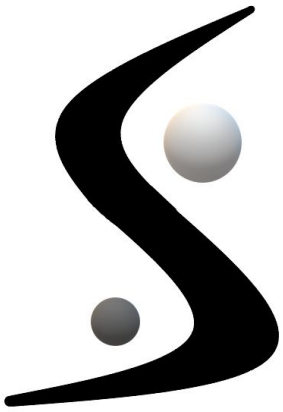
Team Sirius



Requirements Specification

11/6/2020
Project: NPOI Dashboard Web Application

Team Members:
Mario DeCristofaro
Cameron Hardesty
Hannah Park
Matt Rittenback

Sponsors:
Jim Clark
Adam Schilperoort
Peter Kurtz

Mentor:
David Failing

Version 2.1

Accepted as baseline requirements for the project:

_____	_____	_____	_____
Team Lead Signature	Date	Client Signature	Date

Table of Contents

1. Introduction **3**

2. Problem Statement **4**

3. Solution Vision **6**

4. Project Requirements **7**

4.1 Functional Requirements 7

4.2 Performance Requirements 10

4.3 Environmental Requirements 10

5. Potential Risks **11**

6. Project Plan **14**

7. Conclusion **15**

1. Introduction

The Navy Precision Optical Interferometer Web Application project revolves around the central goal of all astronomical observation which is to better understand our universe and how it works. The Navy Precision Optical Interferometer (NPOI) is the largest optical Interferometer in the world. The NPOI is an astronomical long-baseline optical interferometer that has been in operation in Anderson Mesa since 1994.The NPOI uses an array of six mirrors spaced tens of meters apart to gather data rather than a single telescope, and is so large because it combines star light collected to form a high resolution aperture. The currently collected data is invaluable to inform instrument health and performance. However it remains unusable to administrators, engineers, observers, and researchers alike because this data is so dense, located on different machines as well as on directory trees across the network in a generic text format. This project is sponsored by Jim Clark the Director of the Navy Precision Optical Interferometer Naval Research Laboratory, Remote Sensing Division as well as Adam Schilperoort and Peter Kurtz, who are both software engineers at Lowell Observatory.

The FAA estimates that the US space industry was worth about 158$ billion dollars. A lot of that money goes into research and development. With our proposed solution we should be able to play some part in helping lower costs, increase efficiency and reduce time spent. The envisioned solution is a web-based dashboard which quickly and elegantly displays relevant information based on the interest of the user. This dashboard would provide users a system in which they would be able to interact with graphs of star data relevant to building a better understanding of space. This project will at minimum provide a webpage that will be deployed to the server at NPOI with graphs that users will be able to interact with, displaying the most current observational data.

This webpage will also display the instrumentation data as plots organized by date. Users will be able to interact with this webpage to edit the configuration of the NPOI, and view individual values of data points by hovering their mouse over a scatter plot’s data. Each station will display its metadata (name of station, status, comment) on hover as well. This web application will provide administrators, engineers, observers, and researchers a more digestible way to view the data collected. This will allow the administrators, engineers, observers, and researchers to utilize the data collected at NPOI to its full potential.

2. Problem Statement

The NPOI collects and combines light from six aperuretures simultaneously to form a high spatial resolution synthetic aperture. This data is invaluable to administrators, engineers, observers, and researchers. The collected data is extremely dense and currently stored in a generic text format. Although NPOI collectes and produces invaluable data, without an organized way to view, download, and analyze the data, it is very difficult or at times impossible for administrators to utilize the data collected to understand the state of operation of NPOI. Furthermore, these data sets are needed by the engineers and observers in order for them to understand how environmental factors affect the instrument, but unfortunately in its current state, the data remains largely inaccessible and incomprehensible to most.

