

Jake Liu, Bogdan Sotnikov, Joyce Lin, Naomi Kurian

Lost and Found

P00: Half-Quick

10/22/25

TARGET SHIP DATE: 2025-11-14

SCENARIO Σ

Program Components

Landing page - search bar, logout button. navbar

Navbar - help user navigate stories

Search bar - search for stories

Login page - log in to access editing and profiles

Registration page - creates a new user

User base - store users

User - store our user's information (username, password, pfp, path to profile page) or basically each row in the user base

Story base - store stories

Story page - displays story information from specified story

Profile pages - display user information

Logout page - log out of session

Edit Page - allows user to edit story contents

METHODS (subject to change):

- **authenticate()**: Checks if session contains a user's rowid
 - If none present, will immediately run redirect() to login
 - Ran automatically in pages: Landing, Story, Author, Edit, Profile (, Logout)
 - Prevents bug where if a user logs out and then tries to go back to a page through url, they can access the site
- **fetch_data(string *TABLE*, int *ROWID*, string *DATA*, string *CRITERIA*)**: Accesses corresponding table and pulls the requested data
 - Will comprise IF statements corresponding to each value and submit a query using those select statements → fetchall() will be ran afterwards
 - If criteria is 'none' then this section will be ignored
- **check_existence(string *TABLE*, int *U_ROWID*, int *S_ROWID*)**: will use fetch_data() to pull either EDITORS or CONTRIBUTIONS and check if str(ROWID) is in split(EDITORS, ',')

Component Relationships

Landing page

- Only visible for logged in users, if not authenticated (no rowid present in session), send straight to login page
 - Can be done by running `authenticate()`
- Contains search bar
 - Utilizes Story base (through a case-insensitive search) to pull up an article
 - Potential feature: matching stories pop up following each letter
- Contains a random list of suggested stories (changes on refresh) that pulls from the story base
 - Displays nothing if empty
- Navbar will contain an 'publish new story' button (leads to author page), logout button and profile button

Login Page

- Asks for username/password and matches it to what is contained in the user base
 - Takes the username and compares it to existing users
 - If username does not exist **OR** if username exists but the password does not match the one attached to the corresponding User, output 'username or password incorrect'
 - If username exists AND password matches, sends the user back to the landing page and adds the username to the session
- Contains a button to create an account (sends to registration page)
 - Once account is created, adds the new user to the user base and signs them in (sending them to landing page and adding the corresponding **ROWID** to the session)

Registration Page

- Looks basically like the login page except there is an extra confirm password form response box.
 - If submit is pressed and it does not pass the comparison check, the page will display a "passwords do not match" message
 - If submit is pressed and the username is the same as another one (case sensitive), the page will display a "username taken" message
 - If submission passes all checks, user will be sent back to the login page and the new user will be added to the user base
- Will include a button (back button or return to login button) that sends the user back to the login page

Profile Page

- When go to profile button is clicked, use session rowid to pull up corresponding user and sends them to the path stored in the user (profile page)

- We could also cut this part out potentially by using a default format for the path and plugging the rowid in at the end (rowids are unique identifiers)
- Displays pfp, 'hi, [username]!', associated stories, a back button (leads to landing page), and a logout button (leads to logout page) —> both buttons located in a navbar
 - Stories will be pulled from the Story base by checking if user is present as an editor (stored in Story base as string of rowids) or author for any of the stories, and the link will be included (redirect() with the story's path)
 - Potential feature: two separate lists for authored stories and edited stories
- Potential feature: should include a button or something that allows you to choose from a curated list of profile pictures (while the current one is randomly selected when the account is first created)

Story Page

- Only visible for logged in users
- Displays the title of the article at the very top, most recent entry below (unless already edited), and then the editors and authors at the very bottom (structure may change based on whatever's aesthetically pleasing)
- Navbar contains a search bar, a logout button (leading to logout page), a back button (leading to landing page), profile button, and an edit button (goes to edit page) that is only visible if the user hasn't edited yet.
- Potential feature: have a "next page" button that essentially moves the story forwards (current page number can be determined using a variable and content is displayed through pulling with indexes from a list? → split from \n or some random character that DEFINITELY wouldn't be used)
 - AFTER checking that the user has already edited the story

Edit Page

- Through the identity of current story (using the rowid of the story's path to select from story base), send the most recent content of the story (last_entry) that is being edited to be displayed in edit page
 - Have the edit page display form response box below the last entry
- Page will contain a submit button
 - New section will be stored in Story base (last_entry)
 - Will send user to corresponding story with changes published and the user can now see the entirety of the story
 - Once you go back, the user currently in session will be saved to the editor string of the corresponding story (cannot edit again once added to this string)
- In the navbar, a back button will exist that'll send you back to the article you were originally on (changed are unsaved and no further actions will be taken)

