

## CPSC 304 Project Cover Page

Milestone #: 1

Date: 2024-07-12

Group Number: 50

Name	Student Number	Alias (Userid)	Preferred E-mail Address
Donald Lee	30458228	x4z8c	donaldkevlee@gmail.com
David Lim	79941274	u8w8n	daviddlim11@gmail.com
Jae Wu Chun	66321381	f3c7i	jaewuchun@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

# Project Description

## 1. What is the domain of the application?

The domain of our application is social network/media. We will make an application for people to connect in a specific way: through regularly assigned hangouts based on similar attributes such as hobbies, major, etc.

## 2. What aspects of the domain are modelled by the database?

Making friends and meeting new people on campus can be hard. Every year, hundreds of students post on Reddit expressing how difficult it is to make new friends. We are making a social media app that pairs up UBC students so they can hangout. The app is based on the idea of hangouts called “donut dates.” The database will primarily be modelling these dates and the people that will be assigned to them. In nature of our domain being social media, some entities are more abstract (for example, a Profile entity corresponds to a physical person, but a Post entity corresponds to a virtual/media entity)

Each donut date will have a `createdAt` and `isCompleted` attribute. The server will make the pairing every two weeks. For record keeping we will need to know when the donut was created (`createdAt`). The donut will only be active until two weeks from the `createdAt` date (at which point a new round of pairings will go out). If the users paired end up completing the donut, they’re able to mark the donut complete, at which point they can add information about their date; the donut date entity set will be a strong entity set to a weak entity set Post, which models a social media post that contains the aforementioned information about the date, such as photos, a title, and a description. These posts are then shown in a wide feed of all the users.

The type of donut date that the users can go on may (or may not) be specified as a `StudyDonut` or `SocialDonut`. For example, in finals season, people may opt to just want to go on `StudyDonuts`. Contra summer, where people may want to do something fun. Or, during a regular winter term, one could opt for none specifically—they would be down to study or have fun.

The database will need to know certain attributes of the profiles, which will be used at the business logic layer to make the appropriate matching for the dates (e.g. match people with similar hobbies). But if not enough people are found, we’ll fallback on random assignments. Profile will have two relationships with itself: `Blacklist`, and `BeenPaired`. We don’t want duplicate pairings to happen more frequently than they need to, hence `BeenPaired`. We also want Profiles to blacklist other profiles to not be paired with. Perhaps their ex is on the platform.

Hobby is its own entity set as the hobby itself that it models will be enumerated (users will multi-select the hobbies they have, instead of typing in an open text input), and will have a many-to-many relationship with a profile. Comment is its own entity set (as opposed to a relationship) as it will need to have a relationship with itself in order to model Reddit-like threadable comments.

## Database specifications: (3 - 5 sentences)

1. What functionality will the database provide? I.e., what kinds of things will people using the database be able to do.

Students using our app will be able to manage their profile. E.g. they can change their hobbies, interests, preferences, locations etc. Our app will also create group chats for each donut group, or the students who are paired up with one another. Students in the same donut group will be able to send messages to one another. Each donut pairing, and the messages sent in its respective group chat, will also be saved to our database, so a user can look back at previous dates and messages. We will also allow students to create posts to showcase their donut date for other students to interact with through comments and reactions to build more community.

## Description of the application platform: (2-3 sentences)

1. What database will your project use (department provided Oracle)? See the “Project Platforms” section of this document for more information.

We'll be using the department provided DB, with a locally running server SSH tunnelled into it (like the local ExpressJS guide from tutorial 2).

2. What is your expected application technology stack (i.e., what programming languages and libraries do you want to use)? See the “Project Platforms” section of this document for more information.

Ideally our front-end will be a React PWA since we want our end users to receive push notification and our server will be in Node/Express—the setup will be similar to that from the tutorial as mentioned before.

We expect to use some sort of CRON job/scheduler library in our server, as our application will require a function to be run periodically. We will be using GCP to store files such as images (for profile pictures, posts). We may also implement web sockets for communicating in group chats. We will also use OAuth2 with Google for login so users don't need to create a new password and account to use our application, this also saves us time as we do not need to encrypt and store passwords.

## ER Diagram

