DESIGN DOCUMENT

Nalanda Sharadjaya, Farhan Haque, Ziyan Lin, Gabriel Marks

Database Schema

USERS TABLE

Username (TEXT)	**User Password** (TEXT)	**User ID** (INTEGER)

STORY TABLE

Story ID (INTEGER)	**Story Title** (TEXT)	**Latest Update ID** (INTEGER)

UPDATES TABLE

Story ID (INTEGER)	**User ID** (INTEGER)	**Update ID** (INTEGER)	**Update** (TEXT)

Component Organization

LOGIN AND REGISTER

`templates/login.html`

- Two text fields for username and password; two "submit" buttons for login and register.

USERNAME: <TEXT FIELD>	PASSWORD: <TEXT FIELD>
LOGIN: <SUBMIT BUTTON>	REGISTER: <SUBMIT BUTTON>

- Users can login to their account, or register for a new account.
- Functionality is linked with `account.py`.

`utils/account.py`

Contains functions to authenticate login of existing user and register new user.

- `login(user, pass)`
  - If username and password match those stored in Users Table in SQL database, redirect to `main.html`.
  - If username does not exist, password is incorrect, or something invalid is inputted, redisplay `login.html` with error message.

- `register(user, pass)`
    - If username does not exist and password is valid, create account and add information (including generated `User ID`) to Users Table in SQL database. Redisplay `login.html` and prompt to login.
    - If username or password is invalid, redisplay `login.html` with error message.

SQL FUNCTIONALITY

`templates/main.html`

- Users see complete, up-to-date versions of stories to which they've contributed in reverse chronological order.
- Users can click on one existing story in the feed to link to `story.html`.
- In a sidebar, users can look at the titles of stories to which they can contribute and click on one to link to `add.html`.
- Both the sidebar and the main feed will be displayed using HTML tables (will not resemble the table below). These will be constructed using for loops.

STORIES YOU HAVEN'T SEEN YET	STORIES YOU'VE CONTRIBUTED TO
Title 1 (Newest) Title 2 (Older) Title 3 (Even older)	Title 4 (Newest) Title 5 (Older) Title 6 (Even older)

`templates/story.html`

- Users can view the entirety of a single story to which they have contributed.

`templates/add.html`

- Users can view latest update to a story to which they wish to contribute.
- Beneath the latest update there will be a text field for users to add their own update to a story.
- When `add.html` is opened, write a log file. Remove when `add.html` is closed. This way, only one user can add an update to a particular story at a time.

STORY TITLE
LATEST UPDATE (username attached to update will not be made visible until user has contributed to story; then user can view all usernames attached to all updates)
USER'S UPDATE: <TEXT FIELD>
ADD UPDATE: <SUBMIT BUTTON>

`utils/sql.py`

Will obtain and communicate data from SQL databases.

- `get_stories(user_id, viewing_on)`

    - Uses `is_edited()`.
    - Returns list of `Story ID`s attached to `user_id` if `viewing_on` is true.
    - Returns list of `Story ID`s not attached to `user_id` if `viewing_on` is false.
- `add_story(story_id, user_id, update)`
    - Edits Updates Table to reflect latest update (adds line with `Story ID`, `User ID`, and `Update`).
    - Edits Story Table to reflect latest update (edits `Latest User ID` field in line with given `story_id`).
- `get_latest_update(story_id)`
    - Returns `Latest Update ID` for a particular `story_id`.
    - Useful for `add_story()`.
- `get_all_updates(story_id)`
    - Returns list of `Update ID`s for a particular `story_id`.
- `is_edited(story_id, user_id)`
    - If `user_id` is attached to `story_id` in database, user has contributed to the story and the user can view all updates.
    - Otherwise, story has not been edited by user and user can only view latest update.
- `get_title(story_id)`
    - Returns `Title` for a particular `story_id`.
    - Useful for display stuff.
- `get_update(update_id)`
    - Returns `Update` for a particular `update_id`.
    - Useful for display stuff.

