Detecting convoys of molecular and atomic trajectories

DESIGN DOCUMENT

33
Iowa State University
Goce Trajcevski
Claira - Client Interaction, Project Management, Documentation
Jeremy - Individual component design
Daryl - Testing
Jason - Overall design
sdmay23-33@iastate.edu
http://sdmay23-33.sd.ece.iastate.edu/

Revised: 12/2/2022/ Version 1

Executive Summary

Development Standards & Practices Used

List all standard circuit, hardware, and software practices used in this project. List all the Engineering standards that apply to this project that were considered.

- IEEE/EIA 12207
- IEEE/ISO/IEC 90003-2018
- ISO/IEC 27001
- V. Phoha, "A standard for software documentation," in Computer, vol. 30, no. 10, pp. 97-98, Oct. 1997, doi: 10.1109/2.625327.

Summary of Requirements

List all requirements as bullet points in brief.

- The system should be able to graphically display results from user input
- The system should scale with multiple users
- All computations must take place on the backend of the system
- The system should only allow valid filters as inputs
- The user should be able to save their output to view later
- The system UI should be intuitive and user-friendly

Applicable Courses from Iowa State University Curriculum

List all Iowa State University courses whose contents were applicable to your project.

- Coms 309
- Coms 311
- Coms 319
- Coms 339
- Coms 317
- Coms 329
- Coms 327

New Skills/Knowledge acquired that was not taught in courses

List all new skills/knowledge that your team acquired which was not part of your Iowa State curriculum in order to complete this project.

- Use of React.js as a frontend framework
- Use of C#/.Net in the backend
- Use of VMD as a graphical display

Table of Contents

List of figures/tables/symbols/definitions

Figures

Tables

1 Team

1.1 Team Members

Jason Guo

Daryl Kay

Claira Springer

Jeremy Lewis

1.2 Required Skill Sets for Your Project

Skills required:

- Web Development
- Testing
- Documentation
- Web Server
- Database

1.3 Skill Sets covered by the Team

Documentation - Claira, Daryl

Web Development - All Members

Testing - All Member

Web Server - Daryl

Database - All Members

1.4 Project Management Style Adopted by the team

For our project we went with the agile development life cycle due to the fact that it works very well in an all software based project. This will allow us to divide our project into sprints with specific purposes and will also allow us to incorporate feedback from the client very easily.

1.5 Initial Project Management Roles

Claira- Administration, client interaction, project management, documentation

Jeremy- Individual component design

Daryl- Testing

Jason- Overall design

2 Introduction

2.1 Problem Statement

Chemists are wanting to detect convoys of atoms and find interesting interactions between simulated atoms. Interesting interactions are defined by different types of atoms interacting with each other. Current methods include looking through data by hand and using algorithms with interpretations, which is borderline impossible with the amount of data that is created in simulations. We will solve this issue by building a graphical interface and allowing a user to quickly go to interesting interactions that they define. This would make it easier and quicker for chemists to complete their work.

Specifically, one of the pre-conditions for something "interesting" to happen is that atoms from different molecules come close to each other and stay close to each other for a period of time. This, in turn, makes the structure of the so-called convoys of trajectories one that could help focus the search and make it more efficient. What we will have as the main goal is to provide a UI that will enable chemists to define the parameters of such convoys and have a visual tool to inspect them subsequently. A notable difference from the "plain" trajectories of moving point objects is that when it comes to molecules, there are physics/chemistry laws that infuse a degree of the "semantics" (e.g., a bond can be formed only if one of the atoms is hydrogen) – which we will also use to help with the more focused search.

2.2 Intended Users and Uses

User 1: Chemists

Characteristics: Intelligent, Stressed, Busy, Detail-Oriented

Needs: Needs a way to easily filter through data and get easy to understand and visualize results, reduce costs to further extend the funding that they receive, expedite results to secure more future funding

Benefits: Saves time and work for the chemists as they will have a program to do these tasks for them, reduce costs in materials, speeds up the drug development process

User 2: Experts from other fields, collaborating with chemists

Characteristics: Less Knowledgeable about the specific chemistry, more experts in data analytics.

Needs: This user needs a way to have a better understanding of the results of simulations without having to understand the data specifically

Benefits: Will have the ability to look at visualizations of the specific data that is interesting and allows for easier communication between the chemists and them.

User 3: Managers in the pharmaceutical industry/business decision-makers

Needs: They need a high-level interactive tool to be able to understand the potential benefits of turning a simulation into actual experiments.

Benefits: Will be able to minimize the risk for investments into experimental setup(s) which could lead to eventual production.

2.3 Requirements & Constraints

Functional Requirements

- Scale well with multiple users
  - Multiple users should be able to access the system at one time and request individual queries
  - Users should only have access to their specific results and not those of other users using the system
  - Users should be able to get a notification when their request has been processed and completed
- Hard computations on the server
  - No computations should be occurring on the front end, all computations are to occur on the backend
  - Front end should just be to send parameters to the backend and also display outputs of the queries
- The system should not crash with certain input parameters (too large)
  - Frontend should have input sanitization to make sure that values imputed are in the proper ranges
  - Backend can contribute to this as well with checking for variables being outside of acceptable ranges

Non-functional requirement

- Intuitive UI
  - The UI should be very user friendly and require little to no assistance for the users to be able to perform their desired tasks on the system
- When bulk inserting new data, the downtime for users should be minimal
  - New data should be queued to be downloaded/uploaded during off hours which can be set by the users but will most likely be during the night
- Conversion from one data to another should work with .mol2, .dcd

Constraint

- The overall implementation should not exceed 300 dollars
- Server to constantly run and enter information
- The user should be notified within 10 seconds when his computation is complete

2.4 Engineering Standards

IEEE/EIA 12207

Covers the common framework for the software development life cycle

IEEE/ISO/IEC 90003-2018

An extension of ISO 9001:2015 which is also includes the software development cycle but also other factors that will boost customer satisfaction

ISO/IEC 27001

Standard dealing with information security and practices that should be followed

V. Phoha, "A standard for software documentation," in Computer, vol. 30, no. 10, pp. 97-98, Oct. 1997, doi: 10.1109/2.625327.

Based on ANSI/ANS 10.3-1995 a standard used for the documentation of computer software and development

3 Project Plan

3.1 Project Management/Tracking Procedures

We plan to use the agile project management style due to the fact that our project is entirely software-based. We decided that the use of sprints and the development of sections at a time would be the best way to manage our project efficiently and effectively. This will also allow us to easily relay our progress to the client as we get through different stages of development which can be beneficial for feedback related to various components since we will have time to make changes if needed with agile.

As a team, we plan to use Git to track our progress during this project. Using Git will easily allow us to track contributions and progress throughout the duration of the project through user commits and user stories. It will also allow us to set up milestones to clearly divide up what tasks need to be done, which ones have been completed and the time frames they should be completed.

3.2 Task Decomposition

Tasks divided by sprints:

- Section 1
  - Setup a frontend with some UI functionality
  - Get simulation data from the chemist on the backend
  - Start looking at how to read the data and store it
- Section 2
  - Integration with the graphing library on the frontend
  - Get familiar with graphing objects
  - Refine reading objects and start implementing algorithms provided by the chemists
- Section 3
  - Ability to graph potential data values
  - Ability to input queries and send them to the backend
  - Ability to receive queries and send data back to the frontend
  - Ability to manage queries in a queue to allow for more users and potentially also start working on more data inputs
- Section 4
  - Frontend can graph interesting events based on filters provided by the user
  - Frontend is specific to that user or session
  - User can store results for easy access
  - Backend has full functionality
- Section 5
  - Testing frontend
  - Testing backend

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Milestones include:

- Sprint 1/2
  - Having a working graphing library integrated into the frontend that is able to graph spherical objects and have a load time of less than a second
  - Uploading and ability to read/process data from the simulation asynchronously
- Sprint 3
  - Frontend is able to send queries 95% of the time with no issues
  - Able to graph potential data in less than 2 seconds
  - The backend receives and processes queries 95% of the time with no issues
  - The backend has implemented algorithms that run without any issues
- Sprint 4
  - Frontend can read responses from backend and graph results of queries in less than 2 seconds 99% of the time with no issues
  - The backend receives and processes queries 99% of the time with no issues
  - Backend manages queues of queries with no issues
- Sprint 5
  - 90% test coverage on front and back end

