Table 1: Revision History

Date	**Developer(s)**	**Change**
February 1, 2021	Graeme	Team Roles, POC, Technology, Style Guide
February 1, 2021	Anando	Team Roles, Git, POC, Technology
February 1, 2021	Yuvraj	Team Roles, Git, Technology

SE 3XA3: Development Plan
Terminal Social Media App

L03 Team 1
Anando Zaman, zamana11
Graeme Woods, woodsg1
Yuvraj Randhawa, randhawy

Due February 5, 2021
Tag DP-Rev.1

This document outlines the Development Plan Lab section 3, team 1 project - Terminal Social Media App. This document outlines the team meeting plan, team communication plan, team member roles, Git workflow plan, proof of concept demonstration plan, technology, coding style, project schedule, and project review.

1 Team Meeting Plan

1.1 Meeting Agenda & Roles

Our team will meet virtually ~~once or twice a week~~ on monday and wednesday each week via FB messenger or MS Teams. Each meeting will be approximately one hour and will take place at 5pm. Each meeting will have its own agenda and each task will be completed by the user assigned that task. Most of the meeting plan and decisions will be recorded via Teams built-in recorder or FB. These meetings will discuss several major components such as blockers to current timelines, technical or non-technical discussions amongst team members, and potential technical demonstrations between different team members process so that everyone is aware. Roles will be rotational with Anando taking charge of the first meeting followed by Yuvraj for the next and Graeme for the last. This cycle will repeat.

Table 2: Meeting Roles

Roles	**Description**
Meeting Lead	Responsible for keeping the meeting on track and ensuring that the meeting is organized according to a meeting agenda
Recorder	Takes notes of decisions and action items that have been reached during the meeting
Summarizer	Compiles the meeting notes into meeting minutes
Participant	Responsible for contributing towards the meeting discussion

2 Team Communication Plan

As the course is completely virtual, the communication outside of class hours will be held mainly through Facebook Messenger. Through Messenger, team members can communicate about different tasks as well as brainstorm any ideas. This way, each team member is imformed and avoids knowledge gaps. As an alternate option, team members can discuss through MS Teams if they are unable to reach via Messenger for technical issues. A Gantt Chart will be used to organize timelines for different aspects of the work with information regarding what aspects have not begun, have completed, and are in-progress.

3 Team Member Roles

The following table outlines the roles that each team member will be responsible for the course of this project.

Table 3: Team Member Roles

Role	**Member(s)**
Project Lead	Anando
Developers	Anando, Graeme, Yuvraj
Documentation Experts	Anando, Graeme, Yuvraj
Git Expert	Graeme
LaTeX Expert	Yuvraj
Technology Expert	Anando

4 Git workflow plan

The Git Workflow will consist of a master branch initially where all team members will stage commits and make changes. This will be a centralized approach and will likely be effective for the initial documentation and design phases since the team is small, and is unlikely to conflict. In the development stages of the project, team members can move to a branched approach if merge conflicts arise or if commits become difficult to track or manage.

5 Proof of concept demonstration plan

The main challenges for this project will consist of the Firebase API libraries, porting the application from Java to Python, and testing in PyTest. This is because the application relies heavily on Firebase as the back-end which is something some team members might have limited exposure in. This can cause some friction in the development stage as team members have to learn how to correctly use it to develop the product.

We selected Python as our implementation language because all of our team members are familiar with it. However, the original open-source project was written in Java which some of us are not as proficient in. Understanding the Java code and porting it to Python may present some challenges as well, especially if many there exist libraries in Java that do not have a python equivalent.

The app would be executable through any desktop system that is capable of running Python scripts. This can include Mac OS, Windows, and Linux to name a few. This ensures portability as the application can be run on several systems without major system level restrictions.

We plan to implement some base level functionality for the POC such as login authentication and add/delete/view posts. The authentication modules would take 2 parameters of username and password and send it to the firebase back-end via API requests to authenticate the user credentials. The add/delete/view posts functionality is self explanatory as it will update the database respectively to reflect the commands.

