

Selected scenario: Web Log

*Front-end Feature Listing (minimum viable product features are indicated by *)*

| Features only for users logged in; anything logged-out users get is the login/registration page |

Account-local

- Account creation with encrypted password*
- Create blogs*
- Create or edit entries within own blogs*
 - View edit histories for own blogs*
- Profile Page*
 - View listing of own blogs*
- Settings Page
 - Change password
 - Change username

Sitewide

- Every page contains a heading of links to <home>*, <profile>*, <settings>, <logout>*
- View other users' blogs*
 - Directory with all blogs, sorted by timestamp of latest entry of blog*
 - Search for keywords within title
- View edit history for other users' blog entries
- Comment on other users' blog entries*
- View other users' profiles*
 - Contains links to user's blogs*

Database Layout

Table name	Entry content
users	user id, username, hashed password
blogs	user id, blog id, title
entries	blog id, entry id, version id ¹ , time stamp ² , content
comments	blog id, entry id, user id, time stamp

¹Starts counting from 1

²Unix time of the server

HTML Templates

- Login.html
 - Directed from “/” if user is not logged in
 - Displays text input boxes for username and password
 - On click <submit>, redirects user to “/login” through a POST request
 - On click <create new account>, redirects user to “/register” through a GET request
- Register.html
 - Directed from “login.html” on click <create new account>
 - Displays text input boxes for information needed to create a complete account
 - On click <create account>, redirects user to “/register” through a POST request
- Home.html
 - Directed from “/login” or “/register” if credentials are verified
 - Directed from link <home>
 - Displays all blogs in order of most recently edited
 - Displays a blurb for each blog containing the first 20 words of its content
 - On click <blog_whatever>, redirects user to “/blog” with whatever’s blog_id
- Search_results.html
- Profile.html
 - Directed from link <profile>
 - Displays all blogs of that user
- Settings.html
 - Directed from link <settings>
 - Displays link <change password>
- Blog.html
 - Directed from “/blog”
 - Displays all entries of that blog
 - Displays a <view entry history> link for each entry
 - Displays comment section at the bottom, with a text input box for new comments
 - Displays an <edit> link if
 - On click <view_entry_history>, redirects user to “/entry_history” with corresponding entry_id
 -
- Entry_history.html
 - Directed
- Layout.html

Task Division

- Joseph
 - Home.html, Blog.html
 - The python & flask regarding tasks in integrating above-mentioned html pages
- Yevgeniy
 - Everything database related
 - Fetching a blog's entries, fetching a list of blogs, etc.
 - Adding users, blogs, entries, comments to database
- Eric
 - Helping out with frontend and backend tasks when applicable
 - Team management
 - Devblog and updating design doc

Component Map

Frontend	Relationship	Backend
Register account	If the username is not already located in the users table, then add account to users table	Users table
Login	If username typed is in the table and the password of the username matches the password typed, then login	Users table
Create new blog	Find the userid associated with the current user in the users table. Insert values in blogs table with the userid, name of blog, and blog id.	Users table, blogs table
Create new entry	Find user id associated with current user in the users table and only show new entry button if users match. Find the blog id of the blog in which an entry is being created. Insert values in the entries table with the blog id, entry id, version id, time stamp, and content	Users table, Entries table, blogs table,
Edit entry	When editing an entry, the current text in said entry is displayed to user by querying the entry from the database. Find userid associated with current user in the users table and only show edit entry button if users match. Find the blog id of the blog in which an entry is being edited Insert values in the entries table with the blog id, entry id of entry being edited, a incremented version id, the current timestamp, and an updated content section.	Users table, Entries table, blogs table
Change password	Hash the password that user inputs and then change the value of password associated with the current user.	Users table
Add comment	Find entry id of entry in which a comment is being added. Find blog id of the entry Find user id of user making comment Get current time stamp Add these values to the comment table	Users table, blogs table, entries table
Search	Select blogs from the table where their titles match the search request	Blogs table

Sitemap

