

# *Team Greedy Toads*

*Helena Williams, Anya Zorin, Arib Chowdhury, Benjamin Gallai*



p0 Proposal Document

# Components

## Editing Entries

Each entry has the option to be edited, and all contributors including the original author and the editors can be listed at the bottom. For the edit entry the action that occurs when you click the edit button would create a text input box with the value of the already present text. After the editor is done editing and also inputs their name into the entry box, the data will be updated. If the editor's name is already listed for that subtopic, it will not be appended to the list. The text will always be updated.

## Creating/ Deleting Entries

An entry can be deleted or created. Deleting an entry would simply remove it from the data table, but it may be better to simply mark it as not displayed so that we are not actually deleting it and it can be retrieved if needed. Creating an entry would be a form in which the person would give the entry name, and the text in the entry. This would be a separate page accessible through the nav bar and once completed and submitted, the user is redirected to the newly made page, and if deleted the person is given a message saying page successfully deleted.

## Displaying Entries

The home page will list all the topics, one of them being "other", and once you follow the list of links, the user is redirected to a list of subtopics for the topic, which all lead to separate entries that you can edit or delete.

## Database (and Database Organization)

The database we are using is SQLite, which will be locally stored on the digital ocean server. There will be one table for the entries. Each entry has an id, a title, a topic name, a list of contributors/authors, the date/time created or edited, deleted or not, and then the entry itself. If we were to add users later on

## Navbar

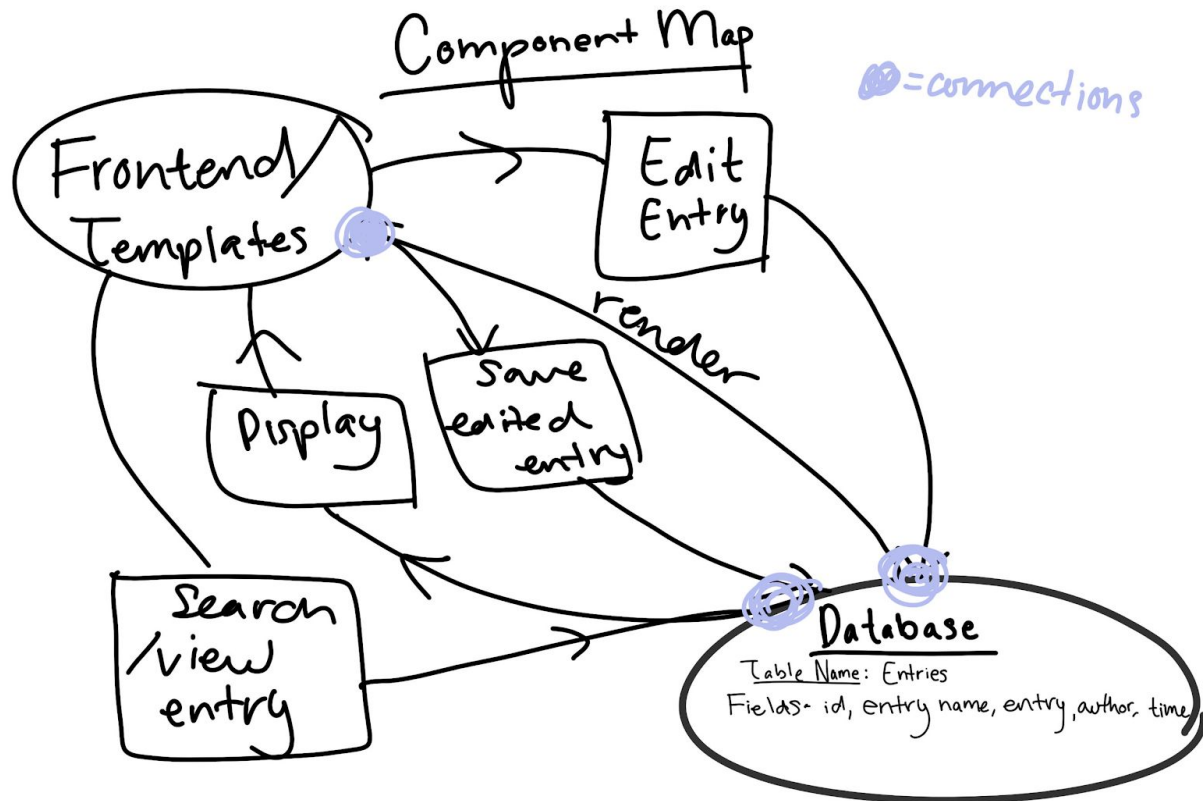
The navbar is accessible from any page and includes navigation to the homepage, a create new entry button, and a search bar to search the existing entries. The search bar connects to the database to get a list of the entry names to search through so that it updates with the database.

