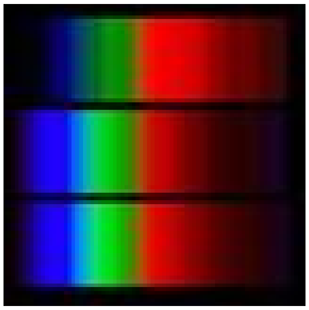


# **Requirements Specification** Version 1.0

2022-10-31

**Project Sponsor:** Joe Llama, Lowell Observatory

**Faculty Mentor:** Rudhira Talla, Northern Arizona University

**Team:** Empyrean- Henry Fye, Nhat Linh Nguyen, Jakob Pirkl, Jacob Penney, Kadan Seward

## Overview

The purpose of this document is to describe our project's functional and non-functional requirements and will form the contractual basis for the expectations to be fulfilled by our development team.

**The requirements elaborated herein are accepted for the associated software:**

*Client*:

| | |
|---|---|
| *Signature* | *Date* |

*Team Lead:*

| | |
|---|---|
| *Signature* | *Date* |

# Contents

## 1.0 Introduction

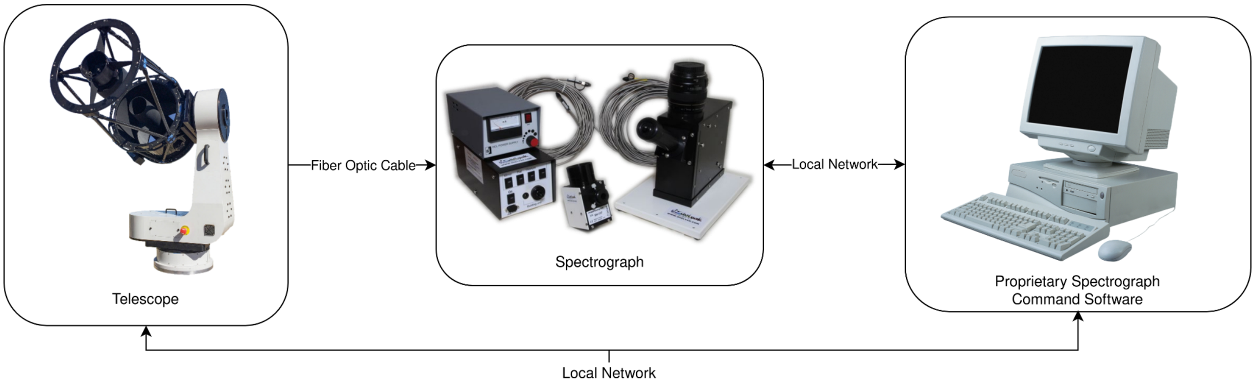
Astronomical research has proven to captivate the minds of modern people: the Space Race of the 20th century wrested the attention of millions across the developed and developing worlds; minds such as Albert Einstein, Carl Sagan, and Stephen Hawking have become icons of science, popularizing not only interest in the heavens but science more broadly; and research institutions and associated discoveries, like NASA and its illuminating missions, Flagstaff's own Lowell Observatory with the discovery of Pluto, the James Webb Telescope with its crystal-clear images, and events like the acquisition of the first image of a black hole [1], all pique and tantalize the public. In fact, astronomical activity is also interesting to and an area of moderate spending for the United States government: for example, NASA's 2022 Agency Financial Report indicates that their net cost of operations this year was $22,639,000,000 [2]. NASA themselves and institutions such as the National Science Foundation and the Marley Foundation provide millions of dollars a year in grants to other astronomical research organizations, like Lowell Observatory [3]. According to one unofficial count, there are a total of 349 other professional observatories operating telescopes in the United States right now [4].

At these institutions, researchers like our client, Dr. Joe Llama at Lowell, spend their nights and days collecting and analyzing invaluable astronomical data. In particular, Dr. Llama's "research is broadly focused on stars and exoplanets, and the interaction between the two" [5]. Currently, he uses a variety of tools "to measure the masses of Earth-sized exoplanets" and, using the data that he collects from these tools, he attempts to "understand how activity from a star from spots, faculae, and plage limits our ability to detect the smallest exoplanets" and "determine methods to remove these signals from the data we collect on other star systems" [6]. This work is undoubtedly very interesting both to the public, with the concept of habitable exoplanets gaining interest as a popular scientific topic in the past twenty years [7], and to the United States Government, who is hunting for these so-called "Goldilocks planets" [8]. Among the aforementioned tools that Dr. Llama uses to conduct this in-demand research are spectrographs, which are "instrument[s] for dispersing radiation (such as electromagnetic radiation or sound waves) into a spectrum and recording or mapping the spectrum" [9]. In Dr. Llama's case, the spectrograph is attached to the end of a telescope, which accepts light from a solar-type star and splits it into its component wavelengths. This allows the astronomer to gather valuable information, such as about the composition of objects orbiting the targeted star. One could see how this tool plays a critical role in the workflow of both our client and many other professional astronomers; with it, the hunt for habitable exoplanets becomes much easier.

