

HAZL - Jayden, Ziyad, Aditya, Kevin
SoftDev
p00 - StoryBoard
Target Ship Date: 2024-11-07

OPTION CHOSEN: Scenario 1

Each Team Member's Roles For Design Document

Components and their Relations: Ziyad, Aditya, Jayden

Site Map: Jayden

Database: Kevin, Jayden

Program Components:

Outline:

- Users, Sign-Up/Sign-In Capabilities
- A Story Manager, something to keep track and store stories to be viewed, edited, and added upon; needs to keep track of contributors
- A homepage, login/sign-up page, story creator page, pages for stories

Flask Framework:

Role: Web server, connection between backend and frontend.

- The flask server will serve as a connecting point between the web and what has been coded. It will allow us to display our implementation on a web browser that the user can use to design or edit stories.

SQLite3 Database:

Role: storage for user accounts, stories, and story contributions. Contains tables for users, stories, and contributions.

- "dataBase.db" will house information for all stories which will be pulled from each HTML page to display text and information.

User Registration and Authentication:

Role: Allows users to create accounts, log in, and log out.

- "Login.html and Logout.html" - simple server logic will be used to create accounts for each individual and keep them securely while also retrieving them appropriately.
- Login will allow users to make new accounts and log in to them, logout will display a confirmation message that the user has successfully signed out

Story Creation Module:

Role: Allows logged-in users to start a new story with a title and initial text.

- “createStores.html” - this webpage will allow users to effectively create their own stories with their own desired media. All information will then be translated from an HTML form to create the page, subsequently storing it in our SQL database.

Story Contribution Module:

Role: Allows logged-in users to add text to an existing story. Users will only see the latest addition to a story. A user can only contribute once to each story as each editors story will be tracked and kept in a database for that user.

- “newStories.html” - this webpage will have the ability to EDIT previously curated stories through an edit button, while also being able to see a list of all stories (with a section for the most recent). The EDIT button will take the user to a page similar to the creating stories page, but will already have the original text and media there to change. After changes, all information will be updated to the SQL database.

Home Page (User Dashboard):

Role: Displays stories the logged-in user has contributed to, providing an overview of their participation.

- “homePage.html” - The homepage will serve as the main route of the entire web application as it will have a description of the website, a place to credit and view stories, as well as login/logout features. If the user has edited any stories, it will appear on the bottom.

Story Display and Navigation:

Role: Allows users to browse and contribute to ongoing stories or view new stories they have started.

- This feature will be implemented in the “homePage.html” and the “newStories.html” page to avoid clutter and unnecessary web pages. We plan to make this efficient and effective for utmost clarity (through elimination of unnecessary clutter and through not overdoing CSS styling).

Front-end Templates (HTML/CSS):

Role: Web Page structure and user interface. Uses Flask templates to display the site and provides a user-friendly way for interaction.

- “storyTemplate.html” - this is an example of one of the templates we plan to use to create the story as it pulls information from “createStores.html” to create a story page. Form submission via post.
- “homePage.html”
- “login.html”
- “logout.html”
- “newStories.html”

- “createStories.html”

*Different stories will have their own unique ID to use in its route.”

Component Relationships:

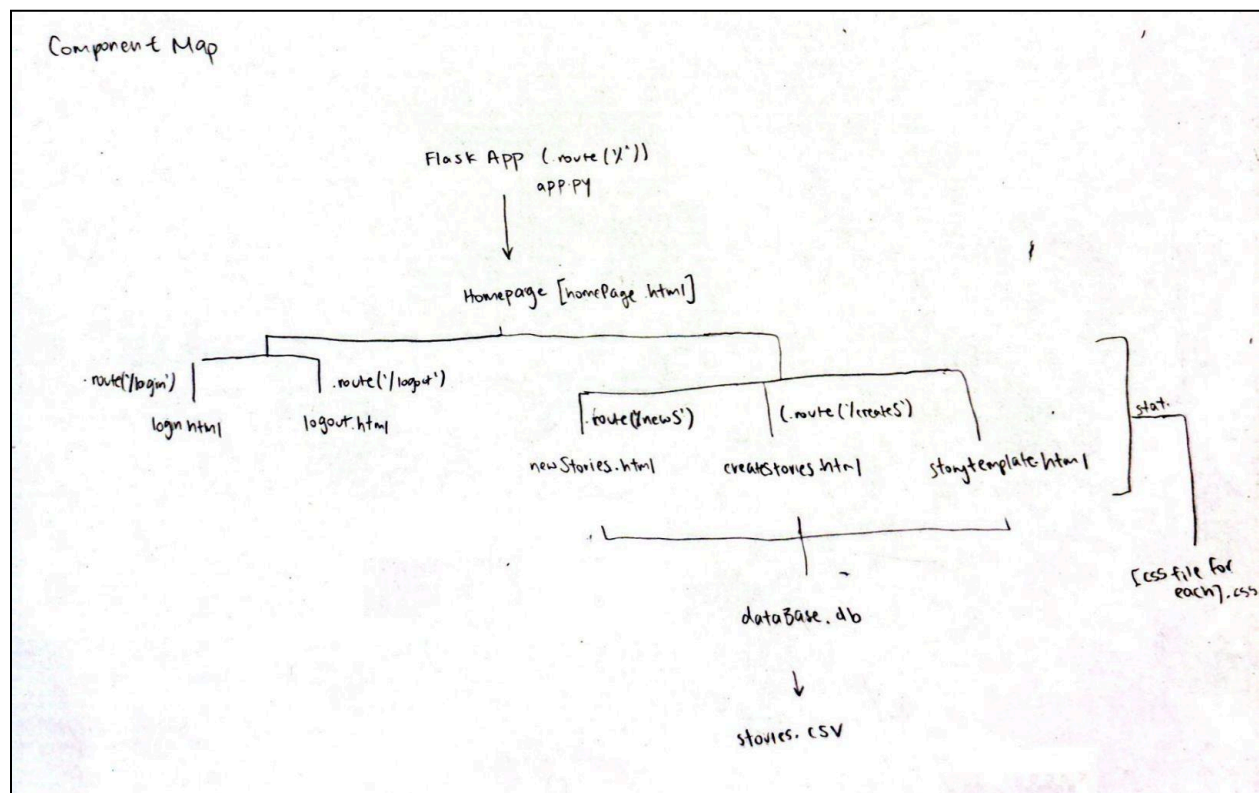
Flask is the central connector.

SQLite3 Database stores all user data, stories, and contributions, which Flask retrieves for display or modification.

