CPSC 304 Project Cover Page

Milestone #: 3

Date: 01/03/24

Group Number: 60

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Sangita Dutta	29933983	sdutta06	sangita7dutta@gmail.com
Delsther James Edralin	26279729	dedralin	dedralin@student.ubc.ca
Aman Johal	40801152	amanj21	ajohal21@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

1. Project Description

The project is a social platform that manages outdoor, dog walking activities. The domain of this project focuses on “pet wellbeing”, which relates to pet care, recreation, and health. This mainly caters for people that are dog owners and want to guarantee their dogs to receive consistent stimulation by regular walks.

2. Timeline and task/breakdown

The format of this section will be based on the main features we currently think are needed in order to identify the queries we have to create and how we should break down the workload.

Website Main Features Breakdown

Pages

Owner’s Profile Page

The URL will be based on the owner’s ID. Whether the user is the owner or not.
Example: http://localhost:8800/owner/23 where 23 is the ownerID of Owner.

- Shows an owner’s name and contact details
- **Edit profile button** - edits name and contact. Hidden when the user is not the owner.
- **Add friend button** - automatically creates a friendID. Hidden when the profile page is the user’s.
- **Friend List** - shows 6 friends with a button that directs to the friends page to view more.
- **Dog display** - shows a few dogs’ names and icons the owner owns. Once a dog is clicked, it will show a modal window. (*See Dog Modal Window component below*).
- **Owner’s posts** - shows either media or walk details via a card shaped component sorted by most recent date. Which when clicked, will go to the link of the post. (*See Owner’s Post Page*)

Owner’s Friend List

The URL will also be based on ownerID.
Example: http://localhost:8800/owner/friend-list/23 where 23 is the ownerID.

- Shows the list of friends, date where they became friends, and how many meetups they had together (we can use meetup IDs for this. If they have the same meetupID, it will be noted via count.)
- Hyperlinks of these friends, to go to their profile pages.
- If the user is the owner in the friend list, they can unfriend them (Should also remove the friend’s friendship with the user. Eg. Friendship(1,2,DATE) is removed, then Friendship(2,1,DATE) is also removed.)
- Sorted by the most recent friend. If possible maybe add a sort selection feature.

  Example:
  {John Doe 1/11/23 Meetups: 5 [unfriend][view profile]}

- Where the view profile button is a link to his page and the unfriend button is shown if it’s the user’s friend list.

Home/Newsfeed

URL: http://localhost:8800/

- Shows a list of the owner's (the user's owner in particular), and their friends' posts sorted in most recent.
- If there is no user logged in, it should redirect to the Login Page.

*Note that it should **not** redirect to the log-in page if the link was not the home's URL. (A non-logged-in user should be able to check post pages, friend list, and profile pages).

Login Page

URL: http://localhost:8800/log-in-sign-up/

- Users should fill up their email, name, and phone number (allowed to be null) if they do not have an account. Otherwise, login.

Owner's Post Page

Example: http://localhost:8800/owner/post/12/ where 12 is the postID from Post_Walk entity.

- Shows the Post with the media, content, and tags.
- Shows the Walk's (or MeetUp's) details particular to that post (time, location, date, dog, etc.)
- If the user is the owner of the post, they can edit and delete this page.
- Once deleted, it should redirect back to the user's profile page.

Example:

{ [x]

[photo]

Alberta Pond

10/25/23

Walked with Peggers.

Went to the pond today with this doggers right here.

#pond #doggers }

Panels/Components

Side Menu Bar

Always shown (I propose to put this at the left like instagram's webpage). Functions below are listed in order from top to bottom for the icons. (Can be rearranged though)

- **Home Icon** - sends the user back to Home's URL
- **Search** - This function would be searching all the owners' names via a query. For example, if the search value was "Ja", it should show owners like "Jake", "Janice" "Jajaja", etc.
    - Minimize the number of results. Meaning the user has to search the full name to find out the results. We will also not have a search page to view all owners to make it simpler.
- **Notification Icon** - pops up a modal window (*See Notification Modal Window Component).*
- **Profile Icon** - sends the user to the user's owner profile page.
- **Logout button** - logs the user out.

Side Scheduling Bar

Only shown when the URL is at the home page. (Right side). Also in order.

- **User’s upcoming walks or meetups** - will show a maximum of 4 with a dropdown toggle. Sorted in the closest upcoming walk with a brief description of each.
    - Delete and Edit button beside. They can also remove or edit a walk if they want.

  Example: {Walk with Peggers. *10/11/24*}
  {MeetUp with Perry and Peggy. *10/23/24*}
  {Walk with Edgar. *5.0 km goal!*}
  {[see more]}

- **Schedule Button** - provides a dropdown form once the schedule button is clicked..

  Example of the form:

Location*: [Pond in Alberta]	**Date: [DD:MM:YY]**
Dogs Participating*: [Peggers] [Jake]	**Owners Participating: [---------]**
Time: [--:--:--]	**Distance Goal: [---------]**

  [submit][cancel]

- **Past walks/meetup list** (also max. 4 with a see more button), shows a brief description of each
    - If the walk/meetup was posted, it shows a [VIEW] button that hyperlinks to the post URL.

      Example:
      {Walk with Peggers. *3/11/24* [view]}
      {Walk with Peggers. *3/11/24* [post]}
      {[see more]}

    - If the walk/meetup was not posted, it shows a [POST] button which provides a form.