3.4 Project Timeline/Schedule

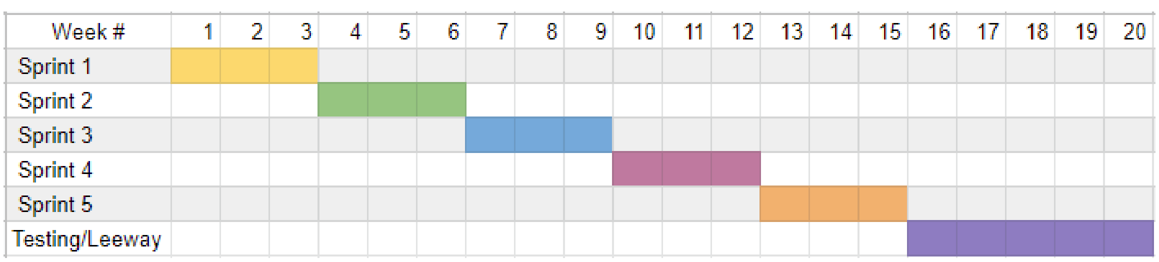

Figure 1: Gantt chart of project schedule

3.5 Risks And Risk Management/Mitigation

Table 1. Risk identification and mitigation plan.

Risk	**Probability**	**Explained/Mitigation**
Front end library fails to function as intended	0.3	We have identified several alternatives to the primary graphing library that can be used in the case that our primary one fails.
Corrupted data	0.1	With the amount of data we are processing, there is potential that data is corrupted
Server unable to handle/is slow in processing the computations required	0.2	Research will be put into what servers to invest in
Issues connecting frontend to backend	0.1	
Testing for accuracy of the system is difficult	0.2	Potentially with the type of problem we are solving, testing could be difficult
The data set is too large to graph effectively	0.5	Mitigation: If the dataset is too large to graph, we will split the data and graph it in sections. This isn't a huge issue because they don't need to see all of the data at one time.

3.6 Personnel Effort Requirements

Table 2. Estimate of effort requirements per task.

Task	**Hours (Total)**	**Explanation**
Working library to graph objects	20	Should not take long to find a library to work with but will take time to familiarize with various libraries and any new IDE's/editors
Upload and read/process data asynchronously	30	Time to understand the format of data and to process it efficiently as well as set up the relevant frameworks for ease of use further in the project timeline
Mostly working queries (Frontend)	20	Have a somewhat working frontend able to send queries to the backend
Mostly working queries (Backend)	30	Receive queries from the front end and process them as well as send the appropriate data for displaying
Backend implemented algorithms	40	Process the data according to specifications and detect the correct parameters
Graph quickly (<2sec)	30	Use data and the correct library to streamline data processing and formatting it to display efficiently
Increase reliability of queries and algorithms (>99%)	20	Find more efficient ways to increase the speed of algorithms and decrease runtime
Majority test coverage (>90%)	40	Set up automated testing for the majority of our code to display functionality and verify that it is working

3.7 Other Resource Requirements

We will need virtual or physical access to the chemist system that stores the simulation data in order to move it over to the backend system our project will be using so that the system will be able to access the data it needs to respond to the queries from users. We would also need the ability to access the system or just the ability to get sample data from the client for testing purposes.

4 Design

4.1 Design Context

4.1.1 Broader Context

Our design problem is going to be used in the realm of chemistry, more specifically to better understand molecular convoys. This includes chemists who will be able to use our software, as well as people who don't work in chemistry but need to use our software to complete their job that is associated with chemistry. Potentially, people who don't understand chemistry at all will need to use our software.

