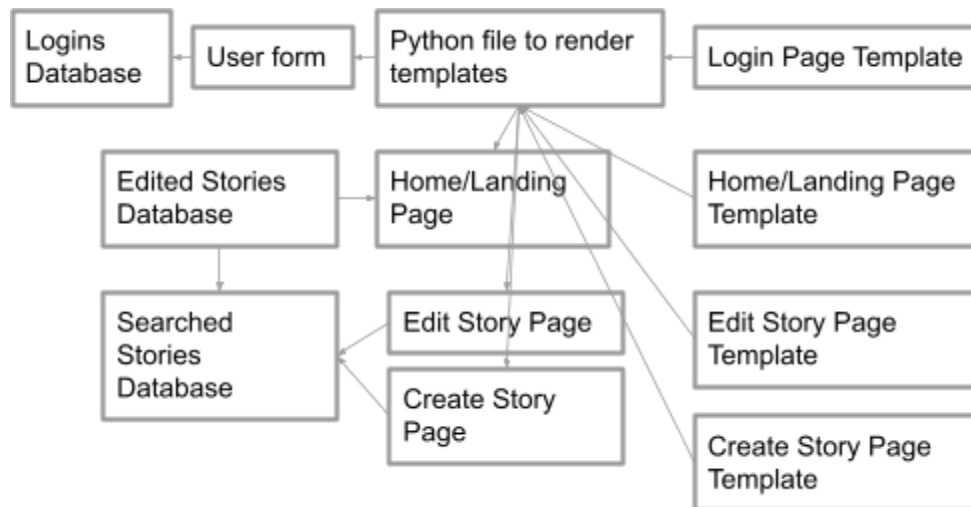Flying Turtles | Nicholas Tarsis, Anson Wong
Softdev
P00 – Scenario 1 Design Doc
2022-11-02
Time spent: 0.5
Target ship date: 2022-11-14

Program Components
- Registration/Login System
  - Provides a prompt that is rendered by an html template that asks for a user's username and password in order to log in.
  - If a user has not made an account, use the same html template as before but now add that user's info to an SQL database.
  - After logging in, take the user to a homepage with all of their stories.
  - If a user is already in a database, return another html template saying that the user did not provide a valid password (optional)
- Story Editing
  - In order to edit a story, the user must have the ability to easily look up an existing story. Thus, an additional database is needed to store the stories and a template search bar can be used to find stories in that database. This search bar should be used in the landing page for the user.
  - When editing a story, the user must only be able to see the most recent change, so another database can be used to store existing changes. So when a user is editing, only the change from the last row will be visible. Make sure that the user is not able to remove rows from the database, i.e. remove prior pieces of some story.
- Story Creation
  - Render an html template that has two boxes, one for the title and one for the actual text being added as the beginning of the new story.
  - When the user submits a new story, add information about the story to another database so that logged-in users can more easily view their prior stories.
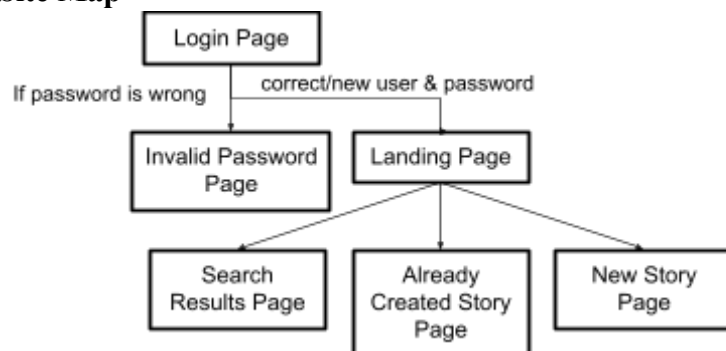
Component Relationships
The python file used to render templates renders the first template (for homepage) at route "/", displaying an alphabetized list of hyperlinks to different stories made by the user as well as a search bar (form). The search bar will use sqlite3 to SELECT and fetch a record WHERE the query is found. These records will be returned and displayed when rendering. Note that these records apply to the database of all stories on the webpage, not the database with the stories made by the user. Different HTML templates should be used depending on if the user is creating a story or just contributing to a previously made story.

Logins Database → User form → Python file to render templates ← Login Page Template

Edited Stories Database — Home/Landing Page — Home/Landing Page Template

Searched Stories Database — Edit Story Page — Edit Story Page Template

Create Story Page — Create Story Page Template

Database Organization

We plan to include four different databases. The first database contains the login information for every user, such as the username and password. The second database stores the respective information of every story on the web page to allow for the search bar to function properly, such as the title and current text used in the story. The third database is meant to keep track of a user's changes to another story, such as the last edit made to the story and also the new addition made by the user. The fourth database will be used to maintain the title and starting text of a story created by the user.

Site Map

Login Page
— If password is wrong → Invalid Password Page
— correct/new user & password → Landing Page
Landing Page → Search Results Page, Already Created Story Page, New Story Page

Task Breakdown

Nicholas: Project Manager, Rendering templates for homepage and adding/editing information, creating SQL databases for creating and editing stories. keeping track of README.md, optional CSS modifications to templates

Anson: Creating SQL databases for login page and search bar, templates for search results and individual story pages, optional template that warns user of repeating username/password