Requirements Specification

Version 2.0

6 December 2021

Team Truthseeker

Garry Ancheta
Georgia Buchanan
Jaime Garcia Gomez
Kyler Carling

Project Sponsor

NOBL Media – Jacob Bailly

Team Faculty Mentor

Felicity H. Escarzaga

Accepted as baseline requirements for the project:

For the Client:

Jacob Bailly

For the Team:

Garry Ancheta

Table of Contents

Chapter 1 - Introduction

Today, misinformation is widespread on the internet. Different platforms intentionally spread misinformation to harm individuals and groups. The internet, however, is just another representation of many businesses through websites. For most businesses, websites are another source of income through the showing of advertisements. Demand Site Platforms (DSPs) are what allow placement of advertisements on websites and currently, DSPs deal with misinformation by blacklisting or demonetizing websites, but only do so when they are actively told to by the advertiser. Thus, an advertiser can be damaged by being associated with misinformation which is a waste of money since it might deter customers from supporting the advertiser when it is found that the advertiser appears to be supporting misinformation.

This is where the Misinformation and Credible News Analysis Tool comes in; the tool allows advertisers the choice not to support businesses that spread misinformation by rating how credible a page is on a website and deciding whether an advertisement should be placed on that page based on its credibility. Unlike the current process where businesses must tell DSPs which websites the advertisers' ads should not be placed on, NOBL Media has the capability to automate the process by allowing advertisers to simply tell NOBL Media what credibility rating a website should have for their advertisements to be placed on.

NOBL Media collects advertisement information for advertisers who choose to use NOBL Media's services. Once that information is collected, advertisers can see how their advertisement is performing in terms of supporting misinformation and how much money the advertiser is losing due to misinformation.

Chapter 2 - Problem Statement

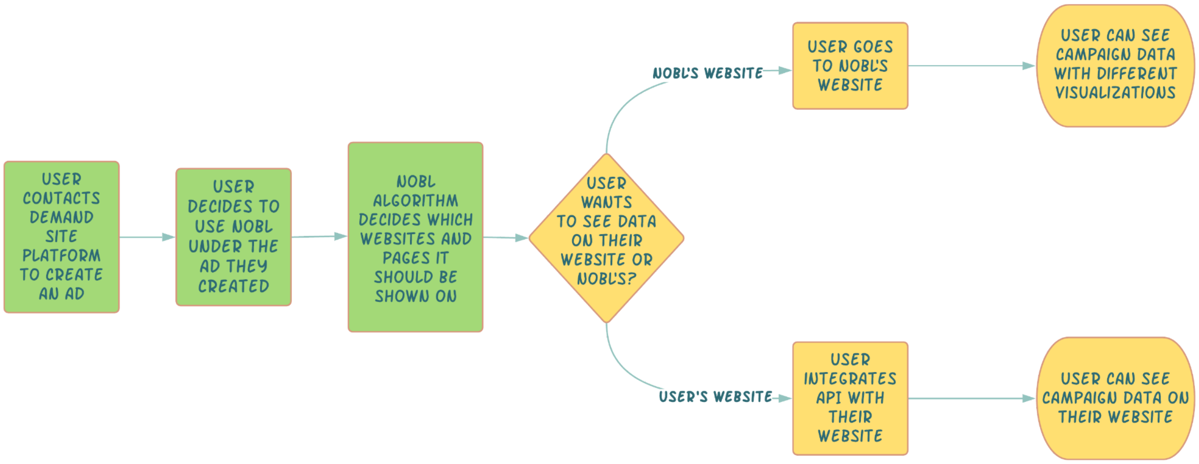
USER CONTACTS DEMAND SITE PLATFORM TO CREATE AN AD
USER DECIDES TO USE NOBL UNDER THE AD THEY CREATED
NOBL ALGORITHM DECIDES WHICH WEBSITES AND PAGES IT SHOULD BE SHOWN ON
USER WANTS TO SEE DATA ON THEIR WEBSITE OR NOBL'S?
NOBL'S WEBSITE
USER GOES TO NOBL'S WEBSITE
USER CAN SEE CAMPAIGN DATA WITH DIFFERENT VISUALIZATIONS
USER'S WEBSITE
USER INTEGRATES API WITH THEIR WEBSITE
USER CAN SEE CAMPAIGN DATA ON THEIR WEBSITE

Figure 2.1

NOBL Media collects advertisement information for advertisers who choose to use NOBL Media's services. Referring to Figure 2.1, the user will decide to use NOBL's service which then leads to NOBL using their A.I. to service the user. The yellow objects within Figure 2.1 shows what is not being achieved by NOBL's business flow. These inadequacies within the business flow can be summed up into the following three points:

1. NOBL Media customers do not have a way to visualize the data NOBL Media collects about the customers' advertisements.

2. NOBL Media cannot abstract the data so that it is easily understood by customers.

3. NOBL Media does not have a way to directly give the customers' advertisement data.

These problems remove NOBL Media's ability to show proof that their service works for the customer. As a result, NOBL Media would be unable to retain current customers and attract new customers.

Chapter 3 - Solution Vision

In order to solve the project's problems, NOBL Media requires two components: the web application and an Application Programming Interface (API). These two components allow NOBL Media to solve the problem with their business flow through the implementation of the following features:

1. The web application must be able to create and authenticate customer accounts to allow secure access to the customer's data using integrated technology Auth0.

2. The API must be able to handle user authentication requests. Once authenticated, the API must handle a request to be parsed into NOBL'S MySQL database.

3. The web application must be able to abstract JSON data coming from the NOBL Media MySQL database and represent these to customers through graphs and charts using technologies such as ECharts.

4. The API must be able to retrieve the JSON data from the NOBL Media MySQL database and return an HTTP 200 level response with JSON data to NOBL's web application or the client's own site.

5. The web application must allow customers to download customer ad data in a formatted file such as in a CSV or Excel file.

Whether the clients of NOBL media decide to use NOBL's website or their own, they will be able to see the worth of their investment. NOBL truly cares about ethical advertising while most companies care about money; sometimes the two do not go hand-in-hand.

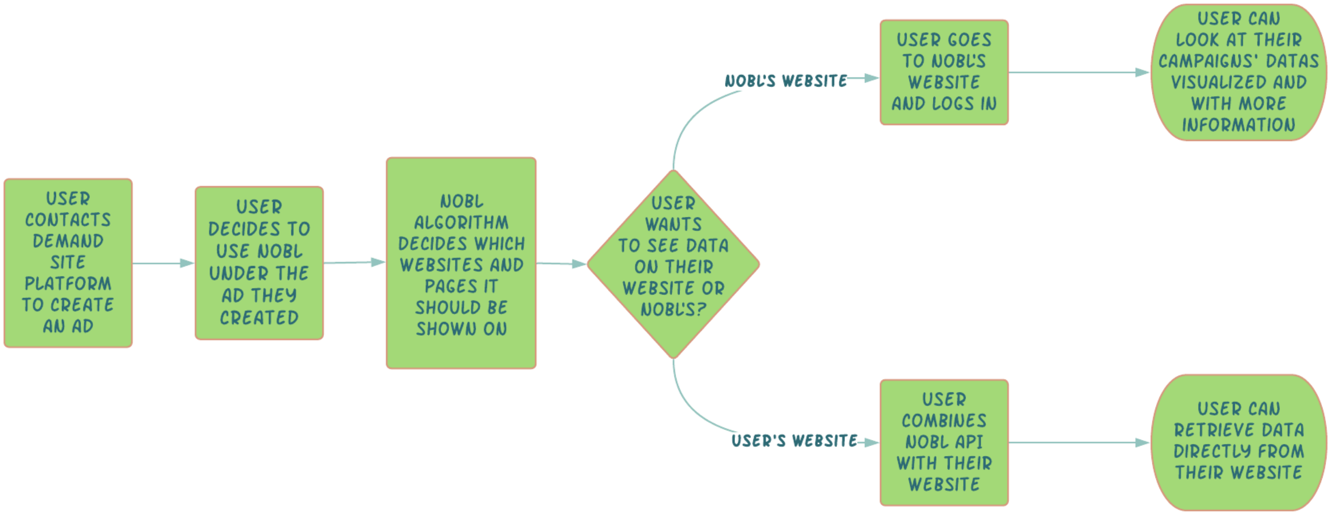
USER CONTACTS DEMAND SITE PLATFORM TO CREATE AN AD
USER DECIDES TO USE NOBL UNDER THE AD THEY CREATED
NOBL ALGORITHM DECIDES WHICH WEBSITES AND PAGES IT SHOULD BE SHOWN ON
USER WANTS TO SEE DATA ON THEIR WEBSITE OR NOBL'S?
NOBL'S WEBSITE
USER GOES TO NOBL'S WEBSITE AND LOGS IN
USER CAN LOOK AT THEIR CAMPAIGNS' DATAS VISUALIZED AND WITH MORE INFORMATION
USER'S WEBSITE
USER COMBINES NOBL API WITH THEIR WEBSITE
USER CAN RETRIEVE DATA DIRECTLY FROM THEIR WEBSITE

Figure 3.1

Figure 3.1 shows the completion of NOBL's business flow once the project is complete. Ultimately, the web application offers a way for clients to actually see that they receive more interaction with their advertisements when they are putting them on more ethical sites. In addition, the API offers a way for clients to pull the same data onto their own site in order to continuously see their ad campaign data while using NOBL's service.

Chapter 4 - Project Requirements

There are a variety of functional and performance requirements necessary for the project. The functional requirements serve as a metric to visualize how the project is progressing while the performance requirements serve as an assurance that the project can actually serve its intended purposes.

Section 4.1 - Functional Requirements

Functional requirements are what the project is expected to perform for the user. These requirements define the features of each component of the project that are either expected to have or a possible stretch goal. The functional requirements for the project are split into three components: user authentication, data visualization, and API functionality.

User Authentication

This section describes the functional requirements for a user to log in to this service. Additionally, user authentication is split into two components based on the user's role within the system: a customer or an administrator.

A - Administrator

This role is reserved for staff members of NOBL Media. Within this role are privileges that allow editing of the user authentication system for the web application.

A.1 - Default Log-in

The administrator must be able to log into the Auth0 dashboard with their email and password.

A.2 - Auth0 Dashboard Access

The administrator role receives privileged access to the actual Auth0 dashboard. Since Auth0 is a separate authentication system that is merely an interface for user authentication, administrators must have access to the Auth0 dashboard instead of having a dashboard on the web application itself. Within the Auth0 dashboard, the administrator will be able to add, remove, edit, and replace users.

A.3 - Multi-factor Authentication

Since access to the Auth0 dashboard is privileged with powers over web application user accounts, it is important that those attempting to log into the Auth0 dashboard have their access as secure as possible. Therefore, as an administrator logs into the Auth0 dashboard, there will be different methods for the administrator to verify their identity. The following are multi-factor authentication methods which the client has expressed a need for:

A.3.1 - WebAuthN Authentication

This refers to the administrator using either a USB-like security device which contains authentication codes generated by the WebAuthN (an API designed for authentication confirmation) that the user will be able to input into the authentication system.

A.3.2 - One-Time Passwords (OTP)

This refers to the administrator using an authentication app like Google Authenticator to receive a secondary password which the administrator must enter into the log in system before they are fully logged into the Auth0 dashboard.

A.3.3 - SMS Notification

This refers to the administrator having a message that contains a code sent to their phone which they must enter after they have entered their email and password to complete the authentication process.

A.4 - Edit Log-In Information

The system must allow the administrator to change their email address and password as needed.

A.5 - Administrator Registration

The system must allow the administrator to register into the Auth0 dashboard. This is done through the main administrator adding access and inviting the intended administrator to have access onto the Auth0 dashboard.

B - Customer

The customer will be the organization who has paid for NOBL's service. Additionally, all users will be tied to an organization within the web application since an organization might want to have multiple users keeping track of their ad campaigns.

B.1 - Default Log-In

The system must allow users to log into the web application using their email and a password.

B.2 - Edit Log-in Settings

The system must allow users to change their passwords and email at any given point.

B.3 - Multi-Factor Authentication (Stretch Goal)

Since the web application handles important and private information pertaining to both the user and their organization, the user must have the ability to use multi-factor authentication within the web application to ensure that access to information within the web application is as secure as possible. In comparison to the administrator role, multi-factor authentication for the customer is a stretch goal rather than an expected feature of the project.

B.3.1 - SMS Notification (Stretch Goal)

This refers to the user having a message that contains a code sent to their phone which they must enter after they have entered their email and password to complete the authentication process.

B.3.2 - One-Time Passwords (OTP) (Stretch Goal)

This refers to the user using an authentication app like Google Authenticator to receive a secondary password which the administrator must enter into the log in system before completion of the authentication process.

B.3.6 - Email Notification (Stretch Goal)

This refers to the user being sent an email that contains a code that the customer must enter before the completion of the authentication process.

B.4 - Private Registration

The system must not allow the user to register on their own volition. For the customer to have an account on the web application, the user must directly request the creation of an account to an administrator by email. The administrator will then log into the Auth0 dashboard to generate a password and account for the user.

B.5 - Selection of Ad Campaigns

The system must allow the user to select a campaign that the user wants to see data for. After logging in, the user will be prompted to select a campaign before proceeding to the dashboard.

Data Visualization

This section describes the functional requirements for the users to view their organization's ad campaign data results.

A - Graphical Data Results

The system must allow users to see their data to be shown on charts and graphs. This abstracts the data so that the users can understand what it actually means for the ad campaign data.

Data Examples

The following are examples of data that could be displayed and the type of charts and graphs that would display them.

1. Campaign NOBL score

This is the NOBL score for a specific campaign; this can be displayed as a gauge chart where the NOBL score is displayed or it is possible to expand upon and show a range of pages rated and their NOBL scores which shows highlights the outliers of pages that extremely affect the NOBL score in a form of a bar chart or scatter plot graph.

2. Average Offensive Words per Page

This refers to the average offensive words per page for a campaign. This can be displayed as a gauge chart like the NOBL score or deepen the level of detail by showing the outliers of pages that extremely affect the average in the form of a scatter plot graph.

B - Customizable Graphical Data Results

The system must allow users to edit different aspects such as how many data points are shown, and show certain data points which could be under a certain threshold in a rendered graph and chart. The user must be able to configure how many data points are shown, zoom in/out on the graph/chart, and highlight a specific data point. The details of the data point will be abstracted so that on-screen it can be understandable. For example, the time the data point was retrieved might not be in the format most are familiar with (Month Day, Year) which means that it might be needed to convert the time to that format. To highlight a specific data point, it is possible that the details may include, but not limited to:

1. Time the data point was retrieved
2. The page URL of the data point
3. The NOBL score of the data point

C - Export Data

The system must allow users to export a graph or data as part of a document or just an image (.png) file. When a graph or data is exported as part of a document, this means that the graph is rendered onto the document or in the case of the data, it will be rendered in an understandable form onto the document. Additionally, data can also be exported not just in .pdf form, but also in different formats such as .csv or .xlsx.

API Functionality

This section describes the functional requirements for the API. The API is required for data retrieval.

A - Retrieve Specific Data Fields

The system must allow users to retrieve specific data fields, not the entirety of a database table. For example, the user would be able to retrieve the nobl_score data field of the NOBL database for a specific page or the number of impressions for a page.

B - Calculate Needed Data

The system must allow users to calculate certain statistics about the ad campaign based on the data retrieved. For example, the API would need to allow calculation of the average NOBL score of an ad campaign.

C - Connect Auth0 certification with Client ID

The system must allow users to match the Auth0 certification to the client id in the NOBL database to ensure that the users can only access data that pertains to their organization’s ad campaigns and no other organizations’.

D - Connect to NOBL Database

The API must ensure a connection to the database without revealing the connection certification which would allow unauthorized access to the NOBL database.

Section 4.2 - Performance Requirements

Performance requirements are what allows the system to serve the users with adequate efficiency. If performance requirements are not met, the functionality of the system will be degraded and the user's experience will suffer as a result.

Data Visualization Rendering Speed

The rendering speed of the graphs and charts must be under 250 milliseconds to ensure that the user is not waiting too much to visualize the ad campaign data. With the challenges the team found in the Technology Feasibility document, the team concluded that it is imperative that 250 milliseconds since this is above average when it comes to the results of the time it took for all the data visualization libraries to render 50,000 data points. This is the main purpose of the project and thus, it is of utmost importance that the system renders the graphs and charts as fast as possible.

- Target Metric
    - Ability to render charts and graphs quickly

- Benchmark Value
    - Rendering of the chart/graph in under 250 milliseconds with at least 50,000 data points.

- Testing Method
    - Generate a user account with data near the upper bound of what the graph/chart is expected to handle and observe the time it takes for the graph/chart to finish rendering

API Response Time

The data retrieval speed of the API must be at a maximum under 1 second. The purpose of this performance requirement is to ensure that the data will be there at a proper time to allow the graphs and charts to populate. If the API data retrieval process takes longer than 1 second, it will become a detriment to the user experience since this is the first step in rendering the graphs/charts.

- Target Metric

  - Ability to transmit data in a timely way

- Benchmark Value
  - Making an API request should take under 1 second for a fresh response and under 0.5 seconds for a cached response under ideal network conditions.

- Testing Method
  - Conduct API queries a time how long they take to complete for both fresh and cached responses.

API Error Rate

There are two kinds of errors that can occur in an HTTP request to a rest API. User error which is typically represented by the API returning a 400 level HTTP response code and server side errors which are typically represented by 500 level HTTP response code. The project must aim to keep both kinds of errors below a certain threshold as it would be a disruption to the service that the project provides to NOBL customers.

- Target Metric
  - Percentage of API requests that return error responses rather than actual data.

- Benchmark Value
  - The API should only produce 1 500-level responses and 1 400-level responses for every 100,000 total requests.

- Testing Method
  - To detect 500 level errors the server side error rate can be monitored during the concurrency tests mentioned previously. 400 level errors will largely be the fault of the website code being faulty or bad user input and not easily tested for.

API Startup Speed

While downtime is undesirable, it is sometimes inevitable. In the case of downtime, it is important to be able to restore service quickly. Discounting the time needed for the server operating system to boot, the API startup sequence is the next most costly element in terms of downtime cost and thus, it is important to make sure that the API is able to quickly begin serving requests after a reboot.

- Target Metric
    - Time from boot up completion until the API is ready to service incoming requests

- Benchmark Value
    - The API should be fully initialized and without user input ready to service requests in under 20 seconds from bootup completion

- Testing Method
    - Reboot the server and time how quickly the API is ready to service requests as the reboot sequence finishes.

Section 4.3 - Environmental Requirements

Environmental requirements are requirements that the team cannot change and must adapt to.

MySQL

The only environmental requirement is having the ability to retrieve data from MySQL databases to store NOBL Media's clients' ad campaign data. That means the Misinformation and Credible News Analysis Tool's API must be compatible with MySQL databases.

Chapter 5 - Potential Risks

Due to the project's need to identify who the user is and display data related to the user, the web application and API have risks involving the manipulation of user data. Furthermore, the risks have different levels of likelihood and severity due to where the risks originate from: the API, the web application, or both. The risks can be categorized into the following aspects:

1. Security
2. Data Visualization

These risks will have their own level of severity and likelihood ranging from low to critical and very low to high respectively. The levels of severity are described as follows:

- Low - Reversible risk; no impactful damage
- Medium - Reversible risk; impactful damage, but can be mitigated quickly
- High - Irreversible risk; impactful damage; affects a user or an organization
- Critical - Irreversible risk; extremely impactful damage; affects all users and organizations

The levels of likelihood range from Very Low to Medium. To have a likelihood above medium would reflect terrible planning and should have been considered beforehand. The levels of likelihood are described as follows:

- Extremely Low - Multiple barriers to break through; security against risk is improved upon over multiple years
- Very Low - Multiple barriers to break through; security against risk is implemented well
- Low - Multiple barriers to break through; avoidance of risk is implements up to standard or have yet to be implemented
- Medium - No barriers to break through; minor risk mitigation implemented

5.1 - Security

Security refers to vulnerabilities in the different components of the project. Specifically, there are three different types of security risks which are:

1. Website Hacked
2. User Authentication System Hacked
3. Data Injection Attack

Each type of security risk will have its own level of severity, the likelihood of its occurrence, and a possible way the project would be able to mitigate the risk.

5.1.1 - Web Application Hacked

Severity	Likelihood
High	Very Low

Table 5.1.1.a

The website being hacked refers to any attacks that are not attacks on user authentication. This attack would happen in the main dashboard where data visualization occurs or even on the user information page as well. Referring to Table 5.1.1.a, the severity of this risk is high due to the fact that it might reveal sensitive user information as well as ad campaign information that users might want to keep to themselves. Additionally, Table 5.1.1.a also states that the likelihood of this risk occurring is very low, this is because to gain access to any page of the web application, one must be logged in.

5.1.2 - User Authentication System Hacked

Severity	Likelihood
Critical	Very Low

Table 5.1.2.a

The project will be using Auth0, which is a comprehensive user authentication system that will be part of the web application. Due to Auth0, the project will not need to create its own user authentication system. This means that this may be an attack point for hackers; the project is relying on the security of Auth0 to protect the information of users.

Referring to Table 5.1.2.a., the severity of this risk is critical because access to the rest of the web application relies on user authentication. If this system is attacked, then the rest of the web application is vulnerable. Table 5.1.2.a also says that the likelihood of the risk occurring is very low. This is due to Auth0's reliable security system and years of security improvements.

5.1.3 - MySQL attack

Severity	Likelihood
High	Extremely Low

Table 5.1.3.a

The project requires a connection between the web application and the MySQL Database that NOBL Media provides. The connection of the web application and the MySQL Database will be the API, which is the other component of the project. To retrieve ad campaign data from the database, the API will need to use queries. These queries can be used maliciously by anyone attempting to either retrieve different data than what the query is intended to retrieve or to change the data that is being retrieved.

Referring to Figure 5.1.3.a, the severity of this risk is high due to the potential theft or manipulation of confidential data relating to a user's ad campaign data. The likelihood of this risk potentially occurring is extremely low since for this risk to occur, the attacker must go through the authentication process and then attack the website to get access to the API. The multiple barriers the attacker has to go through to actually perform a MySQL attack makes this an unlikely target when there are easier ways to attack the system such as directly attacking the web application.

5.2. - Data Visualization

This section refers to any risks regarding data visualization, which is vital to the web application since data visualization is what allows NOBL customers to understand how NOBL Media's service affects their ad campaign in a positive manner.

5.2.1 - Graph/Chart Misrepresents Data

Severity	Likelihood
Medium	Low

Table 5.2.1.a

When the user is on the main dashboard, there will be multiple graphs displayed for the user to understand how their ad campaign is doing. These graphs will be populated with data that comes from the NOBL MySQL database. This risk is referring to the possibility that the data visualization library will misrepresent the data in a way that might make the user believe that there's something wrong with the ad campaign when it's just a problem with how the data visualization library is showing the data. Additionally, this risk also takes into account if the API has been coded to retrieve the wrong data as well.

Referring to Table 5.2.1.a, the severity of this risk is medium since the data visualization library handles how best to show the data and the likely source of this risk is from human error. Additionally, if the wrong data is presented, it might skew the view of the user so that it seems as though their ad campaign is doing something harmful when in reality, it is not. The likelihood that this risk will occur is low since the library is optimized to show graphs and charts in the least biased way possible because the data visualization library is used by different companies and organizations. Additionally, while human error is the source of this risk, the project has implemented several preventive measures to prevent the risk from occurring such as implementing GraphQL for the API which is a verbose query language that allows the declaration of what the API is retrieving.

Chapter 6 - Project Plan

Team Truthseeker is tasked with completing NOBL Media's business flow by developing an API and website dashboard following certain requirements. In order to make sure this plan goes accordingly, multiple milestones have been identified.

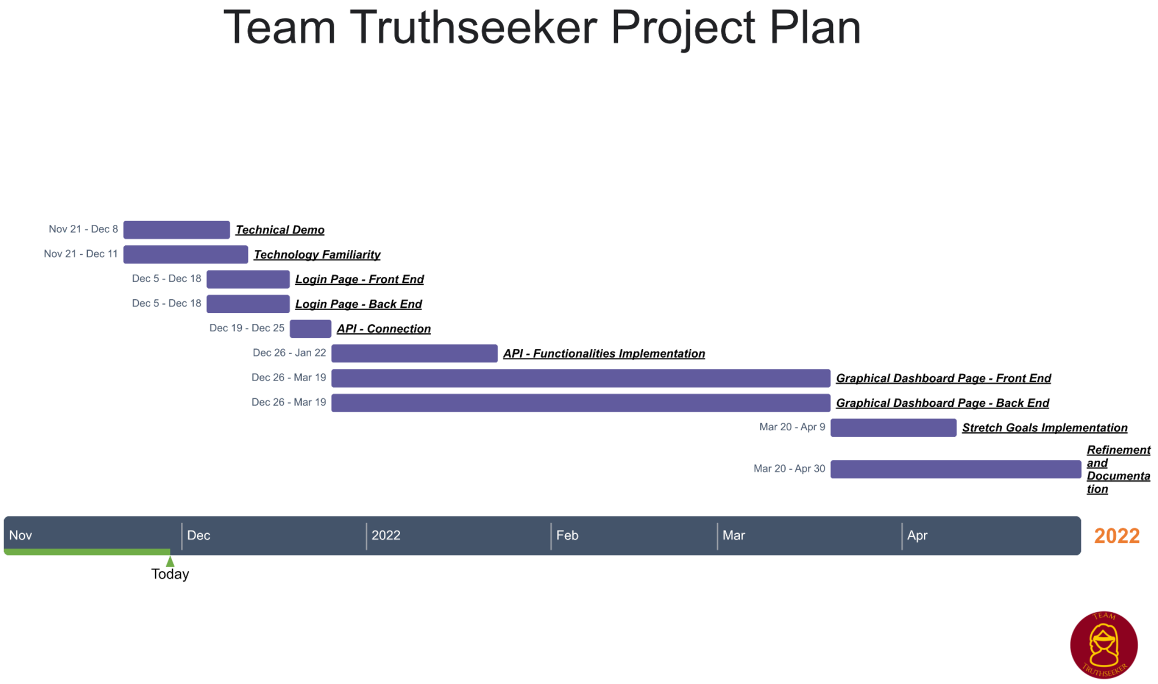
Team Truthseeker Project Plan

Figure 6.1

All milestones are required for the minimum viable product and are therefore going to be worked on heavily throughout December until the end of March. The last remainder of the time will be spent achieving stretch goals for this product. Referring to figure 6.1, this schedule lays out when each milestone or phase should take place; some milestones overlap meaning that those milestones can be worked on during the same time and do not require one to be completed before starting the next. The team plans to start working on the log-in page first, making sure that the front-end allows users to enter an email and password while the back-end authenticates each user sending them to their respective accounts. Once that has been successfully completed, the next milestone is to work on connecting the API. This project requires the team to create an API in order to retrieve client data from NOBL's database; to do that, a connection to NOBL media's MySQL database needs to be established first. After the data has been successfully retrieved, the next milestone is to work on the dashboard page where clients can view this data. Using ECharts technology, we will work on not only creating multiple views of graphs but making user-customizable charts as well. This needs to be accomplished not only for the web application but the API as well, therefore, these milestones can be worked on around the same time. At this point all the requirements for the minimal viable product should be met and the team can begin implementing stretch goals.

Chapter 7 - Conclusion

Due to the lack of proof that can be shown to NOBL Media customers, NOBL Media cannot truly show that their service has any benefits. This produces the problem that the project is intended to solve. The project has two components, the web application and the API. These two components work together to achieve the solution to NOBL Media's problem: show what NOBL's service actually provides.

The solution to NOBL Media's problem requires that the web application be the interface between the NOBL Media customer and NOBL Media while the API acts as the connection between the web application and the NOBL Database. The web application intends to show NOBL Media customers statistics of how NOBL Media's service is affecting their ad campaign by way of graphs and charts. The API is the connection point which transfers data from the NOBL Database to the web application for rendering. Thus, the two components, working together, will be able to solve the problem.

Currently, the team has completed preliminary testing of the technologies that are intended to be used for the project and have proven that all of them are compatible together. The project is at its intended point in its development and the team is extremely eager to start preliminary implementation of the solution with the prototype. Overall, the solution's foundation is set, and now the team has set its eyes on building the framework which will ultimately lead to the project's full completion.