Table 3. Analysis of context considerations for broad areas

Area	Description	Examples
Public health, safety, and welfare	How does your project affect the general well-being of various stakeholder groups? These groups may be direct users or may be indirectly affected (e.g., solution is implemented in their communities)	Our project will have a positive effect on public health as automating part of the research process can allow chemists to develop drugs faster which can help people quicker
Global, cultural, and social	How well does your project reflect the values, practices, and aims of the cultural groups it affects? Groups may include but are not limited to specific communities, nations, professions, workplaces, and ethnic cultures.	Our project will be beneficial to the culture of chemists and researchers in the sense that it will enable a more effective exchange of findings and processes
Environmental	What environmental impact might your project have? This can include indirect effects, such as deforestation or unsustainable practices related to materials manufacture or procurement.	Due to the use of servers our project constantly uses energy, however, our system will enable more efficient separation of potential reactions among atoms without sacrificing the effectiveness
Economic	What economic impact might your project have? This can include the financial viability of your product within your team or company, cost to consumers, or broader economic effects on communities, markets, nations, and other groups.	Potential for faster drug research means that drugs could be on the market quicker which could mean more money for that industry but could also mean more competition in that market

4.1.2 Prior Work/Solutions

There has been some prior work done in the realm of studying molecular convoys. Primarily, simulations have been used but then studying the simulation data is very manual. There are also products that visualize the outcome of the simulations, however, there hasn't been work to then visualize the convoys occurring. On top of the previously stated methods, there has been some exploration of how machine learning can be used to see the outcomes of chemical reactions.

4.1.3 Technical Complexity

1. Frontend
    a. Graph visualization
        i. Use of a third-party library/framework to visualize data that is given to us by chemists
    b. Scaling with users
        i. Ensure that all users are able to access and view data assigned to them
    c. Scaling with data
        i. Ensure that data of any size is able to be handled and displayed
2. Backend
    a. Convoy algorithms
        i. Have algorithms that detect and mark molecular convoys correctly according to specifications given
    b. Upload data
        i. Be able to upload data to process and use
    c. Execute queries
        i. Execute queries from the front end and send data requested in a timely manner
    d. Queues
        i. Queue query requests from the front end
3. Chemist data
    a. Individual motion of atoms vs collective molecules
        i. Most of this information will be given to us in a file and it will be our job to process this information and sort it into various categories for those using the system to look into

4.2 Design Exploration

4.2.1 Design Decisions

One design choice that will play a major role on our front end is which graphing library we wanted to choose to integrate into the system. With that being said we came up with many different options as listed below:

- Three.Js
- D3
- Aframe
- Babylon Js
- Zdog
- Cannon Js
- lightgl.js
- VMD

These options will be discussed in further detail below.

Secondly, another choice was what framework we would use on the front end of our system. We managed to narrow our choices down to either using the React framework or Angular and ended up going with React since our project won't really need more than one dynamic page, it has good ease of use and is also good for rapid development. React also handles variable data very well.

Our last two design choices were to use a MySQL database due to the potential to scale with additional features in the future and to potentially parse through data faster rather than straight from simulation data. We also decided to use queues for queries from the front end to allow the system to not lock up if two users submit a request in succession.