The current workflow of the client is shown in figure 2.1 on the next page, which starts with an astronomer or researcher going to NPOI with a need for data. The astronomer then sends a request to receive the data files for what they are searching for. Upon receiving the request from an astronomer, the technicians at NPOI then collect and email the data to the astronomer. Then the astronomer parses the data they received and then depending on the results of the parse, the astronomer may or may not return to NPOI for more data.

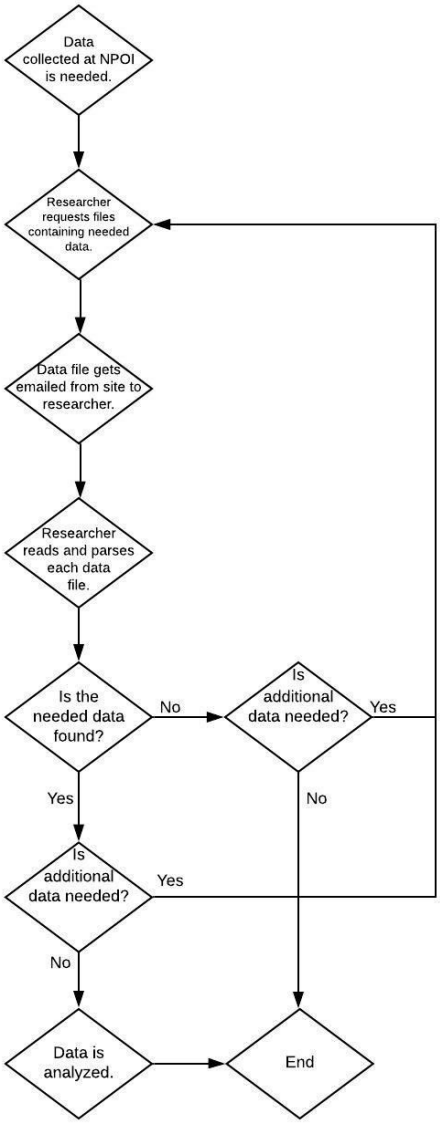


Figure 2.1: NPOI current's workflow

The current workflow for collecting data from the NPOI is not efficient, and causes confusion among end users. StarLog data is not available from the web, and therefore when researchers need access to specific data files, they must request these files be sent via email. These data files are in a generic text format, so the researchers must read through each file to access the data they need. If the data required is not found after reading and comparing each text file, the researcher must request additional files be sent via email, and this workflow begins again. The current workflow leads to delayed data analysis, and further delays the study of the health of the instrument. Studying the health and performance data of the NPOI would make improvements to

the machine possible, but with the current workflow every step of this process is delayed. That's where our solution gets put to play, as our end goal is to help reduce the workload and time spent on those plain text files. Increasing the efficiency of their research while lowering the time spent on viewing the starlog, or other data.

3. Solution Vision

Given the state of the workflow at NPOI the team has been tasked with making a web application to allow the people at NPOI to use the data they collect more efficiently. Most of the problems that are currently at NPOI stem from the fact that there is essentially no support for viewing and using the diagnostic data that they collect about the instruments. The proposed web application we are building is meant to solve these issues by providing:

- Multiple graphs to show vacuum pressure, temperature and other diagnostic readings to be analyzed
- Array configuration to be adjusted by engineers so observers can determine which starts to look at
- Display of star observation data

The application is planned to complete these tasks using the data provided by NPOI. The data is piped onto a single machine in formatted text files that will need to be parsed and have their data stored in a rudimentary database. After the data has been uploaded into a database it will then be analyzed in graphs and tables by the engineering team and the observers to ensure that NPOI is functioning properly. This will greatly improve our client’s workflow, considering any current problems have to be dealt with on site by the engineers at NPOI and there is no guarantee that the problem can be diagnosed easily by the team. With the application we are building engineers and observers alike can determine if there are issues with the array from the data being displayed and determine if the pressure or temperature are causing skewed observations or if an engineer needs to be called to the site to determine repairs. The astrometry done at NPOI is an extremely important study for us to gain new information about the universe. At the current state NPOI cannot function efficiently due to the occurrence of errors from the environment and other sources. The application built by our team is meant to mitigate these errors and continue NPOI running smoothly under conditions that would have previously caused the interferometer to fail or give observation data that is useless.

