**Project Description**

**Overview**

This Capstone project aims to build a sabermetrics-based MLB Fantasy League through a web application, supported by a core set of scripts that will scrape the web for up-to-date scores, stats, team and player information.

*Note: The term "stretch goal" denotes that a particular piece of functionality will be of lesser priority to the main goals, until deadlines have been met with time to spare.*

Our primary aim is to build a web application stack with an operational server box running some form of Apache sever, containing a mySQL database, atop which will be loaded a core of Java-based server-side modules to provide the follwing functionality, in addition to a hosted website providing current information to users with login credentials, as well as a client-side web application/dashboard that will provide the following services:

1)Perform statistical calculations through automation or by request,

2) Service data requests from the user's client, through a combination of PHP(orJSP), javaScript, HTML5 and Java.

3) Run scheduled Python scripts to scrape the web for new statistical data, transform it into well-formed database entities, and insert the information in the the database.

4) Perform daily, all of the overheard calculations placed in storage to keep running totals of all leagues, players, teams, scores, stats and point totals

5) Host a web application that renders all of this information visually with widgets, plug-ins, applets and whatever combination of capabilities we can stably develop before the semester deadline.

6) Provide functionality to store user accounts, securely - with all data tied to this account.

6)Time permitting a JavaFX-based dashboard that links directly to the server-side suite of functionality.

**Project Scope**

Deliverables:

* A stable automated scripting routine that provides refreshed data at a administrator-determined time-interval.
* A live website that presents all of the appropriate data contained within the database,with the following rough page organization: 1) Main Page, 2) Team Page, 3) Sabermetrics Workbench, 4) Account Settings, 5) Data Request Interface, 6) Searchable index of all of the data in the database (Stretch goal)
* A server-authenticated account functionality that ties users and all of their account information to a secure authentication service
* A Java code base that provides the following: **1)**Brokers, bridges and services the gap between the DB and the client-side user reqeuests when necessary, **2)** Provides a dynamic service application which updates league information, performs calculations and administers data mining algorithms against the database by request (Stretch goal) ,**3)** Provides a socket-based, secure direct connection to the client-side web application that will operate independantly of a web session (Stretch goal)
* A JavaFX-based web-application/plug-in that runs on the desktop, providing a cleaner, more powerful sabermetrics workbench interface for the user, with additional functionality (time-permitting)
* Upon project completion, the complete set of modules should be able to perform the following for demonstration:

1. Allow a user to sign up for an account.
2. Begin a league, complete with draft against AI opponents.
3. Compute a weeks worth of game test-data (repeatedly to simulate a week's league-time) to present how the application updates league and user information in real-time.
4. Demonstrate the scripting functionality that is automated on the back-end.
5. Demonstrate web page functionality, by performing all possible requests on each page to show that the server provides what is necessary, in a stable, bug-free manner.

* Demonstrate web appication/dashboard functionality by performing all possible actions with a live SSL connection to the server, in order to demonstrate both functional and non-functional requirements. (Stretch goal)
* All software documentation: Architecture design, class diagrams, database design and additional documents, updated with final implementation notes.

**High-level Timeline**

**1)** 2/16/2015 - Database is designed, normalized and activly consolidating data with the automated Python web scripts.

**2)** 2/23/2015 - The web stack is set up and configured, with a basic web site to access data.

**3)** 3/2/2015 - User account authentication functionality is available, as well as league set-up on the web page

**4)** 3/9/2015 - 50% of the Server-Side java modules have completed initial development, complete with unit tests.

**5)** 3/16/2015 - The website provides the basic functionality outlined above, in order to allow users to peform administrative actions on their league, team and account

**6)** 3/23/2015 - 90% of the Server-Side Java modules have completed initial develoment, complete with unit tests.

**7)** 4/6/2015 - A basic web-application is functional, and connected to the server. If we do not hit this date, we will not attempt this functionality.

**8)** 4/13/2015 - 100% of the Server-side java modules have completed initial develoment, complete with unit tests. If still active, web-application contains 50% of proposed functionality with unit tests.

**9)** 4/20/2015 - All web-site functionality is active, tested and stable. Same for server-side Java modules. 80% of the web application is complete with unit testing.

**10)** 4/27/2015 - Scripting is complete for presentation, along with all documentation, data preparation and diagnostic tests.

**11)** 5/4/2015 - Complete any last-minute functionality (with considerable unit-test before allowed in the final release) and perform a test-run of presentation.

Team Roles and Responsibilites