4.2.2 Ideation

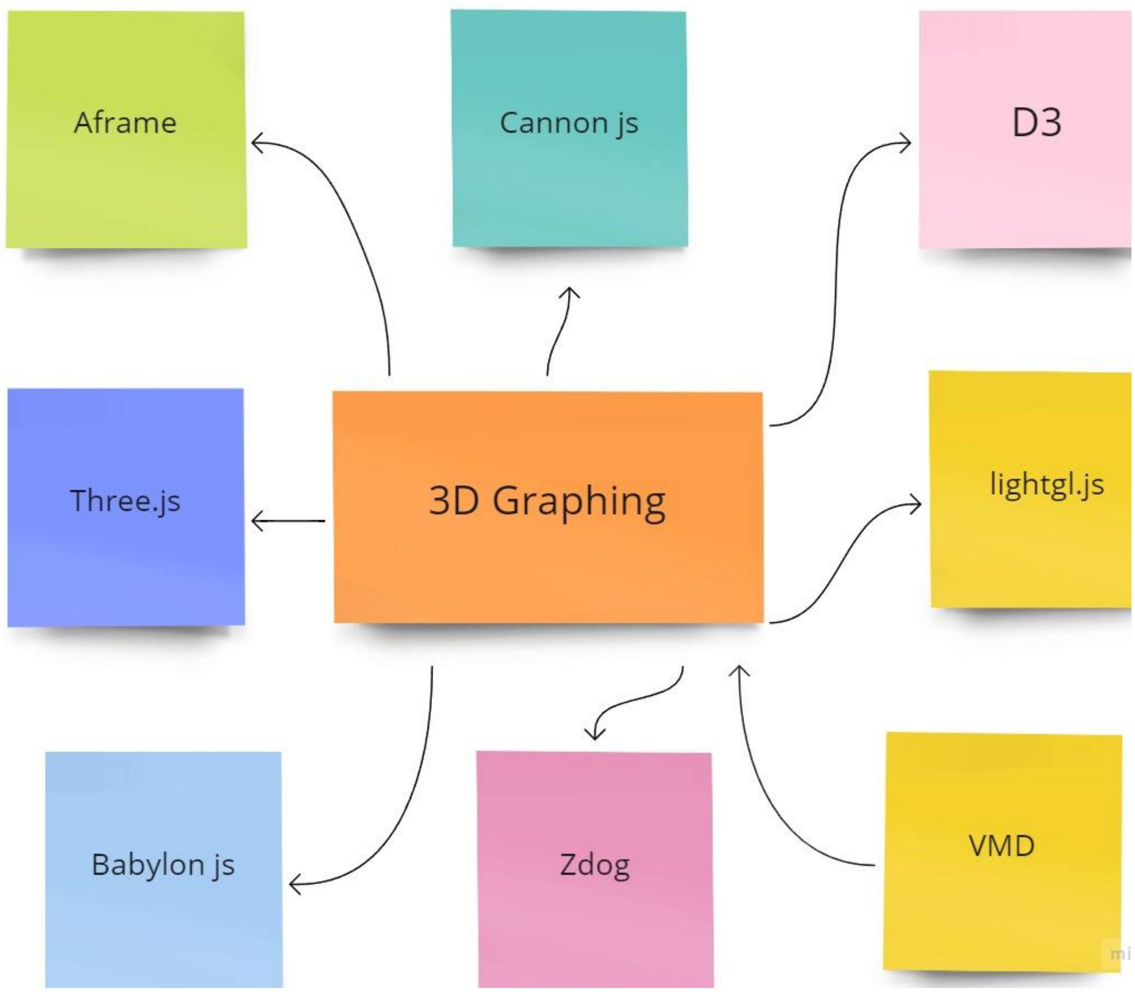

Figure 2: Lotus blossom ideation chart

When it came to looking at different options for graphing libraries that we could use for our project we wanted to look for well-known ones that are based specifically on 3D graphing. With that being said we were able to throw out Zdog and D3 since both mainly dealt with graphing two-dimensional objects rather than three-dimensional ones.

4.2.3 Decision-Making and Trade-Off

Table 4. Weighted decision matrix for tool selection.

Select Criteria	Weight	Three.js		Aframe		babylon js		canon js		VMD		lightgl.js	
		Score	Total	Score	Total	Score	Total	Score	Total	Score	Total	Score	Total
Clean looking 3D visualization	0.3	5	1.5	5	1.5	5	1.5	3	0.9	5	1.5	2	0.6
Able to move through data	0.3	5	1.5	5	1.5	5	1.5	3	0.9	5	1.5	2	0.6
Fast load time	0.2	5	1.0	4	0.8	4	0.8	4	0.8	5	1.0	4	0.8
Documentation	0.2	5	1.0	5	1.0	5	1.0	5	1.0	5	1.0	2	0.4
Total	1	5		4.8		4.8		3.6		5		2.4	

4.3 Proposed Design

4.3.1 Overview

Our design problem is going to be used in the realm of chemistry, more specifically to better understand molecular convoys. This includes chemists who will be able to use our software, as well as people who don't work in chemistry but need to use our software to complete their job that is associated with chemistry. Essentially users will be able to add filters on the front end for which kinds of events they would like to find in simulations of molecules and be able to visualize the result sent from the backend.

