

Disturbed Window Monsters- David Chen, William Guo, Marc Jiang

SoftDev

P00: Move Slowly and Fix Things

2022-11-01

Target ship date: 2022-11-21?

(Scenario 2 - web log):

- 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 and edit their own past entries
 - View the blogs of other users

Necessary Components:

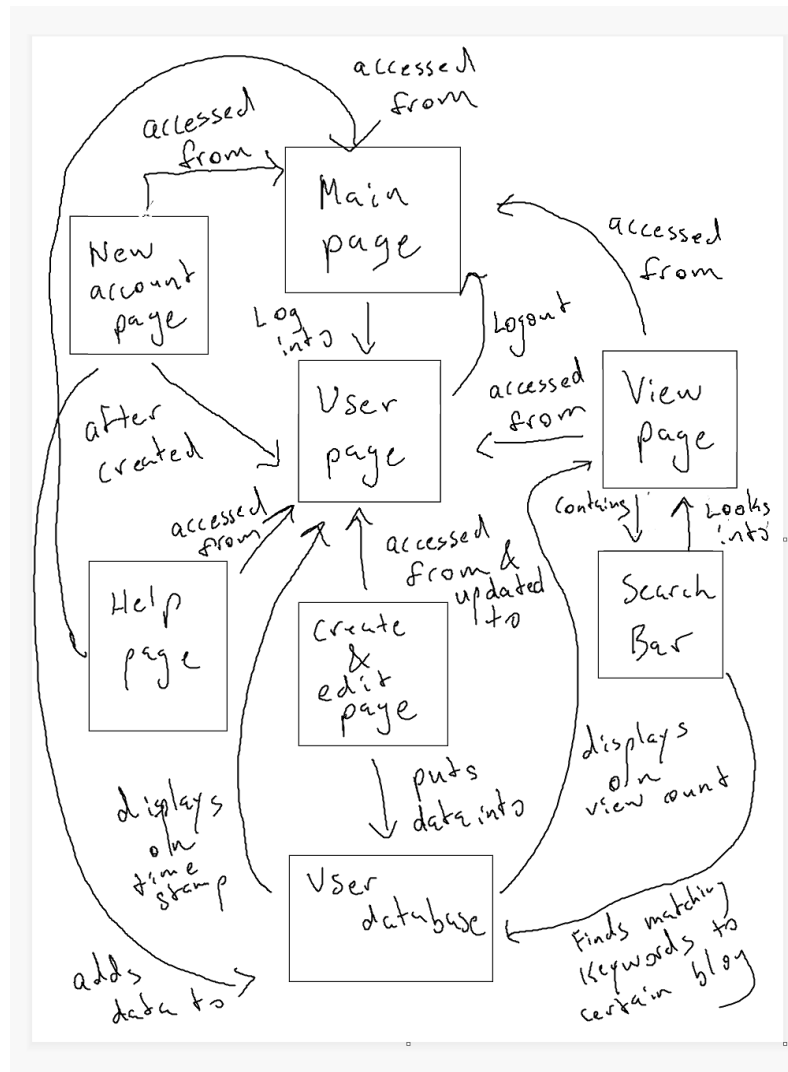
- Main page
 - Login in using Username and Password
 - Create new account
- User page
 - Display all blog posts that you've made
 - Edit blog button for each blog post
 - Navigation tab on side linking to help page, view page, create page
 - Logout
- New account creation page
 - Creates new account to be stored in database
- Create/edit page
 - Creating new blogs presents users with a blank html document where they can write anything, add images, links etc.
 - Editing blog brings up stored information of that post where it can then be edited and saved/updated (can both add and delete what's written before)
- View page
 - Displays full blog post from another user
 - Search bar to view other user's blog posts
- Help page
 - How to navigate this website
 - Helpful tips to start writing your own blog (links to inspirations, ideas)
- Search bar
 - Can search for other users' blog titles that contain specific keywords or dates entered

Relationship between Components:

- From the main page, once you log in it takes you to your own blog page with history log sorted by date posted
 - If you don't have an account, takes you to new page to create an account then redirects to the user page

- Help page can be navigated from navigation bar in user page through or an icon from the main page
- View page can be access from navigation bar in user page or by clicking on blogs pictures/headings when displayed after searching or visiting the view page
- After the user prompted the changes to either create or edit blogs from the user page, the user page is then updated with the latest edits
- After you log out, it takes you back to the main page

Component Map:



Organization:

- 2 tables:
- 1)