Problematically for our client, operating some of Lowell's newest spectrographs is currently inconvenient and costs a great deal of time. With the workflow he has now, our client must be present at the observatory to conduct observations, with no ability to perform them remotely. Furthermore, these operations are complicated because of how "low-level" the interface exposed to the user is, contributing to the aforementioned sensation that the process is inconvenient and costly. These issues constitute unnecessary, and removable, hindrances to the work that he and other astronomers perform.

## 2.0 Problem Statement

The current process of operating the spectrograph goes as follows: the user sets up the spectrograph for use by connecting it to a telescope, which gathers light for it to process, by connecting to it via a fiber optic cable. Once setup is finished, the user commands the telescope to orient its gaze at a celestial object or area of the sky via software. Options which impact how data is gathered can be passed to the various observational tools, such as the duration of exposure to the locus. After the user has selected the relevant options, they execute the observation, during which the spectrograph refracts incoming light from the telescope which is passed by the fiber optic cable. Afterwards, a software solution enables the user to view the output of the operation, which is data about the observation which has been written to a file in a specialized file format, called **FITS**, that astronomers use in their line of work.



**Current Workflow - Hardware**

The relationship between the client's physical tools will be similar to this under the new system. The spectrograph and telescope will be arranged in a longer-term, roboticized system and the requirement to be on the local network will be eliminated, making command over the tools remote.

When considering the underlying details of how the above process is executed, our client feels that it is tedious and wasteful when it could be flexible and powerful. The newest addition to Lowell’s suite of spectrographs, which our client’s workflow and our solution concerns, is one of the finest commercially-available options for digesting visible light that exist today but, at present, he cannot utilize it without incurring significant costs. Starting from the beginning of the operational process:

1. The spectrograph does not have a dedicated system in which it is embedded in the long-term, and, as a consequence, the user must set it up and tear it down for every instance of use.
    a. This is massively time-consuming - our client reports that it can take up to four hours.

    b. The array of devices that make up this total arrangement are delicate and expensive and there is a consequent risk of damage every time the item is moved.

2. Because the observations that the astronomer wishes to make must be performed at night, the entire setup process must happen prior to then so that they can conduct their work when the time is appropriate.

3. After all of this, the astronomer must remain on premises because the software control system that is currently available to them does not permit remote operation.

4. The software solution itself does not meet our client's needs. It is

    a. dated;

    b. unnecessarily complex to use from our client's perspective;

    c. prone to time-consuming and cognitively disruptive faults, such as freezing and crashing;

    d. proprietary, and, consequently, cannot be readily improved upon by the users who depend upon it.

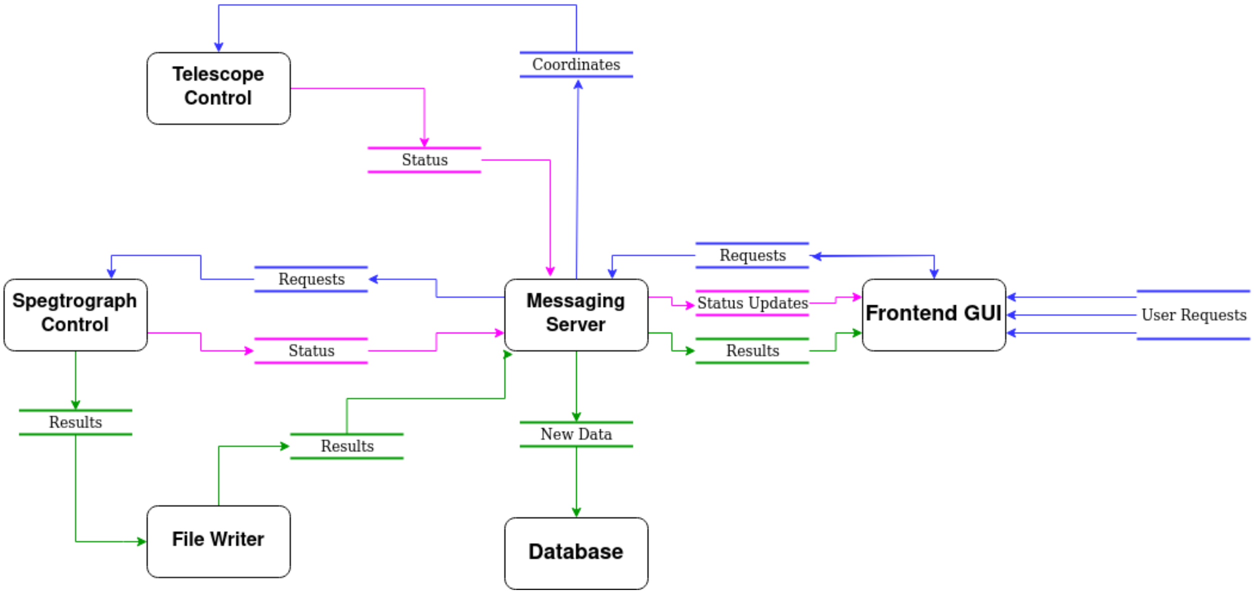
Our client will solve problem items 1 and 2 by embedding the spectrograph in a long-term roboticized system consisting of a state-of-the-art telescope to which the spectrograph will be connected by a fiber optic cable. In resolving these two items, our client will have taken care of the hardware-side of the total system and still requires a tool to manage and operate it. Our team's role in this total implementation is solving this problem by resolving items 3 and 4.

## 3.0 Solution Vision

To solve these issues, Empyrean will design a new system to allow Dr. Llama to command the spectrograph remotely, and then view the data that is subsequently collected. More specifically, Dr. Llama will be able to operate an interface that will first let them select a celestial object, or an area of the sky. Then this request is sent to the spectrograph, where it is processed and carried out. After the observation is complete, the data is sent back to our product, to be cleaned up, and presented on command for Dr. Llama to be analyzed. The following are some key features that will be included:

- An automatic object finder - so that when Dr. Llama searches for say, Betelgeuse, the location in the sky is found without any further input
- A messaging system that delivers the requests from the client to the spectrograph and the results back to the user
- An interface to view all the data collected from an observation. This will also be extended to view data collected on a specific object, or data collected on a given night.
- A message that will perform the setup of the spectrograph, including calibration
- Live status updates from the spectrograph as it operates, including progress and resulting data

Below is a diagram explaining how data will move through the system through the normal operation of requesting an object to observe, and the process of receiving results:



As can be seen, the user inputs the observation they desire into the front end GUI. This request will contain coordinates of the object or location and the duration of the exposure. This data is then sent to the messaging server, where the request is broken up and delegated to various subsystems. To the telescope, it sends the coordinates, so that the dome and telescope can move into the correct position. The spectrograph in turn receives the length of exposure so that it can properly form its observation. Each of these two subsystems at various points return their status, which the messaging server returns to the front end to be monitored. Once the spectrograph finishes data collection, it gives the raw data to the file writer to turn into a FITS file which can be used for analysis. Once finished, it sends this FITS file to the messaging server and finally back to the user. At the same time, these results are saved in the database to be retrieved in the future.

# 4.0 Project Requirements

## 4.1 Domain-level Requirements

Over the course of the design process for our product, we have conducted interviews with both of our clients and utilized documentation that our primary client provided for us which details his preliminary system architecture. From these sources, we've assembled this understanding about the key user requirements that our product must meet. The minimum viable product must provide:

**Remote access to the spectrograph:** a permitted user must be able to operate the spectrograph from any location at any time, provided that it is not currently in use. This eliminates the need for the operator to be present with the machine to operate it.

**Real-time updates from the spectrograph and telescope:** when in operation, the software solution that our team produces must listen to the spectrograph, which emits messages at intervals which discuss its current state or job. This feature contributes to solving our client's problem by making our remote solution informative about the tools it manages, contributing to a responsive and informative interface.

**Secure storage of and access to gathered data:** our solution must provide stable, secure long-term storage and easy access to data gathered through astronomical observations conducted by our client. This feature contributes to solving our client's problem because it abstracts away time-intensive data management, making our system a holistic solution to data gathering, storage, and retrieval.

**Attractive, effective visualization of gathered data:** after an observation has been made, the system must be able to present the collected data in a manner which is meaningful to the user, who we must assume will have deep knowledge of and skill with astronomical data which may contextualize our method of presentation. This feature contributes to solving our client's problem by improving upon their dated solution's interface.

## 4.2 Functional Requirements

Functional requirements are units of functionality that encapsulate the main features that users will interact with when they use the web application. They are considered as high-level functions which are then decomposed into the more detailed, lower level functionality required in the implementation of the application. Correct implementation of following specifications will ensure an outstanding experience among users.

### 4.2.1 Website

**Astronomy Tools Login:** The goal of this project is to create open source software that can be easily used by any astronomers that want to take advantage of a remotely controlled spectrograph. Our client considers the global scope of the product as the key requirement because the spectrograph and telescope are commercially available there are other institutions using similar setups to us, so we could make this control system publicly available for use by others, widening the reach of this project to beyond Lowell and NAU. However, each user is responsible for his/her own observation dataset and team Empyrean proposes the solution as the login feature. For each astronomy tool (spectrograph, telescope, etc.), the website plans to require users to login into their saved accounts in the database, and then if successfully login, the website leads users to their private pages containing user’s information, in-used astronomy tools and observation data. Otherwise, if the user is totally new to the website, they will be asked to finish a signup form including the user's information and reason to use the software. This form will be sent directly to the administration team for consideration and this team has a responsibility for a quick response in approximately 1 week.

**Previous Observations:** When astronomers finish their observation session, the observation is saved in the database and then is retrieved by the front-end to show in the table of observations. It is undeniable that the data of observation is diverse and enormous, so astronomers could access the observation data directly from the website or the compression file of the observation data. Because the number of observations is unpredictable and to avoid overload for the database, Empyrean is considering the removal of the observation data if it exists over 30 days.

**Previous Recordings:** Besides the observation data, astronomers also need the recording of the object, so that they can have an overview of movement or any developments of the stars. The recording will be retrieved from the spectrograph and our product will receive and store inside the database. Meanswhile, the front-end will generate a simple interface showing the recording in the video form and astronomers will access that page by a hyperlink next to the observation title. In addition, astronomers are likely to watch the recording online directly on the website or download to their devices in the MP4 format.

**Observations Reservation:** It is undeniable that astronomy contributed important achievements to space science in particular and humanity in general. To gain these achievements, astronomers and astronomy tools play an important role. In particular, astronomers need to utilize the astronomy tools every night and spend several hours observing the object. Our client realizes this difficulty and team Empyrean demonstrates the reservation feature for observation. More specifically, at the page for observation reservation, there are 3 main information required to be entered from users and they are time, object need to be observed and duration time for observation. After entering this information, users can reserve the observation by pressing the button “Start Observe” shown below the information box. On the observation date, the system will check the availability of astronomy tools (spectrographs and telescopes) and send an email to announce the tools availability to users. Assuming that spectrographs are available to conduct the observation, the system will collect and store the data from the observation in the filled duration time. Otherwise, the email will say clearly the problem and automatically set the observation to the next day compared to the previous booking date.

### 4.2.2 Backend API

**Remote Setup:** It is obvious that spectrographs and telescopes offer great functionality for astronomers to observe the sky. However, the setup for these tools is massively complicated in terms of hardware and software. In addition, our client mentioned the difficulty of setup on-site and his expectations to solve this problem as soon as possible. Thus, team Empyrean considers the remote setup functionality as one of the most vital parts in the MVP. More specifically, the spectrograph and other tools are assembled and installed with all the required hardwares. Then, our product will connect with tools by using an online package which is introduced by our client.

**Ease of Adapt Different Tool:** It is obvious that our product or the system needs to connect various astronomy tools including spectrographs or telescopes. The website is required to connect to different spectrographs or telescopes, which leads to reducing the operation load for one astronomy tool and increasing the productivity of the product. Currently, our client allows the team to work on the spectrograph and telescope if necessary. The back-end server connects to the tool by a package provided by Lowell Observatory and our client suggests using Python as the main programming language to utilize the package.

**Data Configuration Detection:** Besides the login feature in the website section, team Empyrean and our client considers the data configuration file as the substitution for the default login feature. The data configuration file is used to access the observation data without login into the account and astronomers can save it at their computer folders or share it to others by sending only one data configuration file. More specifically, the team confirms that the data configuration file contains the object’s name, observation time and username. User uploads the data configuration file to the website page and then the transition from front-end to back-end is going to operate. When the back-end successfully receives the configuration file, the necessary information is extracted and the back-end retrieves the certain observation data. This feature is inspired by our client, and during the development progress, team Empyrean will adjust including adding new things and delete current steps if necessary.

**Astronomy Tools Availability Detection:** The spectrograph is an un-stopped astronomy tool, which means that astronomers cannot control it when the spectrograph is doing another task. Also, the setup for spectrograph is complicated and astronomers have already set up the working spectrograph. Thus, astronomers need to wait for the current observation, then they can start their observation without resistance. To save their research time, the back-end is responsible to check the availability of spectrographs or telescopes and show the status on the website. If the current tool is busy, astronomers are free to change to others without setting up information from scratch. Currently, team Empyrean will work on 1 spectrograph and plans to accomplish the availability detection for that tool. Besides, the transition to other tools has a plan which the back-end creates and sends the observation configuration file containing the observation's information (date time, object and duration time) to an available tool.

### 4.2.3 Database

**Complex Querying:** The database must offer the ability to query the stored data in complex and variegated ways. Our client has a few definite, simple default queries that must be accessible via the front end interface, but more complex queries may be requested in the future and so should be accessible via the Relational Database Management System that we choose.

**ACID Compliance:** A very important quality of the database we choose is that it must offer strong data integrity and protections. **ACID**-compliant databases ensure that [11]

1. An operation must completely finish, or else the file, document, transaction or database reverts to its prior state, meaning that no partial changes may occur as a consequence of an operation.
2. "Any transaction you complete in the database follows the rules you or others gave to the database", meaning that rules for tables, or structures in general, cannot be broken by a query.
3. Every read and write operation occurs in isolation, meaning that the effects of disparate operations do not "layer", or happen in an overlapping manner. This slows down the database but protects data.
4. Data is written to long-term storage, meaning that operations on the database are not stored in primary memory. This prevents data corruption or loss in the case of some catastrophic failure in which primary memory fails.

The database must offer this functionality to the user to prevent the loss or damage of important data.

## 4.3 Non-functional Requirements

A program that performs all of the functions required of it does not make a complete product. Simply satisfying all of the functional requirements has the potential to leave a job unfinished. To more fully round out the expectations of a finished product, it is useful to know how fast it is expected to perform key tasks, how simple it should be for new users to pick up, how accurate certain measurements should be, or various other markers that can be tracked. The following describes the expected performance requirements that Empyrean should meet as the project is completed.

**Spectrograph Startup**: As it stands, it takes several hours to set up the spectrograph. The automated setup that will be developed will take no longer than that amount of time. This is based on the assumption that this startup task, which includes calibration, is not slow because of human error, or slowness, but because proper calibration of the tools requires significant time. Thus, the overall time of setup will not be reduced by any major factor. However, if this is not the case, and calibration does in fact require a lot of human input, then it would be reasonable to make the case that Spectrograph startup will be significantly faster once automated. This will make remote access not only possible, but preferable to how Dr. Llama's current workflow is structured.

**Response Time/Observation Time**: The time that it takes for this proposed system will take as long to make observations as the old system. Much of the observation time is the exposure, as they can be around ten minutes long for Lowell's purposes, so there is no ability to significantly reduce the time taken for the total observation. What can be measured is the response time of the spectrograph, and Empyrean will ensure that requests made around from anywhere in the world will take less than 3 seconds from the application to the spectrograph that a request is made to. This expands on the previous point to let our astronomers access their equipment at any time, from anywhere.

**Object Selection**: Selecting an object for observation will, on average, take one second. This will be much faster than the dozen seconds that it takes for the current system. Currently, astronomers must select not only the star, but find its coordinates, choose what kind of file they want their output data in, where it will be stored, and other configurations. Because of the structure of the proposed system, many of these configurations will be standardized with no choice by the user, reducing time. Compared to manually finding the place in the sky where the desired object is located, our system will be able to automatically describe the location of this object, reducing the time needed to form a request, without the need for the user to select so many other options. This helps to prioritize the ease of use for astronomers.

**Global Availability**: Due to the nature of Astronomy, much of the work needs to be done at night, meaning it is inconvenient to be at the telescope while collecting valuable information. Also, there is often a need for astronomers to be able to use telescopes they cannot physically access, which can be anywhere. Which is why this product must be able to function anywhere in the world, with no physical restrictions except an Internet connection.

**Data Availability**: Being able to easily measure data has little meaning if it is time-consuming to access said data. Therefore, access to collected data must be quick when compared to the previous method of data collection. Specifically, Empyrean will ensure that the average time to collect each record of an observation will take no longer than one half of a second as long as the user is in the same Internet subnet as the database hosting the data. This guarantee cannot be ensured globally, because as the number of records requested and distance increases, the chance that some of the data gets lost from the server to the user increases, making it inconsistent to promise such a short time frame.

**Real-time Messaging:** To keep track of the various subsystems status updates will be sent from the backend to the frontend for users to monitor. It is critical that these status messages are up to date in case something has gone wrong. Empyrean will ensure that these updates are “real-time” - meaning that within two seconds of the subsystem notifying the messaging server of its updated status, it will then update on the users frontend interface. This is to ensure that Dr. Llama has confidence the spectrograph is operating normally, even when remote.

**Ease of Use**: One of the major goals of designing a new system for managing these spectrographs is that the learning curve is quite steep - in other words, it can be difficult for new users to understand how to use their software at first. For this project, users should be able to have a simple understanding about intended usage the first time they access the app.

Upon entering the homepage for the first time, a user should be prompted to login, sending them to a landing page. This landing page should be able to lead a user to any of the major features of the site: Taking Observations, Viewing Previous Data, etc. Upon selecting any of these options, the resulting page should have plenty of prompts and help to help a new user become instantly familiar with this new system.

**Time to Unfilter Data**: In viewing their previously collected data, astronomers will be able to see the unfiltered data in case they see some data that might be configured improperly. This must be rather quick - users often resend requests after one second of inactivity. Thus, this reprocessing of the data must occur in less than five seconds. We are giving extra allowance due to the difference in user base from the average Internet user. While the average user of any given website might reload the page after one second of inactivity, astronomers know the size of data that they work with and have a more thorough understanding of the data handling and so reasonably will understand it will take time in order to process their request.

**Stalling/Reliability**: Another of the issues with the current system is that it occasionally stalls with no apparent reason. With this new system, it will be ensured that when the app is not functioning, it will either be a network error, or a physical error with the tool, such as when a motor falls off of the tracks of the observatory.