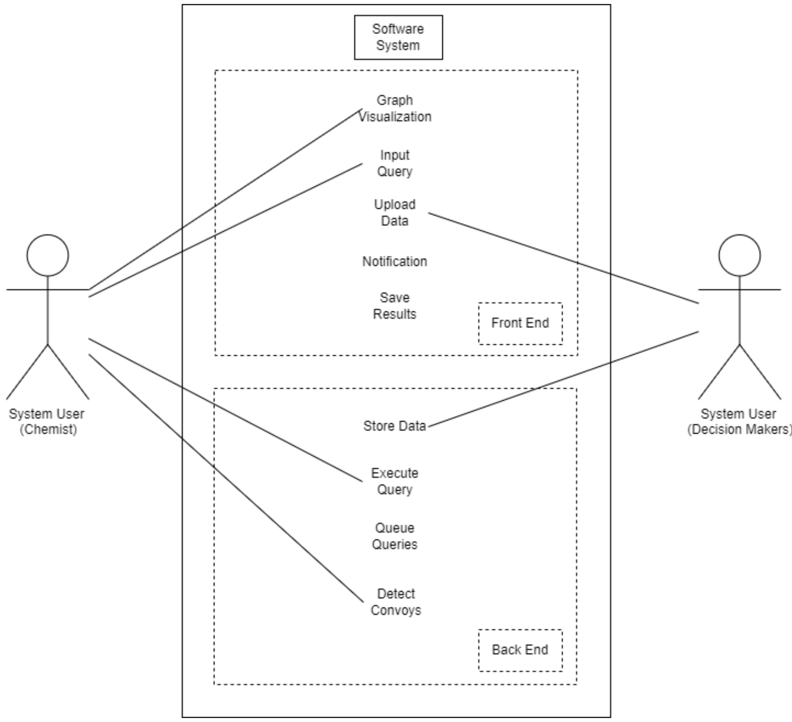

Figure 3: Use case diagram

4.3.2 Detailed Design and Visual(s)

Provide a detailed, technical description of your design, aided by visualizations. This description should be understandable to peer engineers. In other words, it should be clearly written and sufficiently detailed such that another senior design team can look through it and implement it.

The description should include a high-level overview written for peer engineers. This should list all sub-systems or components, their role in the whole system, and how they will be integrated or interconnected. A visual should accompany this description. Typically, a detailed block diagram will suffice, but other visual forms can be acceptable.

The description should also include more specific descriptions of subsystems and components (e.g., their internal operations). Once again, a good rule of thumb is: could another engineer with similar expertise build the component/sub-system based on your description? Use visualizations to support your descriptions. Different visual types may be relevant to different types of projects, components, or subsystems. You may include, but are not limited to: block diagrams, circuit diagrams, sketches/pictures of physical components and their operation, wireframes, etc.

