

bigpro's p00 Design Doc | TARGET SHIP DATE: 2025-11-10

└ Roster: Carrie Ko, Ethan Saldanha, Owen Zeng (PM), Amy Shrestha

PROJECT: SCENARIO Σ - STORYTELLING WEBSITE

| |
|--------------------------------|
| Legend |
| <i>Italic=Optional</i> |
| Green=Completed |
| Blue=WIP |
| Normal=Not started/explanation |

COMPONENTS

- Sessions
 - login/logout
 - Locally stored list of username, password, *picture link*, and email address for each user
 - DATABASE TABLE FOR USER INFO AND PROFILE:

| USER_PROFILE | | |
|--------------|----------------|-------------|
| TEXT | username | PK (unique) |
| TEXT | password | |
| TEXT | email | PK |
| <i>TEXT</i> | <i>picture</i> | |
| <i>TEXT</i> | <i>intro</i> | |
 - Registration: creates a separate webpage
 - Takes in username, email, password
 - Add to USERS a username, email, and password of their choice, so long as the username and email isn't taken
 - Use sessions to keep a user logged in and only log them out if they press a logout button
 - Login: takes in username or email, & password
 - USER MUST BE LOGGED IN TO USE SITE!!!!

- Webpages
 - Profile (PRIVATE, POST)
 - List of websites the user has created /edited with links
 - Allow editing + creation of pages: have user type in title of webpage, content of webpage, and submit
 - *Allow users to delete profile (perhaps as a link placed at the bottom of the page)*
 - *Needs password as a requirement*
 - User created webpages (editable if logged in, if a non-logged in user presses edit, we redirect them to the sign in page)
 - Entire webpage templates (in HTML) using placeholders/variable
 - Use render_template and variables received through Flask code to input values into a templated webpage (so same format)
 - Content refers to the full content of the webpage, including the recent edit
 - Since the editor is only able to access the most recent edit (as our scenario Σ states), nor can they edit the title and text, we only need previousEdit to save the last edit.
 - DATABASE TABLE FOR WEBPAGES

| STORIES | | |
|---------|--------------|----|
| TEXT | storyTitle | PK |
| TEXT | content | |
| TEXT | previousEdit | |
| TEXT | storyLink | |