      Example of the form:
      Post your previous walk with Peggers.

      Media: [upload file] or [add link]
      Content: [---------]
      Tags: [---------]
      [submit][cancel]

    - The post **should** always be connected to some WalkID.
    - *See Media Storage for details on media uploading.*

Notification Modal Window

- Shows notifications that are either walks or meetups (unclickables), and friend posts (links the user to the post url).

  Example:
  {It's time to walk Peggers!}
  {Janice just shared her recent walk.}

- Has an exit button which will slide the window to the left to hide.

Dog Modal Window

- A window when a user clicks a dog icon in some owner's profile page. Shows their walk histories. Users will be able to edit the dog's details.
- Dog's are shown through pre-built icons.

  Example:
  [icon clicked]
  { [x]
  Peggers ***3 years, 4 months***
  Pug
  [photo1][photo2][photo3][photo4]}

- Note that the photo will come from the media from the post that is connected to the walkID and dogID. There would be some algebra to get it, but not too hard.

Other Features

Persistent Storage - helps to know who the current user's owner is. We will implement this using a barebones mechanic in firebase.
Media Storage - used mostly for posts. We need some sort of place to store these images so we only store links in the database. It should generate a link as well after uploading a photo.
Header and Footer components - extra, can be done in later times.

SQL Integration

- We will be using PostgreSQL, instead of the planned DBMS MySQL. It was changed due to an authentication bug from version 8.1.0. It will be integrated using Node.js and Express with React as the frontend framework.
- Each entity from our milestone 1/2 will have their own **controller** and **service** *.js* file to connect to the database. The functions that will be implemented will be based on our current needs for the features in our webpage. Current files will be in the backend folder of our repository.
- The controller file would handle the calls to the service file.

- The service file would send the queries to the server.

Division of Tasks

The proposed division would be an equal End-to-End division. This would mean that if the user created the homepage of the website via react/js, they should implement the function to retrieve the necessary data by developing the controller and service files in the backend.

Task Breakdown

Member	Aman	Sangita	James
Frontend development	Owner's Friend List Page	Home/Newsfeed	Owner's Profile Page
	Dog Modal Window	Login Page	Owner's Post Page
		Side Menu Bar	Side Scheduling Bar
		Notification Modal Window	
Entities' controller and service files needed functions for backend development	Owner- for friend list	Owner - for login and search	Owner - for profile page
	Friendship - for friend list	Post_Walk - for post	Owns_Dog - for profile page
	Owns_Dog - for dog window	Friendship - for posts	Walk - for scheduling bar and post
	Walk - for dog window	Receives_Notifications - for notification window	Post_Walk - for scheduling bar and post
	WentFor - for dog window	FriendPost - for posts	On_MeetUp - for scheduling bar and post
	On_MeetUp - for dog window		Schedules - for scheduling bar
	Post_Walk - for dog window		Post_Media, Video, Photo - for post
	Post_Media, Video, Photo - for dog window		TaggedIn - for post
	TaggedIn - for dog window		Logs - for scheduling
Misc. development		Service rerouting to PostgreSQL	Server, controller, and service setup and documentation
		Frontend/Tailwind Layouting	Frontend setup
		Persistent Storage Integration	Media Storage Integration

*Green text means it is done.

Timeline

Week 1 [Mar 4 - 10]: Setting up the code base.

- Initial setup of PostgreSQL and React. Integration of the database to Node.js and Express.
- Templating controller and service files.
- Media storage integration I (documenting how to develop).

Week 2 [Mar 11-17] : Easy tasks to gradually get in through development. More on functionalities.

- Home Page (as it only needs to fetch the user's owner and its friends posts then sort it to recent)
- Owner's friend list (also easy fetching and crud operations)
- Side Menu Bar - Basic functionalities. Mainly working on search.
- Notification Modal Window (to complement the menu bar)
- Owner's Post Page I - more for data retrieval and not media storage fetching.
- The needed controller and service files for these functions.

Week 3 [Mar 18- 24]: Developing the harder parts.

- Owner's Profile page I (Finish edit profile, add friend, and link to friend list)
- Media storage integration II (implementation)
- Owner's Post Page II (should be complete with media integration. Will also reflect on homepage)
- Side Scheduling Bar I (Finish upcoming walks, plan forms and past walk with dropdown features)
- Login page and functionality through persistent storage integration
- Dog Modal Window I (Display needed data in a window)
- The needed controller and service files for these functions

Week 4 [Mar 25 - 31]: We should probably be nearly done at this point. Design stage.

- Owner's profile page II (finish this)
- Scheduling Bar II (finish this)
- Dog Modal Window II (finish this)
- Styling

April 1 - 5: Extra buffer days to catch up with pitfalls. Other than that: fix bugs, tasks, and more styling.

3/4. Current commit repository history

- Milestones 1 and 2 - Sangita, Aman
- README - Aman
- Codebase setup and documentation - James
- Rerouting of database service - Sangita

Commit history link:
https://github.students.cs.ubc.ca/CPSC304-2023W-T2/project_f6o6r_o4l8z_z1f3s/commits/main

5. In Canvas for M3 submission add link to the repo

- https://github.students.cs.ubc.ca/CPSC304-2023W-T2/project_f6o6r_o4l8z_z1f3s