

Project00: Half-Quick

TNPG: coolbeans

Roster: Evan Chan, Amanda Tan, Danny Mok, Michelle Zhu

TARGET SHIP DATE: 2024-11-7

# DESIGN DOCUMENT

---

## I. Description

This project implements a web log hosting site, with the following features:

- Users will have to register to use the site.
- A logged-in user will be able to
  - Create a new blog
  - Update their blog by adding a new entry
  - View the blogs of other users

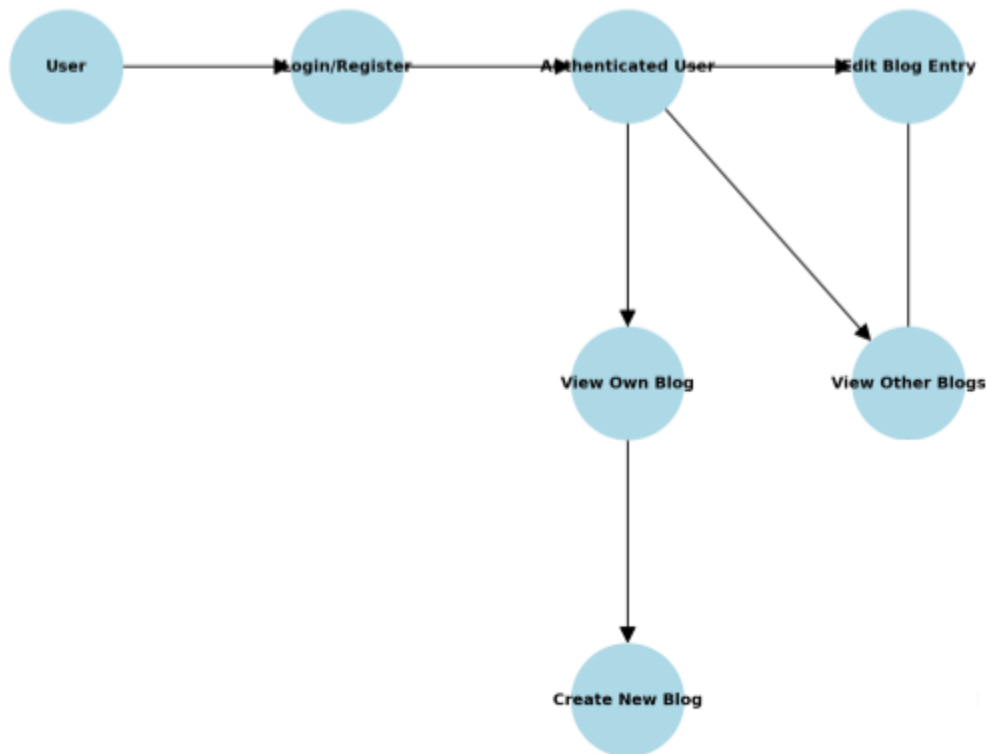
## A. Functionalities

- I. Management of Individual Blog Pages:
  - a. Ability to create blog pages and make further changes after publishing
  - b. Plain text editor to create/format content: the form to submit/ make new text edits
2. Home Page: Dynamically updating page with all the existing blog pages
  - a. Recent blogs show up on top
  - b. Uses links to bring to each page

