Project Plan

for the

Observatory Scheduler

by

Jaime Acevedo

Matthew Bunch

Ryan Sharp

of

Team Observatory Project

Revision 1.0

As of: 14 October 2014

Change Log:

Revision	**Change Note(s)**
1.0	• Initial release

Reviewed and Approved By:

Name	Signature	Date
	______________________________	____________
	______________________________	____________
	______________________________	____________
	______________________________	____________
	______________________________	____________

Contents

ESTIMATION 1
High Level Architecture 1
Milestones 3
Product Backlog 6
Product Burndown Chart 6
RESOURCES 7
Time 7
Client 7
Front-End Framework 7
Back-End Framework 8
Mac OS X Server 8
SCHEDULING 8
COMMUNICATION 11
BitBucket 11
Dropbox 11
Facebook Messenger and Text Messaging 11
Communication with Client 11
QUALITY ASSURANCE 11
Defining Quality 11
Measuring Quality 12
Improving Quality 12
Additional Tools for Approaches 12
Test Plan 12
Risk Analysis 14
CS425 EXIT STRATEGY 15
Core Component 15
CS499 Exit Strategy 15

Figures

Figure 1: Simplified UML Activity Diagrams for System Architecture 2
Figure 2: Initial Project Burndown Chart 6

Tables

Table 1: Project Milestones 3
Table 2: High-Level Schedule 8
Table 3: Team Test Plan 12
Table 4: Risk Analysis Chart 14

ESTIMATION

High Level Architecture

Front-End Web Application

- Allow users to create an account and then use that account to login
- Allow users to submit a form to schedule an observation time with the observatory
- Store user-scheduled times into the database
- Write data from each valid user form to an XML file (format is XML as this time) in a format the observatory automation software can interpret

Administrator Interface

- Allow the administrator (our client) to see what and who are currently scheduled
- Allow schedule changes to be made by admin

Database Back-End

- Store observational parameters specified by user
- Store user and administrator account information
- Store results from an observation to return to the appropriate user
- Parameters will be selected from the database to write to the XML file

Email System

- Notify users of their recorded observations (recorded by automation software)
- Provide pictures taken during time with observatory telescope

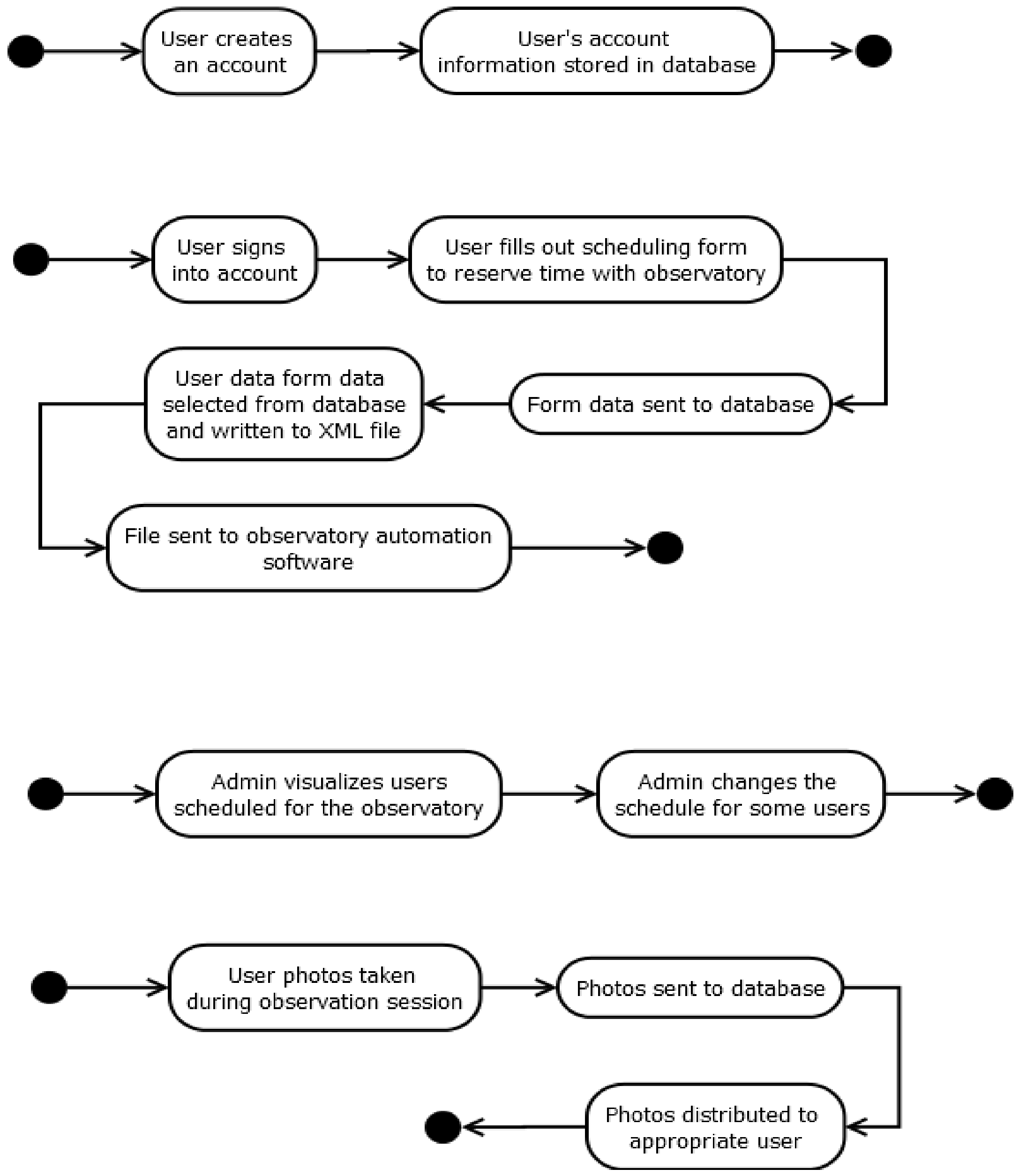
User creates an account
User's account information stored in database
User signs into account
User fills out scheduling form to reserve time with observatory
Form data sent to database
User data form data selected from database and written to XML file
File sent to observatory automation software
Admin visualizes users scheduled for the observatory
Admin changes the schedule for some users
User photos taken during observation session
Photos sent to database
Photos distributed to appropriate user

