **STORY**
  - Purpose is to use to display story options on html page
  - We are using a set because we can easily add every **title** to this set everytime the story table is edited, and there will be no repeats
- **edited\_currentuser** = []
  - List of the **id** of **STORY** that a user has edited, the **edited** column of the **USER** table
- **user\_watchlist** = set()
  - Similar to edited\_currentuser but has **title** instead of **id**

## Helper Functions

- **update\_users()**
  - Populates **currentusers**
  - Called every time a user registers
- **update\_stories()**
  - Populates **stories**
  - Called every time a **STORY** is edited/created
- **update\_edited\_currentusers(username)**
  - Populates **edited\_currentuser**
  - Parameter username is session["username"]
  - Called every time a user logs in or edits a story
- **grab\_stories(story\_ids)**
  - Parameter story\_ids is **edited\_currentuser**
  - Populates **user\_watchlist**
- **grab\_username(id)**
  - Used whenever a user's **username** is needed, as we mostly use user **id**
- **get\_time()**
  - Grabs current time using datetime module and datetime.datetime.now()
  - Splices the output so the smallest unit of time is the minute

# Site Map



## Tasks

# Brian Chen

- layout.html and homepage.html and CSS
- Display story flask method and story.html; story table
- Edit story flask method and edit.html; user interaction with edited stories
- design.pdf finalizations

## Weichen Liu

- CSS for layout
- login.html and login Flask method
- design\_wiki.pdf finalizations
- requirements.txt and Readme

# Vansh Saboo

- Register.html and register Flask method
- user table and user creation in sqlite3
- requirements.txt and Readme