

Jobless Monkeys: Aidan Wong, Abidur Rahman, Brian Liu, Leon Huang

SoftDev

POO: Move Slowly and Fix Things

2024-10-24

Time Spent: 7.5 hrs

TARGET SHIP DATE: 2024-11-07

DESIGN DOCUMENT (VERSION 4)

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

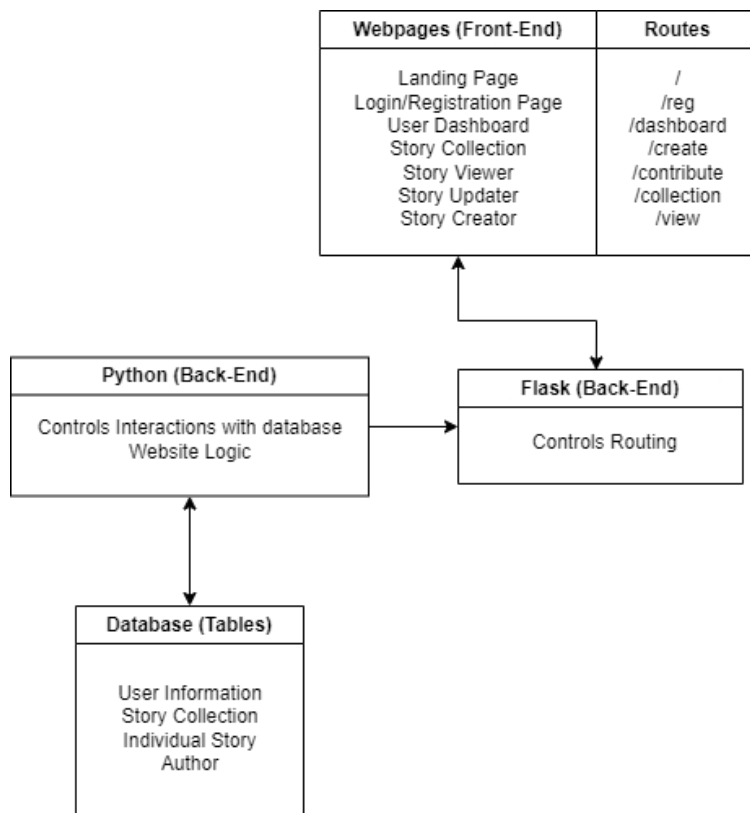
- I. User Accounts:
 - a. Creation of accounts and login/logout functionality
 - b. Sessions
2. Routes to different pages of the website using Flask and Python
3. SQLite3 Database: Stores data of the user and stories
4. Jinja Templates:
 - a. User dashboard: Contains stories that the user has contributed to; Landing page when the user logs in (List 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 content depending on the user's contribution status: only the latest update and a contribute button if they haven't contributed, a full story if they did contribute (with the name of the contributor on the right side of each addition)
 - d. Story contribute: Page where a user is shown the latest update and allows the user to make an addition to the story
 - e. Registration/Login: Small templating for telling the user if they entered incorrect information

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)

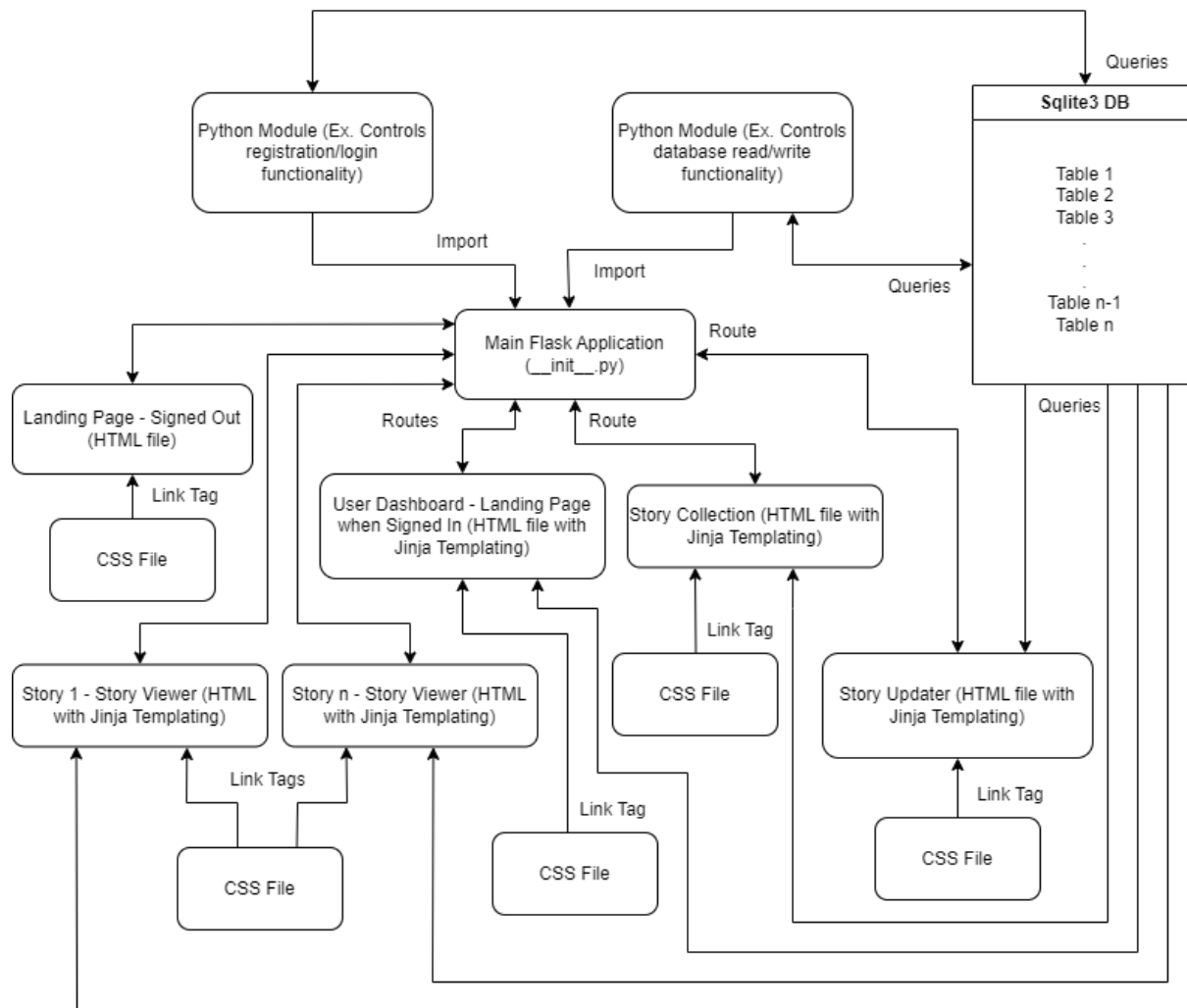
C. Component Maps

I. Simple Version: Basic component Structure



* Inspired by Swedish Fish's Component Map

2. In-Depth Version: More specifics into how different files interact with each other
 - a. Database alterations (write permissions) are restricted to Python modules, which allows for more organization and consistency



