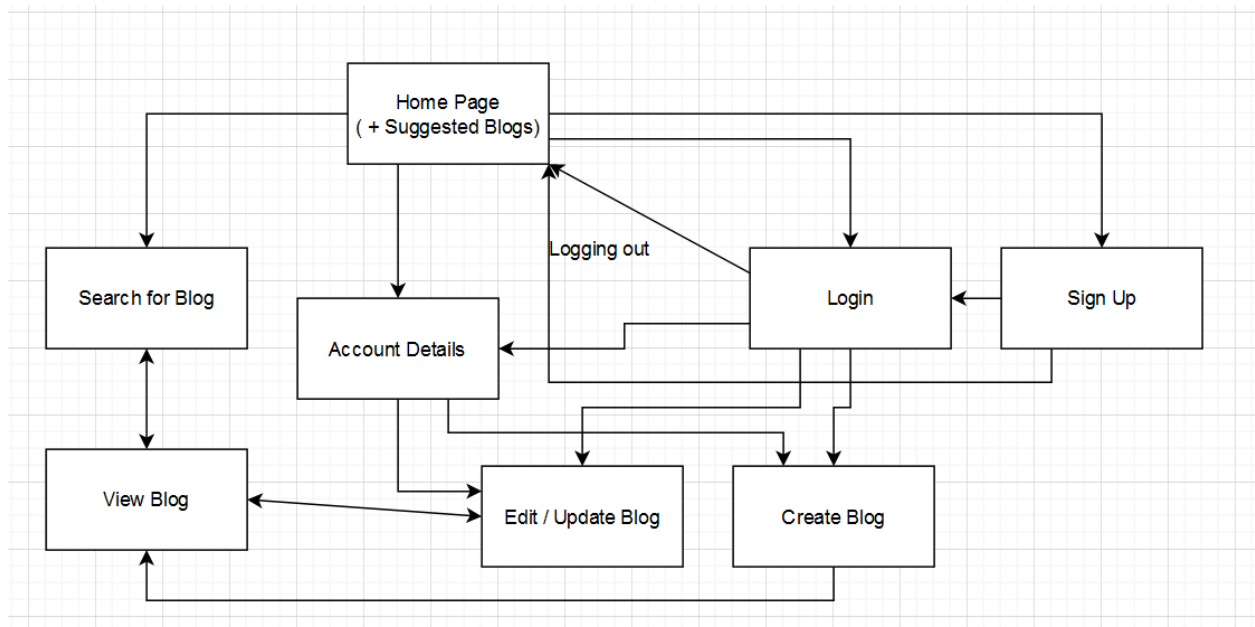# Wiki Project Design Document

Team The Best Cameron -- Cameron Nelson, Jonathan Wu, Alif Abdullah, Kevin Cao

File Tree:

```
flask_app
├── app.py
├── blogs.db
├── templates/
│     ├── layout.html
│     ├── home.html
│     ├── login.html
│     ├── view.html
│     ├── create.html
├── static/
│     ├── css/
│     │     ├── style.css
├── blogs/
│     ├── blog_1.txt
│     ├── blog_2.txt
│     ├── blog_3.txt
```

Front-End Sitemap

Program Component List:
- Python

- ○ The python app would be used for the creation of the database itself and for inserting data / pulling data from it.
- ● SQLite
  - ○ We use SQlite to interact with our database. The commands used in the python app would come from this component.
- ● Jinja
  - ○ Jinja takes the data from the backend for the wiki contents, page title, and other bits and puts them where we want in our webpage.
- ● Flask
  - ○ Flask serves the actual web pages jinja will be rendering (does network stuff).
  - ○ Session allows for the site to allow a logged in user to edit existing pages and add new ones. Someone cannot edit/create a page without being logged in / having a session.
- ● CSS
  - ○ CSS makes our site look pretty to the user on the frontend.

Database Organization
- Tables:
  1. **BLOGS**
     a. Stores the number pointing to where the contents of the page can be retrieved (if ROWID the blog is stored in blog_1.txt), the name of the blog, and the author.
     b. 4 columns, ROWID, NAME, and AUTHOR, and BID.
  2. **USERS**
     a. Stores user data
     b. Holds username, hashed password + ROWID

Member Assignments

- ● Cameron: Working on backend stuff and database.
- ● Alif: Using jinja/flask/python to connect backend to frontend stuff.
- ● Jonathan: Working to create the frontend HTML templates with Kevin and css for later.
- ● Kevin: Also working to create frontend HTML templates and then css for later.