Figure 1: Simplified UML Activity Diagrams for System Architecture

Milestones

Task/Milestone	Description	Person-Hours
Creating user account system	Developing system for a user to successfully create and login to their account	14
Subtask: creating user account creation system	Writing script to allow user to create an account and send to the database	4
Subtask: creating user login system	Writing script to allow user to login to the application	4
Subtask: creating forms for login and account creation	Designing actual forms for user to fill out in order to create an account or login	2
Subtask: creating system for allowing admin control	Writing script to designate our client’s user account as an administrator	4
Testing user account creation and login	Creating dummy accounts to ensure success of account creation and login	1
Testing administrator account functions	Creating dummy admin accounts to ensure appropriate privileges are assigned	1
Creating Database	Developing database tables to hold important data	6
Subtask: creating user accounts table	Creating the table with all attributes relevant to holding user account data	2
Subtask: creating schedule table	Creating the table with all attributes relevant to hold a planned observation session. Will hold parameters	2

	submitted through the form and user ID	
Subtask: creating observations table	Creating the table with all attributes relevant to storing the observations made by a user. The observations will be pictures formatted by the telescope automation software.	2
Integrating database with user scheduling interface	Ensuring our web application's functions correspond to our database's functions and tables	6
Creating observation scheduling form	Designing form for user to fill out to schedule time with observatory	3
Subtask: creating fields for all parameters required	Determining a good layout of fields for the observational parameters to be inserted and submitted for scheduling	3
Testing form submission	Observing forms and their parameters getting sent to the database appropriately	1
Creating observing queue interface	Creating the interface the administrator will be able to use in order to see scheduled users and observing times	10
Subtask: Creating graphical UI	Designing and implementing a good layout for administrator to see scheduled users	5
Subtask: Creating system for arranging users in the scheduling queue	Designing and implementing a drag-and-drop system for re-prioritizing users	5
Testing observing queue interface	Submitting scheduled times with dummy user accounts and observing their occurrences in this UI	2

