

lobo p00 Design Document

TARGET SHIP DATE: {2024-11-7}

Will Nzeuton, Daniel Park, Tim Ng, Yinwei Zhang

Blog - xbxlobog

("suh-buh(respirated/inhaled)-low-bog")

Overview

* "blog entry" is used interchangeably with "blog post"

This project is a web app that we built using Python Flask and SQLite to allow users to create and view blog posts. When starting the app, you will be brought to the homepage in which you have three options: logging in, creating an account, and viewing the blog page as a guest. In order to create or edit said posts, you must have and be logged in to an account. Each blog has its own respective category, title, and creator listed to allow for greater sortability.

Components

1. User Authentication Module (UAM)

- a. Handles user login and signup
- b. Allows users to edit their username, email, and password
- c. Handles sessions for user-based cookies
- d. Handles password encryption

2. Blog Management Module - Routes (BMMR)

- a. Handles creation, editing, and display of blogs

3. Blog Management Module - Utilities (BMMU)

- a. Utilities for HTML generation for blogs, fetching, insertion, etc.

4. Entry Management Module (EMM)

- a. Allows users to create new entries, edit existing entries, and delete their entries on the blogs.

5. Category Management Module (CMM)

- a. Organizes blog entries into categories for easier navigation by users. And filtering

6. Database Module (DbM)

- a. Handles all data storage and retrieval operations using SQLite3. Contains tables for users, posts, and categories.

7. Frontend Interface (FeI)

- a. User-facing component built with HTML (refer below) and CSS. Responsible for site layout and styling.

8. Other utilities

- a. Clear Database
- b. Setup Database

Required HTML Templates

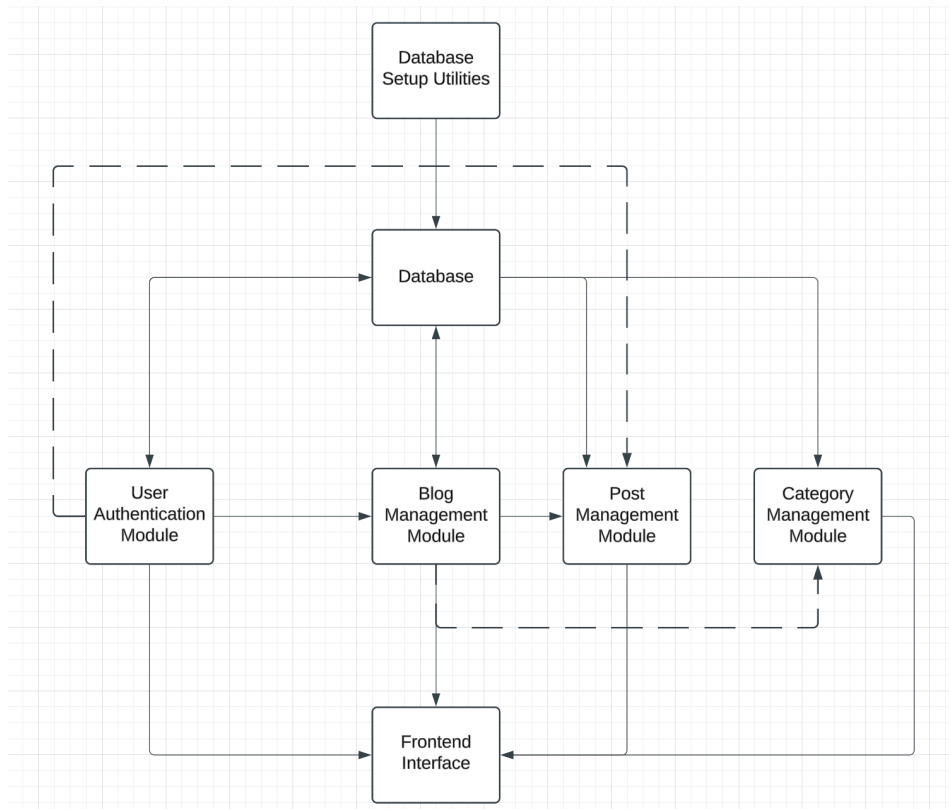
- Images
 - Logo and Icon
 - HTML templates consist of
- Homepage
- Login
- Sign up
- User Profile
 - Settings
- Create Entry Page
- View blogs by category page
- View all blogs
 - Individual Blog Page
 - Individual Entry Page
 - Delete Blog page

