Bourbon Chocolate - Blog Design Document

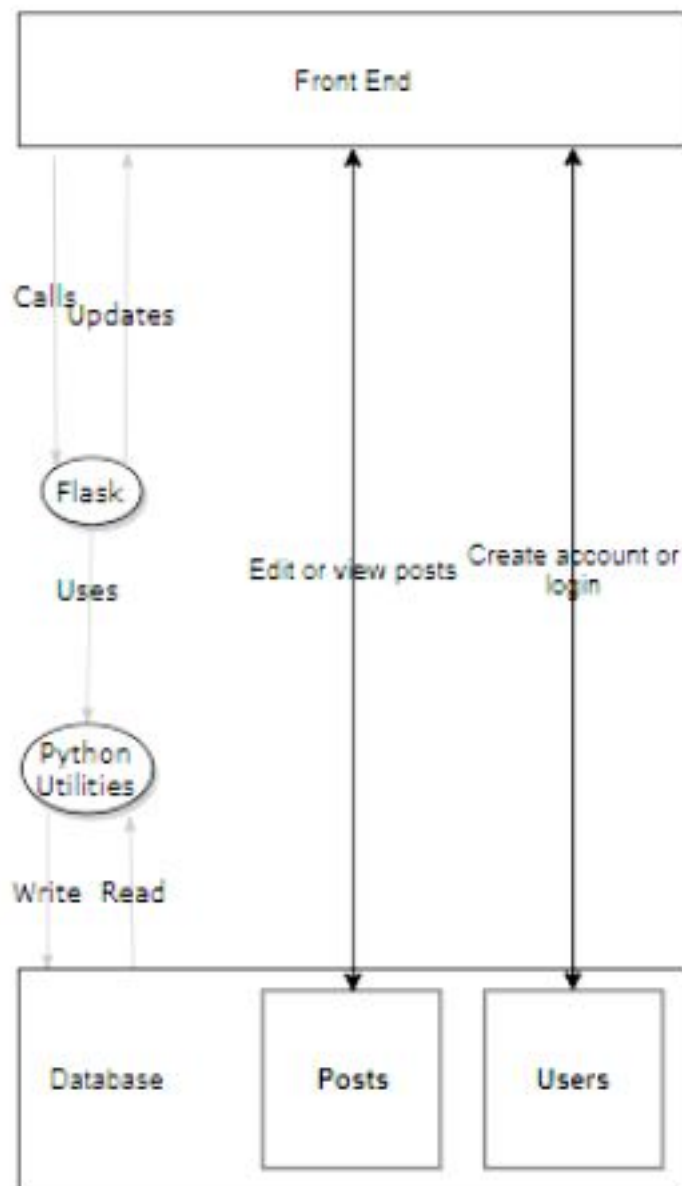Created by: Daniel Keriazis, Karen Li, Vincent Lin, Rachel Ng

To-Do

- Make to-do list
  - Add core features
  - Add stretch goals
  - Divvy up tasks
- Core features (in priority order)
  - Displaying posts
  - Adding posts
  - Editing of posts
  - Homepage
  - Account creation
    - Hashing and Salting
- Stretch goals (in priority order)
  - Search for specific blog (name, title, tag)
    - Also include support for tags
  - Page formatting using markdown
  - Voting on posts

Roles

Date	Component	Person
10/16/18	Roadmap	Everyone
10/16/18	To-Do	Everyone
10/16/18	Assignment Details	Everyone
10/21/18	**Front End** Home page, blog page, edit form, account creation, log-in	Karen
10/21/18	Flask routing	Vincent
10/21/18	Post management	Rachel
10/21/18	Account management	Daniel

Component Map

Front End

Calls
Updates

Flask

Edit or view posts   Create account or
login

Uses

Python
Utilities

Write Read

Database   Posts   Users

Backend

- Database
  - `posts (id PRIMARY KEY TEXT, title TEXT, content TEXT, author TEXT)`
    - `id` is the unique internal key for the blog post
    - `title` is the title of the blog post
    - `url` is the url-friendly version (i.e. no illegal characters) of `title` and where the page should be located
    - `content` is the content of the page
  - `users (username PRIMARY KEY TEXT, pass_hash BLOB, salt BLOB, karma INT)`
    - `username` is the unique name of the user
    - `pass_hash` is the hash-and-salted password for the user
    - `salt` is the salt used in `pass_hash`
    - `karma` is an integer representing the karma level of the user (the upvotes on the users' blog posts minus the downvotes)
  - `tags (id TEXT, tag TEXT)`
    - **Note: This is a stretch goal**
    - `id` is the id of the blog post with tags
    - `tag` is the tag for the blog post
  - `votes (id TEXT, username TEXT, value INT)`
    - **Note: This is a stretch goal**
    - `id` is the id of the blog post with votes
    - `username` is the username of the voter
    - `value` is the vote the user `username` gave to the blog post (can only be -1 for a downvote or 1 for an upvote)