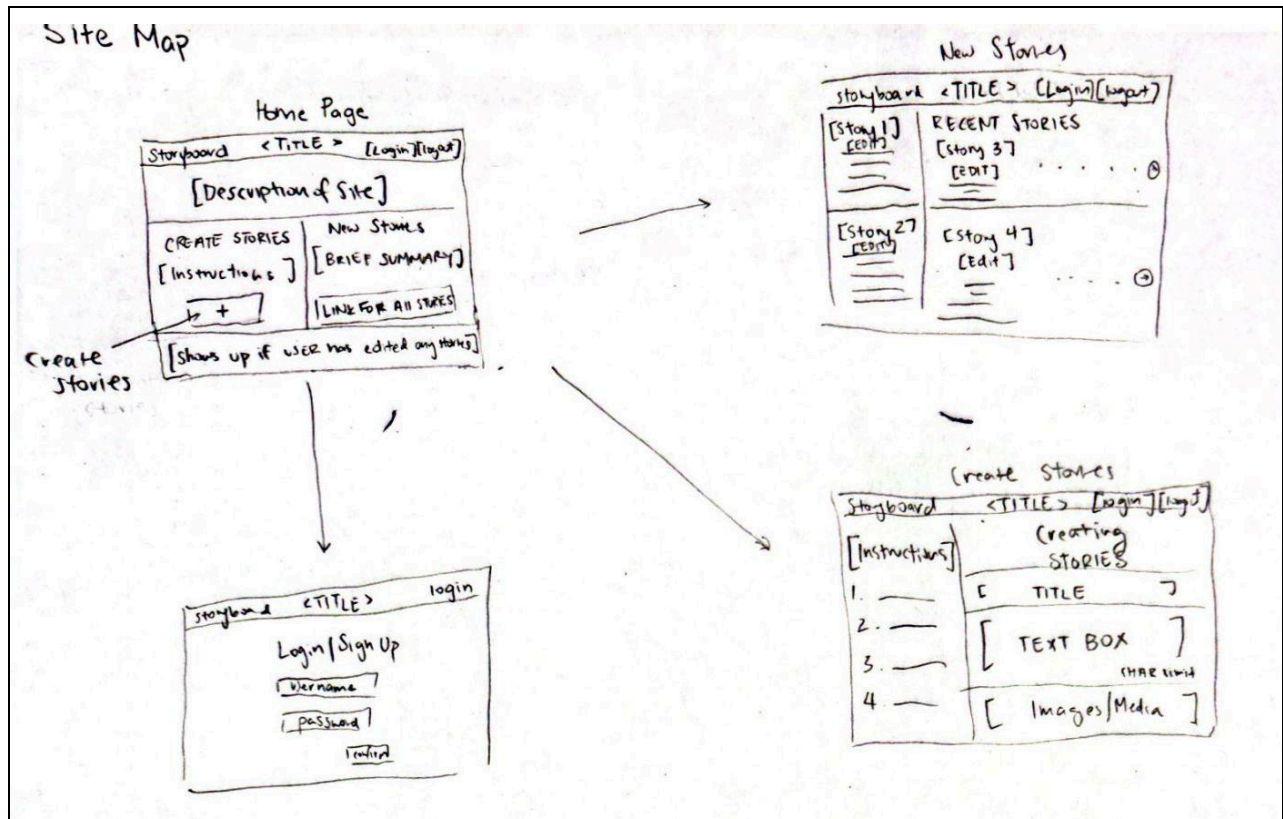
User Registration and Authentication is required before accessing any other feature, including story creation or contribution.

Story Creation and Contribution Modules are dependent on user authentication. Each interacts with the database to either create a new story entry or append text to an existing story.

Front-end Templates render content based on user actions, such as displaying the latest update to a story or showing the user dashboard.



Site Map



Database Organization

Database (tables)

Login(username TEXT, password TEXT)

- Function that adds username and password (stores in database directly)
- Checks if the username exists

Each story will have a table that stores the text and any users that have contributed to the story

- Function that adds username and text
- Checks if they have added to the story
- Let's them make edits (edits will be made on newer versions strictly)
- Checks if the text is below the word limit before adding (through simple if/then logic)

Stories(storyID INTEGER)

- Stores all the stories
- References the individual story tables to display them

The database below is all one database titled "Story" with subtable "User ID" in it.

Main Table

Story	Examples
User ID [who contributed]	[User2, User1, User0]
Text	““Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.”
Story ID	[3218]
Media	[Image1, Video1, ...]
Title	“MY HOUSE CAUGHT ON FIRE”
Timestamp of Last Edit	2024-10-24 8:32PM (UTC) [will always be UTC]

Subtable

UserID	Examples
User ID	User3
Stories Contributed To [VIA ID]	[3218, 3213, 7329, 2317]
Username	“MYHOUSEBURNED2002”
Password	*****

Breakdown of Tasks

- PM duties, README (Description of website/app, Install guide, Launch Codes) - Aditya
- Database SQLite - Kevin
- Flask Server and Backend Python - Ziyad, Aditya
- HTML/CSS - Jayden