**Project  3 - Mara, Belle, Stacey, Casey, Ally**

* *Basic Functionality Expected*
* Build a command line tool (Python or shell script) that takes inputs of different data about systems attached to the UNIX environment. (Input from Keyboard?)
* **Don't require the user to specify the type of input that was given (auto detect it).**
* At the end, print a summary of how many machines are up or down in the UNIX network from the input given.
* Modify above scenario that will allow data to be piped in from stdin – (From a File)
* *Additionally we need to see how much of the below they can achieve after doing the basic requirements*
  + Bonus : Add a CRON JOB which will display the number of Systems up and down at specific intervals.
* Operating System: Linux
* Language used: Shell Script, Python

**Personal Notes**

* at the moment we are to assume “system” to mean “user”
  + bonus: to model a way to detect a system online or offline
  + need to know what “system” means
  + what type of data is used to determine a “system”
* work on “auto detect” the type of input
  + we need to know the structure of username, uid for regex
* be careful of “print a summary”
  + print to stout
    - not too much, don't want to overload the terminal or user
    - pertinent information?
  + print to file
* write BOTH Python and Bash

Update as of 5/5/16

* Client requirements have been done, just need to be reviewed by client.
* Documentation is well underway and almost completed.
* We have a week to make additions to code/production side.
* Consider using python for the following:
* Statistics: find standard deviation of a user or group of users and alert when they are outside of 2 sd of norm
* graph with a heat map each users aggregate use vs time now in green and when alert goes to red
* iPython notebook for scratch work and instruction
* testing suite
* Flask to show all the pieces, especially notebook
* complete by Monday
* Note: basically this is Project 1 but using state of the art industry standards for Akamai and science communities in the area.

**Project  1 - Sara, Kendra, Evelyn, Jess A., Morgan**

* *Basic Functionality Expected*
  + Write a shell script to capture the number of active machines and some data about the machines from the Unix environment that you work in.
  + Write a CRON job to run this periodically and store it in a database .
  + Develop a Web Page to Access this data and display it over time in Tables or Graphs
* *Additionally we need to see how much of the below they can achieve after doing the basic requirements.*
  + Bonus : Capture both the machines that are up and down and show the difference between them at each sample time.
  + Extra bonus: Make the page interactive and pretty
* *Operating System:* Linux
* *Languages used:* Python and Shell Script -CRONJOB need to implement for populating the database with the required information - Background activity. HTML and JavaScript for Client side with AJAX.
* *Database :* MYSQL or SQLITE- since they are already available in UNIX
* *Web Server* - Inbuilt Web server of Python

**Project  2 - Gina, Chelsea, Hetal, Gloria, Cavaille**

* *Basic Functionality Expected*
  + Build a webpage where a user can write an SQL query into a text box and have it executed, and the results shown on the same page.
  + Use JavaScript to do SQL prettification (indentation, line breaks, etc), validation of syntax and syntax highlighting/coloring.
* *Additionally we need to see how much of the below they can achieve after doing the basic requirements*
  + Bonus points: Allow users to save queries and name them as bookmarks, so they can send people to those queries later on.
  + Extra bonus points: Save history of each query into a database automatically so users can pull them out later.
* *Operating System* : Windows
* *Languages used* : Python, HTML ,JavaScript with AJAX
* *Database:* SQLITE
* *Web Server :* Inbuilt Python Web Server

**Project 4 - Trish, Jess H., Lisa-Maria, Ginelle, Jenny, Renee**

* *Basic Functionality Expected*
  + Develop the logic/algorithms/sequence that takes an input of one or more tracking numbers and tracks the given tracking number(s) on various carrier’s websites (FedEx, UPS, USPS).
  + Difficulty: Don't use any existing Perl or Python libraries that do this. The challenge lies in developing logic for this from scratch.
  + Existing Libraries/packages can be used for reference.
  + The User interaction page should be web based
* *This project will not be a working model; this project will be treated as a Case Study*
* *Operating System*:  Linux
* *Language Used* : Shell Script, Python