Jobless Monkeys: Aidan Wong, Abidur Rahman, Brian Liu, Leon Huang
SoftDev
P00: Move Slowly and Fix Things
2024-10-23
Time Spent: 1.5 hrs
TARGET SHIP DATE: 2024-11-18

# I. Description

This project is a collaborative storytelling website where registered users can create stories and contribute to existing ones. When creating a story, users add a title and a story of any length. When adding to a story, users can only see the most recent update by another user. After submitting their contribution, users are restricted from adding to that story again, but can now read the full story. Each user will have a personalized homepage to view all the stories they have contributed to.

## A. Program Components

1. User Accounts:
    a. Creation of accounts and login/logout functionality
    b. Sessions
2. Routes to different pages of the website using Flask and Python
3. SQLite3 Database: Stores data of the user and stories
4. Jinja Templates:
    a. User dashboard: Contains stories that the user has contributed to; Landing page when the user logs in (List/Grid view)
    b. Story Collection: Contains all available titles (but contents are hidden); Different permissions (read/write) depending on if the user has contributed or not
    c. Story viewer: Displays the entire content of any story; Can only be accessed if the user has contributed to the specified story
    d. Story updater: Page where a user is shown the latest update and allows the user to make an addition to the story

Note: Templates c-f all include a button to log out

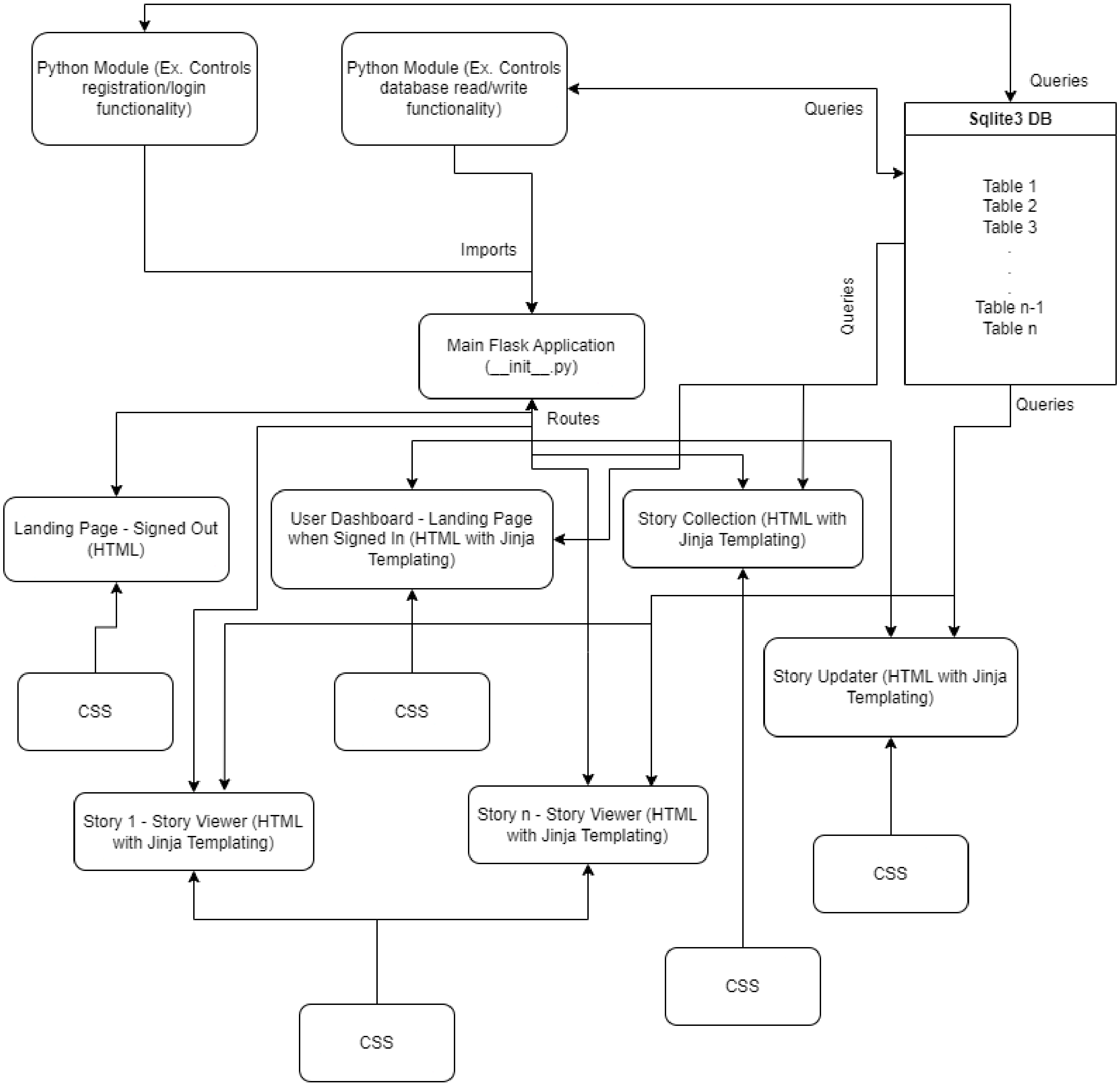
## B. Program Component Connections

1. User accounts: Give access to the entire site (Created through registration, and can be logged in to or out of). Allows for the use of permissions per user for the different stories
2. Routes + Python: Allows users to transverse throughout the website (if logged in). They connect the different pages (HTML documents) of the website. Python

allows for permission control and interacts with the database to determine if a user can view/edit a story

3. Database: Stores information related to the user (ID, permissions, etc) and stories (ID, most recent edit, etc)
    a. One of the main factors for information exchange between components (has all the data)
4. Templates: Allow for dynamic web pages (as they need to update when new stories are created/edited)

