

**Gastric Bypass Train - Sam Cowan, Anna Fang, Sadi Nirloy**

**Poo: Half Quick**

**SoftDev**

**27-10-2022**

**Time Spent: 1 hr**

**Target ship date: 2022-11-8**

### **Design Plans:**

*Homepage:*

- HTML Table with all of the different pages
  - Each entry in the table consists of buttons that input an integer, which is a unique ID number that is used to get the row of a table from a database.

*Page Creation:*

- A pair of textboxes, one for a title, the other for text

*Loaded Page:*

- A way to return to the home page
- A button that leads to the editing page, with the title and text in appropriate parts

*Database:*

- One table that holds all of the pages

Column1: ID Primary Key ("page" number)

Column2: Title

Column3: Text

*Python:*

- Handling the creation of the homepage

- Contains the table with buttons to each page. See Homepage for more info.
- Using sqlite to load pages from the db and stores them in the db

HTML:

- Template for the page where you create a wiki article
  - Like a form
    - Contains two fields – one for the title of the page (text) and one for an information paragraph (text)

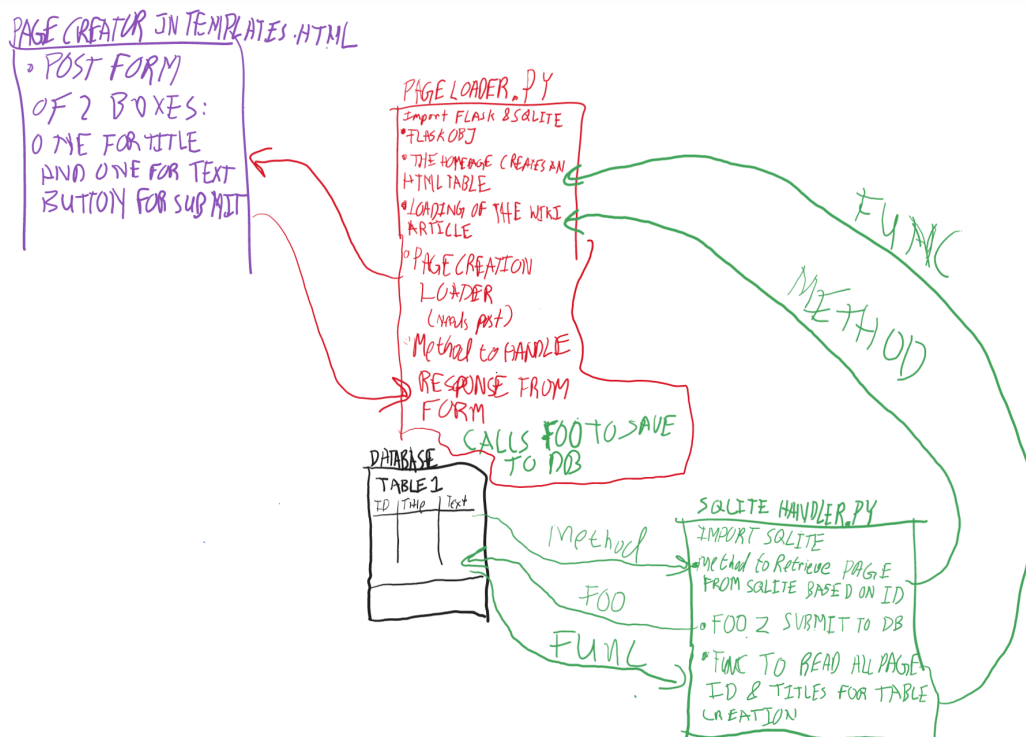
Sqlite database (which is dynamically added to by flask app):

- Contains a table with all (user) inputted elements as text (title and paragraph)

### Explanation of how each component relates to the others.

- The Python app handles post requests when the user adds a new entry to a page and calls another python file which works with sqlite and adds a new row to a table in the database. This table holds the id number, the title, and the text of the page.
- HTML templates for making an HTML table from data in sqlite database. Python app will call on these. One for displaying homepage (because of the links) and another for displaying sub-pages

### Component map

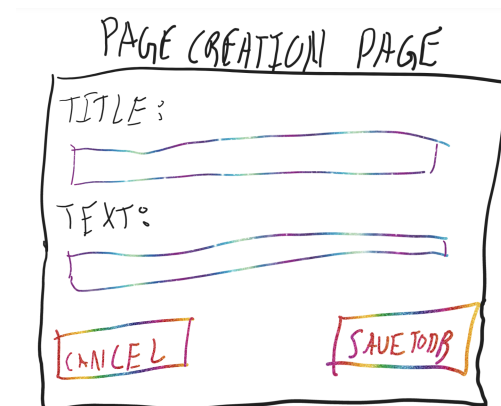
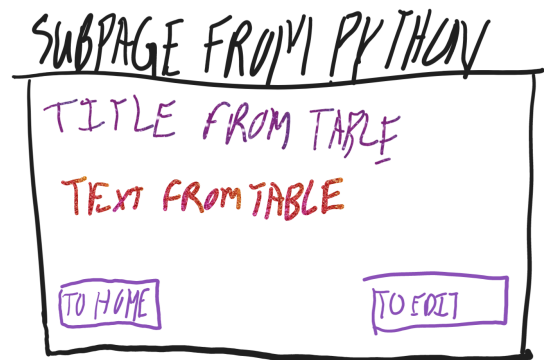


## Database Organization

- Main database storing all pages and their information (in a table)
  - Enumerates the “page number”/location when displayed
  - Contains title name
  - Contains all paragraph text
- Submissions from user inputs added to the database through python file

## Site map

Title	Link/writeup
Title	Link/writeup
Title	Link/writeup
Title	Link/writeup
Title	Link/writeup



## Task Breakdown

MVP:

- Homepage with a table containing title and text for each entry
- flask app which hosts and allows users to add new entries.
- No ex post-facto editing

*Expectation:*

- Homepage with a table of titles and links to different pages.
- Each page has a button to take the user back to the homepage.
- Each page has its own table with entries of title and text.
- Users can add new entries, both as pages within the homepage or entries within pages, and can edit any table.

*Stretch Goal:*

- Configure user login.
  - Using the database, previous wiki entries are saved and can be re-accessed when the user logs in with their username.
- Pages within pages within pages (infinitely) – not fleshed out entirely at this point; it may not even be possible!

## Task Assignments

*MVP:*

- Flask app – **Sadi**
- HTML templates – **Anna**
- Sqlite-managing Python files – **Sam**
  - Making tables editable
- Return button – **Anna**

*Stretch:*

- Login stuff – **Sadi**
- Recursive pages – **TBD**