posts	
PK	id TEXT
	title TEXT
	content TEXT
	author TEXT

users	
PK	username TEXT
	pass_hash BLOB
	salt BLOB
	karma INT

tags	
	id TEXT
	tag TEXT

votes	
	id TEXT
	username TEXT
	value INT

- Flask (For python functions related to page management)
- Util (For utilities to be used by the app)
  - `create_page(id)`
    - Create page `id` if it doesn't exist
  - `get_page(id)`
    - Return contents of page `id`, or `None` if it doesn't exist
  - `delete_page(name)`
    - Deletes page `name` if it exists and the user has the proper access level
  - `edit_page(id)`
    - Edit text in blog entry
  - `hash_pass(password, salt)`
    - Get a hash of `password` with salt `salt`
  - `add_user(user, password)`
    - Create a new user with username `user` and password `password` if it doesn't exist
  - `auth_user(user, password)`
    - Return `True` if user `user` exists with password `password`, otherwise `False`

Site Map
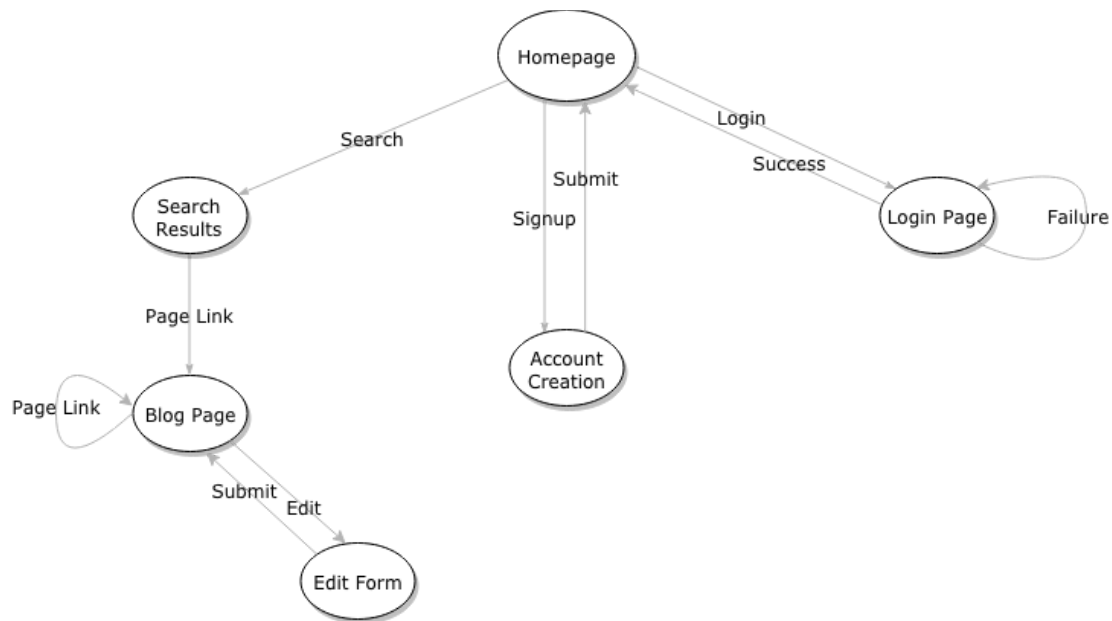
Front End
- Homepage
  - Template: `index.html`
  - A collection of all the posts
- Create blog
  - Template: `create.html`
  - A form allowing a user to create a post
- Blog pages
  - Template: `blog.html`
  - The pages with the information
- Search results
  - Template: `search.html`
  - A page allowing users to search for pages and displaying those searches
- Edit form
  - Template: `edit.html`
  - A page to edit a post
- Account creation
  - Template: `signup.html`
  - A page allowing a user to sign up
- Login page
  - Template: `login.html`
  - A page allowing a user to login