## (Extra Component if there is time) Authentication

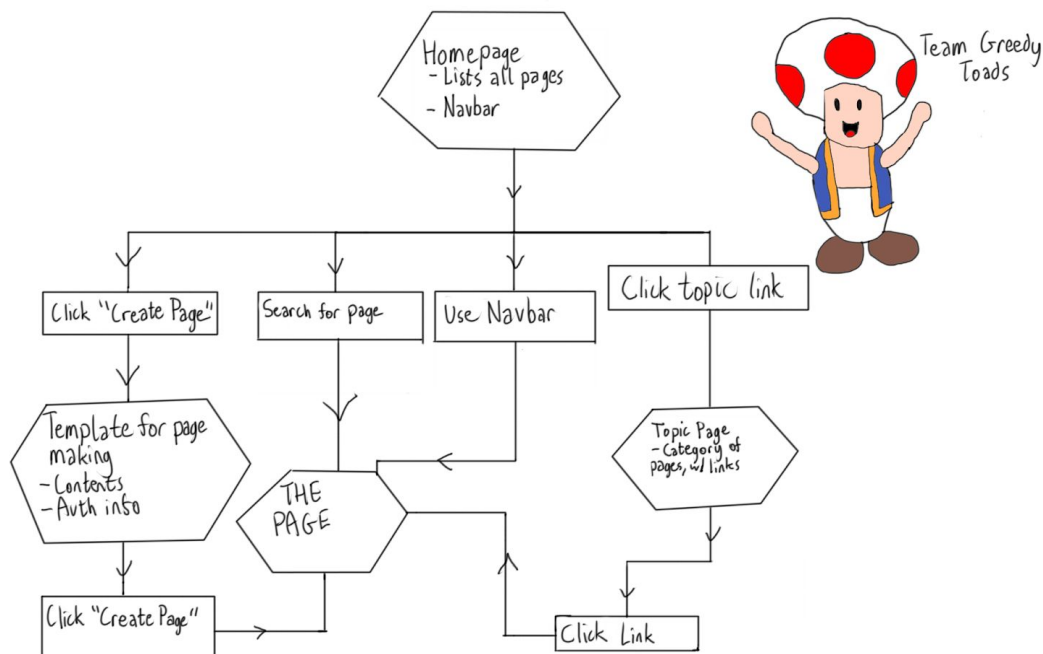
Another table containing the users information would be added to the database. It would have their id, username, email, and password. The entry table would have an array holding the user

ids instead of contributor names directly. It would also be possible to add a page that shows the user all the entries they have created or contributed to.

# Component Map



# Site Map



## Tasks

### Frontend - Helena

Html files using jinja for forms and such. Css added later and must be functional without it. Will be filled with my AMAZING drawings because they're my best talent.

- Edit mode button should have a name given to it with jinja so that when the editing action happens it has something to pass to indicate the title of the entry.

### Rendering Entries - Benjamin

Function to load the entries. Each entry is loaded on the same html file. You always get to the entry by clicking a button/submit with the value being the same as the title of the entry. The function should be inputted this title, then use the title to access the correct row of the database and send all the retrieved information to the template.

## Creating Entries - Arib

This function creates new entries. The user inputs are: contributor name, entry title, and entry. This will be used to create a new row in the database. The date/time will also be added to it, and the id is automatically generated. After the new entry is created, it is then rendered using the same method as rendering entries. They will also create separate functions that create a new row.

## Search Bar - Anya

Retrieve a list of the entry names and (using jinja) put them in a searchable dropdown

## Editing Entries - Benjamin

This requires two functions. The first one enters “edit mode” changes the displayed text to a text box input with the value of the same as the entry (taken from db). It also creates another input box for the contributor name.

The second function is triggered by submit, and if the entry is edited (value is different than the one stored in db) then the db is updated with that entry, and the contributor list is updated, as well as the time/date. Then the newly created entry is rendered.

## Delete Entry + Basic Functions for Rendering/Editing- Anya

Delete entry function simply removes it from the database using the title.

Bas functions are ones used in the other functions, such as retrieving the entry based off title as well as other functions that interact with databases. By creating them separately it ensures uniformity in theses actions, as well as making it easier to find bugs, and making other functions cleaner and easier to understand.

- `getEntryDB(title)`
- `getContribsDB(title)`
- `updateEntryDB(title, entry, contrib)`