4. Project Requirements

Our solution is going to be a web-based application that displays a dashboard with graphs that contain critical information. These graphs will be used to help determine the status of each station or instrument, and also display the collected star data of the active stations.

Through the process of weekly meetings with our client and discussing the desired features of the project, we have created a number of requirements for the project. These requirements can be broken down into 3 categories: Functional, Performance and Environmental. In the following sections, these 3 categories of requirements will outline the requirements we see necessary to implement for a successful product.

4.1 Functional Requirements

The functional requirements of our project contain the core features that the product will need to contain in the final version. These core features are the backbone of the project and we break them down in the following sections.

FR 1. User Accounts

The NPOI dashboard web application requires secure log in capabilities. The log in landing page must allow for new users to create an account to access the dashboard. Based on the user the capabilities or privileges granted will alter. Administrative users will have additional capabilities that regular users do not. Administrative users will have the ability to edit data collected, and edit the capabilities or privileges granted other users. For instance a research scientist will have the ability to view and download displayed StarLog data. However the researcher will not have the capability to change the date of the collected data, or change the “researcher who collected” information. All users will have the ability to view data collected.

- Secure Login page
    - Allow for new new users to create an account
- Different user types
    - Basic user
        - View data graphs
        - View array graph

        - View machine status page
    - Admin user
        - Edit collected data
        - Edit other user's privileges

FR 2. Instrument and Star Data Graphing

The main purpose of the NPOI dashboard web application is to give access to researchers and engineers of a display of graphs that help the client visualize the instrument data collected by the NPOI and parse the observational data it collects. The data plots and graphs will have the ability to be sorted by date, month, and year. The number of data points on each scatter plot should dynamically change based on whether the plot currently displays the day, month, or year's data. The scatter plots and graphs should have the ability to zoom in while the axis adjusts automatically. Upon logging in the most recent nights data will be displayed. When data points are hovered on, the individual value of the data point will be displayed while remaining on the same web page. If additional information from the data point is needed for analysis the user will be able to click on the data point and be redirected to an additional web page. This web page will have all the information collected from the data point listed and categorized for efficient data analysis.

- Graphs
    - Labels on each axis, relevant to the type of graph
    - Options to sort by day, month, or year
    - Scatter plot resolution: dynamically change the number of points displayed based on the search parameter of the date
    - Click on a point to have specific information displayed in a window
    - Hover over a point with the mouse to see the value of the point
    - Ability to zoom with the graph automatically adjusting

FR 3. Array Graph

A very important portion of the NPOI dashboard is the ability to monitor the actual configuration of the array and its pieces. In order to accomplish this the dashboard will contain a map of the overall NPOI array with some identifier at the different stations along the array. For each type of station, sideriostats, fast delay lines, and long delay lines, hovering over their identifier will display basic information on the station. Further information on these stations will be available on a separate page for each type of station after clicking on the identifiers. These pages will contain graphs with the

diagnostic data and identifiers for which station the diagnostic data pertains to. In these pages administrators will be able to select whether a station is currently commissioned and if there are certain criteria active on the station.

- Array Graph
    - Display each station in the array
        - Sideriostats
        - FDL - Fast delay lines
        - LDL - Long delay lines
    - Display information for each station while hovering over with mouse
        - Commissioned/Partially Commissioned/Inactive
        - Vacuum pressure
        - Temperature
        - Bypass/Delay
        - In-play/Out-play
    - Display new web page after clicking a specific station to show more detailed information
        - Shows station history
        - Displays maintenance notes left by admin

FR 4. Computer Status Page

One portion of the NPOI dashboard is to ensure that connectivity between PCs, FDLcomm, FDLcon, Sidcons, and Pavo, in the array is working properly with little hitches. This means that the status of each of the PCs on the NPOI network will need to be checked regularly and posted so the users can ensure that none of the PCs have gone offline. To do this the dashboard will need to ping each of the PCs on the network and gain the data for the memory status of the PC, the ping to the computer and its connectivity to the network. This will all need to be displayed on its own page to show each of the PCs information separately and make it easily readable.

