CSE 485 Project Initialization

Connor Alfheim   Ryan Dougherty   David Ganey   Dylan Lusi

Joseph North   Ben Roos

October 6, 2014

1 Table of Contents

1. Overview

2. Requirements

3. First Semester Plan

4. Conclusion

2 Overview

2.1 Introduction

NOTE: this is stolen from the Bowman document. We should rewrite this a bit

New radio telescopes, including the upcoming Hydrogen Epoch of Reionization Array (HERA; reionization.org), are the next generation of observational cosmology telescopes. In many engineering terms these are not the most complicated telescopes ever built, but they do involve the integration of dozens of independent computer systems and generate petabytes of highly complex data every year a situation that is very new to astronomy projects. Hence, the new telescopes present unique operational and data analysis challenges, many of which stem from the volume of data and the complexity of processing the data. Commanding the telescopes and data processing are much more automatic than with traditional telescopes, but the operational model and user interfaces for controlling and monitoring the telescopes has yet to be well-implemented. The top level user interfaces need to be developed so that users can easily monitor data flow, search meta-data, and initiate pre-defined processing steps on subsets of the data archive. Partial solutions have been developed to allow more efficient and routine observing command and analysis, however they still lack a unified interface – leaving much of the data hidden. When completed, this project will be a foundation that enables international teams of astrophysicists and cosmologists to make important cosmological observations for many years to come.

2.2 Description

The project involves all the major components of a modern web application. The foundations of the application will be on Amazon Web Services, specifically the Elastic Compute Cloud (EC2) Platform as a Service. On this server framework, MySQL will handle the database operations. Python will likely be used for some data processing scripts, and standard web tools (e.g. HTML, Javascript) will be used to handle the user interactions. While not all the technology choices have been made yet, it is clear the project will involve the entire stack.

As important as the functional web application is the ability to easily extend the web application. As this capstone project is limited in scope to just two semesters, the necessary exodus of 100% of project developers will necessitate modifications down the road by others. Since the application will primarily be used by astrophysicists, it is important that it be modifiable without a significant software development background. This will be done in multiple ways. Clean, well-written code, efficient segregation of interfaces, and minimal interdependencies will make the code easy to understand. A comprehensive version control history will make modifications easier to trace. Ultimately, the most important method of achieving future extensibility will be a simple ORM (object-relational mapping) schema and database API which should allow scientists to "plug in" other data sources and allow them to appear on the website.

2.3   Assumptions

No significant assumptions will be required for this project. Because the project involves improving an existing system, we are able to know how the system will be used and tailor the performance to those situations. If assumptions become necessary during the course of development, they will be made known to the project sponsors.

3   Requirements

The web system has the following requirements:

1. The web application should display the current status of the Murchison Widefield Array

   (a) The application should display the status of the array in a visual format

2. The application will support user accounts

   (a) Signing in will be required to act on the website, but not to read it

   (b) Users should be able to customize their experience by subscribing to various feeds, modules, etc.

   (c) Users can create accounts only with permission from a site admin

3. The application should support some data processing

   (a) Data contained in periods during which the MWA had malfunctioning nodes should be flagged

4. The team should generate an API which allows astronomers with limited knowledge of software development to add, extend, and modify the application functionality.