FLASK APP

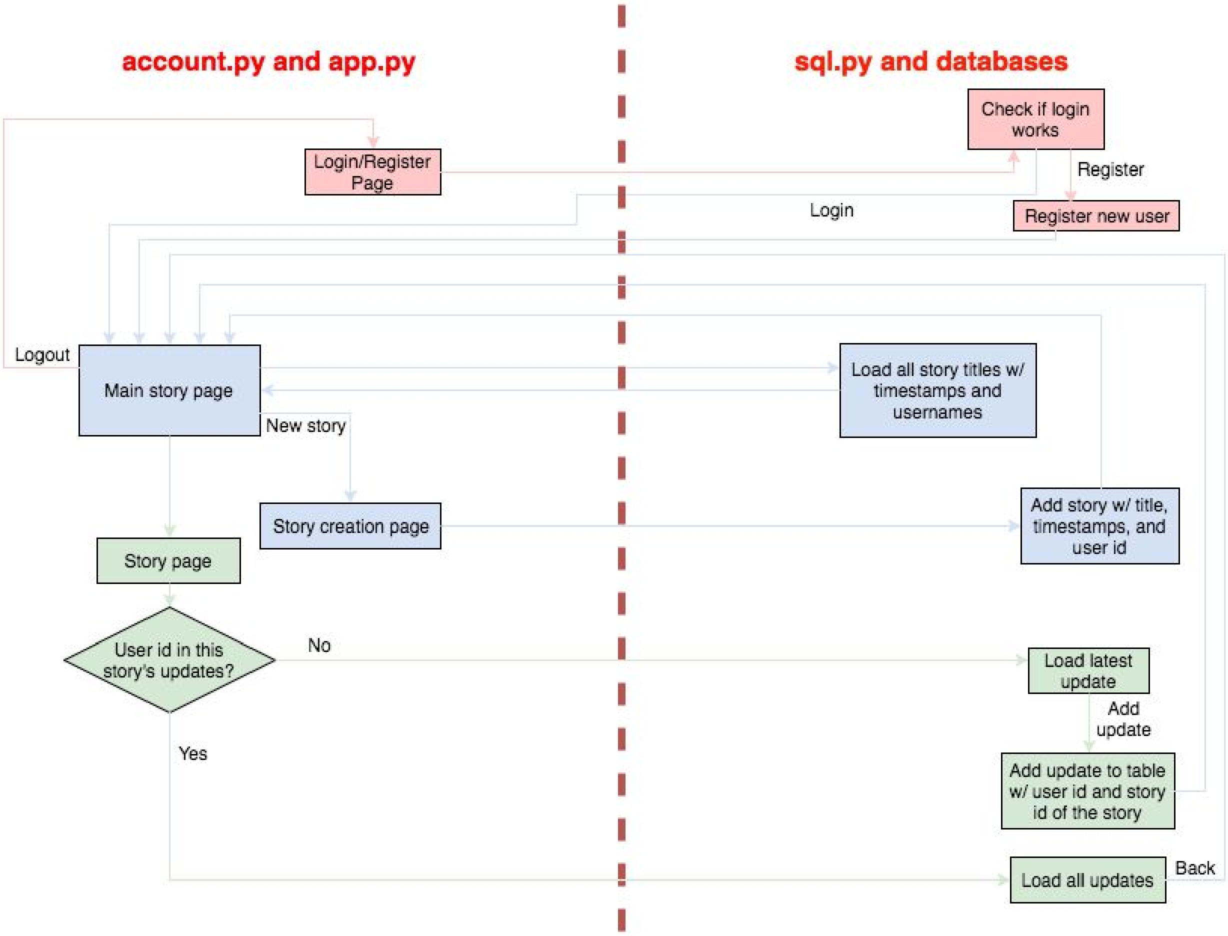
`app.py`

- Has session stuff implemented.
- `authenticate()`
    - Uses `account.py` for back-end, `login.html` as template.
- `main()`
    - Uses `sql.py` to create content for HTML tables (both of sidebar, which will contain stories the user is able to update, and main feed, which will contain stories the user has already updated).
    - Uses `main.html` as template.
- `story()`
    - Uses `sql.py` to display a completed story page, which shows the title of the specific story, and each of its updates in chronological order (with the

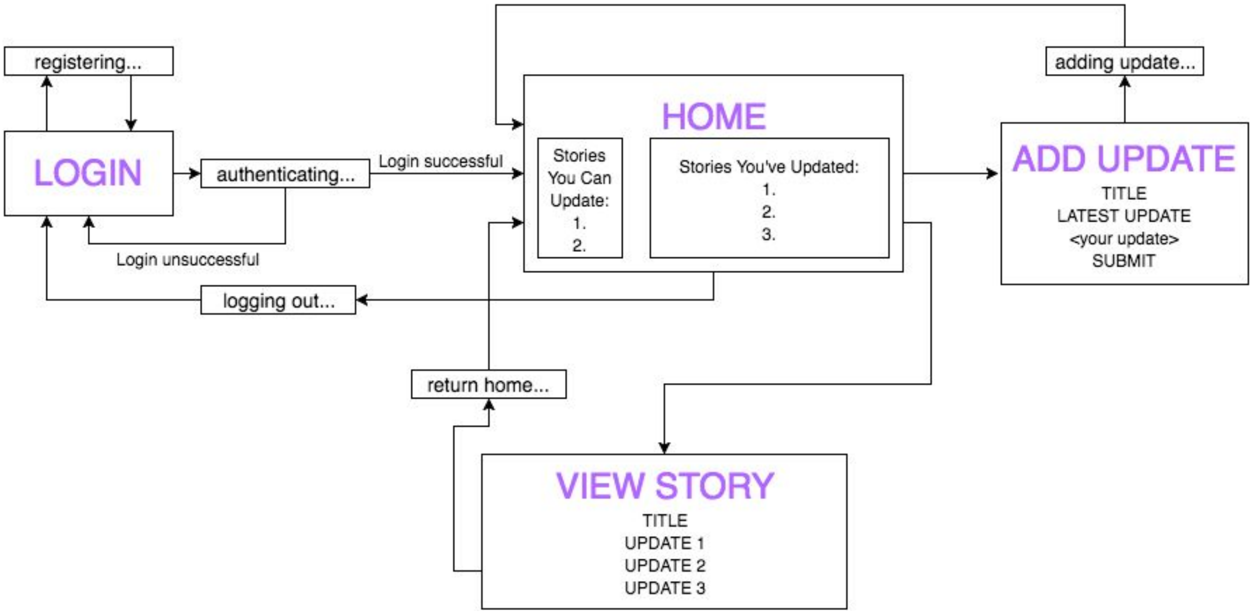
username attached to each update).
Uses `story.html` as template.

- `add()`
  - Uses `sql.py` to display a story-editing page, which only shows the title of the story to be edited and its latest update.
  - Uses `add.html` as template.

Component Map



Site Map



Division of Tasks

Project Manager (ensures all group members are on task; oversees progress of group): **Nala**

Login and registration, HTML templates and Flask apps: **Ziyan and Farhan**

SQL Database management: **Gabriel and Nala**

Front-end design (CSS files, etc), if there is time: **Ziyan**