## C. Component Map

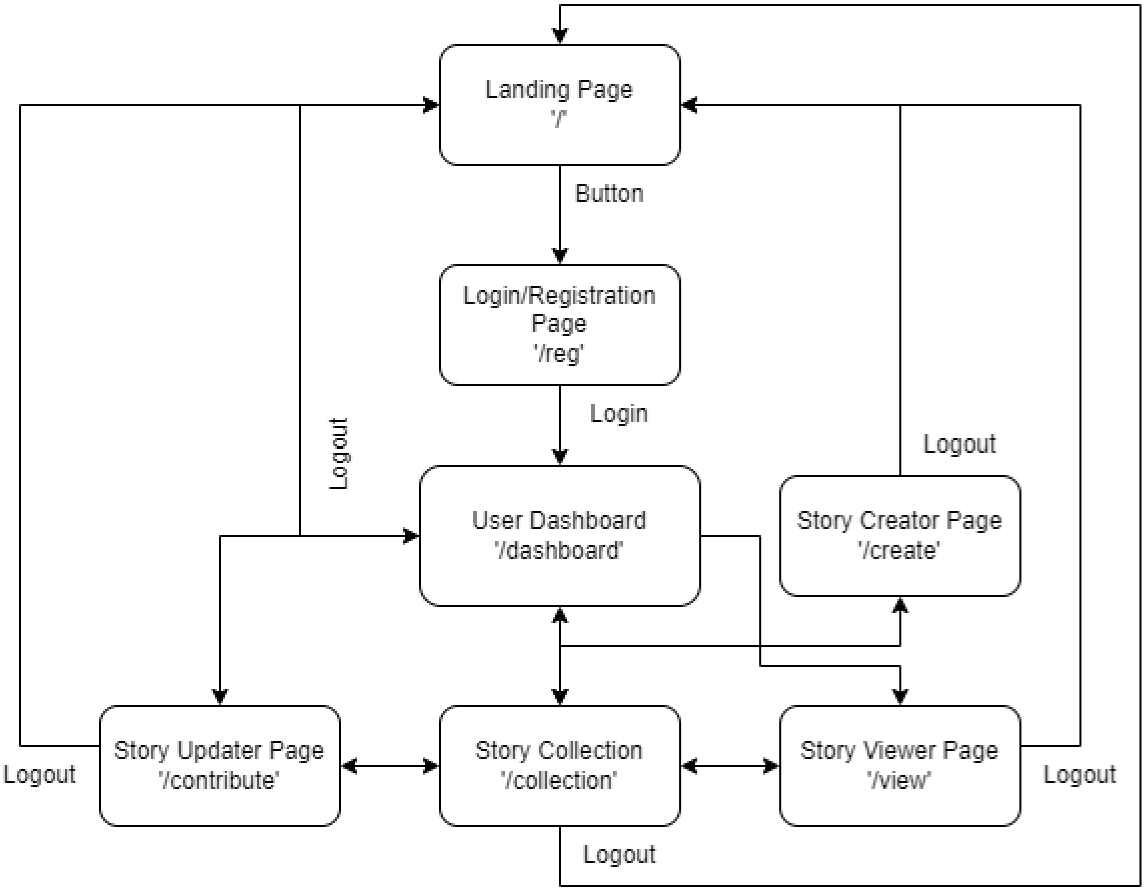

Python Module (Ex. Controls registration/login functionality)
Python Module (Ex. Controls database read/write functionality)
Queries
Queries
Sqlite3 DB
Table 1
Table 2
Table 3
.
.
.
Table n-1
Table n
Imports
Queries
Main Flask Application (__init__.py)
Routes
Queries
Landing Page - Signed Out (HTML)
User Dashboard - Landing Page when Signed In (HTML with Jinja Templating)
Story Collection (HTML with Jinja Templating)
Story Updater (HTML with Jinja Templating)
CSS
CSS
Story 1 - Story Viewer (HTML with Jinja Templating)
Story n - Story Viewer (HTML with Jinja Templating)
CSS
CSS
CSS

## D. Database Organization

1. User Table
    a. Username/User ID (PK)
    b. Password
2. Story Collection Table (Contains all stories)
    a. Story ID (PK)
    b. Titles (PK)
    c. Story Content (Entire)
    d. Version (PK)
3. Story Table (One for each story)
    a. Story ID (FK)
    b. Latest Author - Username/User ID (FK)
    c. Latest Text Content
    d. Version (FK)
4. Author Table
    a. Username/User ID (FK)
    b. Permission Level (read/edit)
    c. Version (FK)

Note: FK stands for foreign key (to link tables), PK for primary key (each row value must be unique)

## E. Site Map + Descriptions



1. Landing page (/): Contains website description and a button to log in/register for an account; Default page when the user is not logged in
2. Login/Registration Page (/reg): Allows the user to create or sign in to an account; Accessed only through the landing page
3. User dashboard (/dashboard): Contains stories that the user has contributed to; Landing page when the user logs in (List/Grid view)
