Scuba-Doo Dog Erasers -- Karen Shekyan, Gabriel Thompson, Russell Goychayev
Softdev
P00: Move Slowly and Fix Things
2022-10-31
time spent: 1.0 hrs

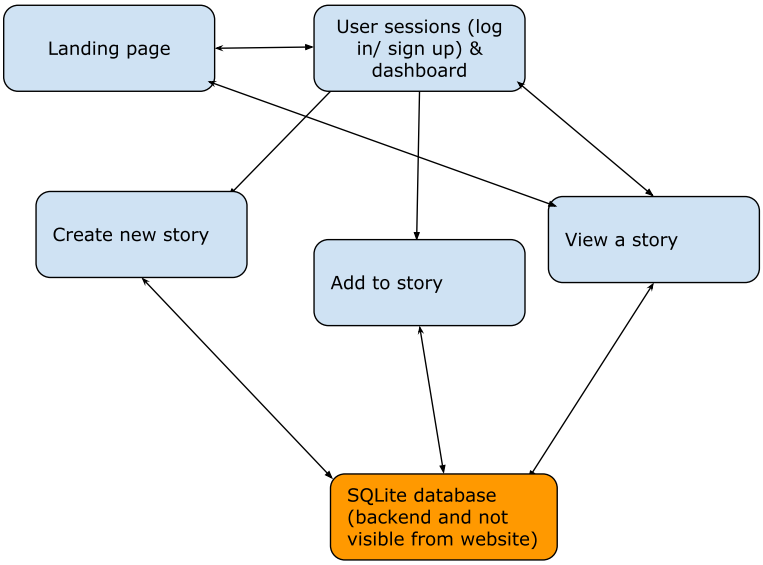
Components:

- User accounts
    - User sessions
    - A log-in and log-out system
- A route that allows users to create new stories (this is a form)
- A route that allows users to add to stories
- A SQLite database that stores the ID of the story, its title, its corresponding latest update, and a list of all the usernames who have edited it
- A route to serve the contents of any story
- A landing page with a button allowing you to log-in to the website and sign-up for website
- A dashboard for the user when they are logged in
- Templates for:
    - Landing page
    - Story-reading page
    - A user dashboard
    - A form to write updates to a story

Connections between components:

- User accounts allow users to log in and log out as well as sign in and sign up. It relates to other components because editing and creating articles are functions of users with accounts.
- A route that allows users to create new stories relates to other components because only users can create new stories.
- A route that allows users to add to existing stories relates to other components because only users can create new stories.
- A SQLite database that stores story-related information relates to other components because it allows for efficient storage of information and also acts as a middleman for information exchange between components.
- A route to serve the contents of any story relates to other components because it allows any internet user to see a story.
- A landing page with a button allowing you to log in to the website and also sign up relates to other components because it acts like the door to the house. Without it, a visitor to the website would not know how to use this website or "enter the house".
- A dashboard for the logged-in user acts like a guide on a journey. It creates a better user experience and allows for a seamless transition between the use of various front-end components.
- Templates relate to other components because it allows for more efficient and more readable code. It is like steroids for a bodybuilder.

Component Map:

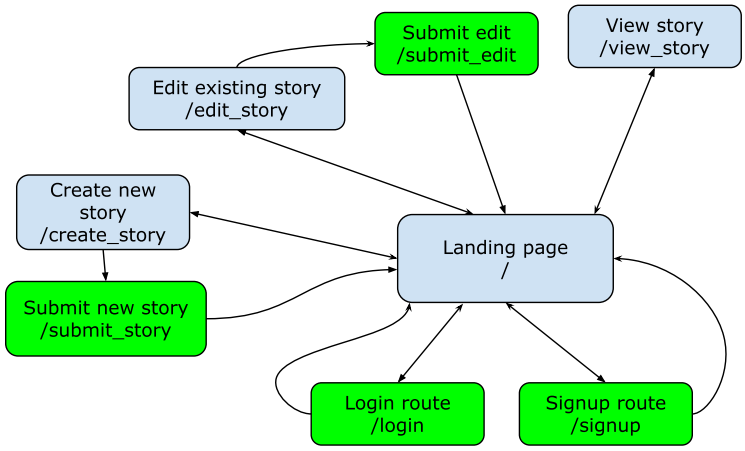


An internet user arrives on the landing page. An internet user can view a story. In order to create or add to a story, an internet user must create a user session. Story creation, modification, and viewing are made possible by retrieving the necessary data from our SQLite database.

Database Organization;

- 3 tables:
  - User info
    `create table user_info(username text, password text, stories_ids text);`
    - This table will be used when logging a user in, to verify that their username matches their password
    - `username` is unique, so we can match edits to their creator and there's no namespace conflicts
    - `stories_ids` stores the ids of all the stories the user edited
  - Pages
    `create table pages(story_id int, title text, content text, edit_ids text);`
    - `story_id` is unique
    - `title` is the title of the article (not necessarily unique)
    - `content` is the html content of the body of the article
    - `edit_ids` is a comma-separated list of edit ids for all of the edits made to the article. For example, if edits #200, #564, and #1337 are the only three edits to the article, then the content of the `edit_ids` would be "200,564,1337"

Site Map:



Key:

- The format for each box is the title of the page, followed by its route
- Two-way directional arrows mean that two pages are accessible from one another
- One-way arrows mean that one web page redirects to another
- Blue boxes mean that this route is accessible via GET request
- Green boxes mean that this route is accessible via POST request
- **Important thing to keep in mind:** Every GET-accessible page has a link back to the root of the webpage (there is a logo in the upper-left corner than guides the user back to the start)

Description of each page:

- / — If you're logged in, then this is the dashboard. If you're not logged in, then this is the landing page. It contains a logo and the login and signup forms. Contains a search box where you can type in the name of every article, and the server will return the content of the article via /view_page. Contains a "Create Story" button.
- /login — The login route. This is the route that the login form on the landing page sends the information to
- /signup — The signup route. This is the route that the signup form on the landing page sends the information to
- /view_story — Contains the content of any requested story (the title of the page is specified within the URL arguments).
- /create_story — This returns a form allowing a user to enter the title and content of a new article they want to create
- /submit_story — This creates the new article from the parameters provided from the form on /create_story that sent the user to this route
- /edit_story — Contains a form that allows you to type in the new contents of the story you're editing. When you press the submit button on the form, the information you entered is sent to…
- /submit_edit — Updates the database with the new story content

List of templates we'll need:

- landing_page.html
- dashboard.html
- view_story.html
- edit_story.html
- create_story.html

Assignments of each task to each group member:

- Gabriel:
  - A route that allows users to create a new story
  - A route to serve the contents of any story & a template for this route
  - A route that allows users to add to a story (form)
- Karen:
  - SQLite database that stores the ID of the story, its title, its corresponding latest update, and a list of all the usernames who have edited it
  - A dashboard for the user when they are logged in & a template for the dashboard
- Russell:
  - User accounts & sessions
  - A landing page with a button allowing you to log-in to the website and sign-up for website & template for landing route