6 Technology

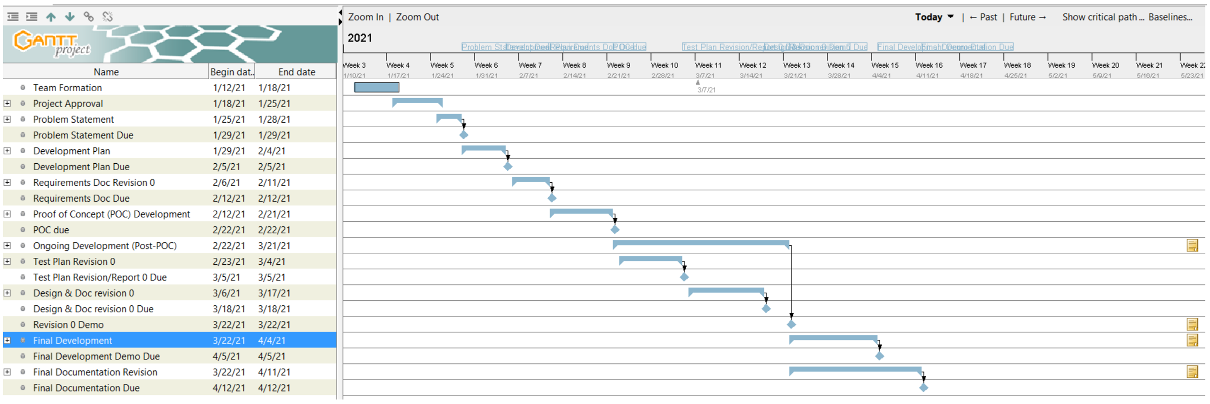
- Language: We will port the existing project to a Python implementation. The team members are all comfortable with coding in Python. Python also contains several built-in libraries that will be helpful in implementing the project.
- IDE: We will use the PyCharm IDE and its included linter to manage the coding style of PEP8.
- Testing framework: The testing framework that will be used is PyTest for the unit/integration tests.
- Doc Generation: We will be utilizing latex and Doxygen to generate documentation.

7 Coding style

We have decided to use the PEP 8 style guide to define our code structure and formatting. The linter, outlined above, will help us automatically match this style guide and keep our code clean and readable. Outside of the linting, we will also have to keep our imports organized, use ”snake_case” for functions, ”CapCase” for classes and add useful comments to our code.

8 Project schedule

The gantt file can also be found in the projectSchedule folder.
URL PROVIDED BELOW:
https://gitlab.cas.mcmaster.ca/zamana11/se3xa3-project/-/tree/master/ProjectSchedule



Name	Begin dat..	End date
Team Formation	1/12/21	1/18/21
Project Approval	1/18/21	1/25/21
Problem Statement	1/25/21	1/28/21
Problem Statement Due	1/29/21	1/29/21
Development Plan	1/29/21	2/4/21
Development Plan Due	2/5/21	2/5/21
Requirements Doc Revision 0	2/6/21	2/11/21
Requirements Doc Due	2/12/21	2/12/21
Proof of Concept (POC) Development	2/12/21	2/21/21
POC due	2/22/21	2/22/21
Ongoing Development (Post-POC)	2/22/21	3/21/21
Test Plan Revision 0	2/23/21	3/4/21
Test Plan Revision/Report 0 Due	3/5/21	3/5/21
Design & Doc revision 0	3/6/21	3/17/21
Design & Doc revision 0 Due	3/18/21	3/18/21
Revision 0 Demo	3/22/21	3/22/21
Final Development	3/22/21	4/4/21
Final Development Demo Due	4/5/21	4/5/21
Final Documentation Revision	3/22/21	4/11/21
Final Documentation Due	4/12/21	4/12/21

9 Project Review

The development of this project went as expected. As a team, the work was split up evenly and everyone was able to contribute. The final outcome fulfilled all the functional and non-functional requirements and followed our Gantt chart time frames correctly. The newly redesigned application added many features that expanded far beyond that of the original project functionality.

With that being said, there are a few places that could have been improved. In terms of the final application, security could have been further improved by obscuring the passwords using the * symbol but was not possible due to some technical limitations of the python console outputs. In terms of workflow and time management, everything was completed within planned time slots but could have been finished earlier instead of waiting until the deadline to finalize the work.