Mid Semester Report

So far for the project, I have been researching Python’s various networking libraries for use in the tracking of when roommates leave the apartment. I have settled on using the basic networking features paired with a dictionary file that will be written to a CSV for use in Excel. There is also a Twitter library that I will use to create a private Twitter account that will update when people leave and return.

I currently have the preliminary network communications done for a basic app. I have been testing out communications using my phone as the client and my laptop as the server. Just for starters, to fully understand how the libraries work, I made a very basic single threaded chat app using the standard Python libraries.

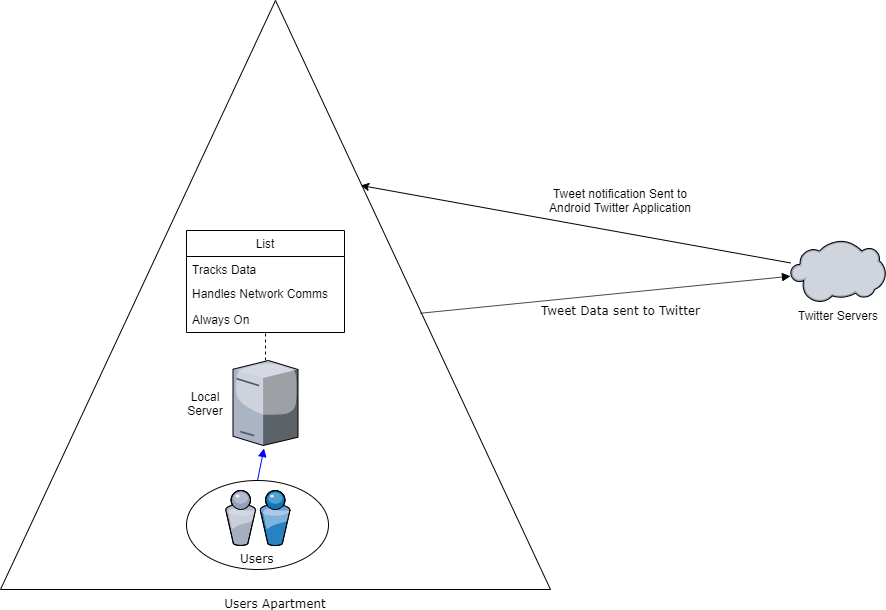
Once I have much of the network portion figured out, I plan to have the data received stored in a dictionary that is written to a CSV file any time an update is received. Eventually I may look into modifying an excel document directly instead of using a CSV file. I think this would be better for data storage since there is more that can be done with an excel document. I am considering having the information presented on a webpage and will be looking into HTTP request in Python as well for future use.

There were some modifications made to the original proposal though to simplify and cut down on cost. Original I wanted to use a raspberry pi with an attached NFC reader to track and notify when roommates left the apartment. It would send updates over the network to a main computer that would then handle updating a file with times of departure and return while also sending a notification via Twitter. Since a raspberry pi, a wireless adapter, and an NFC reader all cost money, I decided to opt for a python script that will run from the user’s phone that will send the same data to the server over the network wirelessly. Most people have a smartphone, so this simplifies things significantly and should be able to be done quicker as it takes out the electronics portion.

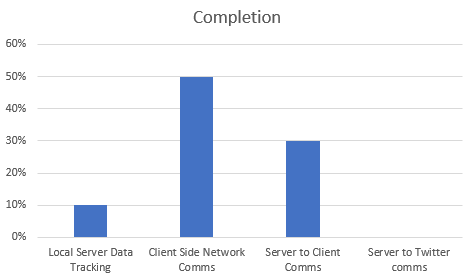
For notifications over the internet, in order to make things easier for myself, it will use Twitter. I will be creating as private Twitter account that will be updated when the server receives information from the user’s phone. This is simpler because there are open source Twitter libraries that are easy to use and implement. The Twitter library I plan to use is Tweepy for its simplistic nature and ability to the normal Twitter API.

This mixing of LAN and WAN communications is one that I think would be interesting as a college student. I am looking into doing more with it at a later date using a native Android app, but as a conceptual thing for this project, a Python script that can be ran via QPython3 will work to get complete a proof of concept and this project. A simple command line based interface isn’t the most impressive showing, but it works.

For protocols, currently I am using TCP for the client to server communications. I hopefully will be able to also use HTTP when I attempt to implement a webpage being hosted on the server that will display the current status of the roommates in the apartment. That will likely be effected by whether I can get threading working properly or if I run two different sets of code on the server computer. One to host a webpage, the other to actually manage communications with the client. I think this is quite doable and should be able to implement it in some way within the next few weeks.

 Currently, the basic script is complete for the Android phones but can be ran on a computer as well so long as Python 3 is installed. Toward the main goal, I believe the client side is probably about 50% complete and most just needs to be set up to prompt the user for the correct information and to receive data correctly.

Server side is currently still in the basic stages with mostly proof of concept code to figure out general Python network communcations. If I had to estimate, server side is maybe 30% complete. I do have a very basic network diagram showing the general flow of data that I expect for the project once everything is finished.



Overall, I believe that more progress will be made in the coming weeks as I have some very basic templates I made for myself to work with. Having these will make work faster and being able to use my phone as a test device makes things easier as I always have my laptop and phone with me, allowing me to work on the project anywhere. Mainly wanted to keep versatility of the code there so I could use it on as many platforms as possible. As I learn more about the networking portions of Python.

When researching how to use the networking libraries, I started out using the Python Documentation to get an idea of what is available to use in the standard libraries (<https://docs.python.org/3/library/ipc.html>). This lead me to look up examples of code to get an idea of how to use it. Python Module Of The Week (<https://pymotw.com/2/socket/tcp.html>) had a great tutorial about basic TCP client/server communcation that allowed me to make my initial basic chat app to get a hold of how to use the network library. Currently I am in the process of researching how to use Python threading with the network libraries in order to allow for multiple client connections to the server. I used the Python Documentation (<https://docs.python.org/2/library/threading.html>) for that but currently am unable to successfully implement it without error. Hopefully that will be fixed within the next week so I can start working on having multiple clients at once.

For future research, I will be looking into webserver hosting via python or possibly hosting a webserver with other software in order to simplify that process. I would prefer to do it via Python as a learning experience, but if needed, Apache is a viable option for this and is something I have used in the past. But based on some minor research into webserver hosting, Python shouldn’t be too hard to use for this. (<https://docs.python.org/2/library/simplehttpserver.html> , <https://docs.python.org/2/howto/webservers.html>)

I am also considering looking into implementing a GUI for the client side, but command line should be functional enough. There are a couple open source GUI libraries that I have found, the most notable being PyGUI (<http://www.cosc.canterbury.ac.nz/greg.ewing/python_gui/>) and it also seems to be the simplest to implement. But this will be a thing that I save for last and if I have time at the end to implement it.

Once the software is feature complete, I will be cleaning up the code and organizing it better for readability and to get rid of redundant code. I have done a bit of that in the server code already, but I think it will be necessary later as I add more to it. Otherwise it will be very cluttered and making commenting the code harder than needed.