Component Relationships

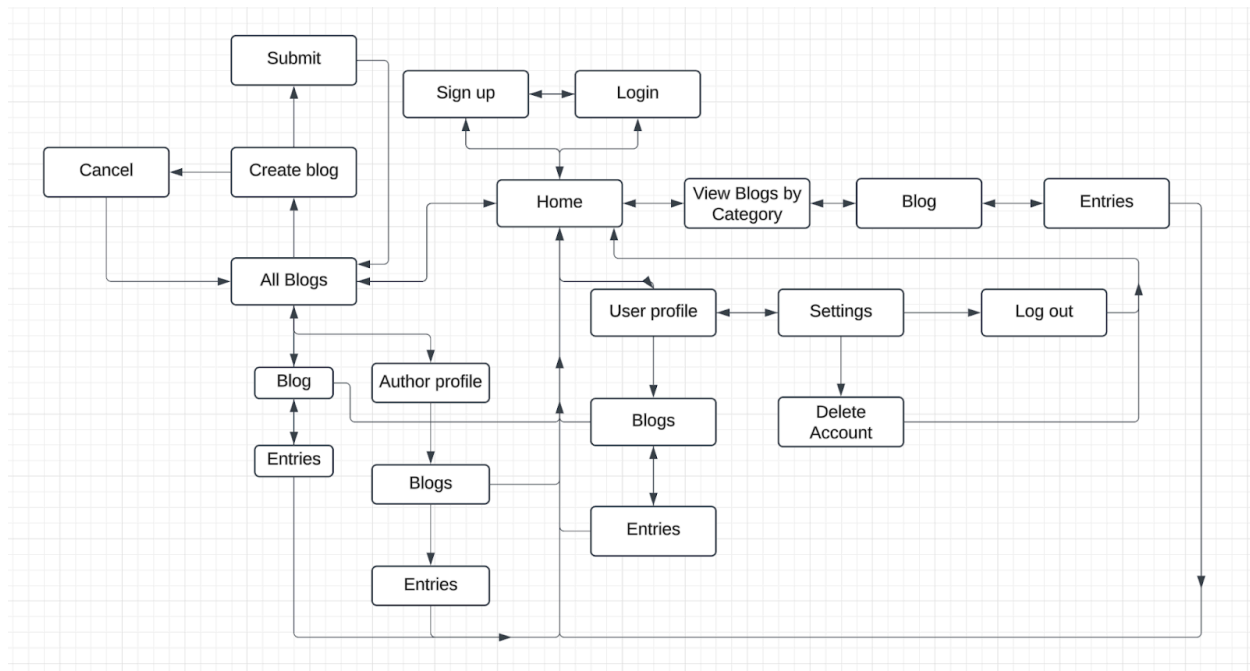
- All modules (excluding database) interface with Frontend Interface to display content, menus, and page layouts
- All modules (excluding frontend interface) interact with Database to fetch and store information
- User Authentication Module
 - Connects to **Database Module** to fetch, update, and insert user data
 - Interacts with **Frontend Interface** to display user profiles, login/signup menus, etc.
 - Interacts with **Blog Management Modules** to verify permissions for creating/editing/deleting blogs
 - Interacts with **Entry Management Module** to verify permissions for creating/editing/deleting blog entries

- Blog Management Module - Routes (BMMR)
 - Interacts with **Blog Management Module - Utilities** to use HTML generation utilities and data-fetching utilities for formatting blogs for display.
 - Interacts with **User Authentication Module** to ensure user authentication and authorization for creating, editing, or deleting blogs.
- Entry Management Module
 - Interacts with **User Authentication Module** to ensure users are logged in and authorized to create, edit, or delete entries.
 - Interacts with **Blog Management Modules** to link entries to their corresponding blog
- Category Management Module
 - Interacts with **Blog Management Module - Routes** to organize blogs by category, making it easier for users to sort and view content

Component Relationship Visualization:



Site map



Database (xase.db)

Tables

users:

id	username	password	salt	email
#	TEXT	PYBYTE	TEXT	TEXT

blogs:

id	title	description	category_id	author_id
#	TEXT	PYBYTE	REFERENCES categories(id)	REFERENCES users(id)

categories:

id	title
#	TEXT

posts:

id	title	date	author_id	blog_id	content
#	TEXT	DATE	REFERENCES users(id)	REFERENCES blogs(id)	TEXT

TASK ASSIGNMENTS

TASK	Will Nzeuton	Daniel Park	Tim Ng	Yinwei Zhang
Set up Flask and SQLite3 environment	X	X	X	
Build User Authentication Module, handle passwords, logins, etc.	X			
Blog Management Module (Routes)	X	X		
Blog Management Module (Utilities)	X	X	X	X
Create Entry Management System	X		X	
Build Categories		X		X
Build Database	X	X		X
Frontend (HTML Templates)				X
Frontend (CSS)				X
Final Testing and Bug Fixing	X	X	X	X