Username (TEXT_UNIQUE)	Password (TEXT)
------------------------	-----------------

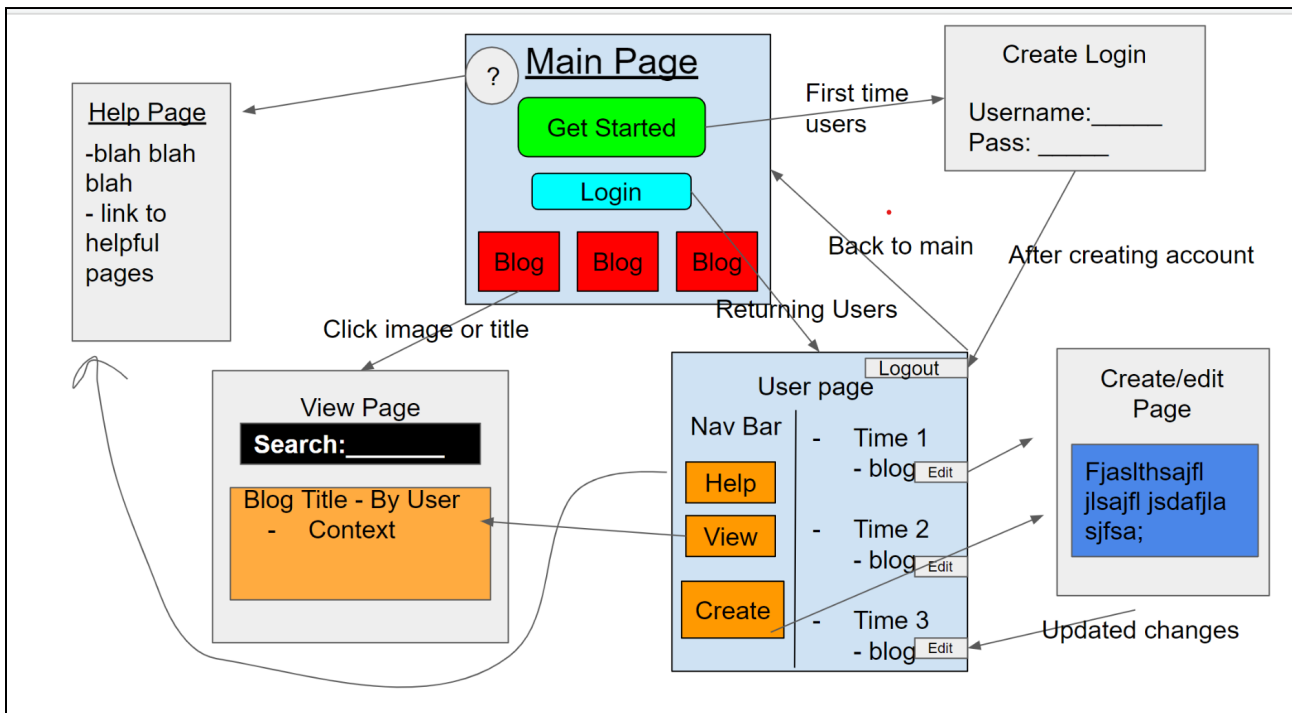
- Username and passwords stored in two column table so when user logs in, it can verify if password matches

2)

Username (TEXT_UNIQUE)	Blog post Title (TEXT)	Blog post body (TEXT)	Date Added (Integer)	Date Modified (Integer)	View Count (Integer)	Time (Integer)
---------------------------	---------------------------	--------------------------	----------------------------	-------------------------------	----------------------------	-------------------

- Date added and modified example: 20221101 (2022-11-01, date without the hyphens)
 - First time date modified is the same as date added but afterwards, date modified only changes if prompted through edit instead of create
- In the user page, the blogs are displayed by the dates that it was first posted.
 - If the dates are the same, then sorted by latest time
 - Time written out in **unix time (ie. '013535' would be 1:35:35 am whereas '133535' would be 1:35:35 pm)** - capped at 245959 before going back to 000000.
- Using the same table above, the search results would be sorted by views (counts how many people visited this site whos username doesn't match up with your own)

Site map (excuse the mess):



Assignments:

William: Database integration

- Creating and displaying blogs sorted by proper time
- Matching login information with user input
- Added new entries of login info, edits, and entries to respective table

Marc: HTML:

- adding links where it is necessary and creating help page
- Redirection of pages

David: Python and flask backend

- Flask forms
- POST and GET requests for search and login
- Update changes on the website after user's edit