Creating email system	Designing and implementing the system for sending users their photos that were captured during their actual observation session with the telescope automation software	6
Subtask: creating system for retrieving captured photos	Designing and implementing a way to identify pictures taken by a specific user	4
Subtask: emailing users their photos	Sending all observations through a download link in their email	2
Fixing Bugs	Discovering bugs in the software and providing solutions to them	10
Testing email system	Ensuring pictures get sent to the appropriate user	1
CS425 Team Project 3 Prototype/Design	Developing a prototype of our software to demonstrate	4
CS425 Team Project 4 Presentation	Preparing PowerPoint slides to demonstrate our prototype	2
CS425 Exit Strategy Preparation	Completing a core component of the final product; Will have the functionality of writing to a file that the observatory automation software needs	5
CS425 Team Project 4 Final Presentation Preparation	Preparing slides to display our project; extra time for practice is needed	8
CS499 Exit Strategy	Writing a document to demonstrate team's understanding of the details of our software project, as well as to document tests, demonstrations, and	10

	usability studies prior to or during delivery	
System Testing and Validation	Testing and documenting the integrated system; Ensuring valid results	10
CS499 Final Presentation	Preparing to demonstrate the totality of our project; much practice is needed	10
Estimated Total Person-Hours to Complete Project	----------------------------------	**118**

Table 1: Project Milestones

Product Backlog

The Product Backlog, which is separate from this document, is based upon the above milestones. The backlog includes the estimated person-hours remaining to meet each of the milestones and their priority among the project. Because the backlog is to be updated often, it is not included here. It is a standalone document to flexibly illustrate unfulfilled requirements.

Product Burndown Chart

Below is the initial Product Burndown Chart. It has a single data point of "118". This value stands for the total estimated remaining person-hours in the project. These person-hours are described in more detail in the milestone section of this document.

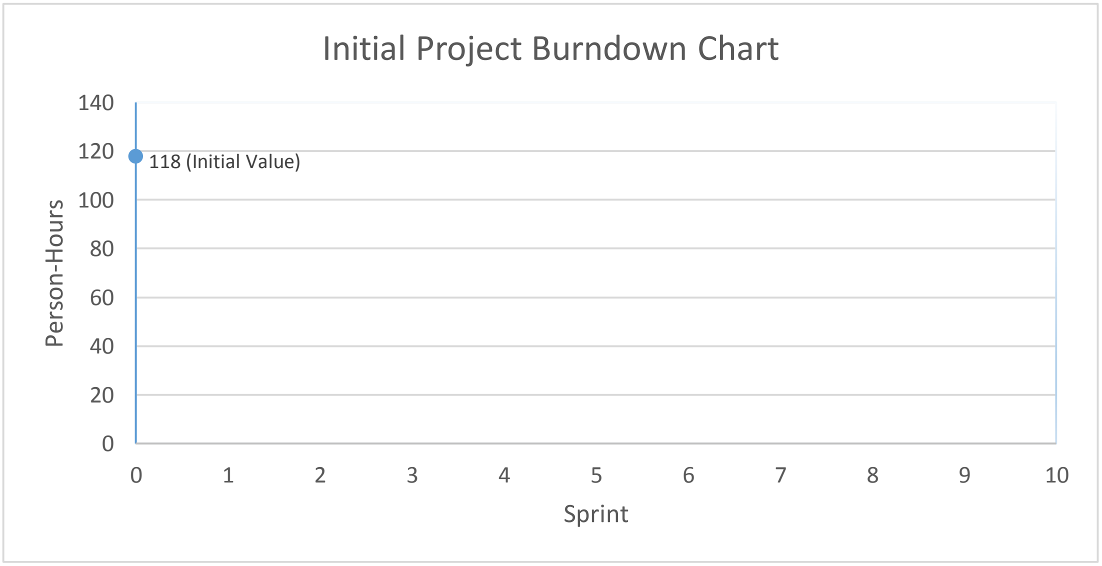

Figure 2: Initial Project Burndown Chart

RESOURCES

The Observatory Project will require resources our team has used before and some that we are not familiar with. In order to develop this project successfully, these resources will need to be understood by every member of the team. An important resource is time and time management. Each of the three members working on this project have different schedules which means individual work will be just as important as a group meeting to work on the project.

Time

Time management is essential for a group project. Each member in the group will have to plan ahead of time to work on this project. We have created an estimation of how long each and every part in the project will take. We also need to understand problems and complications are highly likely to occur and we need to tackle that issue as soon as possible. Time is a resource that is valuable and wasting it will no doubt impact our ability to finish the project. Time is a resource that we as a group need to be aware of and use as planned.

