**User Manual**

Grading Criteria:

1 – The description of the project describing its architecture and functionality is in the word document file “Final\_Project\_Description.docx”.

2 – The User Manual is this file (Final\_Project\_User\_Manual.docx)

3 – The description of the team’s software process can be found in the world document file “Software\_Process.docx"

4 – The description of the status of the project in terms of what deliverables are completed and operational can be found in the word document file "Project\_Status.docx"

5 – Module level descriptions (use cases and the clear connection between a module and its requirements) can be found in the “Use Case” directory. Each module has an independent file within the directory that illustrates this; the Login+Persist module documentation is stored in its own folder as two separate documents. Each module has appropriately formatted docstrings within its code. The docstrings for all modules can be viewed by running the “docstring\_viewer.py” script.

6 - The interface and exports are clearly defined in the code files for each respective module and the associated documentation and use cases. The unit tests are within the Unit Tests directory and are run by running the “Unit\_Test\_Runner.py” file.

7 – There isn’t a centralized Project code file from an interface standpoint. The main “app.py” file can be viewed as project code, although it’s primarily just the different flask implementations merged together into a single file.

8 – The instructions for running the project in full is detailed below; both the actual running instructions for the main interface (ran through flask) and the unit testing instructions are given.

9 – The pip compatible installation file was not able to be created in time.

-------------------

Below are instructions for using the Quiz App created by group O for COMP 2005. Please use the linux partition of a MUN labnet computer for best results. Pull the files from the "project” repo in Clifford Bowe’s repositories and place them in a folder named "final\_project"

APPLICATION RUN INSTRUCTIONS

---------------

1. Navigate to the “final project submission (flask ready)” directory in the final\_project directory in the terminal

2. Run the command "python3 -m venv testenv"

3. Run the command "source testenv/bin/activate"

4. Run the command "pip install Flask"

5. Run the command "python app.py"

6. Navigate to the localhost:5000 address in a web browser.

7. Click Instructor then Click Register. Register a username and password. You will be taken to the Login form.

8. Login with the instructor username and password just created

9. Fill in a Quiz Name, a Number of Attempts and a Start and End Date. Click “Create Quiz”

10. Enter the name of a student that you would want to take the quiz and click “Give Access”. Repeat for as many students as desired. Then click “Add Questions”

11. Type in the question in the “Question” box. Type in the weight you would like the question to have.

12. Type in the correct answer in the “Answer” box and then click “Add Answer”

13. Type in the list of choices, including the correct answer, separated by commas. For example “R,G,B” or “3,5,9”

14. Click “Save Question”

15. Repeat steps 11 – 14 for as many questions as desired.

16. Click “Finalize Quiz”

17. You’ve now created a quiz as an instructor. Click “Log out”

18. Click Student then click Register. Register a username as password. You will be taken to the Login form.

19. Log in with the student username and password just created.

20. You should see the quiz previously created

**It is at this point that the implementation ceases to work; there was no longer time to add in the additional functionality described in the description of the Quiz app and the Modules. From this point, the student would’ve taken the quiz and then we would’ve moved through the separate View Results areas to confirm that the quiz was indeed graded and the functions were working as intended.**

UNIT TESTS

----------

1. Navigate to the final\_project directory in the terminal
2. Navigate to the Unit Tests directory in the terminal
3. Run the command “python Unit\_Test\_Runner.py”
4. Observe the results of the tests in standard python unittest format