Milestone 5

Team Guugle

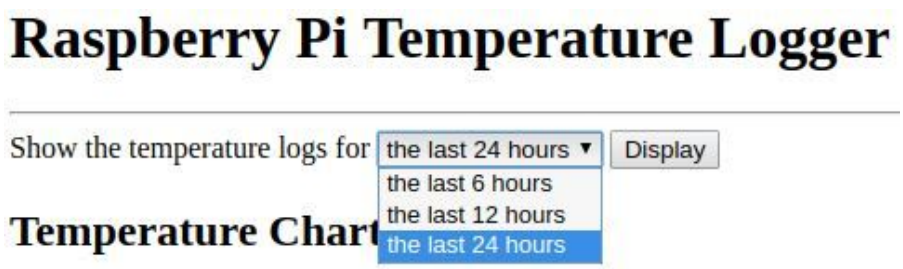
Who: Yu-Shen Chang, Kai Johnson, Lucas Laughlin, Matt Cohen, Andrea Chamorro

Title: Mobile Autonomous Database App

User Acceptance Test Plans

We want to confirm that the system meets requirements and solve any discrepancies.

Feature 1: In the app, a drop down menu gives the user the choice to choose how far back they want the graph to display. We want to test the correct dates are being displayed based on what the user picked from the dropdown menu.



We made sure that things the user would be interacting with, (user acceptance) were bing met. For this reason we implemented the following unit tests:

Feature 1: Testing that our webpage would work in both Firefox and Chrome.
Feature 2: Testing that the dropdown menu and all it's options could be chosen.

Feature 3: Testing that the display button was correctly being interacted with

Firstly we had to download the Selenium drivers for chrome and firefox, which can be found here, and placed them in the same folder as our python scripts. In this case, we found that Chrome version had to be 6.9 for it to work.

```
errorhandler.py", line 242, in check_response
    raise exception_class(message, screen, stacktrace)
SessionNotCreatedException: Message: session not created: Chrome version must be
 >= 69.0.3497.0
  (Driver info: chromedriver=2.44.609551 (5d576e9a44fe4c5b6a07e568f1ebc753f12146
34),platform=Linux 4.15.0-39-generic x86_64)
```

Once the newer chrome version was found and downloaded, the chrome script could work. Running the Selenium unit tests depended on creating "drivers" for the browser for which we wanted to test, running a "GET" request on the locally hosted page (which could only be connected to when the Raspberry Pi was running and when the database was available with data collected). It was for this reason that we did not try to run Continuous Integration on our unit tests, as there are too many nuances with connecting with a webpage that runs in these

specific circumstances. Were our webpage hosted on the internet we could have but this is not in the scope of our temperature sensing app.

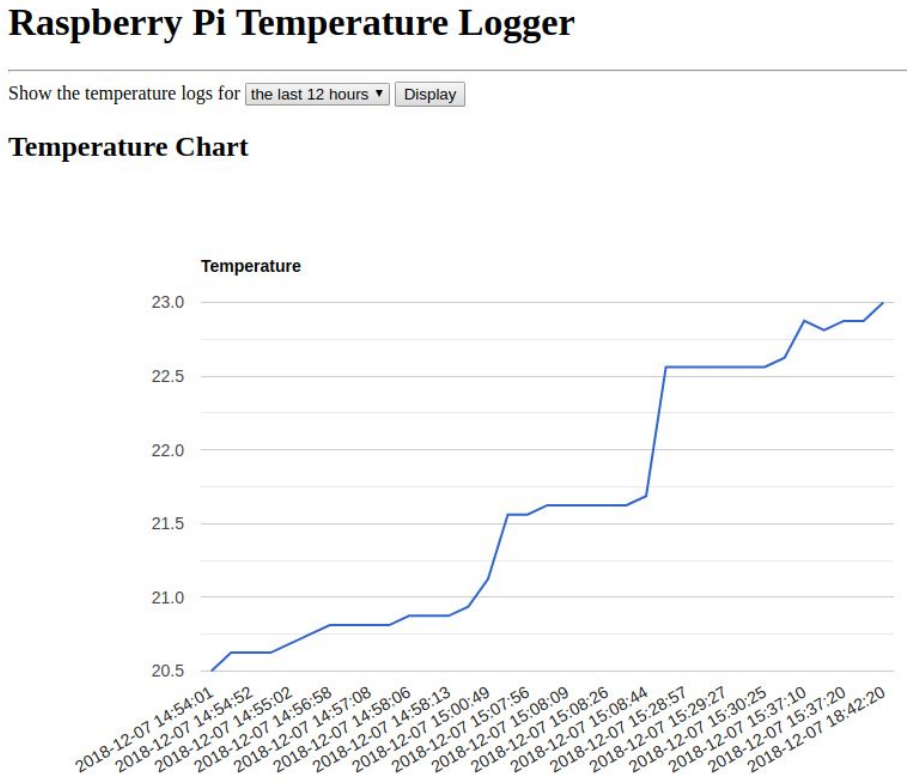
The scripts additionally test that the page is up and running , and the driver closes.
Scripts are run using:

python test_selenium.py
python test_chrome.py

The end result opens the Firefox and Chrome browsers, tests each user interface item, and closes it. Terminal displays:

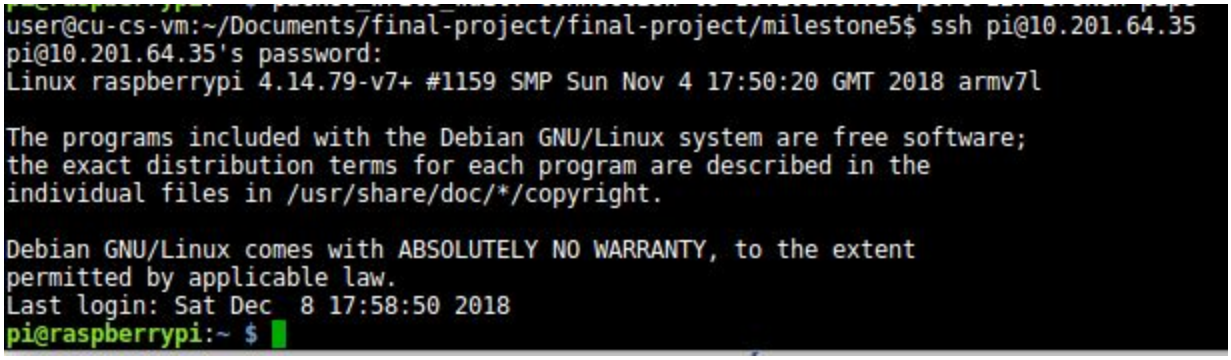
```
user@cu-cs-vm:~/Documents/final-project/final-project/milestone5$ python test_se
lenium.py
.
----------------------------------------------------------------------
Ran 1 test in 17.522s

OK
```

Point browser to http://10.201.64.35/cgi-bin/webgui2.py to get the webpage:



The Raspberry Pi is connected to using the following commands:

ssh pi@10.201.64.35
Password: csci3308



The filepath of our python scripts are:

cgi-bin/webgui2.py

You can find our unit test scripts at
https://github.com/Natty-Laugh/final-project/tree/master/milestone5

and can read the writeup.md for some quick reference information.

Test Cases

For our test cases we employed the following resources:

- **Selenium** - Used for web browser automation, use this for instructions
- **Jupyter** - Used to write the unit/compile tests

Other pertinent links include:
How to use Selenium with Python Tutorial here and here.
Selenium Python Documentation here
WebDriver Chrome documentation here