Client

The client is an extremely important resource for this project. The client is responsible for the explaining the requirements for the project and we need to be attentive to this. The observatory project uses terminology for the interface that not all of us are familiar with. An example would be the coordinates of an object in space. We will need to discuss with him how we need to properly format the user inputs in order for them to be valid parameters. Our client is also allowing us to use his server to host the database for the project amongst other things. Our client is a resource we will use frequently in order to ensure the project is a success and completes the desired requirements.

Front-End Framework

A resource we are using for the front-end side of this project is bootstrap. This framework is useful for dealing with HTML, CSS and JavaScript for the interface. Another resource we plan to use is JSP. JSP will be useful for communicating with the server and it is beneficial to us as it uses the Java programming language. These two resources will be used when we begin implementation on the user interface which is near the very beginning of that phase. Our entire group isn’t familiar with JSP which may require us to uses JSP tutorial websites as another resource. Another resource we will use is XML. We will not be coding in XML, but our generated files need to be formatted properly and read into an XML file.

Back-End Framework

The resource we are using for the back-end side of this project is the Eclipse IDE. We are using Java as the programming language. Our group is confident in our ability to use Eclipse and plugins if needed. The IDE is less familiar to us than NetBeans but we feel it is the best option for a project like this. This is a resource that will be used throughout the course of development for the observatory project.

Mac OS X Server

The two servers being used in the project are both OS X servers. OS X Server is simply an add-on package for OS X. We are familiar with UNIX which makes using OS X Server more familiar. What we are not familiar with is the group management and administration software tools provided by OS X Server. We may try to learn how these additional resources work if it improves the quality of our project as a whole. We may not have time to do this but this would not deter us from our main goal for this project. OS X Server is a hardware platform we are comfortable with and if we have the time and need to use the add-on tools provided by OS X server, we will.

SCHEDULING

Task	Specific Resources Needed	Start Date	Finish Date	Events that may affect work progress
Completion of Project Plan	Microsoft Word	September 22, 2014	October 14, 2014	Individual coursework/work schedule
Creation of Team Presentation (of Product Plan)	Microsoft PowerPoint	October 14, 2014	October 16, 2014	Individual coursework/work schedule
Team Presentation of Product Plan	Microsoft PowerPoint	October 16, 2014	October 16, 2014	Absence of team member (highly unlikely)
Creation of Prototype	HTML, CSS, JavaScript	October 20, 2014	November 4, 2014	Individual coursework/work schedule; concurrent focus on Exit Strategy may hinder progress
CS425 Exit Strategy (Core Component)	Java, Eclipse IDE	October 20, 2014	November 6, 2014	Individual coursework/work schedule; concurrent focus on Prototype may hinder progress

Creation of Prototype/Design Presentation	Microsoft PowerPoint	November 4, 2014	November 6, 2014	Individual coursework/work schedule
Team Presentation of Prototype/Design	Microsoft PowerPoint	November 6, 2014	November 6, 2014	Absence of team member (highly unlikely)
Creation of Final Team Presentation	Microsoft PowerPoint	November 6, 2014	November 13, 2014	Individual coursework/work schedule;
Practice of Final Team Presentation	Microsoft PowerPoint	November 13, 2014	November 17, 2014	Individual coursework/work schedule;
Team Final Presentation	Microsoft PowerPoint	November 18, 2014	November 18, 2014	Absence of team member (highly unlikely)
Creation of Post-Mortem Deliverable	Microsoft Word	November 20, 2014	December 4, 2014	Holiday/Thanksgiving Break (November 22 – November 29) will deter group meetings and project progress; preparation for individual final exams may be another deterrence
Conclusion of CS425	------------------------	---------------------	-------------------	End of the course
Winter Break	Not applicable	December 13, 2014	January 11, 2014	Holiday/Christmas Break; classes are not in session; this will deter group meetings and project progress
Start of Rough Outline for CS499	------------------------	---------------------	-------------------	---------------------------
Start of CS499	Not applicable	January 12, 2015	May 9, 2015	Beginning of spring classes
Creation of Core Components (Listed below)	Eclipse IDE, Java, JavaServer Pages	January 12, 2015	February 15, 2015	These components consist of the milestones and tasks designated in the estimation section
Creation of User Account System	Eclipse IDE, Java, JavaServer Pages	January 12, 2015	January 18, 2015	Individual coursework/work schedule;

