Project Approval – Team 1 L03

Members	Anando Zaman zamana11 Yuvraj Randhawa randhawy Graeme Woods woodsg1
Original project name	Social Network App (Terminal App)
Software purpose	The purpose is to provide a means of socialization whether that be for school related activities or for public conversations without having to meet in-person. This ensures that students have some form of communication with their peers without being able to physically meet them due to pandemic restrictions. Features include ability to add/delete posts, view profiles, & follow/friend users.
Software scope	The overall scope of the project is to implement several commands that can be used via the terminal to add/delete/view posts and follow users. These commands will have their respective modules that together, will result in a smooth user experience and overall functionality of the social media application. Each user will have their own profile which they can fill with general information such as name, city, country which can be viewed by other users. The users can run terminal commands for add/delete posts that will save the changes under their account. Other users can view these posts, if they exist, by running a view command followed by the users' username. All the changes will be updated under the unique userID of each user in a NoSQL database that will be queried each. As this is a simple terminal application, posts will not be encrypted and instead, will be posted as is.
URL for original project	https://github.com/sindoudou/social-network-app
Specialized hardware requirements	N/A
Software licences required	N/A
Programming language	Java, our implementation will be Python
Programming language feasibility	High
Is the domain knowledge understandable within one term?	Yes
Software license	Public on Github

Number of lines of code	1103
License allows public redevelopment	Not explicitly but it is public on Github
Able to compile original project's source code	Yes
Test Case Examples	● Functional/dynamic testing of adding/deleting posts ● **Performance testing** by measuring how long and how accurately can query posts. ● **Usability testing**: test how effectively users can modify their profiles and view their friends lists & profiles. Should be possible within 2 mins. ● **Automatic testing**: The plan for automated testing is to conduct unit tests on each of the individual submodules and assert with expected outcomes. <u>AddPost/DeletePost:</u> Assertions can be run for a given user account to check whether the new post is now visible(addPost assertion) or if it does not exist(DeletePost assertion). If AddPost is run on a user and is found that the message is not written under the specified user, then we can confirm that the module failed. Same idea for DeletePost. <u>ViewPost submodule:</u> Assertions to check if the content of the user matches the content read by the module. For example, if the user posted “I like video games”, the ViewPost module would conduct a string assertion and check if the data queried from the NoSQL DB containing the posts, is the same as the expected message of “I like video games”. If so, then the module is functioning correctly as it is able to accurately retrieve the information posted by a given user. <u>followUser submodule:</u> Assertions take place on the adjacency list. For example, if user1 follows user2, then user2 should be apart of the adjacency list of user1. We can do automated test cases for this to check if the followUser module outputs this behavior.