- DATABASE TABLE FOR AUTHORS

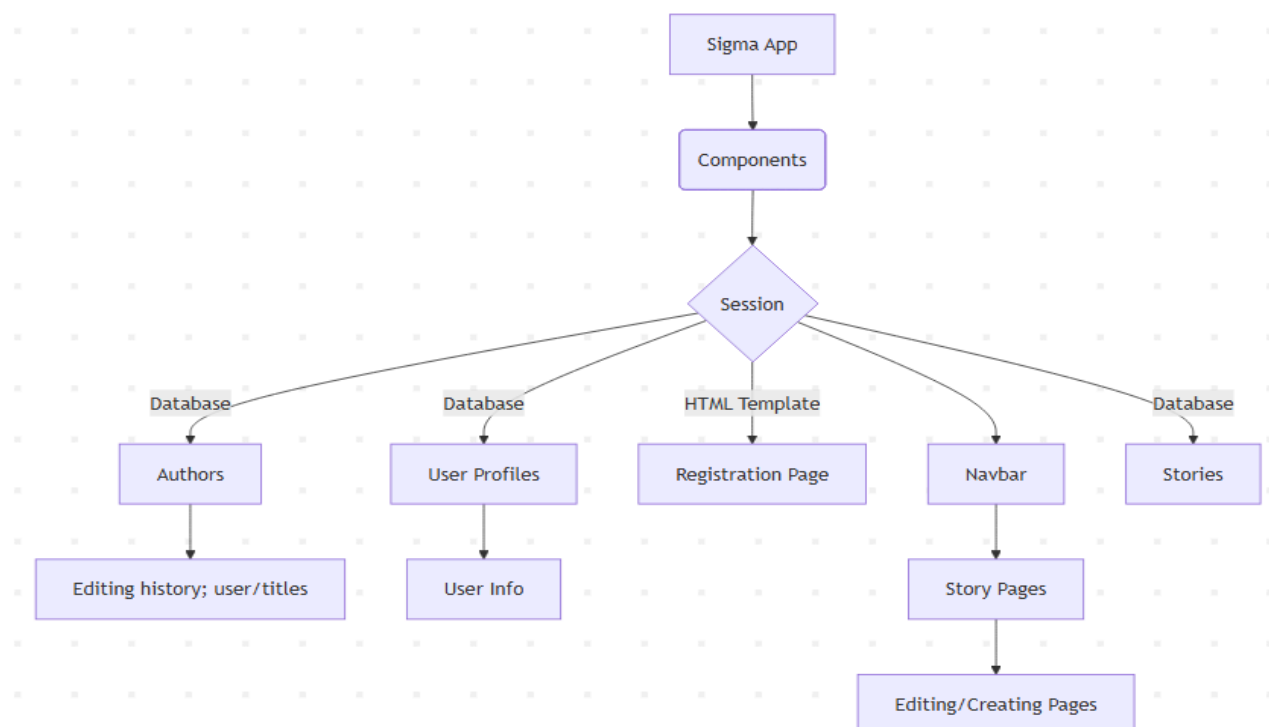
| AUTHORS |
|---------|
|---------|

| | | |
|------|------------|--|
| TEXT | username | |
| TEXT | storyTitle | |

- AUTHORS and STORIES work like STUDENTS and COURSES
 - Use STORIES to access AUTHORS to see who contributed to a webpage
 - Use username to access STORIES to display them on profiles
- Navbar
 - Log in / sign up / logout button
 - Log in / Sign up for if the user isn't logged in, allowing them to either create an account or log in
 - This turns into a logout button once logged in
 - Link to a logged in user's profile page
 - Users not logged in redirected
 - Drop down menu with a link to a page of all stories in alphabetical order, *can jump to a part of the page using links?*
 - *Random page button, Donate button.....*
 - *Search bar?*
- General wiki information (PUBLIC, GET)
 - Includes everything the user types for the webpage
 - Heading
 - Page content
 - Author of page (username)
- Homepage (non-logged in)
 - Description of what the website is
 - Logo (aka team flag :))
 - *Search bar*
 - *Redirect search bar to login page*
 - Footer with our emails and last edited date (maybe)
- Profile page (a user's homepage hahahahahahha)
 - Display:
 - Username
 - Email
 - Link to all stories they've contributed to!!!

- History page
 - Log of all edits on every story; will be accessed by logged in users on the Homepage
 - Format: • username linked to profpage: title of story edited ({commit message})
 - Extra: add timestamp of edit :)
- Database (SQL)
 - Stores all webpages and data
 - Refer to [DATABASE TABLE FOR USER INFO:](#), [DATABASE TABLE FOR WEBPAGES](#), and [DATABASE TABLE FOR AUTHORS](#) for the mock table setup :)
- HTML templates
 - Login/register page
 - Must take user input
 - Logout page (?)
 - User created stories (only need one template)
 - Must take user input
 - Replaces old content with updated content and new update in the same template
 - Homepage
 - Profile page

COMPONENT MAP



The Webstory

└ **Session** acts like a cookie; is the user signed in? To which account?

└ **SQLite3** is required; certain data structures and algorithms will be used to organize and access the databases

└ **Users** and user data will be stored in the **user** database, and authentication may be handled/checked by its own function called upon accessing any page.

└ **Registration** (*possibly account deletion too*) also falls under a similar umbrella. A form will take two inputs, a **username** and **password**, and then check whether the username is taken. If not, it will add it to the table of registered users and the user will be signed in, otherwise it will prompt the user to pick another username. This page will not be accessible if the user is signed in.

└ **Navbar**

└ acts as your standard finder tool, formatted similarly to a directory, with multiple (likely fixed) webpages [home, profile, one or two popular stories] and a search tool.

Said tool will be its own function that combs through the database of **webpages**

└ **The Webpages**

└ nothing too complicated here, a standard display function can work here, that fishes the standard wiki post HTML template and fills in the appropriate data [heading, body, title, optional images etc]. Separate booleans or functions may be needed for the profile and home webpages which may have different formats.

