Department of Electrical and Computer Engineering - University of Auckland

COMPSYS 302 S1 2018 – Python Project (Weighting 50%)

Background

In an age of increasing use of digital communications, there have been privacy concerns over the possibility of electronic surveillance being conducted to gain access to confidential information. The CEO of a company hopes to reduce the risk of unwanted monitoring of their communications by rival companies which compromises their ability to develop strategies for their company. For this purpose, he has approached your software development company to help develop a system that allows their executive team to communicate with each other without fear of being monitored.

A peer-to-peer social media network has been proposed by your manager, and the client thinks that it would be perfect. Your job is now to implement a functional prototype of the system. A functional prototype means that we are mainly concerned about the backend; we are not too worried about the visual design or aesthetics of the system at this stage, although it would be nice to have if possible. The client is not 100% sure about the requirements that they need at this stage, and would like to engage with the class (via the lecturer) to find out what else the developers can offer. This is an individual project, but the class will be developing the requirements and protocol together.

The system will be implemented in Python 2.7, with the help of CherryPy, a simple web framework written in object-oriented Python. To make the program easier to use, a simple web interface will be constructed for the app. Each student in the class will develop their own implementation of the system, which will be able to interact with each other. Therefore, an interfacing protocol will need to be agreed upon between the developers. The system will be used by the client on Ubuntu; therefore, the system will be developed and tested on Ubuntu.

Objective

The overall goal of this project is to design a simple peer-to-peer social media network:

1. Allowing a user to log into the system.
2. The system can automatically find other users on other computers.
3. User can create and maintain a simple profile page.
4. Users can send messages, images, audio, and PDF files to each other.

Additional elements will be decided upon in discussion between the class and the client. During project discussion times, feel free to propose additions to the lecturer, who is acting as the client for the purposes of this project. During week 11, the teaching staff will determine which additions are more valuable, and release a more complete requirements list developed in conjunction with the students.

At a bare minimum, the requirements are:
- The application must run on the Faculty Remote Linux system without us having to download any extra things or configure anything manually. We will read the README, run the application, and test it against our test application and/or other students' applications.
- The application must allow a user to "log in", see who is currently online, and send a message to someone who is online. They should also be able to receive messages and know who they are from.

The purpose of this project is to learn Networking, Web Development, Security, and Software Project Planning & Development. Languages learnt are Python, HTML/CSS, and Javascript (if desired). The project is completed **individually**.

Versioning

The use of the Git versioning system via BitBucket is compulsory and will be monitored. Steady progress from all individuals is expected throughout the semester. Students should create their own private repository ASAP with the name “UoA-CS302-2018-P2-UPI” (where UPI is your UPI) and invite the teaching staff account to access their repository.

Assessment

This project is completed and assessed individually. We do not want to see any identical projects. We will be checking for code plagiarism, and if any plagiarism is found then we will not hesitate to fail you and report you to the University Senate. There are three assessment phases (more details of the specific requirements will be explained in lectures):

1. **Python Assignment (5%) – Due 8pm Friday Week 8**
   This is a simple programming assignment that checks that you can use the Java syntax. It is administered via Coderunner - coderunner2.auckland.ac.nz and comprises five programming challenge questions, each worth 1%.

2. **Protocol Submission (5%) – Due 6pm Wednesday Week 9**
   Students will informally form groups of between 3-10 and submit a protocol proposal. More details will be provided in the class. The document should include a description of how the student proposes that the network should work, and a list of proposed APIs (including the name, description, inputs, and outputs). Submission is via Canvas.

3. **Final Project Demo + Report (40%) – Due 2pm Thursday Week 12**
   On the final demo day (in addition to submitting the code), there will be **individual** interviews with each student, where each student individually presents their social media interface, and discusses the networking elements of the project. Your design should be documented in a final report, the substantive part of which should be no longer than **four pages** (font size no smaller than 10). Submission is via Bitbucket – tag the appropriate commit with PROJECT-B-DELIVERABLES.

Reports

In the report, we recommend that you include (at least) the following elements:

- How the developed system has met the client’s requirements
- A top-level view of how the system works (diagrams help)
- One or two significant issues during development and how they were overcome
- Features that improve functionality of the system
- A brief discussion on peer-to-peer methods and its suitability for this application
- A brief discussion on the protocol and its suitability for the system
- A brief discussion on the process of developing the protocol and its suitability
- A comment on the suitability of the suggested tools (e.g. Python) for the application
- Suggested improvements for future development

A high standard is expected for these reports, including in the presentation of the reports. Feel free to include appendices (not included in page limit) if you have class diagrams, flowcharts, etc. that help aid the explanation of the system.

Final Note

You are expected to do your own planning and find out how to do things as much as possible. There will be some content delivered in lectures to help speed up this process at the beginning, but ultimately it is your responsibility to find out what you need. If you get really stuck and need help, please ask a TA or the lecturer. **Do not leave things to the last week (or last minute).**

Poorly documented or implemented code, while otherwise functionally correct, may not get you full marks for the project overall. Someone else has to be able to work on your code later! Please develop consistent and good coding practices – it will help you later in life.

The Tuesday lab sessions are compulsory, and it is strongly recommended that you interact with other students. This project will require some amount of collaboration with other students, and having others in the same place is helpful. Additionally, TAs will be there and you can discuss requirements with them as well.

Your code will be put through a similarity checker to detect plagiarism. Write your own code – do not copy someone else's! You are welcome (and expected) to discuss the project with other students (developers), particularly surrounding the protocol, but collaboration becomes plagiarism when any code is copied or written for someone else.

Provisional Lecture Plan:

	Monday	**Wednesday**
Week 6 Apr 16		**Java Project Assessment**
Week 7 Apr 23	Intro to Project Python Part 1	**ANZAC Day** Replacement Lecture on **Friday**: Python Part 2 + Part 3 Networking and Models
Week 8 Apr 30	CherryPy + APIs HTML/CSS/JS	Information and Web Security Databases Protocol Discussion
Week 9 May 7	Protocol Discussion	Protocol Discussion
Week 10 May 14		
Week 11 May 21		
Week 12 May 28		**Python Project Assessment**

Academic Integrity Notice

The University of Auckland will not tolerate cheating, or assisting others to cheat, and views cheating in coursework as a serious offence. The work that a student submits for grading must be the student's own work, reflecting his or her learning. Where work from other sources is used, it must be properly acknowledged and referenced. This requirement also applies to sources on the world-wide web. **Do not plagiarise from other students or the internet.**