**Password Security:** Logins must be kept secure, especially when accessing expensive equipment like those kept at Lowell Observatory. Each person should have their own account, unable to access the accounts of other users, of course. Thus, we will ensure that there is secure storage of every user password. Security in this context means that none of the passwords will be stored in plaintext, only their hashed values, as well as all of the communications between the messaging server and the frontend regarding authentication need to be encrypted. This encryption should not be breakable with current non-quantum hardware in less than 10 years.

**Uptime**: Empyrean will ensure that our program will be functional no less than 99% of the time, so that from anywhere, Dr. Llama will be able to access data from past observations, or be able to make observations whenever needed. This is because software that is not operational when the astronomers need to use it hampers the collection of data that is needed to make discoveries.

**Time Between Observations:** Between observations, the data that the spectrograph took during the previous exposure must be written into a FITS file to be stored and used later for analysis. As it stands, this sequence of events can take between one and 30 seconds depending on the amount of data, but considering the data we will be working with, the time between the end of one observation, to the next observation should take no longer than twenty seconds, including the file writing, and storage into the database.

## 4.4 Environmental Requirements

Because of the nature of our open source design and based on requirements given by Lowell, there exist a few well defined environmental requirements for our solution. Firstly, Lowell asked that the technologies we used for our frontend, backend, and database were ReactJS, Python, and SQL respectively. This sets in stone the technology our product will use and eliminates a great deal of the analysis our team would have had to perform to determine which technology would have been most suitable.

**Frontend:** Our client requested that we use ReactJS because it is a popular, mature framework with a large user base and is easy to learn and work with, which is eminently important to him because he suspects that many of the people contributing to the project later on will be the astronomers who are using it. Regardless, we feel that it is an excellent option for what the project requires our team to build and it would have been a top contender in our choices of technology.

**Backend:** Our client chose Python for the backend technology for similar reasons that he chose ReactJS for the frontend: it is stereotypically very well-known by scientists and researchers whose work revolves around data collection and manipulation, is relatively easy to work with, and makes maintenance and building new features relatively painless. To implement Python in our backend, we have chosen to use Flask, a popular and simple web framework. This choice reflects our consideration for our client's reasoning in choosing Python: of the web frameworks available to use for a Python backend, we feel that Flask is the most simple.

**Database:** Our client requested that the database we use to store the data they will retrieve use a query language that is a SQL variant. Many relational database management systems (or RDMS) exist that use SQL variants, but some of these are not open source options or do not use a permissive license and, therefore, cannot be used for our project. For just one example, SQL Server, a commonly used relational database management system, is not open source and cannot be used without a license. There are only a few kinds of open source SQL-based RDMS, with the most commonly used two being mySQL and PostgreSQL. We are choosing to use PostgreSQL because it is the single most popular SQL-based RDMS and is as easy to maintain and scale as is required for this project, considering the large amount of data that the scientists at Lowell are likely to gather and store. As well, PostgreSQL is also installed easily and will work on the servers already employed by Lowell.

The only other consideration for environmental requirements in this project is the proprietary time astronomers have access to their data. Since the information gathered in our project would be used for scientific research, that data should only be able to be accessed by the astronomer(s) who collected it. Normally this time is limited to 12 months but it could be more or less based on the type of data. Ideally our project would allow this data to be automatically accessed after the time has expired, which would allow ease of use by astronomers.

## 5.0 Potential Risks

**Multiple Incoming Requests:** Due to the nature of this software, it is going to be very likely that multiple requests will be made to the spectrograph at the same time. There will need to be some sort of way to handle the possibility of multiple incoming requests to ensure issues don’t occur. There are a few different ways that this could be handled but after talking to our client it seems likely that users will only be able to use the spectrograph when no one else is currently using it. The reason for this is that a single user may want to cancel an observation in the middle of telescope movement which would be very difficult if some sort of queue was implemented for observations. To solve this, while one user account is accessing the software, another user account will see that it is in use and will not be able to use the functionalities. This user will see grayed out boxes and a message indicating that that the software is currently in use. Once the first user has finished, the second user will be able to gain access, blocking anyone else out from then accessing. This implementation is a very simple way to reduce the chances of problems occurring from multiple requests.

**Specificity of Project:** The goal of this project is to create open source software that can be easily used by anyone in a similar position as Lowell that wants to take advantage of a remotely controlled spectrograph. As the team works with Lowell, it is possible that the project will evolve to fit too small of a niche, causing its reuse in other observatories to become extremely difficult or impossible. The team will attempt to avoid this possibility by focusing on the reusability of the software, keeping it open to anyone. The nature of open source also allows for indefinite upgrades or fixes to the software over time, giving other astronomers the opportunity to enhance the code as needed.