└ **Editing and Creating webpage stories**

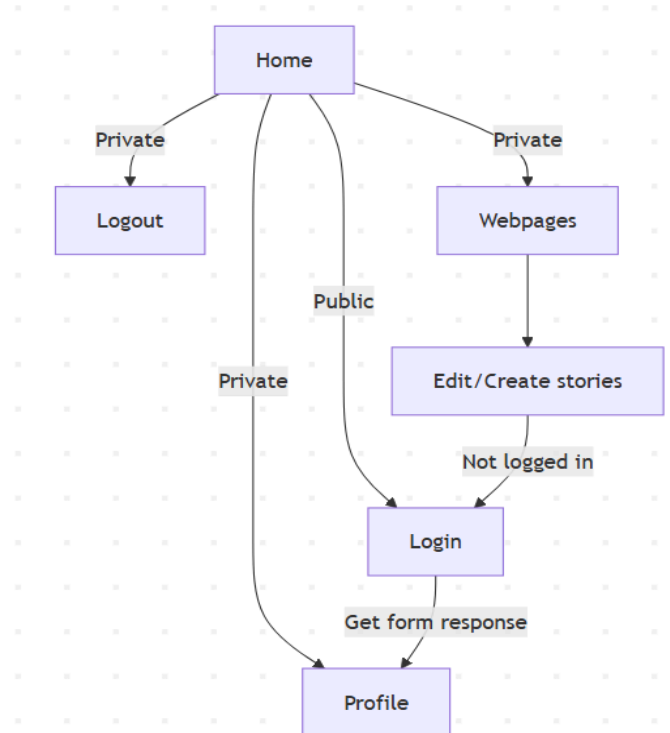
└ If a user is logged in, they may create a webpage through filling out a series of forms that designates its title and information (following a select few inputs like header, body, etc), duplicates not allowed. Following this, it saves the information to the **Webpages** sqlite3 database and refreshes the page. The title may also be saved to an attribute on the **user** database, **webpages_created** so we can display all created/edited webpages on their profile editing a file would be through a form only visible to the user if they haven't edited the file already. If they can see this

|button, they can also only see the last update. Otherwise,
 |they've edited the file already and the whole story is visible
 |while the form is hidden. This is done by an if statement that
 |checks whether the user is in the database of authors or not.

SITE MAP

Home (PUBLIC)

- └ **Login** (PUBLIC)
 - └ redirects to Profile
- └ **Profile** (PRIVATE)
 - └ includes links to all webpages this user created / edited
- └ **Webpage List** (PRIVATE)
 - └ all webpage links sorted by alpha order
 - └ **Create/Edit Webpage** (PRIVATE)
 - └ user inputs webpage title and (new) content for webpage
 - └ redirects to Login if not logged in
- └ **Logout** (PRIVATE)
 - └ redirects to Login or Home



TASKS AND ROLE ASSIGNMENT

| Task | Role | Agent | Deadline | Status |
|-------------------------|---|--------|----------|-------------|
| Database creation (SQL) | Use SQLite3 to initialize tables and values | Carrie | TBD | Not started |
| Flask pathways (Python) | Creating routes for all pages, functions for login/logout/profile, and restrictions | Amy | TBD | Not started |
| HTML template | Build HTML templates for | Carrie | TBD | Not |

| | | | | |
|-----------------|---|-------------------|------------|--------------------|
| | all webpages (home, webpages, profile, login, etc.) | | | started |
| Story functions | Implement function for creating, editing, and getting inputs from user; restrict users that are not logged in | Ethan | TBD | In progress |
| Navbar | Create and link navbar to webpage links | Owen/Ethan | TBD | Not started |
| Profile page | Display necessary information on profile | Owen | TBD | Not started |
| CSS | <i>Creating a style template for webpages</i> | <i>Amy/Carrie</i> | <i>TBD</i> | <i>Not started</i> |

[README.md \(github link\)](#)

First App by bigpro

Roster: Carrie Ko (Back-end/Front-end), Ethan Saldanha (Back-end), Owen Zeng (Project Manager), Amy Shrestha (Front-end)

Description: This website is a wiki, meaning it's a community run encyclopedia for the greater good. Anyone can access its various webpages containing community posts (like a QAF) by searching for a desired topic through a navbar, however users must be registered to create or edit any webpages. Logged in users will have access to a profile page, containing some personal information along with access to any webpages they've created or edited.

Install guide

- 1) Clone the repo into a local directory:
 - a) `git clone https://github.com/Owen-Zeng/bigpro_amy_carrie_ethan_owen.git`
- 2) Enter the app directory:
 - a) `cd bigpro_amy_carrie_ethan_owen/`
- 3) Install necessary modules:
 - a) `pip install -r requirements.txt`

Launch codes

- 1) Run the app through Flask:
 - a) `flask run`
- 2) Open the website at website:
 - a) <http://127.0.0.1:5000>

[requirements.txt](#)

Carrie Ko, Ethan Saldanha, Owen Zeng, Amy Shrestha

bigpro

SoftDev -- P00

2025-10-22

time spent: 0.1

Flask