Working Prototype Known Problems Report

BMI Dashboard, Team Dash, December 3, 2015

Functions not working

1. Module Data Acquisition
    a. The Parser.py file does not parse or simulate the PV data. Only the BMS data.
    b. If there are no files in the directories specified by Parser.py, then the program will crash. The same goes for Sensor.py
    c. If the parser function in Parser.py and the mock function in Sensor.py do not receive an .eso file with semicolon separated values OR a .csv file, then the program will not correctly parse and simulate the data.

```
def parse(tsc):
    print("Cleaning Raw ESO Files")
    for filepath in glob.glob("RAW_ESO\*.eso"):
        print(filepath)
        if "CLEAN" not in filepath:
            cleanESOFile(filepath)
            parseESOKeys(filepath, tsc)
    print("ESO Files Cleaned\n")
```

```
def mock(tsc, linenr):
    sensor_file = openFile("SENSOR\MBNMS.eso").splitlines()
    raw_file = open("RAW_ESO\MBNMS.eso", "w+")

    print("Starting with line %d" % linenr)
    for data in range(linenr, len(sensor_file)):
              d = sensor_file[data]
        if d.startswith("2,", 0, 2):
            if tsc == 0:
                tsc += 1
            else:
                linenr += 1
                tsc += 1
                raw_file.write(d + "\n")
                break
        raw_file.write(d + "\n")
        linenr += 1
        if linenr >= len(sensor_file):
            linenr = -1
            break
    print("Ending with line %d\n" % linenr)
    tup = [0, 0]
```

```
    tup[0] = linenr
    tup[1] = tsc
    return tup
```

2. Data Storage
    a. In Driver.java in regards to insertion to the database we came accross a problem of data duplication. When the program is run again with the same files it will still insert that same information into the database even though there should only be one copy of the data.

```
String insertDB = "INSERT INTO BMSR2"
+"(TimeStamp, Date, Temperature, RelativeHumidity, CO_2, SensibleHeat) VALUES"
+ "(?,?,?,?,?,?)";

PreparedStatement myStmt = myConn.prepareStatement(insertDB);

…

myStmt.executeUpdate();
```

3. Data Display
    a. In file(s) bmsGraph1.js, one of the else if clauses in function updateChart(ds) is not properly reading the data from the BMSR1.csv. Therefore, the chart displays a flat line.

```
else if (ds === "SensibleHeat") {
    x.domain(d3.extent(dataset, function(d) { return d.TimeStamp; }));
    y.domain(d3.extent(dataset, function(d) { return d.SensibleHeat; }));

    line = d3.svg.line()
        .interpolate("basis")
        .x(function(d) { return x(d.TimeStamp); })
        .y(function(d) { return y(d.COtwo); });
  }
```

    b. In the file(s) bmsGraph1.js, the function parseTime(d.TimeStep) is not properly rendering the x-axis for the BMS display. Therefore, the x-axis starts with “1900” as the starting tick instead of starting in a time format. The error is in the expression (“%H:%M) passed to the native d3.time.format() function.

```
var parseTime = d3.time.format("%H:%M").parse;

...

d.TimeStamp = +parseTime(d.TimeStamp);
```

The working prototype of the BMI dashboard is located at this website:
http://kthotav.github.io/CMPS115/