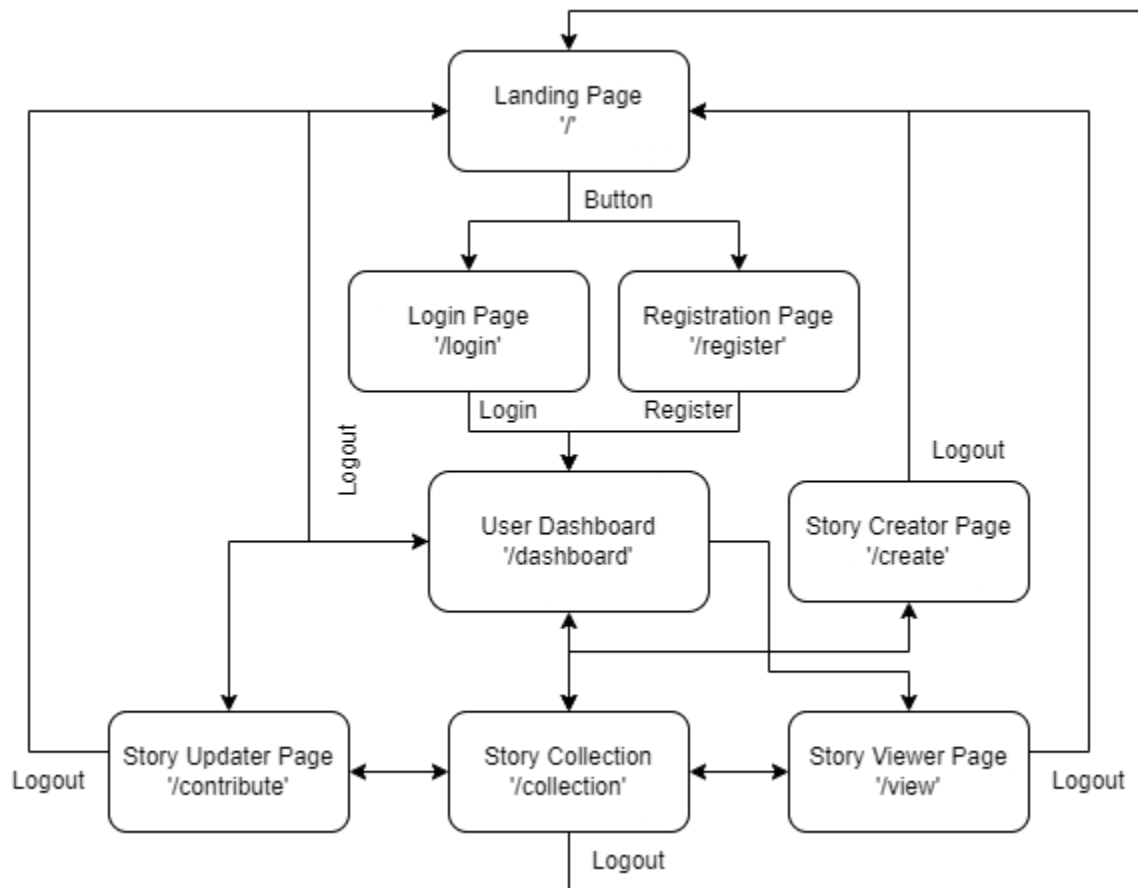
D. Database Organization

1. User Table
 - a. Username/User ID (PK)
 - b. Password
2. Story Collection Table (Tracks most recent data for stories)
 - a. Story ID (PK)
 - b. Titles (PK)
 - c. Most recently added content
 - d. Current version
 - e. Original Author

3. Story Table (Tracks edits for all stories)
 - a. Story ID (FK)
 - b. Version
 - c. Text Content of Version
 - d. Author of Version

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 (/login): Allows the user to sign in to an account; Accessed only through the landing page
3. Registration Page (/register): Allows the user to register for an account and logs the user in upon successful account creation; Accessed only through the landing page
4. User dashboard (/dashboard): Contains stories that the user has contributed to; Landing page when the user logs in (List/Grid view)

5. Story Collection (/collection): Contains all available titles and the original author; Button that allows user to view the text by bringing them to the view page
6. Story Viewer (/view): Displays content depending on the user's contribution status: only the latest update and a contribute button if they haven't contributed, a full story if they did contribute (with the name of the contributor on the right side of each addition)
7. Story Updater (/contribute): Page where a user is shown the latest update and allows the user to make an addition to the story;
8. Story Creator (/create): Page where the user can create a story (Title and contents)

Note: All pages besides the Landing Page and Login/Registration Page include a button to log out

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)