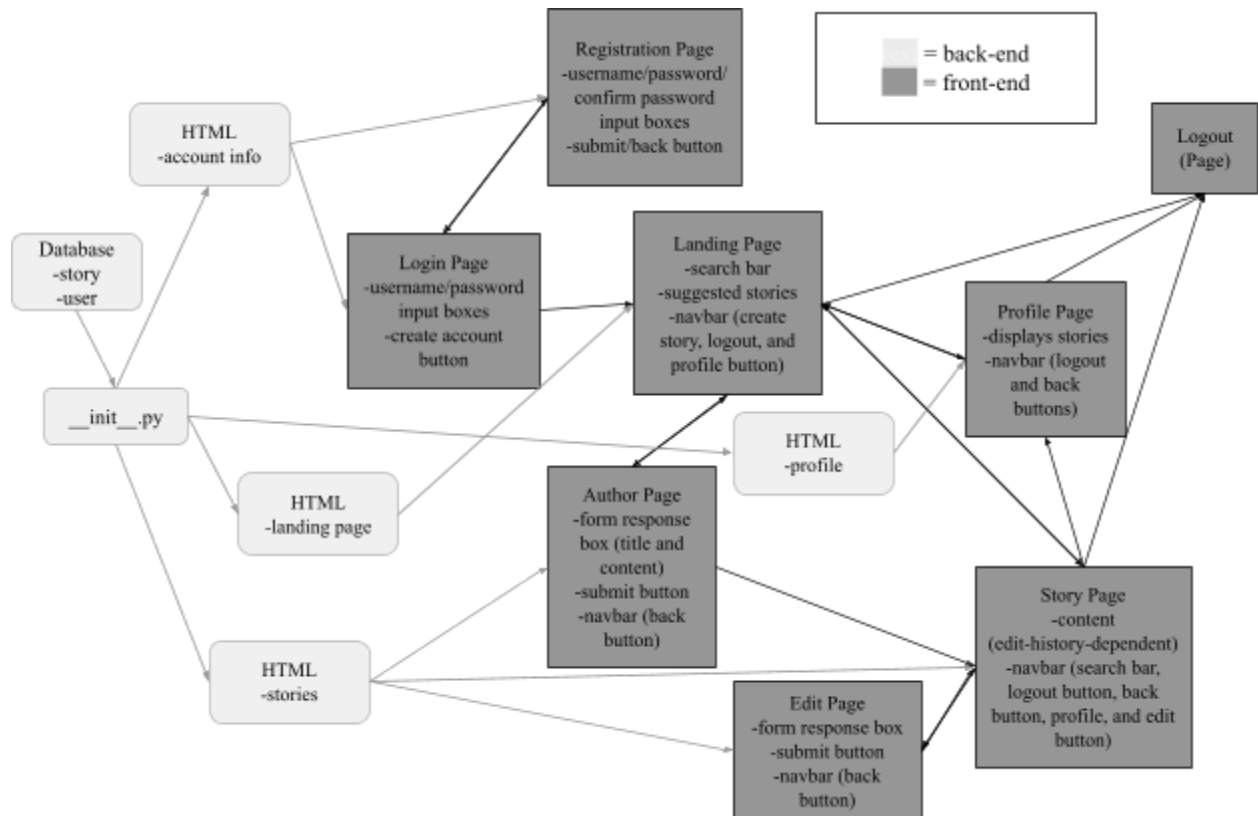
Author Page

- Works similarly to edit page except there are two empty form response boxes (one for title and one for content) → this should be made clear on the page
- Submission (using the submit button) will send you to the newly published story
 - The new story will then be added to the story base and path will be created with the first submission being saved as both content and last_entry
 - Editors will be empty while author will save the authoring user's rowid
- In the navbar, only a back button will exist that'll send you back to the landing page (changed are unsaved)

Potential feature: Logout Page (alternatively, button redirects to login page and removes user rowid from session)

- Removes user rowid from session and contains a button that sends you to the landing page
 - Include a goodbye message

Component Map



Database Organization

sqlite tables will contain the following information:

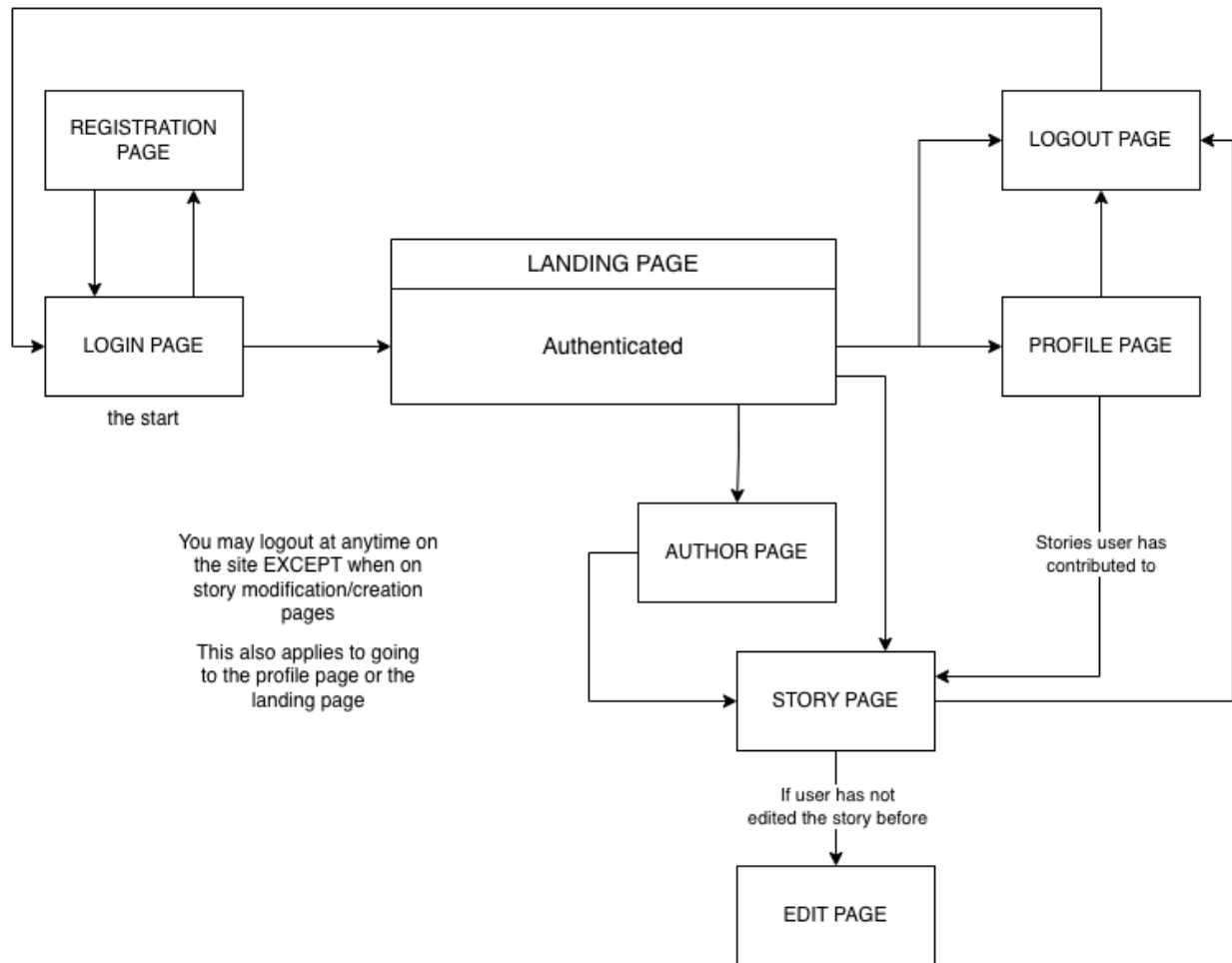
***path will store “[USER/STORY]/rowid”*

Default SQLite PK is used (rowid)

<u>USER BASE</u>		
TEXT	username	
TEXT	password	
TEXT	pfp	
TEXT	path	
TEXT	contributions	Contains story rowid
INTEGER	times_cont	

<u>STORY BASE</u>		
TEXT	path	
TEXT	title	
TEXT	content	
TEXT	last_entry	
TEXT	editors	Contains user rowid
INTEGER	author	Contains user rowid

Site Map



Task Breakdown

Team member	Tasks	Expected Delivery
Jake Liu	<input type="checkbox"/> Story Page <input type="checkbox"/> Edit Page	- TBD - TBD
Bogdan Sotnikov	<input type="checkbox"/> Landing Page <input type="checkbox"/> Author Page	- TBD - TBD
Joyce Lin	<input type="checkbox"/> Profile Page <input type="checkbox"/> Flask Routing (<code>__init__.py</code>)	- TBD - 10/29
Naomi Kurian	<input type="checkbox"/> Login Page <input type="checkbox"/> Logout Page <input type="checkbox"/> Registration Page	- TBD - TBD - TBD