- Computer Status Webpage
    - Display each of the computers in the array
        - FDLcomm
        - FDLcon
        - Sidcons
        - Pavo
    - Display stats for each of the computers of the array
        - Memory status

- Network status
- Connectivity status
- Latency numbers

4.2 Performance Requirements

For the following requirements, we are going to detail the specific performance measures and features that the product will need to achieve. Some of the measures mentioned here for performance were times that we discussed with the client and agreed upon a reasonable expectation or based it on their current performances.

PR 1. Retrieve and Update Data Every 10 Minutes

The application will pull data from the client's current pipeline every 10 minutes to update the database and refresh any of the displayed graphs. To maintain the live stream of new data being pushed through the pipeline.

PR 2. Graph Loading Times

Graphs and tables displayed in the application will load in less than 10 seconds.

PR 3. Page Loading Times

Loading any page on the application will take less than 5 seconds.

PR 4. Page Navigation

Navigating to a specific page or back to the landing page of the application will take a user less than 5 seconds.

4.3 Environmental Requirements

The environmental requirements cover the features and requirements outside of the team’s control. These requirements include external features that the team must

conform to, such as the client's current technologies. Currently data collected at NPOI is stored on the machines at the NPOI site.

ER 1. Database Technology

Once the collected data has been parsed, it will be stored into a database. The database will be built on MariaDB, and the database engine will run on the same virtual machine as the NPOI (Git: 10.10.130.148). The database must interface with phpMyAdmin. Because phpMyAdmin is an administration tool for MySQL and MariaDB, we expect the database will interface smoothly. By utilizing phpMyAdmin and by requiring administration, the data collected at NPOI will remain maintainable under multiple locations.

ER 2. Operating System

The client is requiring that the NPOI dashboard web application must run on Ubuntu version 14.04. By utilizing Ubuntu as an operating system the NPOI dashboard web application will be stored completely or in part on the Web. This allows researchers and engineers to access the dashboard remotely as opposed to the current workflow. Furthermore by utilizing Ubuntu as an operating system administrators will have the ability to access the web application remotely.

5. Potential Risks

In the process of developing these requirements, there are some challenges the team must account for to ensure a working product for the client. In the following section we will outline each of the risks we foresee for the project and their associated likelihood and severity levels. After this we will then explain our plan to mitigate these risks if necessary.

Risk	Likelihood	Severity
Corrupted Data	Moderate	Moderate
Improperly displayed data	Moderate	Major
Unauthorized User Access	Minimal	Major
Data format changes	Minimal	Low
New technologies	Moderate	Low

Figure 5.1 Table that shows the risks of the project and their estimated likelihood and severity

Looking at figure 5.1 above, we will start with the first risk of corrupted data. This risk of potentially corrupted data originates from one of the NPOI instruments being slightly off calibration and producing a stray incorrect character in the text string that contains the instructuments metadata. The likelihood of this risk is moderate because the tools at NPOI are bound to face external factors that disrupt the calibration. The severity of these risks is moderate because if the system runs with corrupted data then it could potentially cause the system to display incorrect data that the user believes is actually correct. To mitigate this risk, we plan to implement methods of data validation and throw out any potentially bad strings of data.

The next risk is improperly displaying data, which is a risk that relates heavily to the first risk we discussed. We rated the likelihood of this risk as moderate because it is expected we will receive potentially bad strings of data from the tools. The severity of this risk is rated as major because it is a problem if the main feature of our product isn't working properly. We will work to mitigate this risk with measures of data validation and messages to the user to alert them if the system detects something wrong with the data it's reading. Through the combination of these mitigation techniques we believe we can

address this high-severity risk and ensure that the main purpose of our product functions as it should.