**Data Corruption:** Data corruption is an aspect of data collection that is always possible and needs to be thoroughly considered. Astronomical data can be very important to the studies and advances made by astronomers. If this data in any way becomes corrupt or lost, it could greatly impede the workings of astronomers and may not be recreateable. The team will attempt to avoid this possibility by potentially creating backups of data and/or by ensuring that the data is as secure as possible using best programming practices.

**Competition:** Competition for a product such as this already exists but has a few issues that this project is attempting to solve. As of now, software that allows for remote spectrograph use can be costly and oftentimes does not fully fit the use case astronomers need it to. The software often has to be modified or worked around to get it to work as needed. This project will be made in the most general sense to ensure reusability and will also be free to use to avoid having cost as a large barrier to entry. Due to the nature of open source, if a new product comes out that heavily competes with this one, the open source software can be modified and evolved to provide support for ever changing needs and compete with other products.

**Observation Ownership:** The tracking of observations to their owners is a vital aspect of astronomy as the owners will likely want to conduct their own research before making the observational data publicly available. Due to the difficulty of creating a website with different accounts that have different levels of administrative privileges, it is possible the website logins will be shared accounts among astronomers. This means there must be some way to keep track of observational data which will likely require user input for who made the observation. Due to possible user error, the owners of observations may be unknown and there will have to be some decision to decide who takes ownership. To avoid this issue, a set of possible owners can be included in the website and the astronomer will have to make a selection of the owner before the data is submitted to the database.

## 6.0 Project Plan

There are three main components of this project that will make up the software: the frontend, backend, and database. The team has decided that development of these three components in the aforementioned order will best lead to our desired outcome. Below, in chronological order, are these three main tasks and their subtasks that will need to be developed.

### Frontend Development

1. Develop website (~2 weeks)

   The group will begin by spending time on developing a solid foundation for the different web pages required to support the rest of the project. This will include placeholders for a login system and the spectrograph/database functionalities.

2. Create astronomy tools login with admin accounts (~1-2 weeks)

   While developing the website an important aspect to keep in mind will be the login page and the ability to create admin accounts. This may mean that different users will have access to or will see different functionalities on a web page than others depending on their level of permissions.

### Backend Development

3. Send requests between backend and frontend (~1-2 weeks)

   Once the frontend has been mostly developed, the first order of business will be to create a messaging server that will interact with all of the moving parts. We will begin by sending and receiving simple requests to the website and gradually connecting it between the entire project.

4. Setup spectrograph/telescope remotely (~2-3 weeks)

   The meat of the project will consist of the ability to set up the spectrograph/telescope remotely. This will likely take the longest to develop as it is the most crucial aspect. The group will add on to the website the ability to submit an observation to the spectrograph/telescope and the backend will take this information and ensure that it arrives at the spectrograph correctly.

5. Enforce unique observer (~1-2 weeks)

   An important piece to implement will be the limitation set onto users while the spectrograph is in use. We want to ensure that multiple users don’t cause crossover/confusion and to combat this we will implement a requirement that only one user can access the software at a time.

**Database Development**

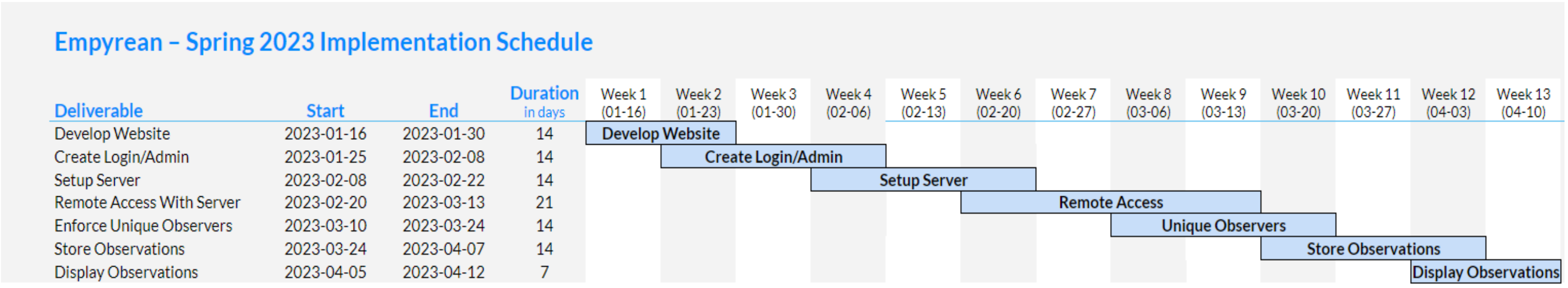
6. Store previous observations (~2 weeks)

   As we begin to develop the database, the ability to store previous observations will be created and hooked up between the UI and the database along with the processing of the observational data. This is essential for the astronomers to be able to study the observations they had created.

7. Access previous recordings (~1 week)

   Access to previous recordings will also be developed. This will need to pass from the database through the backend up to the frontend that will then show it to the user in an easily accessible table format.