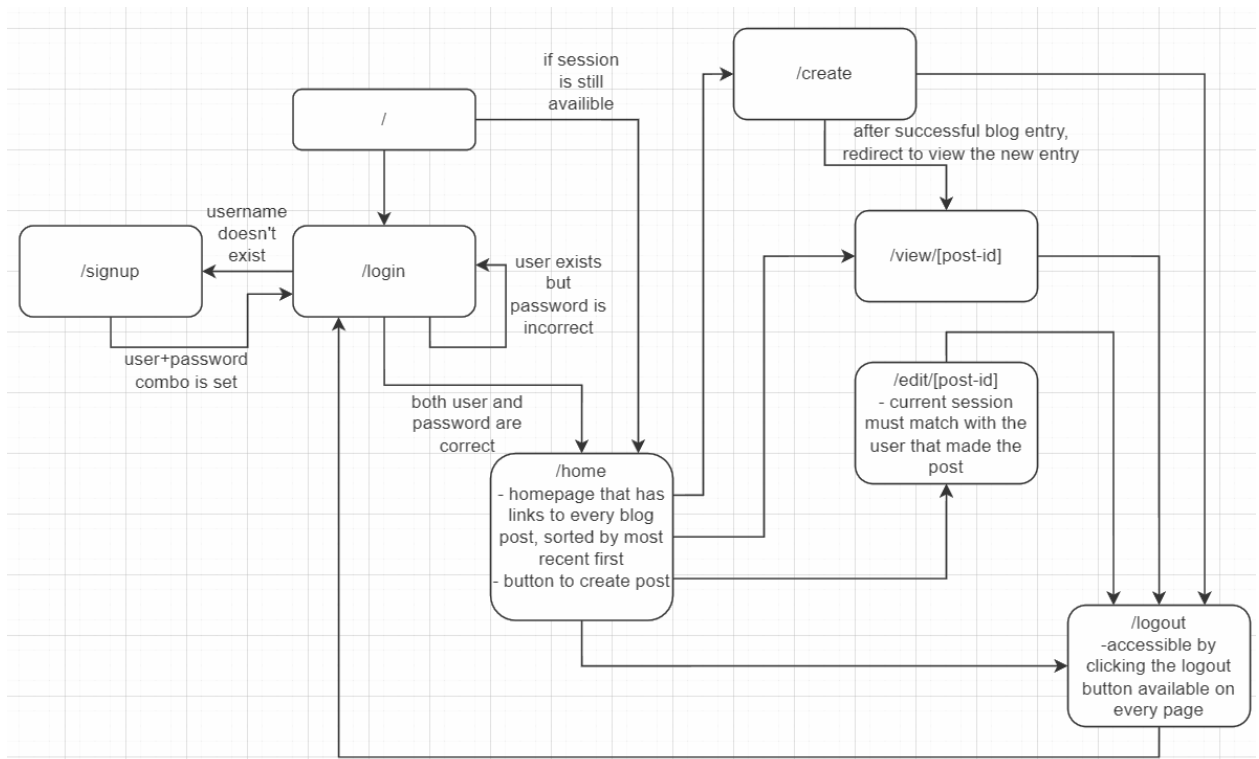
## B. Program Components

- I. Flask/Python:
  - a. Serves the websites, handles HTTP requests, manages user sessions and authentication (login), controls logic surrounding permissions
  - b. Interacts with various tables within the database file to fetch and store user and blog data
  - c. Coordinates Flask routes (between HTML pages) and middleware functions (ex. Module for storing data into tables, creating user sessions, etc) to control application flow
2. SQLite3 (Database):
  - a. Stores user information (usernames, passwords, permissions, etc)

- b. Stores data about each blog page in a unique table → Stores content such as titles, text content, creation time, author, etc
  - c. Data used by middleware functions and HTML templates
- 3. HTML: Provides user interface with links to access different pages (ex. blog pages, blog creation pages, etc)
  - a. Renders content passed by Flask routes/database
  - b. Displays form for user interactions (ex. Creating blog pages, logging into an account, etc) and works with Python to create the various actions
- 4. Jinja: Create templates for blog pages with room for customization
  - a. Used by Flask routes to have blog pages with default formatting (keeps things consistent and makes it easy to create pages for new content)
- 5. CSS: Create a styled plain text editor and make the site look nice across the board
  - a. Works with HTML documents



## C. Site Map



## D. Database Organization

- I. User Table
  - a. User Id/Username (PK)
  - b. Password
2. Blog Page Table
  - a. Page ID (PK)
  - b. Title
  - c. Content
  - d. Username of Creator (FK)

Note: FK stands for foreign key (to link tables), PK for primary key (each row value must be unique)

## E. Task Breakdown

- I. Evan Chan: Full-stack/Project Lead
  - a. Implement user authentication system and sessions
  - b. Manage overall system/file structure: Work with backend and frontend to make sure everything works smoothly
  - c. Pick up any work that isn't explicitly delegated/falling behind schedule
  - d. Assist with roadblocks in any area
2. Michelle Zhu: Backend/Database
  - a. Create SQLite3 database schema

- b. Design database interaction modules (ex. Inserting data, Creating tables, etc)
  - c. Work with Danny to ensure database logic flows/works with front-end
  - d. Create Python modules to handle logic relating to user actions (creating/editing/deleting blog pages)
- 3. Amanda Tan: Backend/Middleware
  - a. Work with Michelle to create a modules (of the blog pages)
  - b. Work with Danny to ensure all related code flows with the front-end
- 4. Danny Mok: Frontend
  - a. Create HTML pages with Jinja templating for dynamic blog content
  - b. Use CSS to make things look nice
  - c. Collaborate with the backend to ensure everything works with the frontend