Testing User Account Creation, Login, Administrator functions	Web browser, Eclipse IDE, Java, JavaServer Pages	January 19, 2015	January 21, 2015	Individual coursework/work schedule;
Creation of Database	Oracle Database, JavaServer Pages	January 22, 2015	January 25, 2015	Individual coursework/work schedule;
Creation of Form and Admin Interfaces	Eclipse IDE, Java, JavaServer Pages	January 26, 2015	February 1, 2015	Individual coursework/work schedule;
Integration and Testing of Interfaces and Database	Web browser, Eclipse IDE, Java, JavaServer Pages	February 2, 2015	February 8, 2015	Individual coursework/work schedule;
Creation of Email System	Eclipse IDE, Java, JavaServer Pages	February 9, 2015	February 15, 2015	Individual coursework/work schedule
Integration and Testing of Email System	Web browser, Eclipse IDE, Java, JavaServer Pages	February 16, 2015	February 22, 2015	Individual coursework/work schedule
System Testing and Validation	Web browser, Eclipse IDE, Java, JavaServer Pages, Server access	February 23, 2015	March 8, 2015	Individual coursework/work schedule
Creation of Additional Documentation Required	Microsoft Word	March 9, 2015	March 22, 2015	Individual coursework/work schedule
Client Review and Presentation	Web browser, Eclipse IDE, Java, JavaServer Pages, Server access	March 23, 2015	March 6, 2015	Available time to meet with client
CS499 Exit Strategy	Microsoft Word	End of March	End of March	Individual coursework/work schedule; Spring break is March 9-15; will deter group meetings and project progress
CS 499 Final Team Presentation	Microsoft PowerPoint	End of April	End of April	Absence of team member (highly unlikely)
CS 499 Post-Mortem	Microsoft Word	End of April	End of April	Individual coursework/work schedule

Table 2: High-Level Schedule

COMMUNICATION

For this project, communication is a requirement. Each member of our group needs to know where to access needed files and documents to do each part assigned to us individually and collectively. We also need to keep tabs on each other individually to allow ourselves to always be up-to-date. Ensuring proper communication with our client is also essential

BitBucket

BitBucket is an online resource that our team will make use of. We add all of our files relevant to the project to it and update accordingly. This form of communication is used for uploading the required documentation and files needed by the course instructor.

Dropbox

Dropbox is an online resource our team tends to use. We use it very similar to BitBucket but we will also have files and documents that we do not plan to use. It is also a nice backup in case our files on BitBucket were modified when we did not mean to modify it or if a file was mistakenly deleted. This form of communication is used whenever a member of the group has a file that is useful for the project.

Facebook Messenger and Text Messaging

Our group communicates almost daily via Facebook messenger or text messaging each other. We use these two technologies to get up to speed with each other and quickly find out what isn't done and what everyone is working. They are also what we use to find out when each of us are free to get together to work on the project.

Communication with Client

Our client, Dr. Sabby, is normally contacted via email. This is how we setup meeting times with him as well as retrieving answers to questions we have for him. This form of communication is used every time we need to contact the client. Our Customer Representative will typically put together the email with a list of items we need to address with Dr. Sabby. The Customer Representative will then forward his response to the rest of the group once the client has replied.

QUALITY ASSURANCE

Defining Quality

In order to achieve quality for our product we will need to define quality, measure quality, and improve quality. We will foremost, define the expectations of the product before we can define quality. As stated in the project specification, the application must be a web based and compatible on safari web browsers. The application must also be able to successfully schedule a reservation for the telescope end system. The client expects this to be a smooth and fast process. The user must immediately get feedback from the application any time a reservation is made. Finally the client has addressed that the user must be able successfully log in and out of the application.

Measuring Quality

The quality of the final end product will be measured based on the following quantitative numbers: the amount of time it takes a user to schedule an observation, the amount of time it takes a user to log in and out, the amount of time it takes the user to receive an e-mail notifying the scheduled reservation. Our team will conduct various test to make sure that these quantitative times are reasonable with the client and the application's users.

Improving Quality

Throughout the testing and implementation phases we will perform integration testing on actual future users. Based on the user's body gestures and feedback (questionnaire) of the product we will address areas of the product to improve on. Also, we will conduct meetings with the client to assure his satisfactory with the product. His feedback and criticism will be a critical part for our quality assurance. As a team it our top priority to make sure the client is happy with the overall product. The client will have the final say to any major/minor changes despite the feedback from users.

Additional Tools for Approaches

The Observatory Project will be implementing most of the application in the Eclipse IDE. The Eclipse IDE offers many potential solutions for quality assurance testing. As a team we do not have any experience using these specific tools. We will investigate what plugins/evaluation software will be appropriate for our final product.