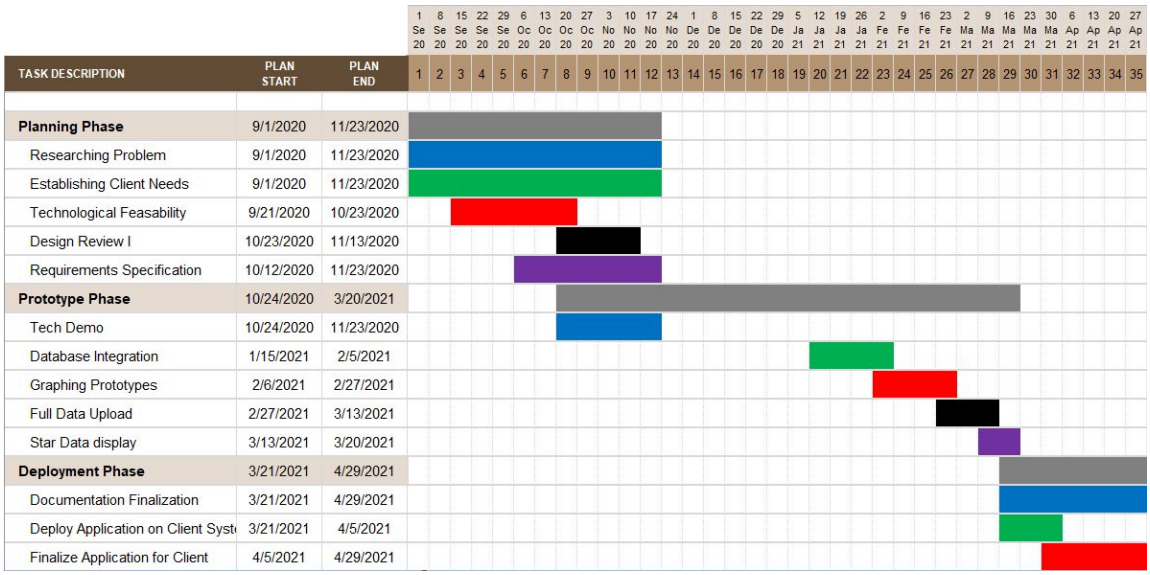
Following the risks on the table in figure 5.1, the next risk is unauthorized user access. With the client working for the Navy, we need to ensure a high level security for the system. However in our selection of the technology for building our user system, we selected a tool with excellent security features. With this selected technology we rated the likelihood of this risk as minimal. For the severity of this risk however, we rated it as major because if an unauthorized user retrieved data from our system that would be a security breach for the client in the Navy. However, as we mentioned, we mitigate this risk by selecting a web framework that boasts strong security features.

Next, the risk of data format changes is another risk we discussed with our client. In the discussion of this risk with our client, we were informed that if they were to change the format of the data from their instruments, that they would modify their data pipeline to still fit to our dashboard as we built it. The client mentioned that the data format might change if they introduce new technologies to the NPOI, so we rated the likelihood of this risk as minimal. As long as we build proper documentation of our dashboard and how it retrieves data, then the severity of this risk is rated as low because it is in the hands of the client to handle this risk. The team does not have to build the system to potentially work with future data formats that we don’t know.

For the final risk on the table in Figure 5.1, is the risk of the client implementing new technologies. This was another risk we discussed with our client, however in that discussion they informed us that it would be their responsibility to ensure that their new technologies conform to the format of our product. With this knowledge we rated the likelihood of this risk as moderate as the client indicated they would likely add new technologies in the future. For the severity we rated it low, as the client informed us that they would work to meet the specifications of our dashboard, so we don’t need to build the system to potentially accommodate for future unknown systems.