The Gantt chart below displays the estimated timeline for our project milestones:



Empyrean – Spring 2023 Implementation Schedule

| Deliverable | Start | End | Duration in days |
|---|---|---|---|
| Develop Website | 2023-01-16 | 2023-01-30 | 14 |
| Create Login/Admin | 2023-01-25 | 2023-02-08 | 14 |
| Setup Server | 2023-02-08 | 2023-02-22 | 14 |
| Remote Access With Server | 2023-02-20 | 2023-03-13 | 21 |
| Enforce Unique Observers | 2023-03-10 | 2023-03-24 | 14 |
| Store Observations | 2023-03-24 | 2023-04-07 | 14 |
| Display Observations | 2023-04-05 | 2023-04-12 | 7 |

## 7.0 Conclusion

The problem that our work seeks to address is that professional astronomers who rely upon spectrographs to conduct their research do not have intuitive, robust software tools to operate them in a convenient manner. Current solutions are not only closed source, meaning that the researchers who use them (around 1,000 users, according to our client) cannot contribute to their development or maintenance, but are dated and have many issues, such as stalling, crashing, and being unwieldy to use when they do work correctly. Our solution to this problem is to develop an open source web application which will not only seek to eliminate the unwieldiness and error-prone quality of past solutions but also offer new quality-of-life enhancements, such as allowing the operator to work remotely. The consequences of successfully completing this project are that professional astronomers who use spectrographs will not only have more reliable and intuitive tools to conduct their work, but their work will no longer be bound by geographic location; conducting observations using their institution's spectrograph from the comfort of their own homes at night becomes the norm.

The web application that our team will build is tripartite. It will include:

1) A **frontend** employing a log-in system for astronomers for each observational tool attached to the system; an interface for selecting stellar objects and taking observations of them; and an interface for searching for and viewing relevant data, such as recently conducted observations, objects that have recently been viewed, and operational logs.

2) A **backend** with the ability to delegate tasks to the spectrograph, accept commands from the frontend, and store incoming data in the database.

3) A **database** that can store the astronomical data in an efficient manner and will remain scalable over its lifetime.

As displayed in the preceding sections of this document, there are many conditions that we must take into account while building each of these subsystems. Our client is preeminently concerned with creating an application that abstracts away the complexity of spectrograph operation and is easy to maintain and develop by people who are not professional software engineers. The requirements that we've outlined here recognize these interests and elaborate how we intend to accomplish them; not only do we have a firm understanding of the client's envisioned solution but also of the path to that solution. Empyrean feels confident that we can produce a tool which will provide our client and other users with a sense of reliability, trustworthiness, and satisfaction.

## 8.0 Glossary and Appendices

**FITS -** Acronym for “Flexible Image Transport System”; According to NASA’s latest standards guide, it is “the standard archival data format for astronomical data sets” [10].

**ACID -** Acronym for “Atomicity, Consistency, Isolation, Durability”; a model for database transactions which ensures data integrity through behavior specifications.

## 9.0 References

[1] Solar System Nasa. 2019. First Image of a Black Hole. Retrieved from https://solarsystem.nasa.gov/resources/2319/first-image-of-a-black-hole/.

[2] Nasa. 2022. Agency Financial Report. Retrieved from https://www.nasa.gov/sites/default/files/atoms/files/fy2022_afr_version5_111522_c.pdf.

[3] Lowell Observatory. 2022. Lowell Observatory: About Us. Retrieved from https://lowell.edu/discover/about-us/.

[4] Go Astronomy. 2022. Go Astronomy: Observatories. Retrieved from https://www.go-astronomy.com/observatories.htm.

[5] Lowell Observatory. 2017. Joe Llama: Research. Retrieved from http://www2.lowell.edu/users/joellama//#research.

[6] Lowell Observatory. 2022. Lowell Observatory: Dr. Joe Llama. Retrieved from https://lowell.edu/people/joe-llama/.

[7] Google Trend. 2022. Habitable Planets. Retrieved from https://trends.google.com/trends/explore?date=all&geo=US&q=habitable%20planets.

[8] Jet Propulsion Laboratory. 2013. In the Zone: How Scientists Search for Habitable Planets. Retrieved from https://www.jpl.nasa.gov/news/in-the-zone-how-scientists-search-for-habitable-planets

9. Merriam-Webster. Spectrograph Definition & Meaning. Retrieved from https://www.merriam-webster.com/dictionary/spectrograph

10. Fits Working Group. 2018. Definition of the Flexible Image Transport System (FITS). Retrieved from https://fits.gsfc.nasa.gov/standard40/fits_standard40aa-le.pdf

11. Preston, R. 2022. Complete guide to an acid database: Definition and benefits. *Indeed*. https://www.indeed.com/career-advice/career-development/acid-database.