VirusWatch

Technological Feasibility Analysis

Kevyn Sisante
Anas Albedaiwi
Colton Barboro
Bryan Stahman
Ziang Zhou

Project Sponsor: Dr. Crystal Hepp
Project Mentor: Volodymyr Saruta

October 23, 2020

TABLE OF CONTENTS

1. Introduction

COVID-19 has proved to be one of the most dangerous and destructive viruses humanity has faced in over a century. With more than 9 million confirmed cases and more than 230,000 deaths in the United States alone, the severity of this pandemic cannot be overstated. As the outbreak in the US approaches its third peak and a potential vaccine still months away, the American public is eager to return to some sense of normalcy. The US is administering over one million tests a day, yet there are still many ongoing challenges associated with large-scale testing. Policymakers are looking for alternate methods to identify potential outbreak hotspots in order to predict the spread of this disease.

Our client, Dr. Crystal Hepp, is an infectious disease expert that has proposed a novel solution. Because the virus is shed in human waste, she and her team have been collecting and testing sewage samples to identify new cases of COVID-19. Using the amount of the virus present in the wastewater, they can measure the extent of an outbreak in a given area. A substantial barrier associated with this detection method is how to convert wastewater test data into a streamlined and easily accessible platform. Currently, data is distributed to various agents in the form of manually formatted excel spreadsheets, which is time-consuming and slow, and it is impossible to provide information to the agents. VirusWatch aims to create a web application that will allow automated analysis of the data from these studies and host it on a secure server and be accessible to appropriate institutions seeking relevant data.

A secure web application will be built that can automatically collect and store virus data submitted by officials. The data will be organized by location, and will be viewable over specified time intervals. The application will be secure and use a roles-based architecture, so sensitive data cannot be viewed without permissions. The application will provide graphical analysis in order for stakeholders to identify potential new outbreaks.

2. Technological Challenges

The goal of this project is to create a secure web application that can host and compare any type of pathogen surveillance data so that stakeholders can identify an upcoming outbreak. The following challenges been have determined while implementing this application:

- **Roles-Based Permission Architecture** - The application needs to be secure and accessible to authorized users only. Additionally, the application will need to restrict a user's access to data based on their location. For example, a stakeholder from Flagstaff cannot view data from Sedona, etc.
- **Elegant Web-Based GUI** - The team will need to create an easy-to-use GUI for our website.
- **Visualization of Data Analyses** - The application will need to create a functional graphical viewer that allows users to analyze datasets.
- **Processing and Interpreting Data Sets** - Ability to process excel spreadsheets formatted in a certain way and store that data in our database in an organized way is needed.
- **Notifications to Stakeholders** - Ability to support sending emails and text messages is needed to sample-associated stakeholders immediately after results are available to view on our website.
- **Search Tools** - Functionality for authorized users is needed to not only retrieve datasets uploaded but can specify various criteria to develop new datasets from data points drawn from the entire database.

3. Technological Analysis

3.1 Databases

A reliable database will be needed to store user information and pathogen data for different areas across Northern Arizona. The datasets will be uploaded to our web application often, so a way to store the information in a systematic and organized manner is required. Datasets must be easily accessible, as they will need to be compared to different datasets from other areas. Additionally, our application will have to support different user roles, which will limit/grant access to certain data specified by our client. Given our application's requirements, databases will be analyzed and ranked using the following criteria:

- **Easy integration** - Our chosen database must be able to be easily integrated with our chosen framework, Django.
- **Ease of use** - Given the relatively straightforward nature of the data that must be stored, a database not overly complicated or difficult to learn will be useful. Some of our members have had experience working with SQL databases, so there is a slight preference for them.
- **Security** - Given the sensitive nature of the data will be dealt with, security of our database system is important. Therefore, the database that will be chosen should be able to handle secure authentication and data access restrictions.
- **Scalability** - With a long-term vision in mind, the project will need a database that will allow for the expansion of the application. The ability to add multiple pathogens to track, as well as the ability to add new areas and users is needed. Therefore, the database chosen must be able to handle large amounts of operations and activity efficiently.

3.1.1 MongoDB

MongoDB is a NoSQL database where each record is a document consisting of key-value pairs. This is similar to JSON objects. MongoDB is flexible and allows its users to create schema, databases, tables, etc. Mongo shell provides a JavaScript interface through which the users can interact and carry out operations such as querying, updating records, deleting records, etc. MongoDB is commonly used for real-time analytics, Big Data, mobile applications, Internet of Things (IoT). Major companies like Google, Cisco, Facebook, Expedia, Adobe and eBay utilize MongoDB in their applications. The following metrics were decided on after researching the MongoDB website, user reviews, and tutorials with example code:

- **Easy integration (4/5)** - MongoDB is not officially supported by Django, however, there are still third-party connectors that make integration possible. One of those connectors is called Djongo, which is very easy to use because it supports all django contrib libraries. Using Django, all the project would need is for the team to set up and change the model's base import.
- **Ease of use (3/5)** - Although our team members do not have much experience working with a NoSQL database, there are many free tutorials online working with and learning MongoDB, The difficulty in learning this technology would be minimal.
- **Security (5/5) -** MongoDB provides various features, such as authentication, role-based access control, encryption, and vulnerability reports. Communication may be encrypted, as well as the data. The documentation on the MongoDB website even provides a security checklist to follow to make sure our deployment is secure. All of the documentation that is needed on security is on the MongoDB website.
- **Scalability (5/5) -** MongoDB, is a distributed database, meaning there is high availability, horizontal scaling, and geographic distribution. MongoDB supports horizontal scaling through Sharding, distributing data across several machines and

facilitating high throughput operations with large sets of data. This is exactly what the application would need if it would be looking to expand to accommodate different regions and pathogens.

3.1.2 MySQL

MySQL is a relational database management system. As with other relational databases, it organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. Programmers can execute operations on the database using SQL, a domain-specific language used specifically for managing databases. MySQL is open-source, meaning it is completely free to use.

- **Easy integration (5/5)** - Django officially supports MySQL so integration should be easy.
- **Ease of use (5/5)** - MySQL is a very popular database and therefore is well documented, with many free resources our team could utilize. Some of our members are familiar with relational databases and have experience using SQL. MySQL workbench is a visual database design tool that allows us to develop, administrate, and design the database from an easy-to-use desktop application.
- **Security (5/5)** - MySQL includes several plugins that implement security features. There is a password-validation plugin for implementing password strength policies and assessing the strength of potential passwords. The MySQL Enterprise Firewall allows database administrators to permit or deny SQL statement execution based on matching against lists of accepted statement patterns. This helps thwart SQL injection attacks. However, some of the security features are only available for the "Enterprise Edition" meaning it would cost the team in order to use them.
- **Scalability (4/5)** - MySQL is able to be scaled both horizontally and vertically. However, SQL databases are harder to scale horizontally compared with NoSQL databases.

3.1.3 Oracle

Oracle Database is a multi-model database management system that uses SQL. It is a database commonly used for running online transaction processing and data warehousing workloads. Oracle offers Oracle Autonomous Database providing fully automated operation procedures. Oracle is free for download and use in a development environment, however, payment is required when the product is deployed.

- **Easy integration (5/5)** - Django officially supports Oracle so integration should be easy.
- **Ease of use (3/5)** - From what the team has gathered from forums, Oracle is less straightforward and more difficult to learn than the other options. However, there are still plenty of free resources available to utilize.
- **Security (5/5)** - Oracle offers top-of-the-line multi-layered security including controls to evaluate risks, prevent unauthorized data disclosure, detect and report on database activities and enforce data access controls in the database with data-driven security.
- **Scalability (1/5)** - With Oracle, cost is a big inhibiting factor in expansion. The personal edition is free to use, but there is a data limit. The enterprise edition's exact cost depends on how many features would be desired, but it is ultimately way too expensive.

3.1.4 Chosen Approach

Table 3.1 is a summary of the results of each database researched based on the set criteria chosen:

Table 3.1 Ranking of different databases that were considered based on the set criteria from 0 (worst) to 5 (best)

	Easy Integration	Easy Of Use	Security	Scalability	Totals
MongoDB	5	5	5	4	**19**
MySql	4	3	5	4	16
Oracle	5	3	5	1	14

MongoDB was ultimately chosen as the database for this project. It meets all of the specified criteria. It is directly supported by Django, as well as secure and scalable. Furthermore, there are many free online resources and documentation available to utilize.

3.2 Web Application Framework

Alternatives to Node.js were highlighted from programming forums on the internet, from users who tried to develop the given characteristics using Node.js and two other alternatives: Ionic and React Native.

3.2.1 Node.js

Building a web application that meets client's objectives of user-friendliness, saving order items, outlining the costs, discounts, coupons, and vouchers to the clients and issuing promotion information is not easy. For these to be fulfilled, specific functions need to be used in developing the web application (Lal, 2017). For the final product, these functions play the role of enhancing the customer experience and making for a viable return on investment (ROI).

Desired Characteristics

The web application's desired characteristics should also meet the business objectives (Haesen et al., 2008). These characteristics include a save function for saving the customer orders, a summary page for highlighting the order costs, aside pane for highlighting the available discounts, promotions, and vouchers, an easy to navigate user interface to improve user experience and push notification to allow for sending of messages. Node.js features include scalability, speed, asynchronous APIs, a single-threaded model, zero buffering, open-source, and has an available license. These features should enable the desired characteristics to be programmed within the web application.

3.2.2 React Native

React Native is a mobile application framework that was developed by Facebook Inc. Facebook is one such application that was also built using React Native to improve the mobile application development, which supports operating systems such as Android and iOS. It is touted to boost performance while maintaining smooth animations.

Table 3.2.1 Rankings and comparison of Node.js and React Native based on the set criteria from 0 (worst) to 5 (best)

Framework	Application characteristics					
	Save function	summary page	side pane	Easy to navigate user interface	Push notification	Totals
Node.js	3	**4**	**4**	2	**5**	18
React Native	3	3	3	**5**	**5**	**19**

Based on Table 3.2.1, React Native is the best option to go with, given its high ranking. One outstanding quality with React Native, from the discussion forums, is that it is touted as excellent in developing the user interface. This is important in improving customer experience with the web application.

Proving feasibility

React Native can be tested on specific units in a piece of code, such as a function, specific files, strings, and arrays. This makes it possible to test each of the desired characteristics while under development.

3.3.3 Express.js

Express.js is a web application framework. It is also a minimal Node.js framework used in building hybrid, multi-page, and single page websites. The framework helps build applications that can handle requests such as DELETE, POST, GET, and PUT. Express.js framework is suitable for the fast-tracking development of server-based applications. Thus, this framework alone may not be suitable for developing a fully-fledged website. This may not necessarily be a major downside given that the website application that needs to be developed requires four key functions that may be executed using it. For example, the DELETE request can be used in search options and the saved items, while the GET request can be used in search engines. However, the user interface and the side pane may present a challenge in this case.

Desired characteristics

The desired characteristics for the web application includes functions that can handle four key features: save function for saving the customer orders; a summary page for highlighting the order costs; aside pane for highlighting the available discounts, promotions, and vouchers, an easy to navigate user interface to improve user experience (Steed and Oliveira, 2010), and push notification to allow for sending of messages. Express.js may prove useful in handling requests that involve its four key functions

DELETE, POST, GET and PUT. However, more is required in terms of the web page display.

Alternatives

Alternatives to Express.js were highlighted from programming forums on the internet, from users who tried to develop the given characteristics using two other alternatives: Ionic and React Native.

Analysis

The analysis of each of the given features was done by checking online forum discussions and comparing each of the desired characteristics.

Table 3.2.2 below highlights the findings of the analysis carried out on each of the chosen alternatives.

Table 3.2.2. Ranking of Express.js based on the set criteria from 0 (worst) to 5 (best)

Framework	Application characteristics					
	Save function	summary page	side pane	Easy to navigate user interface	Push notification	Totals
Express.js	3	3	2	4	5	17

React Native has the highest rating from the table in terms of the five key characteristics chosen for the web application. Express.js has some good ratings, better than those in Ionic. React Native is thus the option to choose from among these three highlighted frameworks.

Proving feasibility

React Native can be tested on specific units in a piece of code, such as a function, specific files, strings, and arrays. This makes it possible to test each of the desired characteristics while under development.

3.2.4 Django

Introduction

Django framework was created to ease the creation of complex websites that were also data-driven. Some of the well-known websites that run on Django include Instagram, which uses a large number of images and texts exchanged between clients on the platform. Django is especially important for developing web application, given its SEO optimization, high security, and speed. Django offers more and from the outset, which seems to be a suitable framework for developing the e-commerce web application.

Desired characteristics

The desired characteristics for the web application includes functions that can handle four key features: save function for saving the customer orders, a summary page for highlighting the order costs, aside pane for highlighting the available discounts, promotions, and vouchers, an easy to navigate user interface to improve user experience, and a push notification from allowing for sending of messages. Django has numerous features that can help in fulfilling these desired characteristics for the web application. For example, Django's templates are an excellent way of building a user interface, while its python programming capabilities can create the sidebar for promotions and coupons.

Alternatives

Alternatives to Django were highlighted from programming forums on the internet, from users who tried to develop the given characteristics using two other alternatives: Ionic and React Native.

Analysis

The analysis of each of the given features was done by checking online forum discussions and comparing each of the desired characteristics.

Table 3.2.3 Ranking of Django framework based on the set criteria from 0 (worst) to 5 (best)

Framework	Application characteristics					
	Save function	summary page	side pane	Easy to navigate user interface	Push notification	Totals
Django	4	3	4	5	5	21

From Table 3.2.3, Django proved to be a better framework from the given three choices. It proved to be better than React Native, which took the lead from the previous two frameworks analyzed in the above sections. Hence, two clear winners are obtained from this analysis: React Native and Django. However. From the ratings, Django scored higher than React Native, making it the best option to choose for the web application.

Proving feasibility

The next step in this stage is to test Django's feasibility for developing the web application. From previous studies, it is noted that testing Django is done within a framework that supports request simulation, inspecting the application’s output, code verification, and insertion of test data. The plan is first to ensure that the computer used for development meets the minimum requirements for downloading and using the Django framework. This will then be followed by creating a rough sketch of what the application should look like and its wireframes. Django will then be tested for feasibility by developing, separately, each wireframe to ensure that it is running smoothly and as intended.

3.2.5 Chosen Approach

From all of the frameworks that were talked about Node.js, Express.js, and Django. Django is the best framework to use for our website. Django has the advantage of being scalable, has community support, and has built-in administration. Django's main disadvantage is that it has many plug-ins and features that bloated for small and big projects. React Native has advantages in optimum performance, reusable code, cost-effective, simple user interface, supports third-party plugins, handy library and solutions, and more stable applications. The cons were that it is still immature, lacked security robustness, and not good at computation. Ionic has the advantage of supporting many UI components, can be developed for both iOS and Android at once (Majchrzak, Biørn-Hansen and Grønli, 2017), and can be developed in the browser. The disadvantage is that native plugins can conflict with each other, random crashes can occur, and debugging can be difficult. Table 3.2.4 below rates this comparison as per the desired application requirements.

Table 3.2.4 Rankings and comparisons of all discussed approaches based on the set criteria from 0 (worst) to 5 (best)

Framework	Application characteristics					
	Save Function	Summary Page	Side Pane	Easy to navigate user interface	Push Notification	**Totals**
Django	**4**	3	**4**	**5**	**5**	**21/25**
Express.js	3	3	2	4	**5**	17/25
Node.js	3	**4**	**4**	2	**5**	18/25
React	3	3	3	**5**	**5**	19/25

3.3 Front-end Libraries

Choosing an appropriate front end library is just as critical as a back end framework as one is needed that can work well with what is chosen as a back end and the library that is used should be able to create a clean and user friendly UI that makes usability of the website simple and clean. In order to choose the best front end library for this application, the following criteria will be observed:

- **Documentation** - As with other tools for development, having a comprehensive documentation reference is critical to navigating one's toolset with the least amount of issues during development. Having a good documentation reference is going to be a key thing to look for when choosing front end tools.
- **Features** - Front end libraries are tricky, as some of them are more comprehensive than others in terms of what tools are under the hood. Some may better support form validation and processing, and some may specialize in creating front end layers and leave the validation processing to the developer to implement in the backend.
- **Ease of Integration** - Web development is tricky because virtually all of the aspects of implementation are interconnected. You need to know what frameworks support which language, and what language is best for the job. The same issue is understood for the front end. The front end libraries used are easily integrated into the back end support of the web application.

3.3.1 React

React is a JavaScript library for building user interfaces. It lets you compose complex UIs from small and isolated pieces of code called "components", which allow us to keep the coupling between sections of UI low and avoid unnecessary dependencies in the UI design. React is developed and backed by Facebook, and in that respect the library has been documented well, with tutorials and definitions for DOM elements, as well as API references and testing. It is also a popular front end tool, which will help in troubleshooting through community FAQ as well.

In regards to features, react is heavily focused on UI design and functionality. It's not so much a full framework, as it only represents the view layer of an application, leaving other things like form validation, processing, and HTTP communication to us. This is fine as the VirusWatch project involves building custom validation and processing according to the roles based access implementation that is created for users. It is also easily integrated with API's of other frameworks as it can connect to the API endpoint of the framework and work seamlessly with it.

Table 3.3.1. Rankings of React based on the set criteria from 0 (worst) to 5 (best)

	Documentation	Features	Ease of Integration
React	5	4	4

3.3.2 Vue

Vue is another front end library similar to React in that it is designed to primarily focus on the view layer aspect of a web application. It describes itself as a progressive framework in that it is easier to pick up and utilize with other existing libraries. Like React, it has a nice comprehensive documentation, as well as a comprehensive starting tutorial and a page comparing it to other front end libraries, React being one of the subjects in this page. In it's comparison, it notes that both libraries are similar in design and functionality, with it's differences based on its user friendliness and learning curve. Its features are the same as React, in that it's viewer level only functionality leaves the implementation of the rest of the UI functionality up to us. Vue is also easily able to connect to the API endpoint of a back end framework to create a reactive web application.

Table 3.3.2 Rankings of Vue.js based on the set criteria from 0 (worst) to 5 (best)

	Documentation	Features	Ease of Integration
Vue	5	3	4

3.3.3 Chosen Approach

After looking at the options that suited our project, both React and Vue come extremely close to each other given that they both serve view layer UI implementation and both do it in a similar fashion. After weighing the pros and cons, React will be the front end that will implement the UI design as it's component based approach to design will allow us to develop a solid and responsive UI that will limit interdependencies in the design and functionality of the application. Table 3.3.3 illustrates the ratings of the above technologies.

Table 3.3.3 Rankings and comparisons of Vue.js and React.js based on the set criteria from 0 (worst) to 5 (best)

	Documentation	Features	Ease of Integration	Total
Vue	**5**	3	**4**	12/15
React	**5**	**4**	**4**	**13/15**

3.4 Data Analysis Tools

Due to the nature of this project, it is important the team understands the tools being used for data analysis. What this web application will have to do is take in data that will be uploaded as an excel sheet and shown to a client as some graphical representation after it has been processed by the application. Important features to consider for these tools is the reliability of them, as well as their ability to process and manipulate data according to a client's specifications. In order to evaluate the languages for the ideal data analysis requirements, the team will gauge them through the following criteria:

- **Reliability** - This web application will be taking in data constantly and return reliable data analysis results to the appropriate users. When considering the languages to use for this task, one will be needed that has a reliable library of tools that have been tried and tested to ensure accuracy and consistency.
- **Flexibility** - It's likely that the language being used to conduct analysis is also what will be used to conduct the rest of the back end framework implementation. Because of this the team will also need to keep in mind that what is chosen needs to be able to branch out to more than just data analysis.
- **Documentation** - Finally, because of the role the language will play in both data analysis and programming for the back end, the language will need to be documented extensively enough so that the team can reliably reference official documentation and avoid guessing and bug fixing.

3.4.1 Python

Python is one of the most widely used programming languages in the realm of data science. It is an alluring choice as it is considered a language that is easy to learn and quick to implement. It's perfect when data analysis tasks involve integration with web applications or when there is a need to incorporate statistical code into databases, and as a full programming language, it's perfect for algorithmic implementation. Another advantage to python is that it contains libraries crafted for data analysis that are well known such as Numpy, Pandas, and StatsModels.

Table 3.4.1 Rates the python backend framework based on the set criteria from 0 (worst) to 5 (best)

	Reliability	Flexibility	Documentation
Python	4	4	5

3.4.2 R

In terms of popularity, R is the runner up in data science analysis languages. R is a language that was built by and for staticians more so than programmers. What R specializes in is the output of statistical analysis. Utilizing the library "knitr," the communication of findings in presentations or documents is easy. However, one of the issues that arises in R is the restrictions in deployment for larger applications. Because it is not a full scale programming language like Python, it's usability is limited strictly to data analysis and manipulation in certain fashions, making deployment in a web application difficult if it is not using a compatible framework.

Table 3.4.2 Rates the R backend framework based on the set criteria from 0 (worst) to 5 (best)

	Reliability	Flexibility	Documentation
R	4	2	5

3.4.3 Node.js

An up and coming tool for data analysis is Node.js. Node.js is a back end javascript platform that acts as a backend framework for web applications. A perk to using Node.js is that it uses javascript, which is considered the "lingua franca" for web development, for all it's back end development. Node.js also has libraries that act as data analysis tools, such as D3.js, to parse, manipulate, and plot data. This would be ideal to use as it can work seamlessly with a javascript based front end library. An issue with this, however, is that the libraries provided are not as mature or specialized as other languages, making it an unsure choice for this project.

Table 3.4.3 Rates the Node.js backend framework based on the set criteria from 0 (worst) to 5 (best)

	Reliability	Flexibility	Documentation
Node.js	2	3	4

3.4.4 Chosen Approach

Looking at Table 3.4.4, all the languages that have been researched are theoretically capable of performing in depth data analysis and graphing, the primary issue is how they work with a given framework and how flexible the language is as a backend language tool as well. Given the options, the most feasible language to use for data analysis in this project is Python. Python is a full programming language that is capable of also being used for regular back end programming outside of data analysis. Pair that with its friendly learning curve and mature libraries of data science tools, and it seems like the most appropriate tool for our needs. R is an extremely powerful tool for data analysis and visualization, but because it was made explicitly for that, it leaves much to be desired in the realm of flexibility. Node.js is potentially capable of data analysis and back end programming as well, but it’s libraries for data science are newer and the reliability of these tools is not as well understood as Python’s.

Table 3.4.4 compares all of the ratings for the backend languages based on the set criteria from 0 (worst) to 5 (best)

	Reliability	Flexibility	Documentation	Total
Python	**5**	**5**	**5**	**15/15**
R	4	2	**5**	11/15
Node.js	2	3	4	9/15

3.5 High Performance Computing

High Performance Computing, or HPC, refers to the capability of processing enormous amounts of data and making complex calculations at incredibly high speeds. This can be done with the use of clusters, hundreds to thousands of individual computers working on solving the same problem. Each computer in the cluster is considered a node containing its own set of processing cores and graphics processing units (GPU's). Together, these computers can perform calculations up to a quadrillionth of the speed of an average desktop computer.

The importance of high performance computing in regards to VirusWatch comes from the fact that the server must be capable of handling large amounts of data that will be received from the stakeholders. This data needs to be securely processed and stored while being able to be easily retrievable by an authorized user. With the amount of data VirusWatch will have to process, the need for an HPC cluster becomes a necessity. Once implemented, using HPC clusters have the potential to create highly scalable, efficient applications.

Although there are many HPC services being offered today, each cluster may be configured differently in both hardware and software. Choosing the right HPC cluster will make the program more efficient in storing and processing data from the stakeholders. In order to choose the correct service provider, the alternatives that will be researched must include key characteristics. These key characteristics of the ideal HPC cluster include:

- **Affordability** - Because HPC clusters use a great amount of resources, access to a cluster can become very expensive. The web app must be scalable, while paying for only what is needed.
- **Security** - A variety of stakeholders will be using the web application from different areas of Arizona. Data stored in a cluster will not be able to be accessed without proper authorization.
- **Capacity** - The client's work is ever expanding, collecting data on a consistent basis. Storage for the foreseeable future must have enough capacity.
- **Documentation** - Clear, concise information about the specific cluster. Well written documentation can help us establish the reliability of the cluster.

3.5.1 AWS

Amazon Web Services offers users a number of different microservices in multiple areas of business. AWS is reputable and it is known that its services are used within many different companies worldwide. For researchers, Amazon offers HPC tools that can be very useful when looking ahead in the project timeline.

Affordability

Performing HPC in the cloud is great due to the fact that configuration of a certain cluster can be adjusted and changed with nearly endless limits. Although there is a free tier, AWS is not cost-efficient. If AWS were to be used to store and process data, it would charge a few cents for every GB used. It may not seem like much, but it can quickly become very expensive. Our team and client may not have the available resources to deploy and maintain code in AWS.

Security

It seems that almost anyone can make an AWS account and try out some HPC services that they offer. Specific research accounts are given to people who requested but that process remains unclear. AWS has another service to help with overall security, but our team is looking for something that does not necessarily rely on us using another service.

Capacity

A huge factor that plays a role in our team's decision is the amount of physical and processing capacity that can be accessed. Being that Amazon is one of the leading companies in the world, their HPC services include the best of the best. From state of the art processing cores to hundreds of types of GPU's to choose from, Amazon does not fall short on the amount of resources that could be taken advantage of.

Documentation

AWS is extremely well documented. With so many different services offered, documentation must be clear, concise, and accurate. In regards to Amazon's HPC service, documentation is well covered especially geared towards new users and beginners. AWS has been around for the last 14 years and the services along with the documentation has only been improving.

Table 3.5.1. Ranking of AWS based on the set criteria from 0 (worst) to 5 (best)

	Affordability	Security	Capacity	Documentation
AWS	1	4	5	5

3.5.2 Google Cloud

The Google Cloud platform hosts a number of services for their users. Much like AWS, Google Cloud contains a multitude of different tools from analytics to data storage. Included in the suite of Google services; HPC Cloud Computing. Google is using HPC Cloud Computing in association with companies and universities from topics such as genome analysis to medicine discovery.

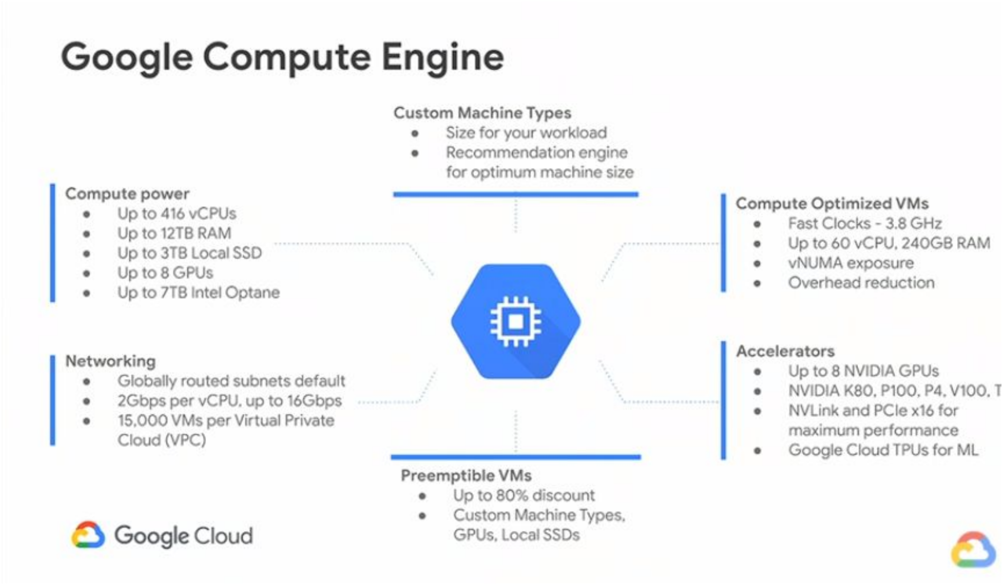
Affordability

Although Google Cloud has more leniency in terms of a free tier, payment still occurs in the form of a rate depending on the amount of data that is being processed. Like AWS, it is not cost-efficient in terms of deployability and maintainability.

Security

Google Cloud is very user friendly when it comes to privacy of user data and security. The interesting thing about Google Cloud is that it uses its own security implementations for programs. Meaning that the security and privacy that you have come to expect on site such as Gmail, can be experienced within one’s own program. Overall, our team felt very comfortable with how secure the platform was.

Figure 3.5.1 Hardware and network specifications of Google Compute Engine



Google Compute Engine

Custom Machine Types
- Size for your workload
- Recommendation engine for optimum machine size

Compute power
- Up to 416 vCPUs
- Up to 12TB RAM
- Up to 3TB Local SSD
- Up to 8 GPUs
- Up to 7TB Intel Optane

Compute Optimized VMs
- Fast Clocks - 3.8 GHz
- Up to 60 vCPU, 240GB RAM
- vNUMA exposure
- Overhead reduction

Networking
- Globally routed subnets default
- 2Gbps per vCPU, up to 16Gbps
- 15,000 VMs per Virtual Private Cloud (VPC)

Accelerators
- Up to 8 NVIDIA GPUs
- NVIDIA K80, P100, P4, V100, T
- NVLink and PCIe x16 for maximum performance
- Google Cloud TPUs for ML

Preemptible VMs
- Up to 80% discount
- Custom Machine Types, GPUs, Local SSDs

Google Cloud

Capacity

Similar to AWS, Google has developed a cluster filled with enormous amounts of resources containing some of the most advanced hardware and software systems currently available. Aside from computing power, Google’s Compute Engine also takes advantage of using networking to create custom and virtual machines. Refer to Figure 2 for a detailed overview of the Google Compute Engine.

Documentation

Documentation for Google Cloud is unsurprisingly thorough. It includes demos for all of the tools needed and great beginner examples on how to get started as well as simple videos that walk users through every step of their project. Google Cloud documentation covers most services available to the average user.

Table 3.5.2. Ranking of Google Cloud Compute based on the set criteria from 0 (worst) to 5 (best)

	Affordability	Security	Capacity	Documentation
Google Cloud	2	4	5	5

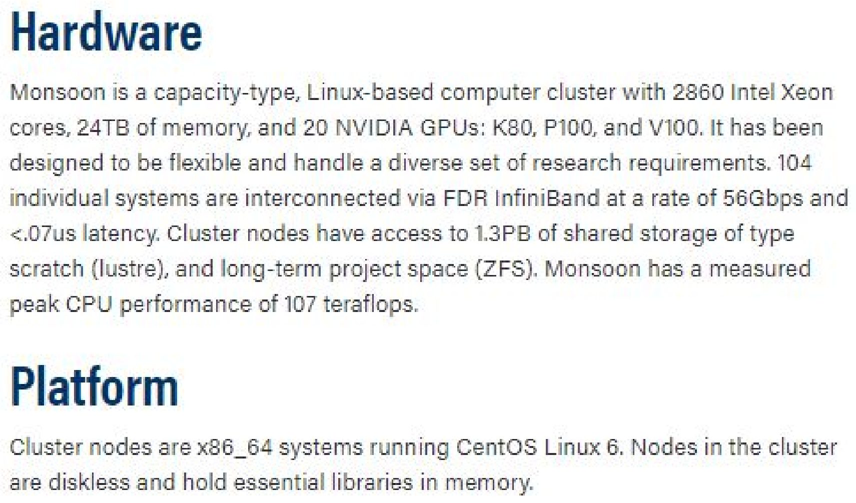
3.5.3 Monsoon

Monsoon is a high performance computing cluster owned and operated by NAU. It is mainly accessed by researchers and graduate students. Purposes can include data analysis or exploration, computational method development and testing, and other tasks specifically related to research projects., which would make it a great fit for the purposes of our client.

Affordability

Although Monsoon does offer payment plans for long-term storage, the cluster and its resources are generally free to use with an authorized account. This is incredibly cost-efficient for both our team, our client, and the stakeholders. Monsoon will only charge for long-term storage greater than 10TB which is much more than our team currently needs. This is undoubtedly the most cost-effective cluster currently available to us.

Figure 3.5.2 Hardware specifications of HPC cluster, Monsoon



Hardware

Monsoon is a capacity-type, Linux-based computer cluster with 2860 Intel Xeon cores, 24TB of memory, and 20 NVIDIA GPUs: K80, P100, and V100. It has been designed to be flexible and handle a diverse set of research requirements. 104 individual systems are interconnected via FDR InfiniBand at a rate of 56Gbps and <.07us latency. Cluster nodes have access to 1.3PB of shared storage of type scratch (lustre), and long-term project space (ZFS). Monsoon has a measured peak CPU performance of 107 teraflops.

Platform

Cluster nodes are x86_64 systems running CentOS Linux 6. Nodes in the cluster are diskless and hold essential libraries in memory.

Security

As of now, only authorized NAU accounts are eligible to use Monsoon. This drastically reduces attacks from other random accounts from different locations. According to their website, connections to Monsoon are allowed only via SSH with NAU credentials. This ensures that authentication and sessions are encrypted. File transfers can be done either via scp or samba. Both of these methods require an NAU username and password.

Capacity

As mentioned above, Monsoon does not require a payment unless long-term storage greater than 10TB is being used. For the purposes of our project, 10TB is plenty of storage space. As far as processing capacity, Monsoon currently runs on 2860 cores, 24TB of RAM, and up to 20 GPU's. These resources should meet if not exceed our computing expectations. A detailed list of specifications can be seen in Figure 1.

Documentation

Monsoon is very well documented. It's documentation page contains numerous links concerning different topics. Monsoon's documentation includes a thorough Introduction section for new users as well as an Advanced Topics section for users who want to optimize their programs.

Table 3.5.3 Rankings of Monsoon based on the set criteria from 0 (worst) to 5 (best)

	Affordability	Security	Capacity	Documentation
Monsoon	5	5	5	5

3.5.4 Chosen Approach

After closely researching these technologies for high performance computing, it has been decided that Monsoon will be used. In terms of high performance computing, all alternatives that have been examined have the capability to perform our desired calculations and host/store our data. While both AWS and Google Cloud are great tools for high performance computing the deciding factor for choosing Monsoon came down to affordability and requirements. Since Monsoon is NAU's cluster, it really makes sense to take advantage of our resources as NAU students, faculty, and staff. This takes care of usage costs and frees up capital for other priorities regarding our research. Being that our client is NAU faculty, deployment on Monsoon is expected to be a smoother transition than any of the other platforms. Table 3.5.4 below gives a brief ranking of each HPC cluster examined.

Table 3.5.4 Rankings and comparisons of all Chosen Approaches for HPC's based on the set criteria from 0 (worst) to 5 (best)

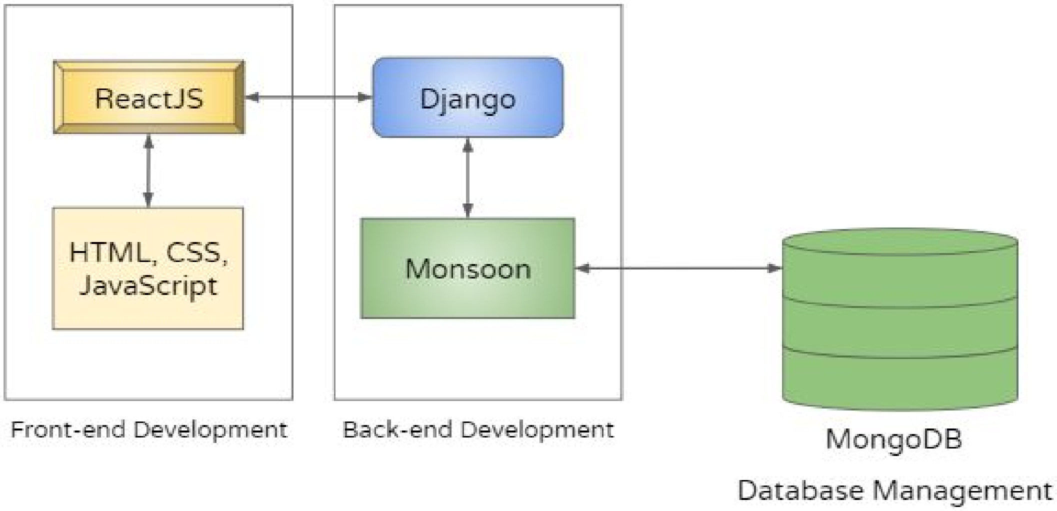
	Affordability	Security	Capacity	Documentation	Total
Monsoon	**5**	**5**	**5**	4	**19/20**
AWS	1	4	**5**	**5**	15/20
Google Cloud	2	4	**5**	**5**	16/20

4. Technological Integration

Since all of the challenges in the previous pages have been chosen, integration of all approaches must be dealt with. All of the solutions meet the requirements to build the website and to make the goal easier and clearer to what needs to be done. In order for us to make that happen, the website must be flexible enough to enable users to access the data that need to be access, either as numbers or as graphs to show how the level of the data increasing or decreasing, allowing users to trick data based in the selected location, where users can compare their data with different location. Our website will have a graphical user interface that will make it our website's better way of using it when you want to register new users and when you want to inter new data. Our application will allow users to analyze datasets, also store data and can review data at any time.

As seen in Figure 4.1 below, ReactJS will be our primary front-end framework. React will be used to create a responsive UI that will be dynamically displayed through a combination of HTML, CSS, and JavaScript and can easily communicate with our backend framework. Django will be used to analyze the intensive amount of data expected from our client and other stakeholders. Django's primary function in our project is to be able to calculate the data and display said data into clean, readable graphs. From there, Django will communicate with Monsoon, a HPC cluster capable of storing and running the scripts is needed to process the data. Finally, MongoDB will be used to store all of this information in a secure database that can be easily retrievable by an authorized user.

Figure 4.1 Technological Integration of all of the Chosen Approaches



5. Conclusion

Without a clear end to the pandemic in sight, effective ways to identify and isolate potential outbreaks must be developed as quickly as possible. Wastewater testing may be one of the most valuable tools officials have to mitigate the spread of COVID-19 in Arizona. Our web application will provide a secure place to upload, store, and view wastewater data for authorized officials so they can react as quickly as possible to new potential outbreaks.

In terms of a database management system, MongoDB, MySQL, and Oracle were compared. MongoDB is the best choice for this application because of its robust security features and ease of use. For the web application framework, four possible options were compared: Node.js, React Native, lonic and Django. Django was chosen because it has the advantage of being scalable, has community support, and has built-in administration. For the selection of data analysis tools, Python, R, and Node.js were compared. Due to the completeness of Python, its friendly learning curve and mature data science tool library, Python will be chosen as our data analysis tool. High-performance computing is a computing system and environment that uses many processors or several computers organized in a cluster. Three different HPC tools (AWS, Google Cloud, and Monsoon) were compared. Due to the affordability and client specifications, Monsoon will be used as the HPC.