Deliverable 1

Introduction

For our first deliverable, our group needed to evaluate several H/FOSS projects to work on designing an "Automated Testing Framework" for. The H/FOSS project we choose will determine how we design our testing framework. The criteria involved for deciding on which project to move forward with include:

- Code base is designed to compile and run on a Linux machine.
- The code base is available within GitHub.
- The code base can compile and run within a Linux virtual machine.

Once we were able to decide on our H/FOSS project, and compile the project. We then needed to run what-ever existing tests the software had and collect the results. In addition to the previous tasks, we were instructed to complete a team exercise which involved writing the contents of any folder a file called "myList.sh" was executed in to an HTML file.

Accomplishments

During project development for Deliverable 1, our group has decided which H/FOSS project as a candidate we would proceed with. Our group has already attempted to compile the code base to our best-known candidate, Epidemiological Modeler (STEM). We also successfully completed the team exercise and are able to execute "myList.sh" within a terminal using the BASH programming language command, "./myList.sh".

Learning Outcomes

Our team has learned a basic understanding of how to write code in BASH, and HTML. How to cooperate and communicate as a team to accomplish simple and semi-complex goals. We also learned how to operate within a Linux environment when dealing with a complex software. Lastly, we learned how to formally build a code base from a given repository.

Problems Experienced

During the writing of our wiki for our project's deliverable 1, we received feedback from our instructor to host the wiki on GitHub instead of Fandom. During the compiling of our best candidate's code base, the system would continuously report errors to where we could not execute any build-in tests available with the software. Some of the plugins in the tutorial for compiling the project are outdated, and presumably covered by newer plugins. However, upon attempting to launch STEM, a "Framework Error" stopped the launch. The developer build is still a work in progress. The standalone STEM release should work for a number of pre-programmed scenarios, however the program hangs up when starting a scenario. A screenshot of one of the errors is provided below.

Going forward

Ultimately, once one machine can properly install the packages and sort through the dependencies of STEM that contain bugs. Then we can begin developing scripts to start testing our project. If we cannot get the software running properly, then we will need to proceed to compile our second candidate, OpenAQ. In preparation for our next deliverable we need to write a test plan for our automated testing framework.

Team exercise source code is linked below:

Team Exercise: https://github.com/csci-362-01-2019/Team3/blob/master/myList.sh

Candidate breakdowns provided below:

Candidate 1: Epidemiological Modeler (STEM)

Language used: Java

- Simulates the spread of various infectious diseases throughout different populations.
- Huge library, will take some time to learn.
- Well documented with tutorials on how to get started.

Candidate 2: Open Air Quality (OpenAQ) - Languages used: JavaScript

- Reports how fresh the air quality is in a given zip code using many measurements.
- The measurements are collected from many sensors across the world.
- The software is decently documented and lengthy.
- Language will take some time to understand in order to work with project.

Candidate 3: Nightscout - Languages used: C, Java, HTML

- CGM (Continuous Glucose Monitoring) software.
- Compatible on many Operating Systems such as Windows, IOS, Android, Linux.
- Setup guarantees errors with too many dependencies to properly demonstrate from a fresh setup.
- Formal setup guides, however error bound.
- Many applications developed for the Web Browser, Android, IOS, and even several smartwatches.

Candidate 4: Drone 4 Dengue - Languages used: JavaScript, HTML

- Image processing software that detects mosquito breeding sites.
- The images are grabbed by a drone in the area.
- Not very well documented.

- Project appears to be incomplete.

  A screenshot of an error with STEM can be seen below: