IOWA STATE UNIVERSITY

ELECTRICAL AND COMPUTER ENGINEERING

Design Documentation

Senior Design Team 10

Visualizing Probabilistic Whereabouts of Moving Objects

Client/Faculty Member - *Professor* Goce Trajcevski

April 16, 2024

Team Members & Roles:

Nathan Thoms - Team Lead & Frontend Developer

Ryan Cook - Full Stack Developer

Mara Prochaska - Backend Developer

Eric Jorgensen - Documentation

Team Website:

Senior Design Team sddec24-10

Executive Summary

Development Standards & Practices Used

IEEE and web software development best practices, utilizing GitLab as the primary tool for code pushing and documentation. Please see Section 2-2 for a more complete view of relevant project standards.

Summary of Requirements

Create a web-based application that enables users to predict the location of objects ("Object Whereabouts") utilizing Conical and Bridgelet Algorithm methods. The application shall have a working database, functional UI frontend, and backend containing the algorithms.

Applicable Courses from Iowa State University Curriculum

COM S 309 – Application Software Development

COM S 228, COM S 227, COM S 311

CPR E 288, EE 285

New Skills/Knowledge Acquired that was not taught in courses

Javascript, Node.js, Vue.js, Database Manipulation, Whereabouts Algorithms and Theory

Table of Contents

Table of Figures and Tables

1. Defining the Problem

1.1 Problem Statement

Traditional location tracking technology such as Global Positioning Systems (GPS) often have a latency time between pings, meaning that the exact location of objects are unknown in between pings. One problem with the current method of estimating location is that it assumes that the moving object travels along a straight line between the two end-points and with a constant speed. In actuality, there is an infinite number of possible scenarios for the motion plan starting from Point A and ending at Point B: (1) the object could stay at point A until just before t_B, and then move to point B instantaneously; (2) conversely, it may quickly travel to point B, shortly after t_A, and then spend the rest of the time at rest until t_B; (3) in-between are many options for the objects starting faster or slower and moving away or towards point B. Thus, more realistic constraints are needed to capture the object's behavior which, in turn, would enable the creation of a more accurate (probabilistic) model for the potential whereabouts of the object. Although traditional location estimation methods may be effective in most typical scenarios, many research-based applications require a significantly more precise methodology.

To improve the estimation for the positioning of objects Dr. Goce Trajcevski has developed an algorithm utilizing conical geometry to efficiently determine the optimal semantically diverse Points of Interest (PoIs) and the optimal route between them given user constraints like maximum travel distance, preferred PoI categories, etc. [1]. There is also an additional algorithm developed by Dr. John Krumm that estimates the location of objects from large data sets in a 2D graphical map, known as the Bridgelet method [2].

The main objective of this project is to design and implement a web application that creates a visualization of the possible whereabouts utilizing the algorithms described above, where:

- only discrete location-in-time data is available
- in-between two such data items, the only known value is the limit on the velocity that the object can take

This web application will allow these algorithms to be used by a wider audience and reduce the effort to apply each algorithm to different applications individually. Creating an integrated system will also allow visualizations from both algorithms to be shared efficiently with others.

With this, the model of the possible whereabouts of a given moving object at a time-instant T between t_A and t_B can be created using the bounds on the jointly-limited locations with respect to both points A and B, which is, the respective maximum distances from each, for any

given time t_X, where $t_A < t_X < t_B$. This idea of utilizing object behavioral constraints determining probabilistic whereabouts of moving objects between known locations is the center of our project, in which we aim to develop a system to visualize this idea. The figures below provide an example visualization of the conical method, which will be implemented in the application. In these figures, each cone represents the possible locations of an object over time based on starting location and maximum velocity.

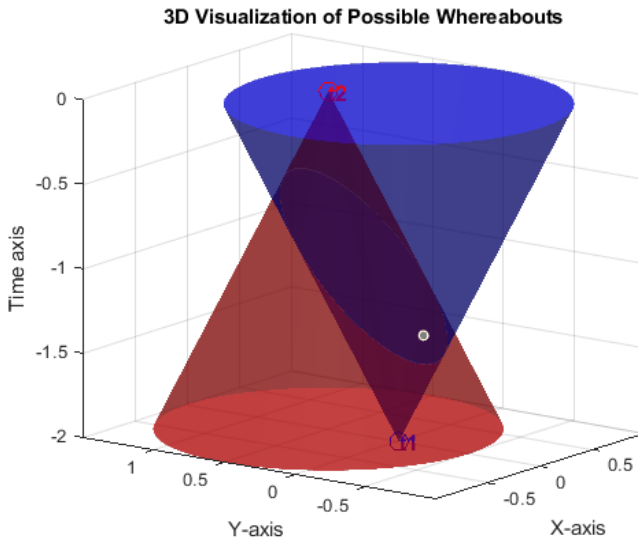

Figure 1 - Visualization of a given whereabout model to capture possible intermediate locations.

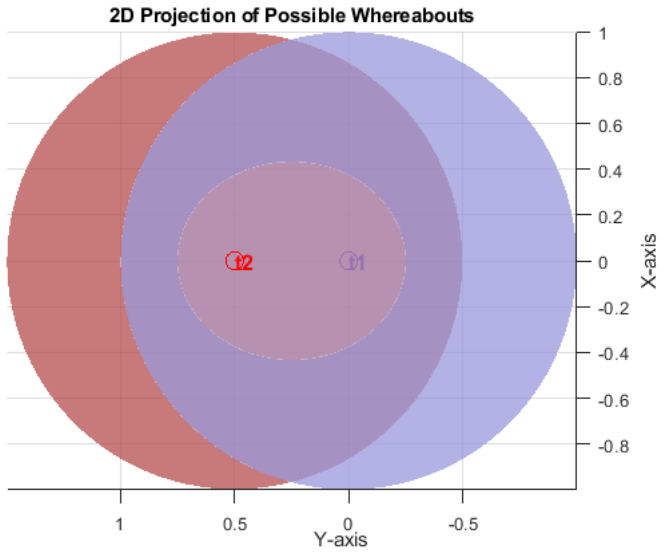

Figure 2 - 2D projection of a possible whereabouts at a given time-instant.

There are currently no applications that implement the algorithms described above with a visual display. The intention of this system is that users will be able to input data sets and choose settings that allow their sets to be visualized using the appropriate algorithms. This project aims to develop a prototype system that enables users from different domains to explore the (impact of the) motion model represented with the incorporation of the uncertainty of the possible whereabouts. The main functionalities of the can be summarized as follows:

1. Provide an informative landing-page for the users
2. Identify datasets from a particular application domain (to be coordinated with the client) that suit the user's interest(s)
3. Enable the user to:
    a. Select a particular dataset
    b. Enter parameters of interest (maximum speed; limit on time-intervals of interest; etc.)
    c. Identify queries of interest (e.g., possible intersection of motion of two objects; possible intersection of a particular object with a region).

d. Access the corresponding datasets and extract the subsets corresponding to the user-provided parameters of interest.
e. Perform the necessary calculations and provide visualization of the outcome for the user (cf. Fig. 1 and 2 above)

4. Provide options for the user to continue the interaction; store the outcomes; etc.

1.2. Intended Users

The following intended users were chosen as representatives from a broader spectrum as the ones who may interact with the system: (1) Zoologists are intended to represent the group who will primarily use the Bridgelet or 2D grid algorithm for predicting the location of objects [2]. (2) Chemists are intended to represent the group who will primarily use the conical, 3D method of determining probable locations of objects [1]. (3) Researchers who are intended to represent an intersection between these groups, who may want to use both methods and need more customization features for different projects.

Zoologist

A biologist who specializes in the study of animals, their behavior, evolution, and habitats. They conduct research, observe animal populations, and analyze data to understand animal life and contribute to conservation efforts.

- Key Characteristics
  - Want to preserve animals and be able to learn about their behaviors
  - Want to input large amounts of data that may track separate animals of the same species
- User Needs
  - Flat, map view with Bridgelet style positioning algorithm
  - Ability to input large data sets and share graphs with other Zoologists internationally

Zoologists will benefit from this application since it will give them a concrete way to visualize and share data they have collected. It will have customizable mapping outputs and allow for additional data such as maximum speed which will improve current tracking capabilities.

Chemist

A scientist who investigates the composition, properties, and behavior of substances at the atomic and molecular levels. They conduct experiments, analyze data, and develop theories to understand chemical processes and reactions.

- Key Characteristics
    - Want to get other Chemists and Graduate Researchers interested in their work
    - Like to share visualizations of their discoveries with other Chemists
- User Needs
    - 3D graph ability to visualize location of molecules with optional 2D ellipse "slice" view
    - Ability to change maximum velocity and start with unknown exact locations within a certain uncertainty

Chemists will benefit from this application since they will be able to utilize the 3D conical positioning algorithm in order to predict the position of molecules. This tool is specifically helpful since it is customizable and can take in large amounts of data. The application will also have a share feature, allowing data sets and visualizations from one user to be seen by another.

Researcher

Someone who systematically investigates topics to discover new knowledge or solve existing problems, through rigorous methodologies and analysis. Their findings advance understanding in various academic disciplines or practical applications in industry and society.

- Key Characteristics
    - Wants to track movement and probability of location for objects that cannot use traditional tracking methods such as GPS
    - Likes to be able to visualize the data he collects in order to increase funding from Donors
    - Likes to switch between many subjects and methods of tracking/predicting locations
- User Needs

- Both 2D (Bridgelet algorithm) and 3D (Conical algorithm) options to choose between for different applications
- Ability to share large data sets and completed analysis with other researchers

Researchers will benefit from this application since they will have a new way to visualize their gathered data. They will be able to input several data sets and store the different visualizations produced using both or either algorithm integrated with the system.

2. Requirements, Constraints & Standards

2.1 Requirements & Constraints

This section describes the requirements and constraints that our team has identified and categorized as main prerequisites for successful implementation of our project. The requirements and constraints listed will be used as guidelines throughout the development process, ensuring that we meet the necessary criteria to provide a meaningful solution to our users.

Functional Requirements

Requirements that fall under functional requirements pertain to the application's capabilities.

- Application shall allow for users to create and login to a personal account.
- The user shall be able to load datasets from a variety of supported formats from their device to a database.
- The user shall be able to select an active dataset from the datasets they've uploaded to the database.
- The user shall be able to select from supported whereabouts algorithms based on their current user type.
- Based on the currently selected whereabout algorithm the user should be able to create a query from a selection of query types.
- Query creation should allow the user to select subsets of data from the active dataset if desired.
- The interface should allow for query creation/configuration through interaction with the visualization space/window.
- The application interface should attempt to constrain queries that would otherwise be uncomputable within the listed constraint time as seen in the constraint section below.
- Visualization will take several forms depending on the current user type and whereabout algorithm (i.e. map overlay, 2D, and 3D white-space backgrounds).

Experiential Requirements

Requirements that fall under experiential requirements pertain to increasing application usability.

- User's preferences should be stored and recalled to create a seamless experience.
- Based on the current visualization tool, inclusion of a set of navigation features (Pan, Zoom, Rotate, ect.)
- User notifications and updates to inform the user of current application state or status.

Aesthetic Requirements

Requirements that fall under aesthetic requirements pertain to the application's visual appearance.

- Application’s user interface will take on a utilitarian aesthetic.
- The interface shall be designed in such a way that promotes intuitive interaction and navigability.
- The visualization tools provided by the application to the user should cater to the current user type.
- Visualization tool - color customization for points and probabilities - where applicable.

Non-Functional Requirements

Requirements that fall under non-functional requirements pertain to application’s construction.

- The application should take the form of a web application that can be accessed from a domain outside the hosts domain.
- Utilize exclusively well-tested open-source libraries.
- Develop with modular coding practices in mind.
- Maintain adequate code documentation.

Constraints:

The constraints outlined determine the level to which functionality is supported.

- Time from query request to visualization rendered should take no longer than thirty-seconds.
- Input datasets must adhere to a supported format.

2.2. Engineering Standards

Table 1 - IEEE Standards

Standard	Rationale
23026-2023 - ISO/IEC/IEEE: Engineering and Management of Websites for Systems, Software, and Services Information	These will be the standards we follow for our development of the web application.
15288-2023 - ISO/IEC/IEEE International Standard - Systems and software engineering--System life cycle processes	This standard will help us to follow a correct lifecycle of software from conception to retirement
1012-2016 - IEEE Standard for System, Software, and Hardware Verification and Validation	This standard we will use to help with requirements to best create the software and risks to make sure they are mitigated correctly. Along with testing and other areas.

Table 2 - Other Standards/Conventions

Standard	Rationale
Oracle Java Coding Conventions	We will be using a Spring Boot mysql server for the backend. Following these conventions will keep all code similar and easy to read.
Vue.js Style Guide	Using the Style guide from Vue.js will keep all code easy to read. This will also keep certain issues to a minimum following the same style.
Scrum Style	We need a style to develop our project. We will have Scrum Sprints that last 2-3 weeks.

3. Project Plan

3.1 Project Management/Tracking Procedures

Our project will implement the Agile methodology for project management and task completion. We chose this method because it is centered around frequent scrum meetings which allow for group discussion. Since we have such a small team and our project is mainly software development related, these scrum meetings will allow our team to gather insight from other team members as software-related obstacles arise.

Since many of the tasks we have designated will have tasks that build on each other, the Agile methodology will allow us to work from the beginning and power through each task in the scrum without strict deadlines. Although we have attempted to make a rough estimate of time that will need to be allocated for each task, it is difficult to predict where specific obstacles with integration and programming will fall. Because of these reasons, the Waterfall method was less appealing to our team since the time estimates are not as reliable with a mainly software development project where we are utilizing new tools we are not familiar with.

Our team will track progress through Github with individual folders for frontend and backend work. Since our project will utilize a VM and MySQL database, we will continue documenting progress in Google Docs and communicating via Text for information that is not on Github. We will follow the task decomposition outlined in section 3.2 for knowledge of which task should be started after completing a previous one.

3.2 Task Decomposition

Our whereabouts web application can be broken down into two major subsections - frontend and backend. Where the frontend, client side, will be responsible for managing user input, while the backend, server side, will run whereabout algorithm computation and handle database interaction. Since our team is adopting an Agile approach, team members will begin by working on the basic user authentication tasks and continue down the tasks listed in the subsystem legend - Figure 3. The major task groupings and the corresponding sub-tasks are highlighted below. Figure 3 illustrates the major tasks and how they interact from a systems-level perspective.

- Frontend
    - User authentication
        - Create login display, and form for accepting user input
        - Submit credentials to backend for authentication
    - Dataset uploads and format specification process
        - Create dataset upload and configuration display
        - Complete local file browse and select functionality
        - Check uploaded files formatting for compatibility
    - Query creation, configuration, and submission requests

        - Create query creation and configuration display
        - Query validator, prevent “long” query runtimes as specified in constraints section.
    - Visualization window and underlying engine
        - Setup inclusion of necessary plotting libraries
        - Add visualization window the the interface display
        - Create plotting handlers for datasets and whereabout data

- Backend
    - Setup Database
    - Authenticate login/signup
        - Query database for submitted credentials or add new user
        - Send error message or authentication token
    - Commit datasets to database along with the format specification
        - Query database and add new entries
    - Setup query endpoints for supported query types (i.e. range, contact)
    - Implement whereabout algorithms and package results to be sent to the frontend for visualization.

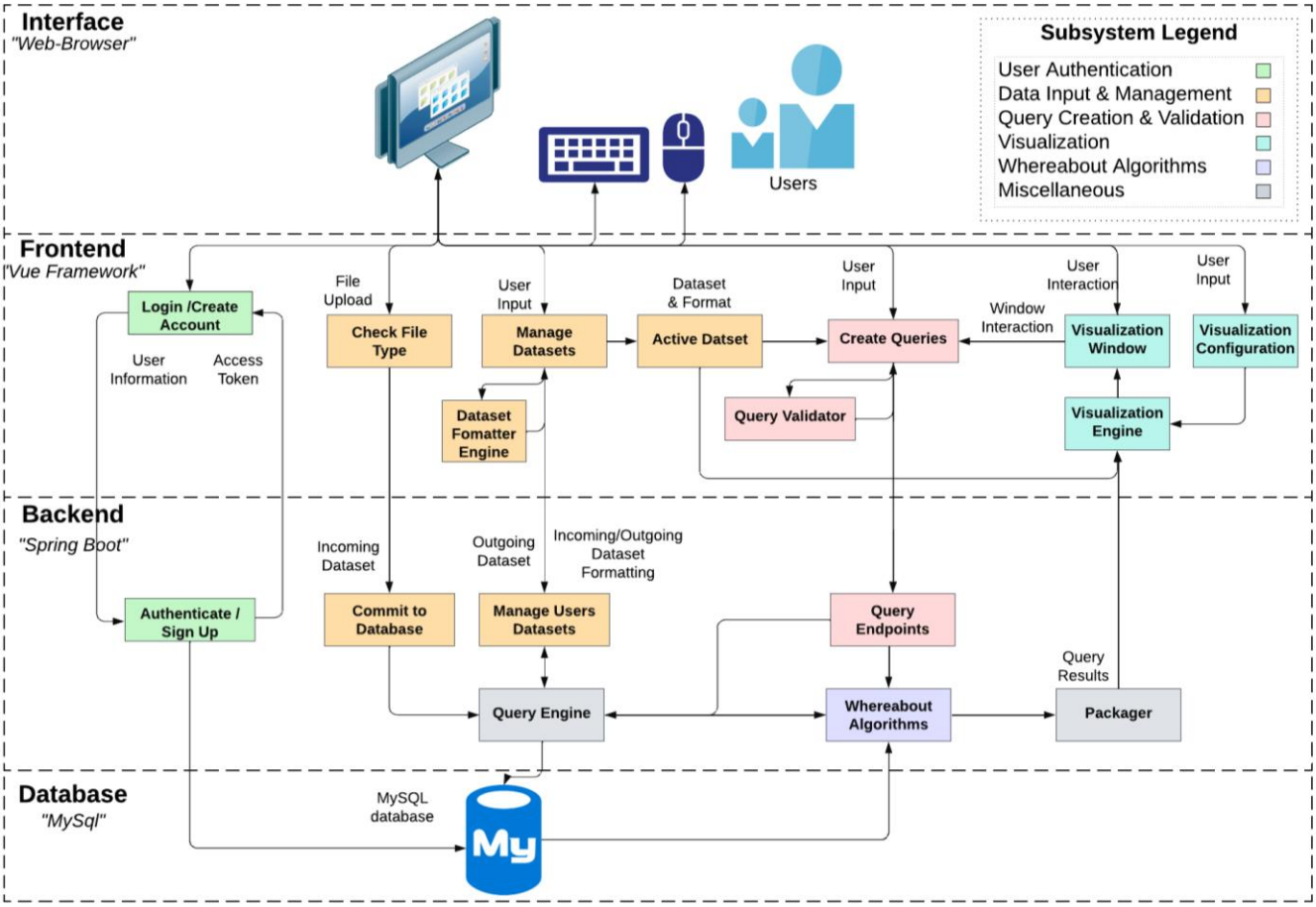

Figure 3 - *Systems diagram perspective of task decomposition.*

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

This section seeks to set progress goals and determine what successful accomplishment should look like.

Milestones

Shown below are the planned milestones we expect to accomplish. The milestones generated are based on the tasks and subtasks listed in section 3.2. For estimated completion of each task, reference the Gantt charts included in section 3.4.

Spring Semester

- Complete project frameworks and libraries
- Define and identify backend API, the contract between function calls from frontend and backend.
- Setup code base (Git, Vue Project)
- Develop preliminary versions of:
    - UI components
    - Database setup
- Finalize design documentation

Fall Semester

- Frontend
    - Render HTML passed from backend to frontend interface
    - Update the HTML from Vue application instance
    - Successfully make a request from frontend to backend
    - User can load file through interface
    - Can distinguish a loaded files format compatibility (i.e. able to be parsed with single delimiter specified)
    - Users can view a dataset through the interface and specify columns/rows (i.e. what data is - positional, timestamp, label, etc.)
    - Displaying a visualization window using plot libraries provided examples
    - Producing an aesthetic visualization window with data points from user dataset
    - Working whereabout query interface that supports a single query algorithm and type where data is manual entered
    - Visualization engine can take whereabout algorithm output and render meaningful interpretation to the visualization window
    - Whereabout query interface supporting multiple query types
    - Visualization engine supports visualization of multiple query types
- Backend
    - Store and retrieve test credentials in JSON format to database
    - Successfully register a new user to the database
    - Correctly authenticate an existing user by serving authentication token

- Store three datasets with unique formatting into database
- Successfully upload a new dataset and store to database
- Whereabout query endpoint can successfully take an input query from frontend and fetch necessary resources needed by the corresponding whereabout algorithm
- Single whereabout algorithm implemented and producing correct output
- Whereabout algorithm output interpreted and formatted in accordance to frontend needs
- Backend supports multiple query types and produces correct output

Metrics and Evaluation Criteria

To measure the progress of our efforts and milestones completion we will be using the following metrics:

- User interface components will complete the given action within a timeframe of a few seconds as the upper-bound. Action will be verified through use of print statements or an equivalent.
- Executing algorithms on the backend should complete their action within a timeframe given by our constraints section - no longer than 30 seconds. A timer will monitor the duration of execution.
- Visualizations of whereabout algorithm output should be correct. This will be evaluated by using known scenarios (i.e. test data provided by the client). This methodology will also be used to verify algorithm implementation correctness.

3.4 Project Timeline and Schedule

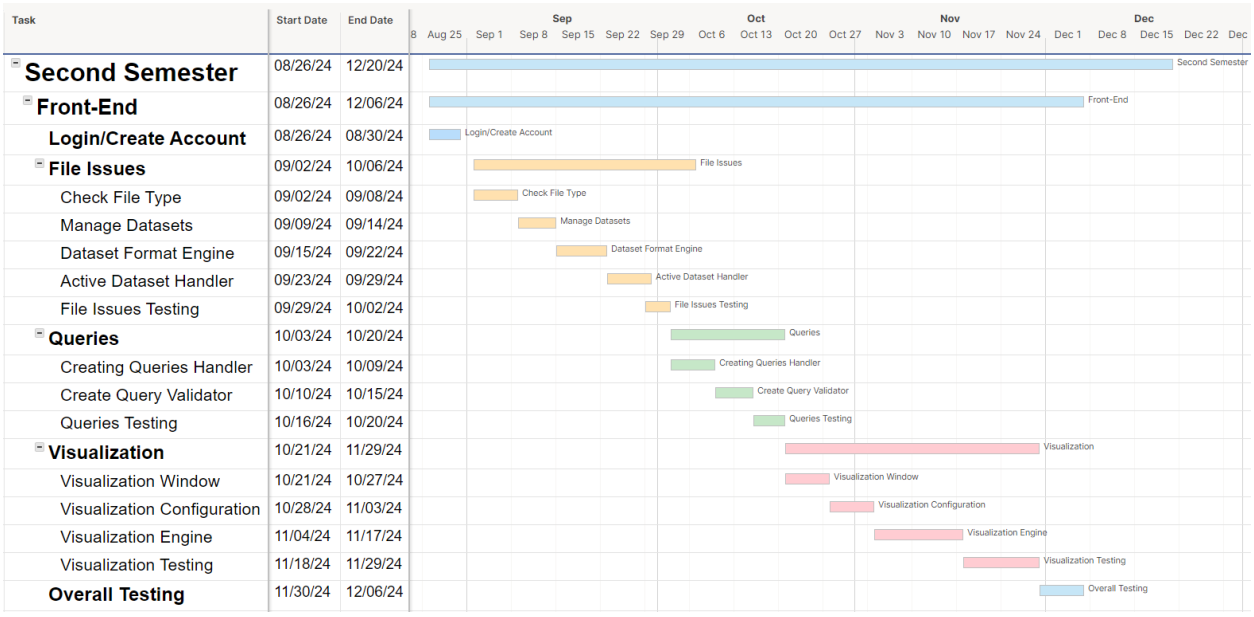

Figure 4 – Frontend Gantt Chart

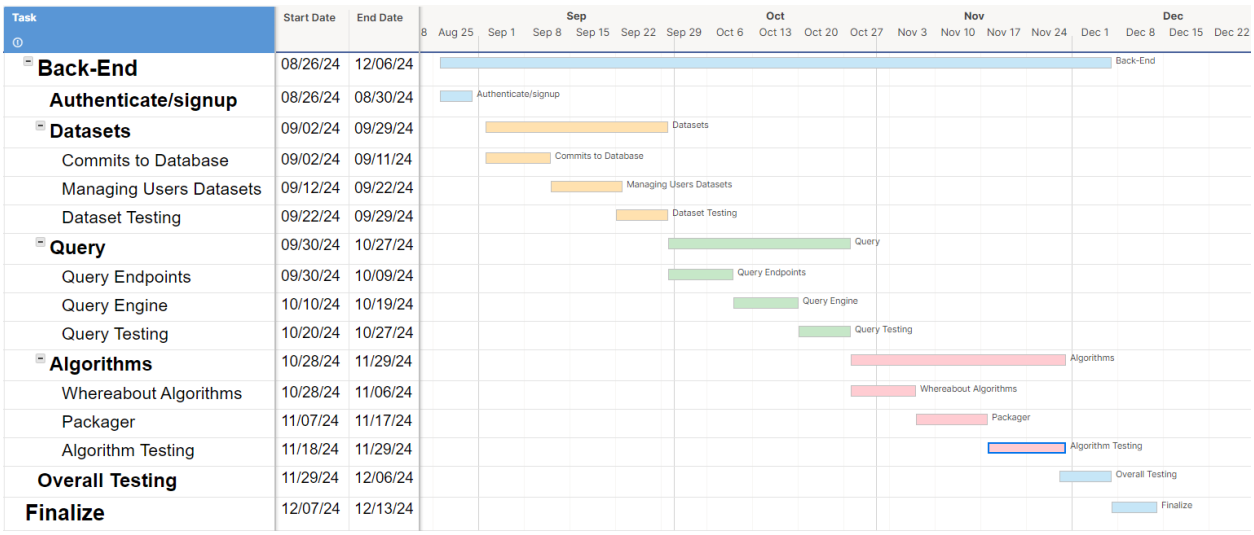

Figure 5 – Backend Gantt Chart

3.5 Risks And Risk Management/Mitigation

The following section contains a breakdown of the possible risk, description of risk, probability of occurrence, and correlated mitigation.

- Frontend
  - User authentication
    - Security vulnerabilities/issues, compatibility issues with certain devices or browsers.
    - Probability: .25
    - Testing across multiple platforms during development to ensure compatibility.
  - Dataset uploads and format specifications process
    - uploading large datasets could lead to eventual performance issues, also file formatting might become problematic.
    - Probability: .3
    - Limit to only accept one type of file for uploading data, optimize server side processing to help handle larger data sets.
  - Query creation, configuration, and submission requests
    - More complex/extensive queries may lead to performance issues, failure to validate and process certain types of query configurations for different desired outcomes.
    - Probability: .35
    - Optimizing query execution and extensive testing of all intended end user combinations/types of queries from data sets.
  - Visualization window and underlying engine

        - Performance degradation with larger data sets and local device constraints, integration issues with visualization tools such as pan, zoom, and move in several axes
        - Probability: .45
        - Testing tool compatibility and implementing strategies to manage larger datasets more efficiently without sacrificing performance.

- Backend
    - Setup database
        - Issues with scaling in terms of volume of data, limitations in speed might create bottlenecks.
        - Probability: .5
        - Performance testing and optimization of both database queries and indexes.
    - Authenticate login/signup
        - Security vulnerabilities in terms of authentication
        - Probability: .1
        - Ensuring correct security measures in user login/signup.
    - Commit datasets to database (and format specification)
        - Data integrity issues during insertion into database, compatibility issues with database configurations
        - Probability: .15
        - Testing compatibility across different database environments during development, implement transaction integrity checks for data entering the database.
    - Setup query endpoints for supported query types
        - Inefficiency processing of different query types might lead to performance bottleneck issues, full support missing from all different required query types.
        - Probability: .4
        - Continuous refinement and adjustment of query endpoint functionalities based on user feedback and issues/concerns discovered during overall development.
    - Implement whereabouts algorithms to send to frontend for visualization
        - Ensuring correct visualization for easy user interpretation of data sets, greater complexity of algorithm can result in performance degradation, integration issues with backend architecture.
        - Probability: .6
        - Performance optimization in optimizing algorithm(s) to ensure most efficiency when it comes to computation, involving continuous monitoring and logging methods to detect anomalies in behavior of algorithm(s), implement extensive documentation to aid in understanding and troubleshooting eventual issues.

3.6 Personnel Effort Requirements

The table below contains a list of tasks, estimated hours necessary to complete, and a brief explanation of each task. Each task's hours may be divided between multiple team members as will be determined by the completion of blocking tasks.

Table 3 – Tasks and Estimated Hours to Complete

Task	Total Estimated Hours	Explanation
Login/Create Account	10	Setting up GUI and Backend Calls to the Spring Boot server.
Check File Type	15	Make sure the file type is correct and formatting in the new file is correct. Send dataset to Spring boot server.
Manage Datasets	20	Receive datasets from the database that the user selects. Update selected dataset if necessary.
Dataset Format Engine	25	Checks that formatting in the updated dataset is correct.
Active Dataset Handler	20	Handles Current dataset being used for visualization.
File Issues Testing	15	Test all File Issues from Check File to Dataset Handlers works individually and when combined together for the frontend.
Creating Queries Handler	25	Selection GUI and background processes for creating Queries related to current dataset.
Create Query Validator	20	Checks that Query from user is valid to be sent to backend and processed in Query endpoints.
Queries Testing	15	Testing For Creating and validating to see that they work individually together and with the file modules.
Visualization Window	20	Create the window for the visualization of the datasets with interaction for the user and with the created query for it.
Visualization Configuration	15	GUI and Configuration methods for visualization of Current dataset with options for how the user wants to customize the visualization

Visualization Engine	40	Setting up Mapbox tool to use the modified query from the backend, the active dataset, and the visualization configuration from the user to visualize in the way the user wants
Visualization Testing	30	Testing visualization parts individually and checking that it integrates with the other front end modules correctly and effectively
Authenticate/signup	10	Create backend request receivers for Authenticating user signin and creating new users. Create a MySQL database.
Commits to Database	20	Receives new datasets from the front end and sends them to the query engine to be stored in the database.
Manage Users Datasets	20	Receives dataset request from user and sends info to query engine to get dataset. Sends chosen dataset from users database, and receives user updated datasets from the front end and sends them to the query engine to update dataset in database.
Dataset Testing	20	Test that all Receiving and Sending from the front end is working as required and will work with the front end.
Query Endpoints	25	Takes received query from front end and sends to query engine for saving to database and use in the algorithms.
Query Engine	25	Saves received data to MySQL. returns requested info from MySQL database.
Query Testing	20	Test that query modules work together and with all connected modules
Whereabout Algorithms	40	Uses user selected algorithm to ready query for visualization.
Packager	40	Packages algorithm output to be sent to Visualization in front end
Algorithm Testing	25	Tests that all algorithms in backend works with query options from the front end

3.7 Other Resource Requirements

Below is a preliminary list of resources that will be necessary to consider in the completion of this project:

- Github Page
- VM
- MySQL database
- Javascript/Springboot Libraries
- Website/Domain Name

This project will have no hardware resource requirements. Additional resource requirements may be identified during the implementation stage of the project.

4. Design

4.1 Design Context

4.1.1 Broader Context

As an introduction to our section on design, we begin by addressing the intended purpose and usage. The application is designed for research communities – either in academia or industry. It is designed to be used as a tool to be used by researchers from a wide range of disciplines with the commonality of needing to quantify the uncertainties related to position in between discrete measurements in time.

With many possible use cases, the societal impact can be challenging to state precisely. Direct needs being satisfied are that of the researchers and their work. It aims to streamline their existing work via a pre-built tool that can be used for analysis out of the box provided a dataset and a positional query. Indirectly, their findings may lead to new discoveries that shape human progression. Potential societal impacts are shown in the table below.

Table 4 – Societal Impacts of Our Application

Area	Example
Public health, safety, and welfare	A zoologist can collect location information from a local deer population to determine if the deer are likely to cross a public road – if yes, they can take action in installing road sign warnings in the area.
Global, cultural, and social	Generally, use of the application could be used to solve a plethora of problems found in the natural world – indirect effects propagate to society.
Environmental	Management of animal conservation movements, new discoveries in chemistry leading to cleaner energy sources. Con - resources used for computation expend energy.
Economic	The pre-built functionalities of the application reduce the time required to meaningfully analyze collected data.

4.1.2 Prior Work/Solutions

Our solution falls under the market sector – information technology, and more specifically can be classified into the sub-category data visualization tools. In the data-driven modern world the need for software assisted tools have become necessary to analyze the collected data in a way meaningful to humans.

Following market research we have identified several existing solutions shown here:

MathWorks - MATLAB

- Highly customizable, leverage many pre-built functions and utilities [11]
- Rich toolboxes for various applications such as signal processing [11]
- Interactive plotting functions through built in GUI [11]

Microsoft - Power BI

- Native integration with Microsoft ecosystem enhancing interoperability [9]
    - Excel, Azure, and Office 365
- Natural language querying, asking questions related to data in plain language [9]
- AI/Machine Learning capabilities generate automated insights [9]

Salesforce - Tableau

- Drag-and-Drop interface for building visualizations quickly [10]
- Supports live connections to data sources [10]
- Patented VizQL technology translates drag-and drop actions into database queries [10]

These products provide extensive functionality related to the visualization of data each catering to a given target audience. MatLab is a programming language with many built in functionality that can be used to create custom visualizations for almost any scenario – this comes at the cost of a knowledge barrier to entry and time to implement. Power BI and Tableau cater to a broader general audience who wants to go from dataset to visual assisted analysis quickly. Power BI achieves this through its integration of AI tools into its platform, while Tableau leverages a drag-and-drop feature to customize a data processing pipeline.

The closest of the three to our solution is the Tableau solution to data visualization. It allows for customization of the data processing while minimizing the barrier to entry. Our application, although not able to cater to a broader audience of potential users – is the only entirely pre-built solution for whereabout queries.

4.1.3 Technical Complexity

The design and implementation of our project has many technical challenges that will draw from engineering principles and mathematics.

Many of the challenges related to our web-application development include:

- Frontend

    - Knowledge related to HTML, CSS, and Javascript
    - Ability to read and understand API documentation for library integration. Example libraries include:
        - Vue.js
        - Plotly.js
        - Mapbox
- Backend
    - Knowledge related to Java and backend web development
    - MySql database querying
    - Mathematics driving whereabout algorithms.

4.2 Design Decisions

4.2.1 Design Decisions

One of the design decisions our group had to navigate was choosing which application(s) the end product would be accessible over – like, for example, whether that be a web-based, desktop, or mobile application, as this will ultimately heavily influence both the level of work required for implementation and the accessibility/useability of the project. Similarly, a decision involving the acceptable formatting of entered data files needed to be made, with complementary concerns as: on the one hand, limiting different acceptable file types would decrease complexity but , on the other hand, it would also decrease end user accessibility and ease of use.

When analyzing data and utilizing entered data for processing specific queries, necessary considerations needed to be made regarding the back end library and compiler, as these aspects can have an instrumental role in determining the maintainability, scalability, performance, and overall complexity for the application. Complementary to this, decisions for the frontend also had to be made regarding the certain primary concerns, as well as regarding the issues of ensuring compatibility between both frontend and backend.

A different category of concerns that needed a consideration related to the overall design pertained to the structure of the backend from the perspective of the types of queries the end user would utilize most frequently. These were also necessary for ensuring that the overall performance and scalability would not suffer due to particular types of queries.

While considering what functionality was necessary for the front end visualization, several decisions had to be made regarding what basic tools will be accessible to the user, such as pan, zoom, and move, as well as more advanced settings/configurable options such as selecting specific cuts/sections of the visualization, transparency and color options. The main trade off being again added complexity for user versatility and accessibility, but also performance while these tools are in use within the high level user interface.

4.2.2 IDEATION

For the design decision of choosing the web application framework, we utilized the lotus blossom diagram/technique. Based on this, we were able to narrow down our search of which framework/language we planned to utilize. We identified several different potential candidates for what we could use on the frontend or backend framework through open brainstorming. We ensured that all possible ideas were heard from the entire team in order to thoroughly analyze and consider all types of options available.

Java- Containing a vast array of libraries, tools, different potential frameworks, and scalability, as well as several team members having experience with this language.

Javascript- Utilization on both frontend and backend would allow for simpler implementation, but by itself, with the team having extensive experience with this language, along with extensive resources and high performance, it's clear why this is one of our top options.

C#- Both incredible speed and scalability come at the tradeoff of limited libraries compared to other options and unfamiliarity with the group.

HTML- While offering great compatibility with modern web browsers, limited functionality and lack of scalability could offer unique challenges if utilized.

Python- While offering both clean syntax and the most extensive/available libraries, simplicity and flexibility, the drawbacks of lack of group experience with projects would prove to be extremely detrimental to progress given our agile methodology.

By exploring these different options via ideation techniques - i.e., the lotus blossom method, we were able to evaluate the pros and cons of each approach and make informed decisions aligned with our project goals and team capabilities.

4.2.3 DECISION-MAKING AND TRADEOFFS

The main ideated options were related to the application format, including mobile, desktop, and website. With these options, we were able to quickly narrow down our options to desktop or web-based, since a majority of our users will have data sets on their computers. Since data sets are relatively large, it would be impractical to utilize a mobile app as the primary format since most users will not have access to research data on their phones. Additionally, mobile applications for Apple and Android devices utilize different languages, so we would likely only have the breadth to choose one. This would be less accessible to diverse categories of users and limit the functionality of our application.

Between the desktop and website options, we determined that a website will be most accessible and effective since it can be accessed from any type of computer. It is also practical since users will not need to download software permanently and can instead access their visualizations from any device with an Internet connection. When looking at the skillset of our team, it was also apparent that either a mobile or web application would be most suitable for the skills we had developed. See Figure 6 below for the Weighted Decision Matrix accounting for the factors described above.

		20	25	38
Options		Mobile	Desktop	Web
Decision making factors	Weighting	Your Score	Your Score	Your Score
Breadth	3	2	3	5
Skillset	2	4	2	4
Data Access	3	2	4	5

Figure 6 – Weighted Decision Matrix for Application Format

After determining the format we would use, we also had to take into account the languages, libraries, and compilers we would primarily use in our design. We ended up choosing SpringBoot for our backend on the Intellij compiler, since some of our team members utilized these tools in previous projects. For the frontend, we ended up deciding on Node.js and Javascript languages since they were most similar to other languages we had knowledge of and were compatible with our backend choices.

4.3 Proposed Design

4.3.1 Overview

The current design for our web-application consists of five primary subsystems and a few secondary or helper subsystems. The primary subsystems are as follows:

1. User Authentication/Registration - Allow users to create/login to their account to access private dataset previously uploaded.
2. Data Input & Management - Allow users to upload new datasets and specify the format to ensure proper parsing.
3. Whereabout Query Creation & Validation - Allow users to interact with their data and format whereabout queries.
4. Whereabout Algorithms - Performs calculations based on a submitted whereabout query.
5. Visualization - Plot data and whereabout algorithm to a window for the users to interpret.

Secondary subsystems include:

1. Database Query Engine - Given user information such as credentials and datasets, fetch and store results as required.
2. Whereabout Algorithm Result Packager - Take the raw results from the algorithm and package it in a way that is compatible with the visualization tool.

Most of the primary subsystems have both a user interface component seen by the user, and a hidden portion that operates behind the scenes. This is true for all except the whereabout algorithm, completely hidden, and the visualization, entirely visible subsystems. Together these systems work together to produce the web-application. Figure 7 shows several high-level components of different user-interface functionalities displayed to the user.

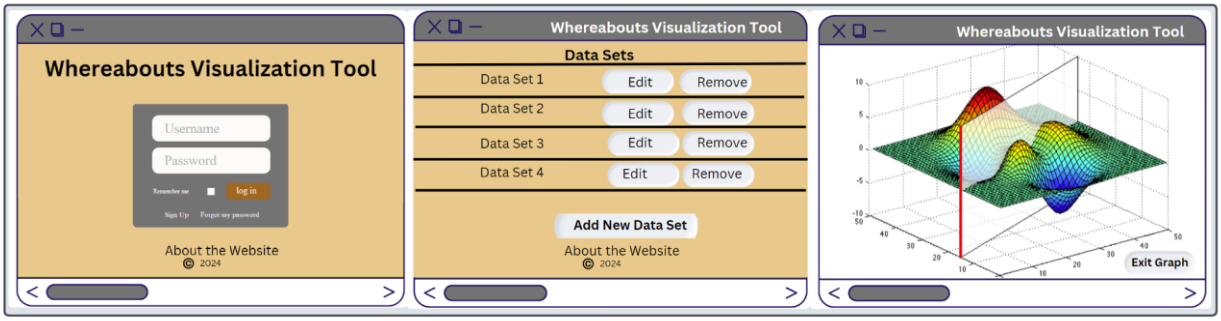

Figure 7 - Several High-Level Functionalities; User Perspective

4.3.2 Detailed Design and Visuals

Describing the design in detail requires a more in depth view of how the subsystems, introduced in section 4.3.1, are interconnected and joint functionalities. Figure 3 above from section 3.2 illustrates the interconnections between blocks of which are color coded on the basis of the

subsystem they belong to; see included subsystem legend. You will also see the before mentioned separation of visible and non-visible components denoted by the larger categorization of frontend and backend. Where frontend contains predominantly visible user interface elements, and backend comprises hidden functionalities or resources.

In further explanation of the diagram we are making two assumptions, firstly that all components falling under frontend and backend categories will be developed by the same groups of individuals, as mentioned in the task decomposition section of design document 3. Secondly all components in their respective front or back end category can readily pass information to one another. For these reasons we will not describe intra-categorical relationships in detail and will direct our attention to the inter-categorical relationships, the backend API. Referring again to Figure 3, there are essentially 6 lines or pathways that bind the front and back end together. Symbolically, they resemble HTTP request methods, and are used to request and send data between the front and back end. In order to effectively create a layer of abstraction, well defined data structures have been proposed for the 6 pathways. Seen another way, the 6 links can be grouped into three bi-directional links and related to the subsystems- they include user authentication, data input and management, and whereabout queries and results.

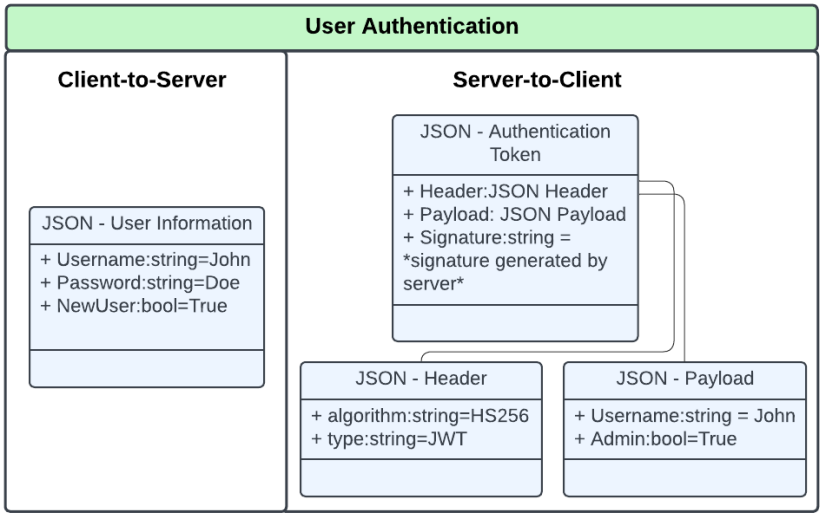

Figure 8 – User Authentication JSON Objects

Information passed over a link is structured using JSON objects, where each entry in an object comes in a “key” “value” pair. Additionally, objects can be nested within parent objects, allowing for complex data structures to emerge. Figure 8, shown above, gives the object details for the information being passed during user authentication. The user submits their username and password to the server. The server fetches user information from the database and if the credentials match the submitted input the server generates a unique signature and embeds it within the JSON authentication token to be returned to the client. In subsequent requests for

information from the client requiring authorization to access, the client will provide token information.

Slightly more involved objects are necessary for the data input and management links. In reality there are three routes for information – the loading of the raw dataset in its entirety, the passing of the dataset format, both of which are moving from client-to-server, as well as a parsed dataset returned by the server to the client. Three routes require three JSON objects to carry this information, laid out in Figure 9 below.

The user uploads their dataset through interface interaction, which can be checked locally in frontend for initial verification; this information is passed using the "Raw Dataset" JSON object. The backend will store all information provided in the file. The user will next specify the layout of the dataset (i.e. What rows and columns contain relevant information for analysis, and how the application should interpret the time stamp). This will be stored alongside the initial dataset within the database.

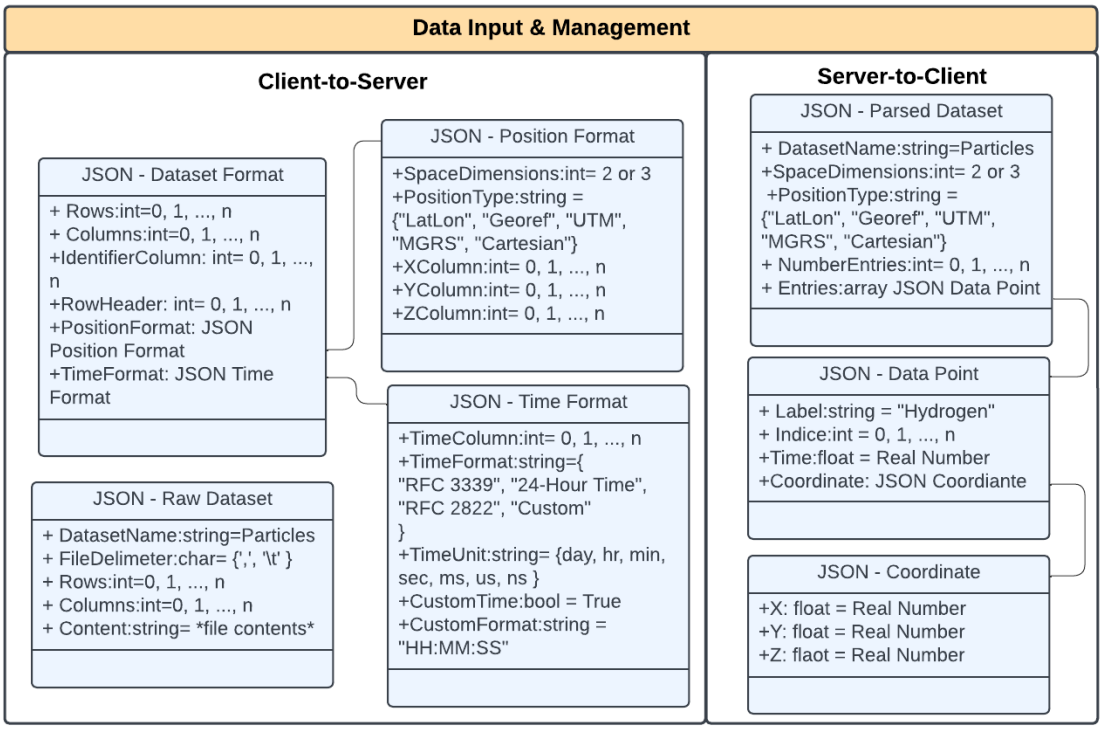

Figure 9 – Data Input, Formatting, & Parsed Dataset JSON Objects

Given the "Raw Dataset" and "Data Format" objects the backend will produce a parsed dataset object to be sent to the client. It will contain nicely packaged data points alongside additional information relevant to the visualization window.

The last of the links to be explained are for that related to whereabout queries and the corresponding results returned after completion. This portion may be the most obfuscated for the layman viewer, for it is related to the resources required by the whereabouts algorithms. We note that the JSON objects shown in Figure 6 are not final, they largely only support a single algorithm, “value” labeled “Cones” for “key”- Algorithm of JSON object “Query”. Support for the “Bridgelets” algorithm can easily be adapted without needing to modify the priori by adding another nested JSON object “Bridgelets Information” within the “Query” object.

The “Cones” algorithm in its simplest form can be observed from the figures included in design document 1, but can be generalized to a series of positions in time. The output of the algorithm in this case will be chain of ellipses or “beads", also referred to as a “necklace”. The intersection of the area taken up by the necklace with the area of interest divided by the area of the necklace gives the probability of the object residing within the area of interest at a given time in between the first and last points of the necklace.

Given the premise of algorithm operation, the algorithm specific queries have JSON objects of their own; the “Query” object can be expanded on to support more query types and only two types are provided in the figure. The range query adaptation takes two series of time varied positions, finds their necklaces, and determines the range of possible distances separating the two objects during the specified time interval. This is why under the “Range Information” object you see two arrays of “Data Point”.

The contact query adaptation is nearly identical to the algorithm descriptor case. The “Contact Information” object translates contains a single array of “Data Point” to be used to create the necklace while the shape information, namely shape type, points, and radius for the circle case makeup the region of interest.

The results of the array will be compiled into a JSON “Packed Query Results” object. This object contains entries containing resulting information from all algorithms and query types in the outermost portion of the structure. This is not extremely scalable and is subject to change to a format similar to the client-to-server query submission – where each query type has its own object. Regardless, the main contents include quantitative results such as probabilities and distances, as well as shape indices to be used by the front-end visualization engine (“Area Data” entries).

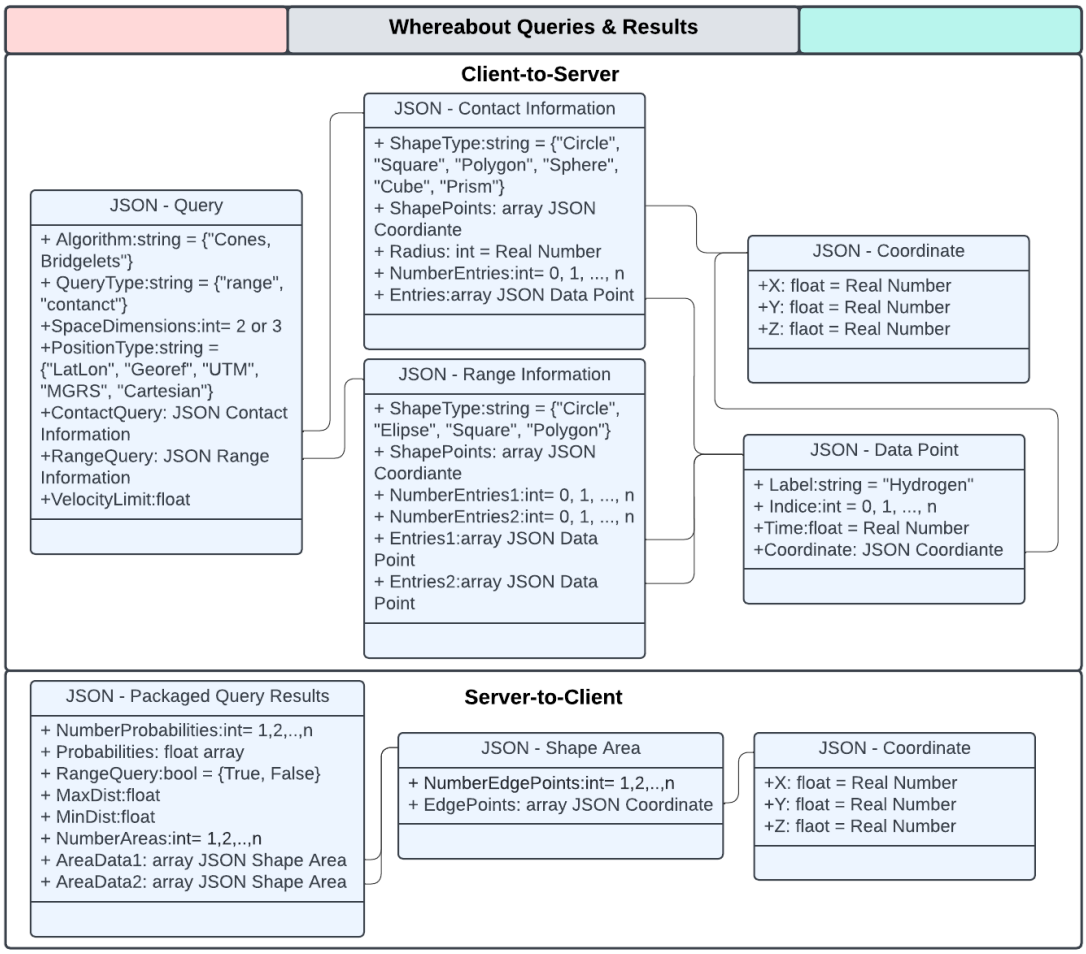

Figure 10 – Whereabouts Queries & Results JSON Objects

The visualization engine and window work together to view the data and algorithm results. The window is where visualizations take place while the engine is composed of logic driving the updating and management of the window. The window may contain a 2D map overlaid with data points and output algorithm necklaces or a 3D white space with equivalent volumes based on the dimensionality of the dataset provided by the user.

4.3.3 Functionality

The user would start by logging in or signing up and the frontend diagram for that process is shown in Figure 11, which goes over what the system would be doing when the user first starts the application. Figure 12 is the backend side for the same part of the system. The backend handles the checks and storing of user info as shown in the figure.

The next main functionality is dataset handling, and Figure 13, Figure 14, and Figure 15 all describe how datasets are handled in both the frontend and backend. Figure 13 describes how the Front-handles new and existing datasets and how the user chooses what active dataset they want. Figure 14 shows how the backend takes in and saves new datasets sent from the front end. Finally, Figure 15 shows how the backend sends saved datasets back to the frontend for use.

The final large functionality is how the visualization works for the frontend and backend. Figure 16 describes at a high level how the frontend will work with visualization and what the user will go through when starting it. Figure 17 describes how the backend will handle visualization with the different choices for the user and that what will be sent back will be packaged then sent to the visualization tool and brought up with the users settings to be seen.

4.3.4 Areas of Concern and Development

Our current website application design satisfies many of the requirements and user needs we have established. The application will have the ability to store datasets of a specified type, allow users to manage saved data sets, and display the requested visualizations using one of the two implemented whereabouts algorithms. These functionalities address most of the primary requirements and user needs that we have addressed in previous sections. In the future, our application could likely be expanded to implement additional algorithms and views for visualizations which could appeal to additional users.

As of now, it is unknown how long processing and saving data will take, which may be one area of concern relating to user expectations. Additionally, we have a relatively small team with limited experience in website design. Significant research and learning may be necessary in order for our team to implement the requested features of the application. Because of this, our team may not have enough time to implement all functionalities and algorithms requested by the client.

Our strategy to mitigate this concern will be to work incrementally and prioritize a single algorithm in order to make sure we have a complete and working application by the end of the semester. A question for our TAs and our client would be: how can we prioritize client requirements given that we have a limited amount of time for implementation?

4.4 Technology Considerations

The three main technologies that we will be using are Intellij, Vue.js and Mapbox. We will also have a VM from ETG to run our backend server. Some strengths for using Intellij is it has many built in Spring specific features to work with a Spring Boot server. Another advantage with Intellij is it has many debugging features to help fix errors that come with the creation of a server. Some disadvantages of Intellij is it is not friendly for new users, due to the complexity of some of the features. The trade off of many built in features to the learning curve of Intellij is one that we think is worth it as it is one of the best softwares for the design. An alternative to intellij could be the Spring tool suite as it is a specific tool for creating spring boot servers.

A strength of Vue.js is the fact that it is component-based architecture as this makes it easier to create the frontend code as the components can be reused in different functions. Another strength is that it integrates easily with Mapbox making what we need to create our visualization tool. A weakness of it is the performance as large visualizations could cause issues and with that our plan is if performance suffers to split the visualization up into multiple segments to be shown as to not slow the performance of the tool. A trade off that we will see is the component-based architecture with performance as the architecture will speed up the development process but with the performance issues finding a balance between the quickest development and the correct optimization will be key. An alternative to Vue.js is React, another Javascript based system that can easily integrate with Mapbox.

The last main technology used in the design is Mapbox. It will have most if not all of the features that will be needed, and it will be tested to ensure all functionality that it will need is part of the software. It also is known for having good performance with large datasets helping to negate some of the performance issues that could happen with Vue. An issue that could arise is that Mapbox does not have all the functionality needed for our design so a backup that can be used possibly with Mapbox if it does not have the functions needed is plotly.js which is also easily implemented into Vue and is our planned backup software if the need arises.

4.5 Design Analysis

So far, we have focused on researching the languages and libraries available for geographic, 3D, and 2D graphing. As we have discovered new tools or information about our chosen libraries, some of our choices have changed subtly, although our overall design plan has remained the same. Currently, we are focused on finding tools that save us additional time during implementation, and we believe that the tools we have chosen will allow us to complete the whereabouts application successfully. We hope to be able to build a more complete prototype that can handle simple data inputs once we have received a VM and website. Once we have set

up our server and have a VM running, we will be able to start attempting to connect frontend and backend in a simple prototype website application to prove proof of concept.

5. Testing

5.1 Unit Testing

Unit testing will be conducted on most functions and methods as completed in order to ensure accurate outputs and proper error handling. Since unit testing deals with individual functions, focusing on unit testing as the preliminary software testing method will be more feasible.

- Correctness and accuracy of critical functions/methods
    - Conical Algorithm Function (Backend)
        - Success Criteria: Comparison of output calculated data points to expected values from externally completed calculations
    - Bridgelet Algorithm Function (Backend)
        - Success Criteria: Comparison of output points of interest to externally calculated expected values
    - Getting, Posting, and Deleting in the Database (Backend)
        - Get Success Criteria: Comparison of received data from data base through Get command to expected value stored in database
        - Post Success Criteria: New posts to database store values, tested using subsequent get and delete unit tests
        - Delete Success Criteria: Deletes data that is already stored in database, tested using post and get methods as necessary
    - Login Functionality (Frontend and Backend)
        - Success Criteria: Frontend uses login function call to send input login and password information to backend
        - Success Criteria: Backend uses login function to call to database and validate input password and security token, returning successful and unsuccessful login attempts
    - Graph Visualization (Frontend)
        - Success Criteria: Tool graphs input data accurately and has proper error handling if data points are invalid or do not match input format

5.2 Integration Testing

Integration testing will be emphasized when considering communication between the frontend and backend components. Examples of this include:

- Sending and receiving data (Frontend to Backend)
- Storage and retrieval of data from the database (Backend to Database)
- Receival of data points from algorithm to visualization tool (Backend to Frontend)

These tests will be written and implemented after major components and unit testing have been completed. Tests will be considered successful if they make proper function calls and output the expected data based on what was stored or given. Successful results of these tests will ensure system testing goes smoothly, and major functions created by different team members communicate as intended.

5.3 System Testing

System testing will be conducted once all major components of the system have been implemented, unit tested, and integration tested with related components. System testing will involve running through all phases of application use, such as:

- User account creation
- User login and 2 factor authentication
- Data input and saving to database
- Receiving output from algorithms
- Graphing received data using the visualization tool
- Saving and exiting the program
- Deleting data sets from user account

The goal of system testing is to ensure that the application functions across the board as expected and is usable for its intended purpose of visualizing the expected location of objects.

5.4 Regression Testing

To ensure any new functionality added does not break any current functionality we will do testing to ensure everything works as expected before adding the new functionality to the main application. We will do this testing using separate branches in GitLab to ensure that the new functionality will be built and tested separately from the main application as to keep the applications current state functional. The critical features that cannot be broken on an update are login, all algorithms and visualization added, storing and getting of datasets, and uploading a file all these must be in working order before an update can be applied.

5.5 Acceptance Testing

To ensure functional design requirements are being met we will use the use cases from the functionality section and make sure the design and any addition to the design falls within those use cases. If a use case must be updated to the current design it will reflect that. To ensure that

the non-functional design requirements are being met we will meet with our client and update him on the state of the application to ensure it fits with what he wants, and we can provide as an application.

5.6 Security Testing

Since our team has limited software development experience and minimal cybersecurity experience, our application is unlikely to be rigorously security-tested. However, we will still attempt to create as secure an environment as possible, such as by using 2 Factor Authentication tokens and creating a login with password. Unit testing for these components will also relate to security testing since they are security-oriented features.

Additionally, we plan to run automated security tests that are available in Intellij in order to understand possible vulnerabilities that are detected, especially in our backend software. We also plan to continue risk assessments for possible security vulnerabilities as we develop and finetune our security architecture.

5.7 Results

We will use the above testing in tandem with our requirements to ensure that all functionalities will be provided to our client. Also, with the above tests we will use the results to reevaluate any issues we come across and change the design accordingly to create the best possible product for our client.

6. Implementation

As per section 3.3 we are following along the milestones specified in that section. We have decided on our frameworks and started setting up our codebases in GitLab. Also, we have received a VM from ETG and are in the process of setting it up to run both the backend and frontend processes. Then to start our implementation we plan on starting out with getting a basic call from the frontend to the backend as well as having the backend connect and use a MySQL database. Then also setting up a return to the frontend from the backend/database. Along with those two another stretch goal is we plan on getting a basic GUI set up. This will be the start to our implementation and will help provide us with a good starting point for our application in the fall semester following the Gantt charts in section 4.3.3.

7. Professional Responsibility

7.1 Areas of Responsibility

Table 5 – Defined Areas of Responsibility: NSPE and IEEE

Area of Responsibility	Definition	NSPE	IEEE
Work Competence	Perform work of high quality, integrity, timeliness, and professional competence.	Perform services only in areas of their competence; avoid deceptive acts.	To maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;
Financial Responsibility	Deliver products and services of realizable value and at reasonable costs.	Act for each employer or client as faithful agents or trustees.	N/A
Communication Honesty	Report works truthfully, without deception, and understandable to stakeholders.	Issue public statements only in an objective and truthful manner; avoid deceptive acts.	To avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist;
Health, Safety, and well-being	Minimize risks to safety, health, and well-being of stakeholders.	Hold paramount the safety, health, and welfare of the public.	to hold paramount the safety, health, and welfare of the public
Property Ownership	Respect property, ideas, and information of clients and others	Act for each employer or client as faithful agents or trustees.	To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, to be honest and realistic in stating claims or estimates based on

			available data, and to credit properly the contributions of others;
Sustainability	Protect the environment and natural resources locally and globally.	N/A	To strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment;
Social Responsibility	Produce products and services that benefit society and communities.	Conduct themselves honorably, responsibly, ethically, and lawfully so as to enhance the honor, reputation, and usefulness of the profession.	To improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies, including intelligent systems;

7.2 Project Specific Professional Responsibility Areas

- Work Competence
    - Ensuring a high quality product with well written code and an ultimately user friendly and understandable user interface.
- Financial Responsibility
    - The only applicable aspect of our project comes with acquiring the VM we intend to use, and how to utilize one to best fit our project's needs without wasting university resources.
- Communication Honesty
    - Meeting with our advisor weekly and being open as a team play a critical role in the success of our project, in order to collaborate effectively and to create a finished project on par with the expectations set for us by our advisor.

- Health, Safety, and well-being
    - Protecting user login information and uploaded data from being accessed by outside parties is critical to the integrity and safety of our project, and is a priority when developing our final product.
- Property Ownership
    - Keeping the data uploaded to our final product intact, respecting the sensitivity, as well as any ideas/intellectual property present within that data.
- Sustainability
    - With future scaling in mind, the allocation of resources for our product efficiently and without waste is critical.
- Social Responsibility
    - This product will be openly available to a broader community and can be utilized for the visualization of several different types of data sets, and as a result our users will have a better understanding of their entered data.

7.3 Most Applicable Professional Responsibility Area

Social responsibility is the most applicable area of professional responsibility related to our product, as it encompasses the main object of our project itself which is to create a tool that serves to better visualize moving objects than other existing comparable products. The main focus points of our project being improving user accessible tools for ease of access, and increasing the speed at which our project runs, both are oriented towards improving this type of tool which will ultimately benefit our large range of intended users.

8. Closing Material

8.1 Discussion

The completion of the project, defined by satisfying the specified requirements, will yield a web application accessible to researchers of varying backgrounds and disciplines. It will provide them utility by streamlining the visualization of their data and configuring a query from supported queries. We hope our project will reduce the time required for data analysis of moving objects for our users.

8.2 Conclusion

The project's design has come a long way since we first began working on it in January. We have established the frameworks we will be using for front and backend development as well as a database. We have established a set of libraries that best fit our needs for UI component development, data visualization and user authentication.

We have created a high level system diagram to organize the interconnections between subsystems and created lower level documentation for how these subsystems pass information to one another.

Overall, our efforts in establishing the project's design framework and documentation have laid a solid foundation for the subsequent development phases. We are confident that our meticulous planning and strategic decisions will contribute to the successful execution and delivery of the final product.

8.3 References

[1] Trajcevski, Goce, et al. “Uncertain Range Queries for Necklaces.” 2010 Eleventh International Conference on Mobile Data Management, IEEE, 2010, pp. 199–208, https://doi.org/10.1109/MDM.2010.76.

[2] Zhang, Bing, et al. “Towards Fusing Uncertain Location Data from Heterogeneous Sources.” *GeoInformatica*, vol. 20, no. 2, 2016, pp. 179–212, https://doi.org/10.1007/s10707-015-0238-6.

[3] Krumm, John. 2022. Maximum entropy bridgelets for trajectory completion. In Proceedings of the 30th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '22). Association for Computing Machinery, New York, NY, USA, Article 79, 1–8. https://doi.org/10.1145/3557915.3561015

[4] Vue.Js, "Introduction," Vue.Js, Available: vuejs.org/guide/introduction.html. [Accessed Mar. 19, 2024].

[5] Oracle, "Code Conventions for the Java Programming Language: 1. Introduction," Available: www.oracle.com/java/technologies/javase/codeconventions-introduction.html. [Accessed Mar. 19, 2024].

[6] "ISO/IEC/IEEE International Standard - Systems and Software Engineering -- Engineering and Management of Websites for Systems, Software, and Services Information," in ISO/IEC/IEEE 23026:2023(E), vol., no., pp. 1-70, Jul. 18, 2023. doi: 10.1109/IEEESTD.2023.10186263.

[7] "ISO/IEC/IEEE International Standard - Systems and software engineering-- System life cycle processes," in ISO/IEC/IEEE 15288:2023(E), vol., no., pp. 1-128, May 16, 2023. doi: 10.1109/IEEESTD.2023.10123367.

[8] "IEEE Standard for System, Software, and Hardware Verification and Validation," in IEEE Std 1012-2016 (Revision of IEEE Std 1012-2012/ Incorporates IEEE Std 1012-2016/Cor1-2017), vol., no., pp. 1-260, Sept. 29, 2017. doi: 10.1109/IEEESTD.2017.8055462

[9] Microsoft, "Introduction to Power BI," Microsoft PowerBI Explanation, 2023. [Online]. Available: https://powerbi.microsoft.com/en-us/what-is-power-bi/. [Accessed: April 16, 2024].

[10] Salesforce, "Learn About Tableau Features," Salesforce - Tableau Explanation. [Online]. Available: https://trailhead.salesforce.com/content/learn/modules/learn-about-tableau/learn-about-tableau-features. [Accessed: April 16, 2024].

[11] MathWorks, "MATLAB Documentation," MATLAB Explanation. [Online]. Available: https://www.mathworks.com/help/matlab/index.html [Accessed: April 16, 2024].

8.4 APPENDICES

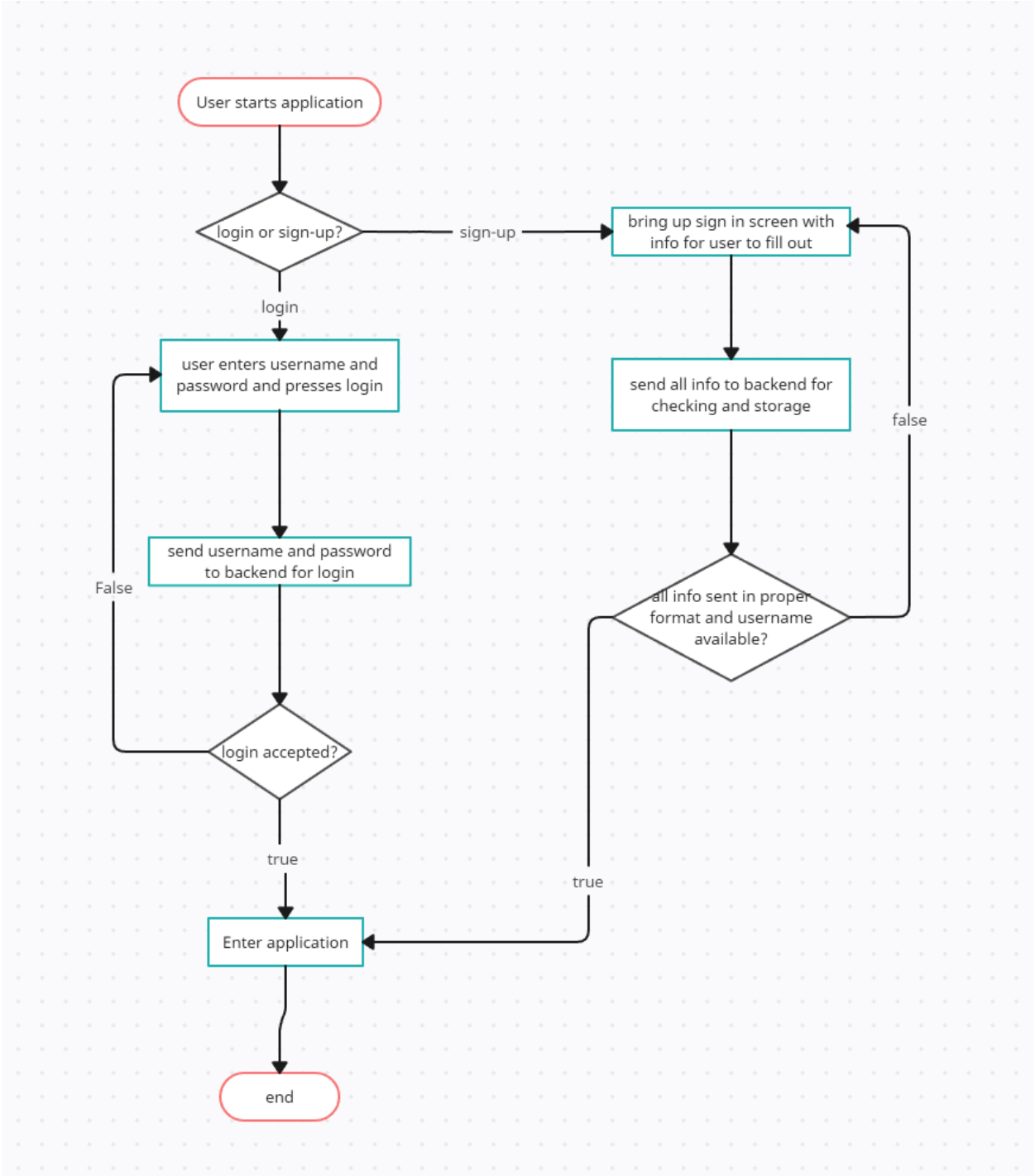

Figure 11 – Frontend login and sign up

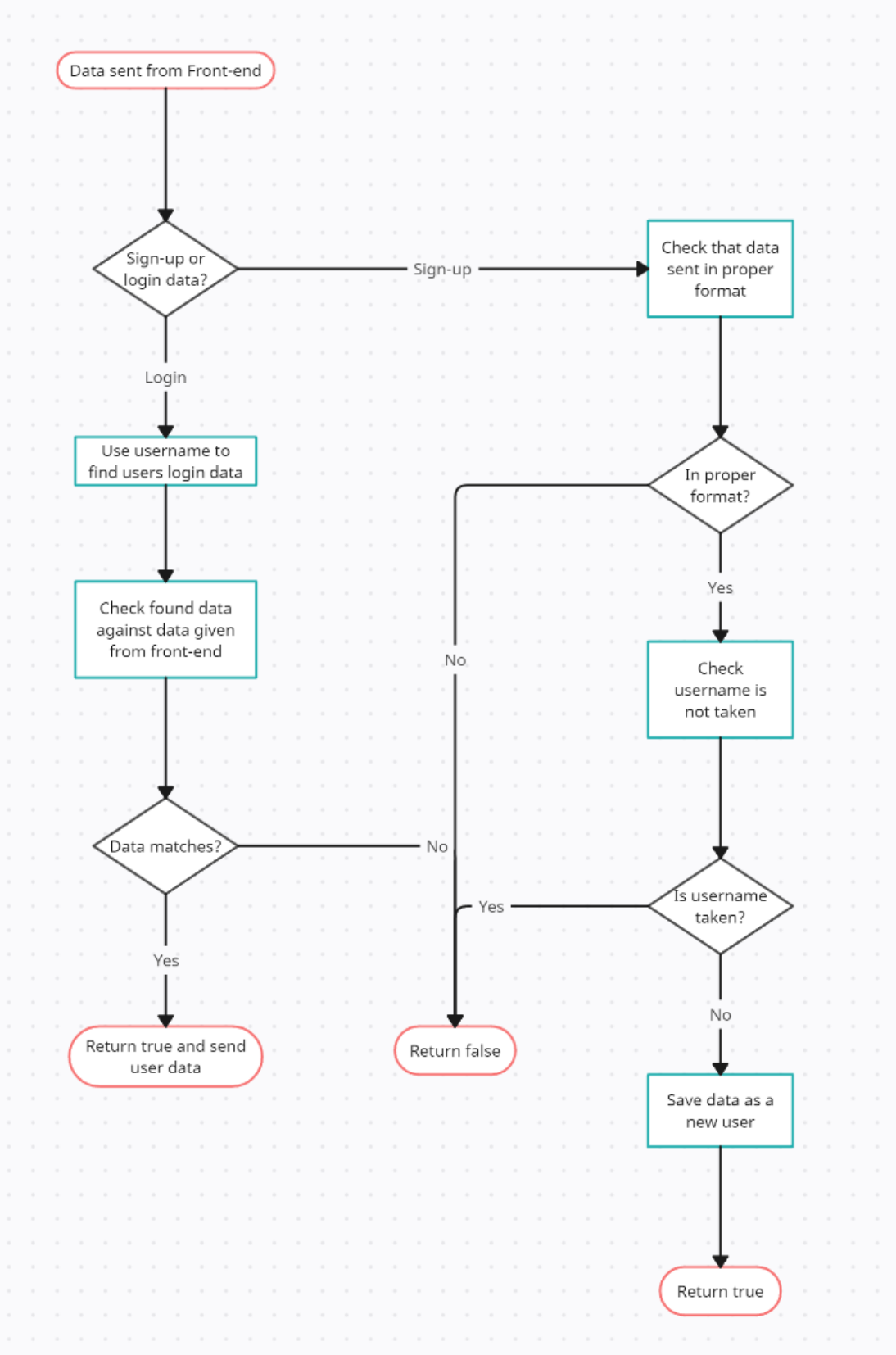

Figure 12 - Backend login and sign up

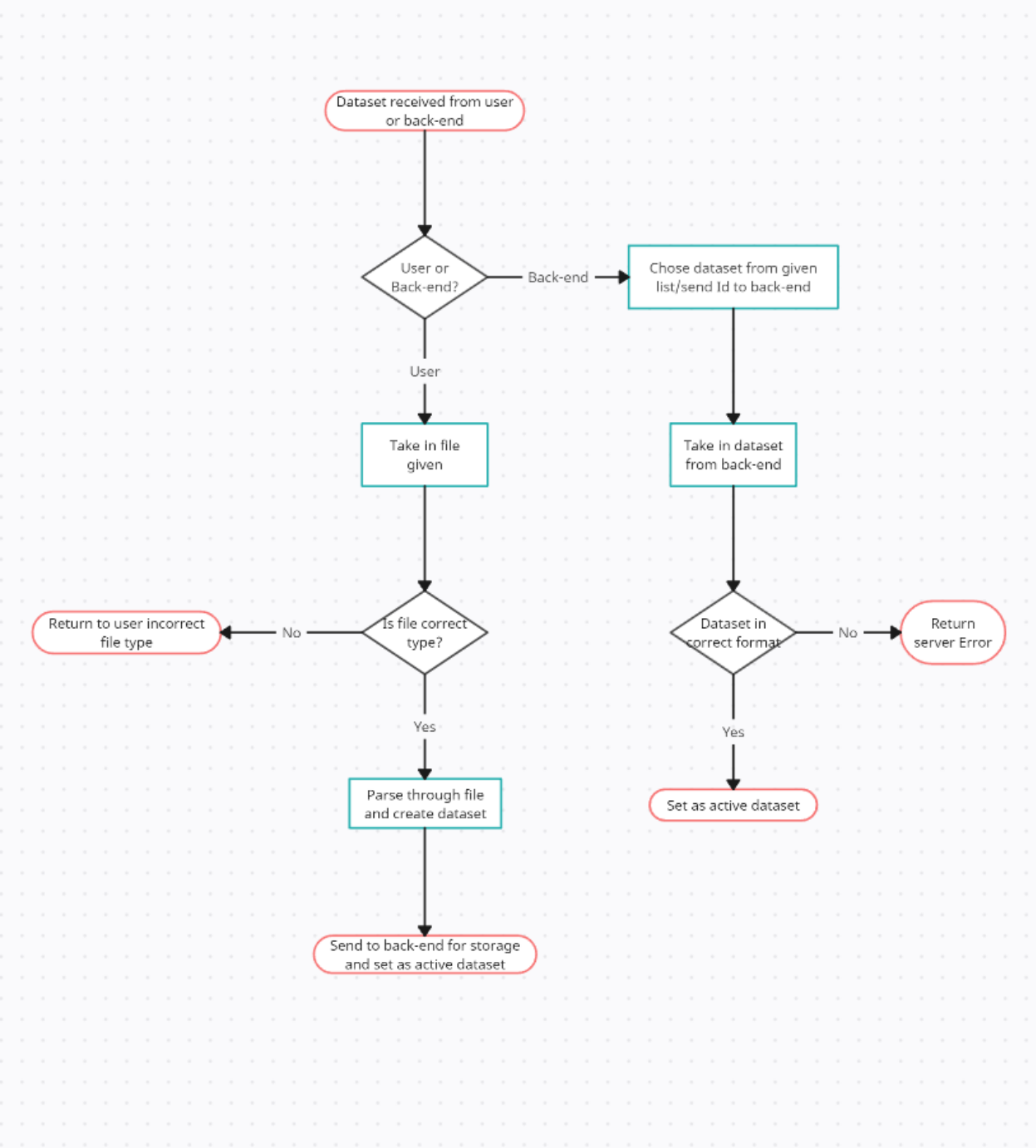

Figure 13 - Frontend file and dataset handling

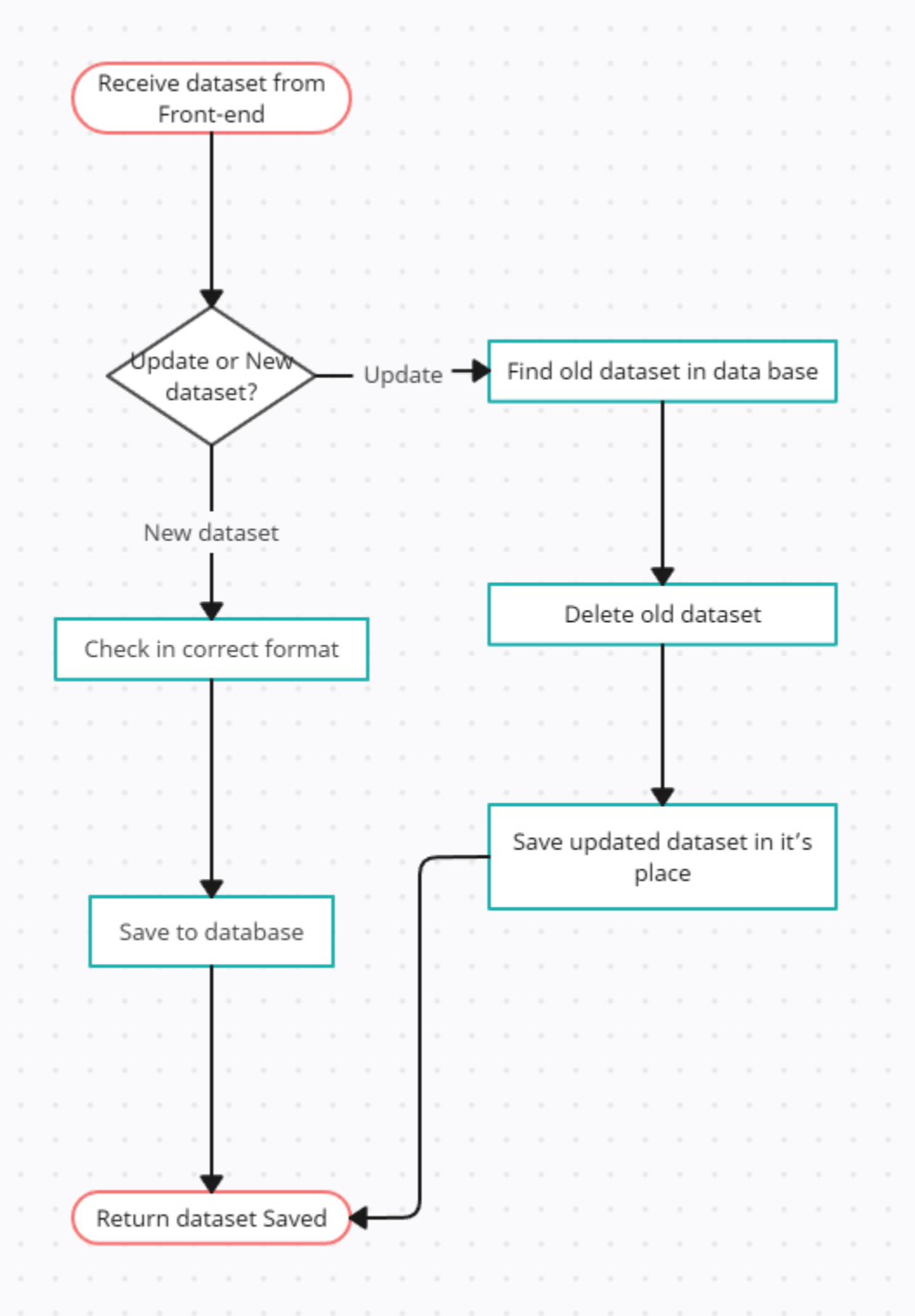

Figure 14 - Saving and updating datasets

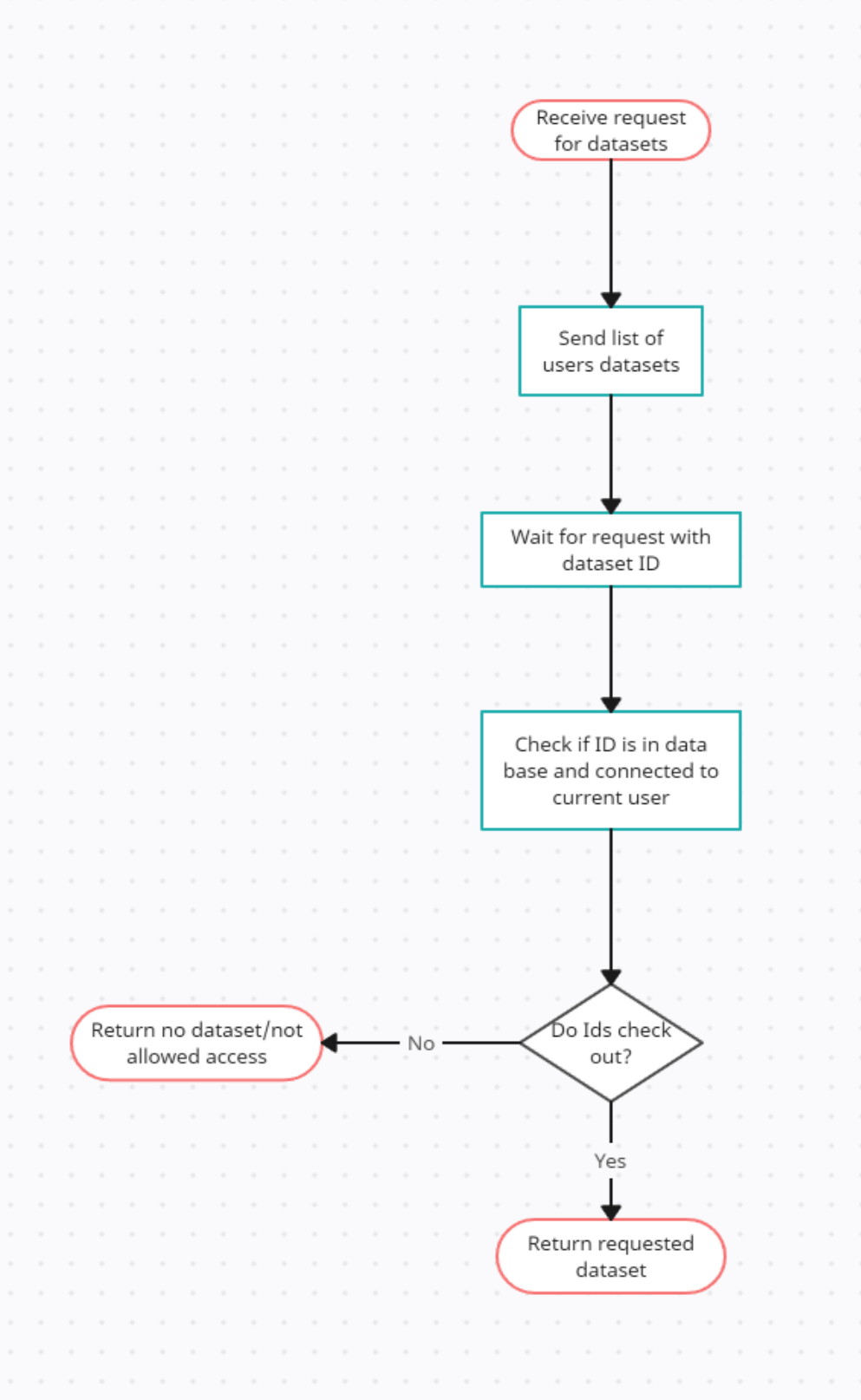

Figure 15 - User requests dataset from database

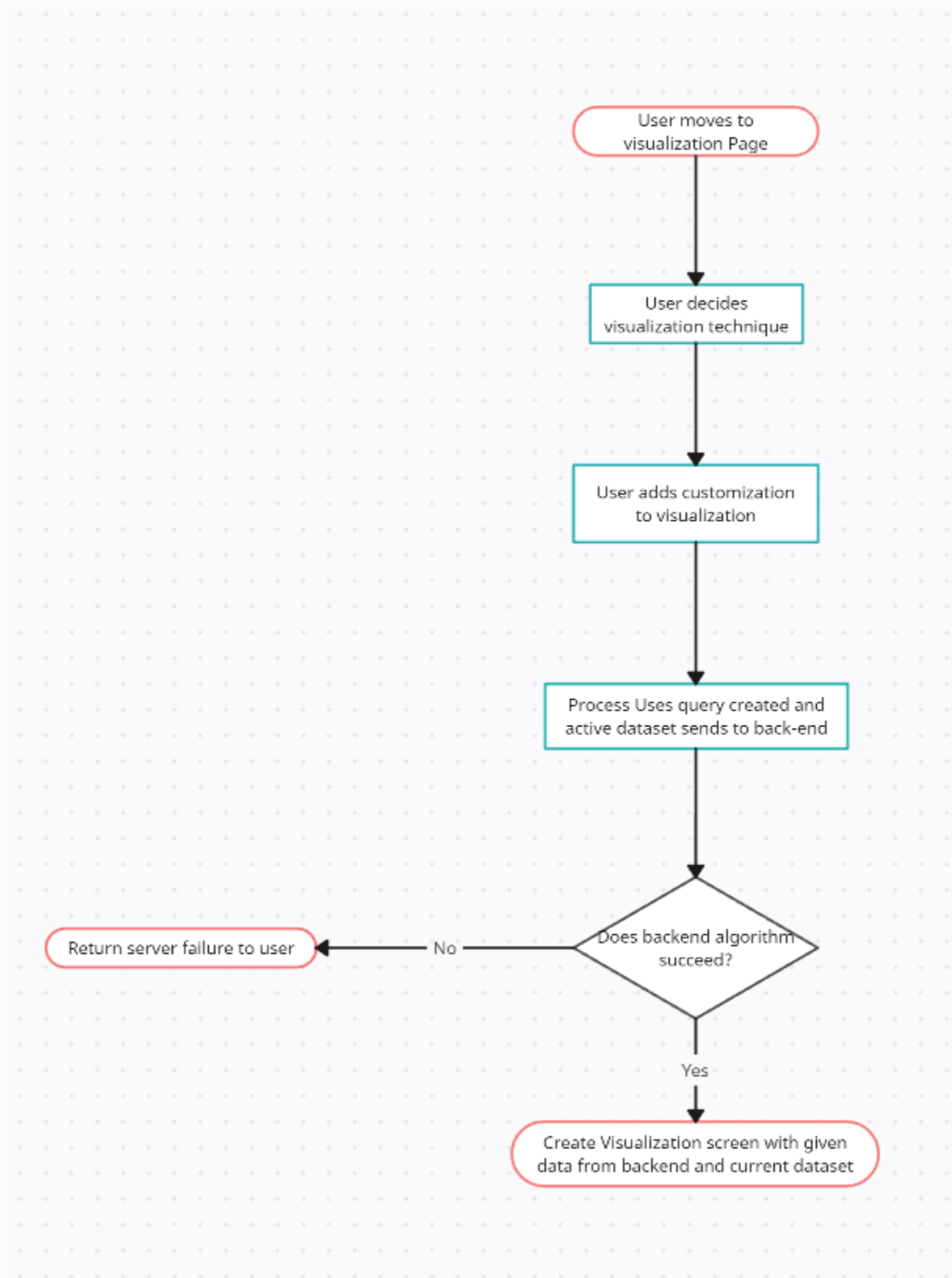

Figure 16 - Frontend Visualization

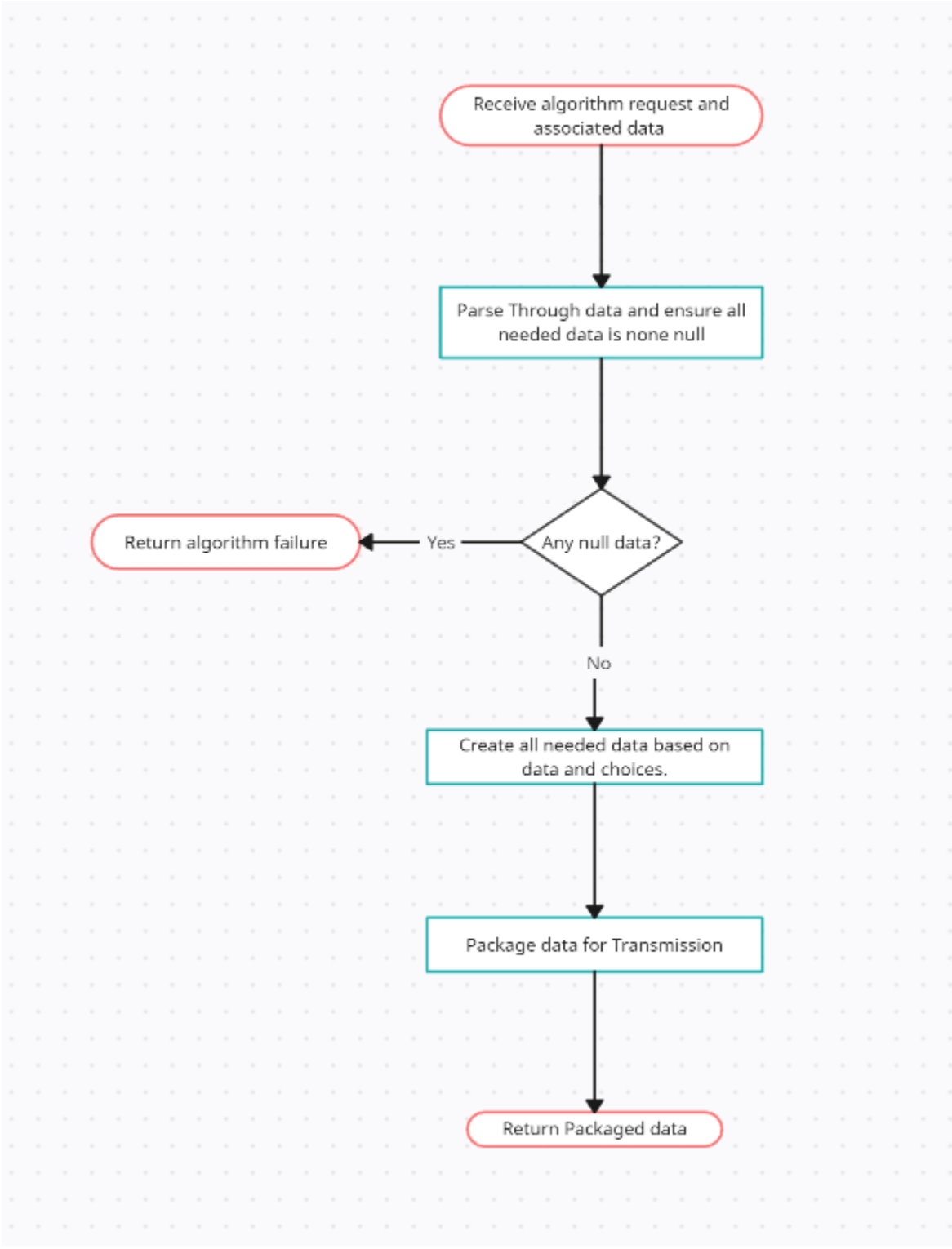

Figure 17 - Backend Algorithm Request

9. Team

9.1 Team members

- Eric Jorgensen
- Mara Prochaska
- Nathan Thoms
- Ryan Cook

9.2 Required Skill Sets for Our Project

- Java
- Javascript
- Intellij use experience
- Vue.js experience
- Database interaction experience
- Http request experience

9.3 Skill Sets Covered by Our Team

Ryan:
- Java experience
- Intellij experience
- Http request experience
- Database interaction experience
- Basic javascript experience

Nathan:
- CSS, HTML, Javascript experience
- Embedded system design
- Industrial automation systems

Mara:
- Major: Computer Engineering
- Systems and Safety Verification Internship Experience
- Other skills: Technical writing and documentation, C/Java/Python/Springboot

Eric:
- Majors: Electrical Engineering/Mechanical Engineering
- Co-Op: Controls, instrumentation, Power distribution and manufacturing design experience
- Ladder logic, Autocad, Inventor, Solidworks, AB Software, Karel, and some C experience

9.4 Project Management Style Adopted by our Team

We have chosen Agile for our project management

9.5 Initial Project Management Roles

- Nathan Thoms - Team Lead & Frontend Developer
- Mara Prochaska - Backend Developer
- Eric Jorgensen - Documentation
- Ryan Cook - Fullstack Developer

9.6 Team Contract

Team Members:

1) __Nathan Thoms_____ 2) __Mara Prochaska_____
3) __Eric Jorgensen_____ 4) __Ryan Cook_____

Team Procedures

1. Day, time, and location (face-to-face or virtual) for regular team meetings:

   We plan to meet regularly on a weekly basis with our project advisor, Dr. Goce Trajcevski on Tuesdays at 4:30 PM in 347 Durham face-to-face. We will also have team meetings on Friday in order to prepare weekly assignments and discuss our next steps. We may also schedule additional online or extra meetings as/if necessary.

2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):

   Our preferred method of communication will be mostly text in addition to weekly face-to-face meetings. We will communicate with our project advisor via email. We will also plan to attend class and discuss our progress in person there as well.

3. Decision-making policy (e.g., consensus, majority vote):

   We will primarily attempt to come to a consensus on major decisions, but we may implement majority voting if necessary for less important decisions. For large decisions, we will attempt to come to a consensus and take into consideration the advice from Dr. Trajcevski since he is our client.

4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):

   We will create a shared spreadsheet keeping track of weekly meeting discussions and other project milestones. We will each be responsible for tracking our individual weekly time contributions which will be documented in our weekly reports.

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:

   All team members are expected to attend all meetings on time unless they have a conflict that they have previously shared. If a team member cannot attend a meeting, they should try to let other team members know as soon as possible. All team members should participate in meetings and have the ability to share their ideas.

2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:

   The team should stay on top of the workload, whether that is organizing and attending meetings or learning required content to implement and design deliverables. Individuals on the team should be responsible for completing assigned portions by deadlines, and the team should meet all team deadlines.

3. Expected level of communication with other team members:

   All team members are expected to respond to messages to the team either via text message or email within 24 hours, excluding special circumstances. If communication lapses become an issue, we will openly discuss the issue and identify the cause and resolution.

4. Expected level of commitment to team decisions and tasks:

   All team members are expected to commit the necessary time it takes to accomplish the tasks that they have taken on. If they need additional time, they should inform the team beforehand. The main point is to communicate how much time is available and be reasonable about what can be accomplished in a week.

Leadership

1. Leadership roles for each team member will be the following, although they may be updated once more information about the project is known:

   Team Organization - Eric Jorgensen
   - Responsible for organizing team meetings
   - Ensures documents are submitted by deadlines
   - Identifies communication issues and assists resolution
   - Coordinate between developers of separate areas

   Client Interaction - Nathan Thoms
   - Send emails and communicate with advisor
   - Document needs and desires of the client
   - Lead decisions for tools and design choices

   Application Design - Ryan Cook
   - Responsible for primary app design
   - Will implement tools, delegating technical tasks to other team members

- Lead design for User Interface

Testing - Mara Prochaska

- Identify scope of testing and type of testing to be performed
- Lead documentation of test cases for application
- Create visual and gather data to demonstrate thorough testing

2. Strategies for supporting and guiding the work of all team members:

   Not only limiting individual team members to their specific leadership areas, but utilizing team-wide input with ideas and suggestions in the different areas of the project.

3. Strategies for recognizing the contributions of all team members:

   Have a brief "stand-up" section of team meetings where team members can briefly discuss what they have worked on and accomplished in the previous week.

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the team.

Eric:

- Major/special focus(es): Electrical Engineering & Mechanical Engineering- Applied robotics, Controls, Digital Electronics, Semiconductors.
- Internship/Co-Op Experiences: Bridgestone- Manufacturing Controls/Power dist./Mechatronics Co-Op.
- Other skills: Ladder logic, AutoCAD/AutoCAD Elec., Karel (Fanuc), VFDs, PLCs, AB/Rockwell software/hardware, 3 lines/1 lines, Elec. Panel design, HMI design, Switchgears/I-Lines, LV & MV Xfrms.

Nathan:

- Major/special focus(es): Electrical Engineering - Signal Processing, Control Systems & Embedded System Design.
- Internship/Co-Op Experiences:
- Employed - Custom-Pak: Designed web-based HMI for interfacing with 6-axis industrial robots. Research and development of machine learning based technologies for anomaly or defect detection.
- Other skills: Programming experience - Javascript, Python, C, C++. Web-Based Application Design. Familiar with several ML algorithms.

Mara:

- Major/special focus(es): Computer Engineering
- Internship/Co-Op Experiences: Northrop Grumman, Systems and Safety Verification
- Other skills: Technical writing and documentation, C/Java/Python/Springboot

Ryan:

- Major/special focus(es): Computer Engineering
- Internship/Co-Op Experiences: Collins Aerospace - Systems Engineering

- Other skills: Technical Documentation through internship, C/Java/Springboot/MySQL

2. Strategies for encouraging and supporting contributions and ideas from all team members:

   Being open to ideas from other team members and open to different points of view from those with differing levels of experience and exposure to technical areas.

3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)

   In order to identify and resolve collaboration or inclusion issues, team members should try to maintain open communication and try to speak up for themselves about issues they have seen. All team members should try to be respectful of others and uplift other team members that are being excluded. If a team member does not feel comfortable with speaking up in a group setting, they should try to discuss on a more individual basis with other team members, faculty mentors, and/or professors.

Goal-Setting, Planning, and Execution

1. Team goals for this semester:

   Our main goals are to understand the goals of this project, document a design plan, and learn more about the algorithms that will be used to power the application. We would also like to learn more about new technologies through hands-on experience with an actual project.

2. Strategies for planning and assigning individual and team work:

   Our strategy to plan and assign individual and team work will be to first create a plan for what tasks should be completed and also create an estimated timeline for how long each task may take. We can then divide tasks up depending on available time and skills that each team member can contribute.

3. Strategies for keeping on task:

   Adhere to the weekly timeline provided by the faculty member. As well as create a finer detailed timeline for sub weekly goals and deadlines. Communicate changes or obstacles as they arise.

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?

   Communicate with the person directly, then if the problem persists, bring it to the attention of the rest of the team.

2. What will your team do if the infractions continue?

Bring it to the attention of the professor if all else fails, as well as attempt to understand the reason(s) why the team member is failing to meet the obligations stated in the team contract.

****

a) *I participated in formulating the standards, roles, and procedures as stated in this contract.*
b) *I understand that I am obligated to abide by these terms and conditions.*
c) *I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.*

1) __Nathan Thoms_____ DATE _1/29/2024_____
2) _____Eric Jorgensen_____ DATE 1/29/24
3) _____Mara Prochaska_____ DATE _1/29/24_____
4) _____Ryan Cook_____ DATE _1/29/24_____