University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: ___3____

Date: ____Oct 26_______

Group Number: ____119_____

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Stuart Chen	48414957	r4v4j	stuartcc6@gmail.com
Allan Xing	28532901	p7i4r	allanx01@students.cs.ubc.ca
Eric Fu	57440844	y2c8w	ericfu55@student.ubc.ca

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

University of British Columbia, Vancouver
Department of Computer Science

Summary:

We’re making a social media platform similar to applications like Instagram, Twitter/X, and Facebook. The primary function is to enable users to share and interact with different kinds posts, through liking or commenting. Other functions include messaging, advertised posts, and changing permissions of users (e.g. through blocking or subscribing to premium features).

Frontend functionality:

- Create account:
    - Users can use a sign up/log in button to create account
- Login
    - Use the login button, user enters credentials, accept or deny request
- Create post
    - Users can click the create post button, prompts the user to choose video/text/image post, and prompts the user to upload files as necessary
- Feed / **HOMEPAGE**
    - Displays the posts of accounts the user follows in chronological order from most recent, limit of 10 posts on fyp, with more upon user clicking (show more posts button)
    - Feed page contains many components of type “Post”
    - If user is not logged in, display some random filler
- Post
    - Like button, showing number of likes
    - expandable comment section, user can comment and like if they’re signed in
- Account page
    - Has a follow button, shows number of followers and following, as well as number of posts
    - Under account summary, displays the feed of that account, based on most recent.
    - Users can click on follower/following to see a list of the accounts that specific account followers or the ppl that follow it
- Profile
    - User can change their account settings and see some stats
    - Users can change their username as well, providing the new username is unique
- Direct Messages
    - Users can send DMs like in discord, instantaneous local updates for new messages using a websocket (Socket.io)
- Hashtag
    - Users can associate posts with hashtags (if they try to use a hashtag that currently doesn’t exist, it will automatically be created and used)
- Purchase Subscriptions
    -
- Dev Page

- A page that displays useful information about the application’s database, such as the various relation tables
- Allows us to query the database and view updates to the data being stored

Backend functionality:

- Create account:
    - POST form processing, takes in the arguments email / username and password.
    - Generates a Bcrypt hash to be stored into the database tied to the username / email.
- Login:
    - POST form processing, takes in username and password
    - Passport.js local authentication based on the stored Bcrypt hash in the database
    - Authenticates the user, redirecting the user to /home
- Create post:
    - POST form processing, takes in video / text / image and stores it into the database.
- Delete post:
    - Deletes a post (might be POST?) requires user id to match the post owner id, OR requires user permissions to be “admin”.
- Create Comment
    - POST text processing, takes in text and stores it into the database.
- Logout:
    - POST Logout function to log users out.
- GetPost
    - GET request for a specific post based on some key
- GetAccount
    - GET request for some account based on some key
- GetChat
    - GET request for some chat based on some key
- GetTable
    - GET request to take a specific SQL table and return the information based on some key. Used for the dev page.
- SendChatMessage
    - POST, takes in some chat key and then sends the message to the chat.
    - Could also just be a websocket event listener
- Purchase plan

Timeline:

- Nov 1st: TA Meeting
- **Nov 1-5th Initialization**: Initialize project, set up Database, Login page functionality (frontend + backend) up and running
    - Eric:
        - Login page backend
        - Logout
        - Authentication class / logic
    - Stuart:
        - Backend database script planning (Login and potentially Account)
        - Middleend responsibilities:
        - /logout endpoint for Passport.js
        - /login endpoint
    - Allan:
        - Login page frontend
        - Login form
        - Logout button
        - Set up database
    - All of us: Write SQL script to actualize the database
- **Nov 6-10th Basic posting functionality**: Dev page (to view SQL tables), Create Post frontend / backend, Delete post frontend / backend, GetPost (used for populating the Home page when we implement that)
    - Eric:
        - Dev page frontend
        - GetPost backend (for fetching posts)
    - Stuart:
        - Create and delete frontend
        - Dev page backend
    - Allan:
        - Create and delete post backend (populating the database)
    - All:
- **Nov 11-15th Viewing posts functionality**: Home page frontend (simply viewing posts, no like/comment features yet), Account page frontend / backend (simply viewing an account with its posts, other account elements like “following” will come later).
    - Eric:
        - Account page frontend
    - Stuart:
        - Account page backend
    - Allan:
        - Homepage frontend
    - All:
- **Nov 16-20th Account/Post Details**: Account/Post interaction functionality - followers, likes, comments (editing account settings will also be implemented here), Hashtag
    - Eric:

        - Post interaction: comment backend
        - hashtag frontend
    - Stuart:
        - Hashtag backend
        - Likes frontend and backend
    - Allan:
        - Account information backend
    - All:
- **Nov 21th-26th Chats**: Chat functionalities
    - Eric:
        - Chat backend
    - Stuart:
        - Outer chat page frontend (viewing all conversations)
    - Allan:
        - Inner chat page frontend (viewing a single conversation)
    - All:
- **Nov 27th-30th Finishing touches**: Polishing ui styling, final touches, QA
    - All: Everyone works on polishing, QA, UI styling