Test Plan

No.	Type	Part type	Analysis	Archived	Results	Authorization	Objective
1	U	Creating User Account	OR	UTD	V	T	To assure account is created properly.
2	U	Log In	OR	UTD	V	T	To assure log in system functions.
3	U	Schedule Reservation	OR	UTD	V	T	To assure user gets feedback when scheduling a reservation.
4	U	Assign Role	OR	UTD	V	T	Assure the Admin has capability to assign roles.
5	I	Database Verification	OR	ITD	V	T	Assure database is properly getting written

							to and read from application
6	I	XML file creation	OR	ITD	V	T	Assure reservation module is properly communicating with the XML module
7	I	E-mail Verification	OR	ITD	V	T	Assure reservation module is properly delivering e-mails
8	I	View Pictures	OR	ITD	V	T	Assure user is able to view pictures.
9	A	Prototype	W	FTD	Q	C	Overview current prototype with client and gather feedback.
10	S	Usability	OR	FTD	Q	U	Overview current product with user and gather feedback.
11	S	Full System	W	FTD	Q	C	Final Overview of product with client and gather feedback
*Note: The below abbreviations correspond to this test plan for readability purposes							

Table 3: Team Test Plan

OR - Observe & Report
W - Walkthrough
U - Unit
I - Integration
UTD - Unit Test Documentation
ITD - Integration Test Documentation
FTD - Final Test Documentation
V - Verification Purposes
Q - Quality Improvement Purposes
A - Acceptance
T - Team
C - Client
U - Users

Risk Analysis

No.	Risk Label	Description	Likelihood	Impact	Score	Mitigation Strategies
1	Feature Creep	Client is unsure if he wants to fully automate the application.	7	8	56	Plan for extra time in the project to accommodate this request.
2	Learning/Training with JSP	For a majority this is our first web application using JSP.	5	3	15	We will consult to online resources and book documentation.
3	Incompatible Operating System	Client has mentioned that the current version of his UNIX server is in beta and may not work well with latest java compilers.	4	2	8	Apple is releasing the new and fixed version of his O.S. soon. We will conduct various test to make sure it will work.
4	Unavailable Team Member(s)	If at any time a team member cannot participate in the project.	3	8	24	We will have to revise the project plan and come up with a new strategy to finish the project.
5	No feedback from telescope	Currently we are not aware how the telescope will communicate with our application.	8	9	72	Strategize with client on how we can get some type of feedback from his telescope.
6	Database Integration Difficulty	Oracle may not be suitable to handle all aspects of our application.	2	6	12	We will consult with different types of database platforms that will fit our application needs.
7	Communication software for telescope	At the moment the client is still unsure what kind of software he wants to use to feed the telescope.	6	5	30	We will keep in constant communication with the client to figure out which software he will use.
8	Uncertainty about telescope parameter file type	It is up for debate if the telescope parameter file type is in XML or a text file. At this time XML format is assumed, as this was provided	1	2	2	File type will be changed to whichever the telescope needs to take

*Note: Likelihood and Impact ratings are on a scale of 1 (Low) to 10 (High). The score is the product of the Likelihood and Impact.

Table 4: Risk Analysis Chart

CS425 EXIT STRATEGY

Core Component

Our team has chosen to complete a core component of the final product as our exit strategy for CS425. One of the major functionalities of the observatory scheduling software will be to write user-specified observational parameters to an XML file. The XML file will get passed to the observatory automation software for interpretation. This is how the telescope knows what to look at and when to look at it. We will focus on the component of generating this specific file, given test data, and validating that this file format will work for the final implementation.

CS499 Exit Strategy

For each component of the software we will perform unit testing. This will occur before the integration of the system. Our main method of testing is by “observe and report” testing. We will observe the current functionality and report any errors we find. Once errors are found, we will immediately try to find a remedy. Detailed individual unit tests will be devised for each module. Since our software is essentially cordoned into four main components, we can determine each is working as it should be with relative ease. White box and black box testing will result. Our database will be tested by performing specific queries that will be used often. Selecting data will ensure we have the correct output and this will also be demonstrated when the XML file is written to. Form data can be tested if it had been successfully submitted by examining the data inside our database. Once integration happens we may have some physics students tests the software to determine if everything is correct. Stress testing is also a plan. We want to ensure that the system can handle many users at once. In the final system, a thorough walkthrough will be performed.