The potential risks we face, whether they be data formatting, improperly displayed data, or unauthorized access may yield some unsatisfactory results, however with proper mitigation these risks should not be a concern. The security level at NPOI to access the data is very secure and will require a user account. Also, the formatting of the data has been extensively outlined to us and we are not expecting it to change unless NPOI adds new technologies that record more data. Through a discussion with the client however, if they were to add new technologies and change the data format, they expressed that they would handle this issue of making the new changes work with our dashboard.

6. Project Plan



TASK DESCRIPTION	PLAN START	PLAN END
Planning Phase	9/1/2020	11/23/2020
Researching Problem	9/1/2020	11/23/2020
Establishing Client Needs	9/1/2020	11/23/2020
Technological Feasability	9/21/2020	10/23/2020
Design Review I	10/23/2020	11/13/2020
Requirements Specification	10/12/2020	11/23/2020
Prototype Phase	10/24/2020	3/20/2021
Tech Demo	10/24/2020	11/23/2020
Database Integration	1/15/2021	2/5/2021
Graphing Prototypes	2/6/2021	2/27/2021
Full Data Upload	2/27/2021	3/13/2021
Star Data display	3/13/2021	3/20/2021
Deployment Phase	3/21/2021	4/29/2021
Documentation Finalization	3/21/2021	4/29/2021
Deploy Application on Client Syst	3/21/2021	4/5/2021
Finalize Application for Client	4/5/2021	4/29/2021

Figure 6.1 Gantt graph of passed and upcoming milestones of project

Our plan is to perform a progressive prototyping of features after building a basic webpage to host our application. The first milestone that needs to be accomplished is the integration and creation of the rudimentary database backend we will develop. The team hopes to accomplish this by early February at the latest to allow ample time for the implementation of other major features. Following the database integration we will implement graphing prototypes using test data that mirrors the format of the real data that we will be using in the final product. We hope to complete the prototyping of these graphs by the end of February. Once we finish creating graphing prototypes we will create a system for uploading real data from the client and ensuring that it is correctly interfacing with the graphing prototypes previously made. The data uploading will be done by the middle of March at the latest to allow extra time afterward for any bug testing that would be needed. The final portion of the prototyping phase is to create a display for the star observation data that allows the scientists that work with the data to access it in a much simpler manner than before. The goal is to complete the prototyping phase in early April to allow the team to finalize the project for the client and have it fully pushed to the clients' system by the end of the spring semester.

7. Conclusion

The goal of this project is to help the clients at NPOI build a web-based dashboard that will allow them to view and monitor instrument data of the tools working to collect observational data. Currently, the client at NPOI has no easy or viable method to view the health and status of the various tools working in unison on site. Monitoring each of the components is critical because if one of the tools is even slightly out of calibration, it can lead to errors in the observed data which defeats the purpose of collecting the data in the first place. Team Sirius hopes to build a solution to address this critical issue.

In order to ensure that the team builds a proper solution to the specifications of our client, we have worked with the client in an ongoing discussion of the core requirements. In this document we outlined and detailed the requirements that we have drafted to help define the core functionality of our product. Each feature we plan to implement has been broken down into specific pieces of functionality that finely match the functionality the client is searching for.

In the process of developing the requirements for this project, we have also identified some potential risks and challenges to the project. In identifying these risks, we have analyzed them as a team to determine their severity and likelihood in the project. Then we discussed how to mitigate these risks if necessary and how we would do so.

With these requirements laid out, we are confident that the team will be able to build a web-based dashboard for our client at NPOI. This product we are creating will be vital to our client in updating their workflow and addressing the key problems of their current workflow. The dashboard will provide a new solution that allows the client to better maintain the calibration of the NPOI with much less effort and cost of time. With this more streamlined workflow, the NPOI will produce more reliable data for astronomers and researchers to pull from and utilize.