3.1   Deliverables

The team will deliver a functional website running on an EC2 instance. The team will be agile, so as the requirements change, so will the specifics of the delivered website. Accompanying the web platform will be documentation, to maintain extensibility in the future after the conclusion of the project.
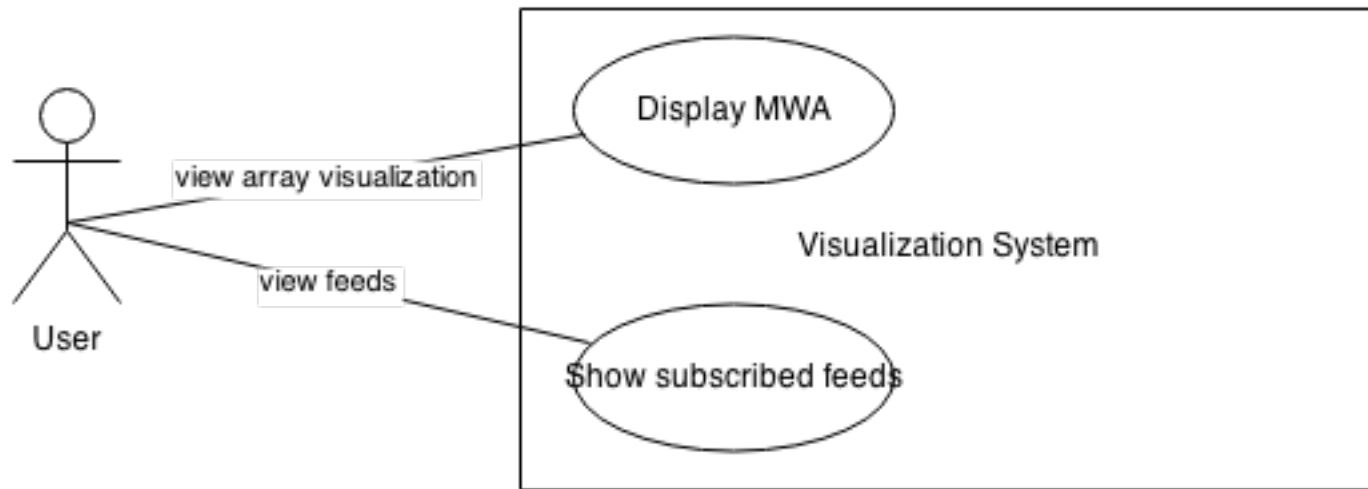
3.2   Use-case diagram

The use-cases for the website are fairly straightforward. Since the application is primarily a data viewer and status monitor, most use-cases simply involve observing data.

3.2.1 Figure 1: User with account
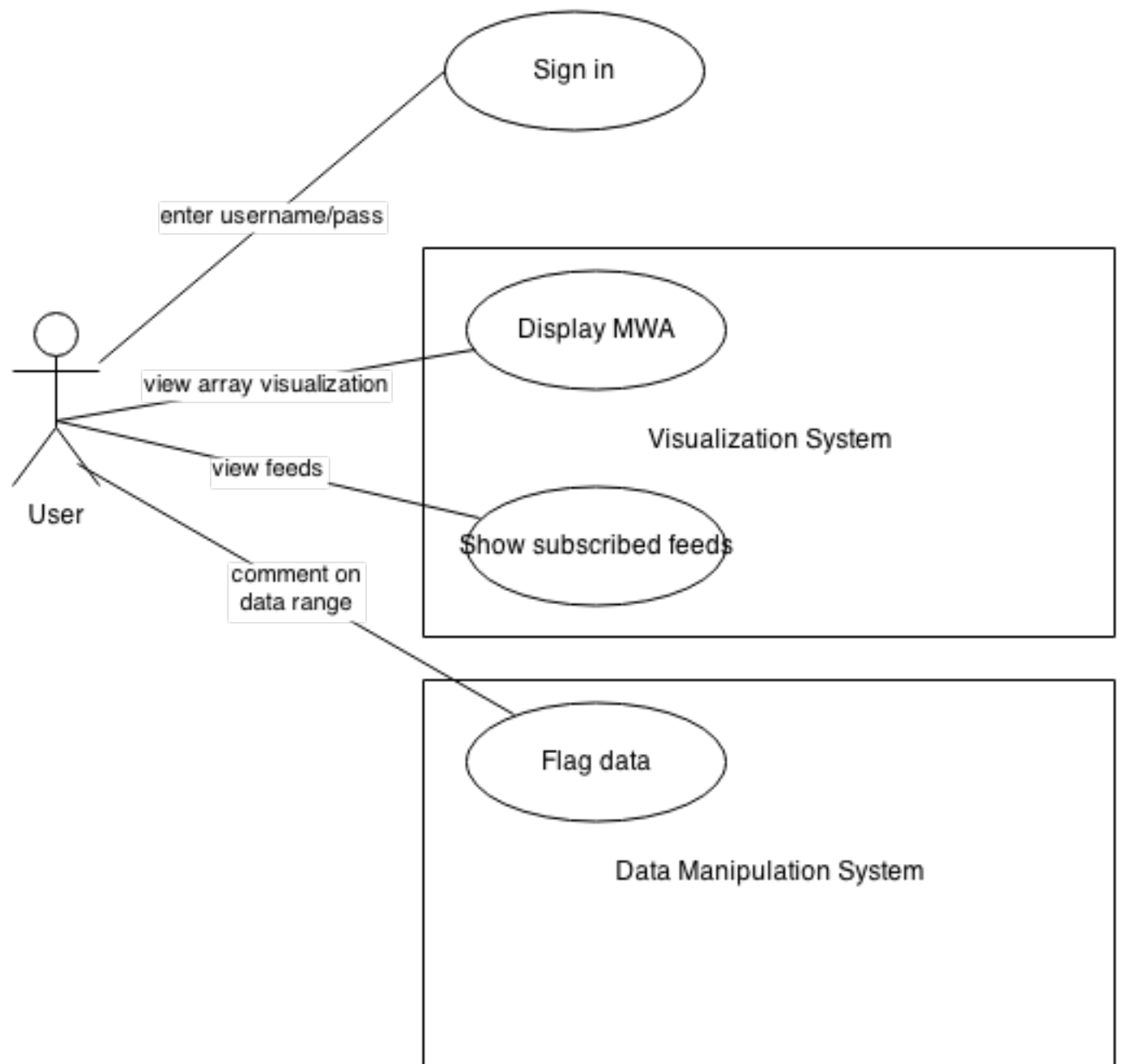
Users who are not logged in are not permitted to edit the data:

3.2.2 Figure 2: User without account

3.3 User stories

3.3.1 User not signed in

Users who navigate to the website will see a graphical representation of the Murchison Widefield Array. The visualization will make it clear if any nodes are malfunctioning. Users will be able to see other default data streams on the main page.

3.3.2 User signed in

Users who are signed in have additional privileges above standard users. Signed in users have the option to choose which data streams they see through their subscription settings. Additionally, these users can leave comments on data streams to communicate with other users.

3.4 Non-functional requirements

The primary goal of the project is *extensibility*. The API developed by the team should allow individuals who are not software developers to modify and add functionality to the website. As a result, scalability and longterm maintainability are significant nonfunctional requirements. Additionally, standard constraints on security and efficiency apply. The website should not be vulnerable to standard exploits (e.g. no passwords stored in plaintext), and should scale appropriately to deliver a pleasant experience to all users.

4   First semester plan

During the first semester, roles will be assigned to the team. All members should be functional in all areas, but the sponsors and the team agree that some team members will specialize in specific areas (e.g. the database). After dividing roles, the database team will meet with the sponsors to discuss the database schema. Due to the complexity of the database, a small portion of it will be copied to our test instance to be used in testing.

Once the EC2 instance is up and running with the database, the team will clone the existing code repository to use it as a base. Whether any of the current code remains in the website is a decision the team will make after reviewing the source code itself. The team can then begin to build features. It is likely that the basic login/logout functionality will be built first, followed by the primary visualization dashboards.

4.1   Initial scope

The initial goal of the project is to build the framework for the website. Once user functionality is enabled, different view modules will be added. During all this work, the team will make easy extensibility a high priority.

4.2   Milestones

4.2.1   Table 1

Date	Goal
10/3/2014	Project Initialization form sent to sponsors
10/11/2014	Project Initialization form complete
10/20/2014	Database copy completed
11/1/2014	Begin work
12/6/2014	When does class end?