4. Story Collection (/collection): Contains all available titles (but contents are hidden); Different permissions (read/write) depending on if the user has contributed or not
5. Story Viewer (/view): Displays the entire content of any story; Can only be accessed if the user has contributed to the specified story
6. Story Updater (/contribute): Page where a user is shown the latest update and allows the user to make an addition to the story; Button that changes the page to display what the text would look like with options to edit more or submit; Includes any formatting the user inputted
7. Story Creator (/create): Page where user can create a story (Title and contents); Button that changes the page to display what the text would look like with options to edit more or submit; Includes any formatting the user inputted

## F. Task Breakdown

1. Aidan Wong: Full Stack/ Project Manager
    a. Create routing and logic between pages (permissions, etc)
    b. User session management
    c. Glue frontend and backend code (Making database work with forms/python modules, etc)
2. Leon Huang: Frontend
    a. Create HTML pages with Jinja templating - Includes any forms required for logging in/signing up
    b. Design site style with CSS
3. Brian Liu: Backend/Database
    a. Create SQLite3 database schema
    b. Work on database interaction modules (python) → General operations (ex. Inserting data, creating tables, etc)
4. Abidur Rahman: Backend/ Python
    a. Logic for story contribution - Can only contribute once and able to view the full story once contributed
    b. Permission management (who can read/write depending on contribution status)