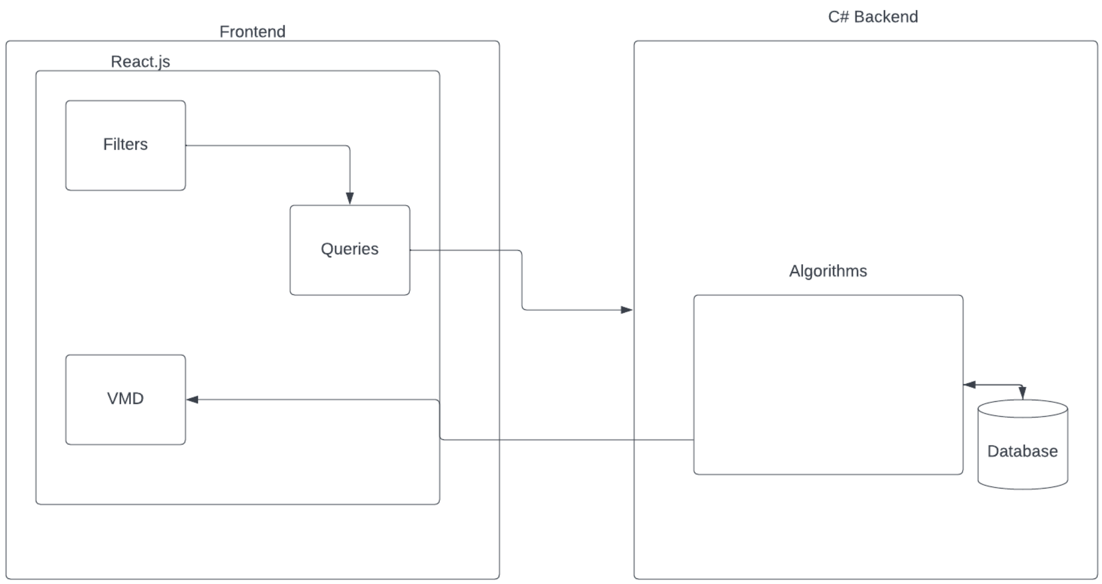

Figure 4: Domain diagram.

Filters/Queries - We consider this part the first step in the process. During this step, the user is inputting data that they want to see back in the VMD module. These two modules are part of our front end in react.

Algorithms - This part of our design is where the CPU-intensive algorithms are run. We find out what queries we need to run from the filters/queries module.

VMD - This part of our design will handle the graphical display of the molecules that we get back from the backend by taking the file data that we get and passing it into the graphing software. This software will have to be integrated into our front-end part of react in order for it to be rendered and still allow the user to send new queries

Database - This section will store our information from the simulations in order to be able to be used by the algorithms

4.3.3 Functionality

Describe how your design is intended to operate in its user and/or real-world context. What would a user do? How would the device/system/etc. respond? This description can be supplemented by a visual, such as a timeline, storyboard, or sketch.

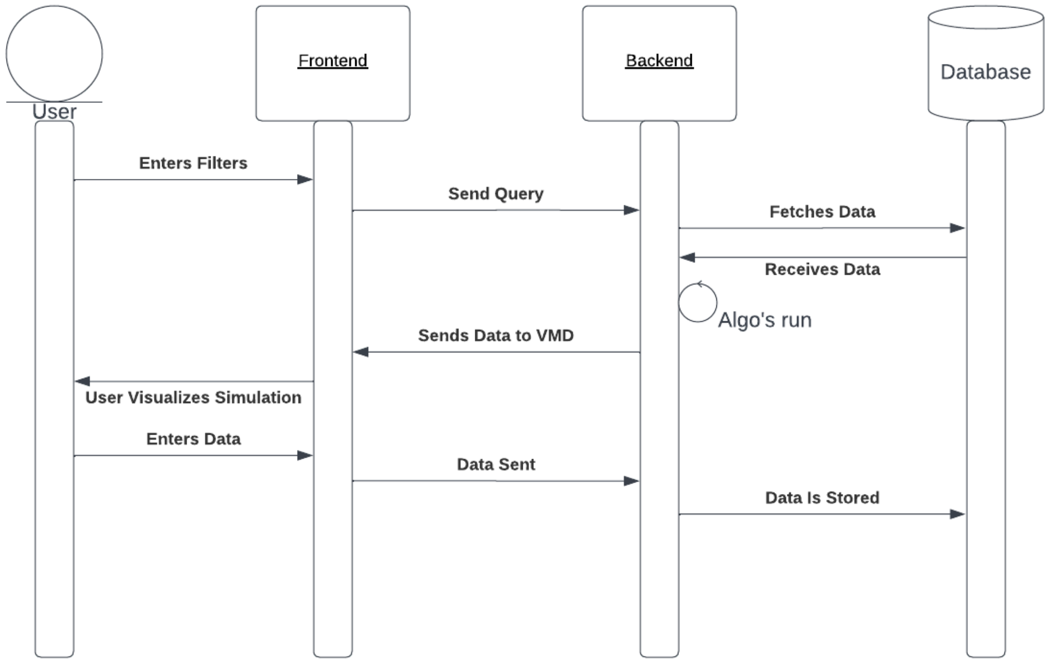

Figure 5: Sequence diagram.

We now discuss how the system discussed in 4.3.2 will actually work when interacting with the users. Envisioned structure of the communication between our components is listed below. The user interface will consist of prompts for filters that are then realized by the VMD graphing software which are then accessible to the user after the query is processed. The user can then save the responses that they got out of the filters for later access.

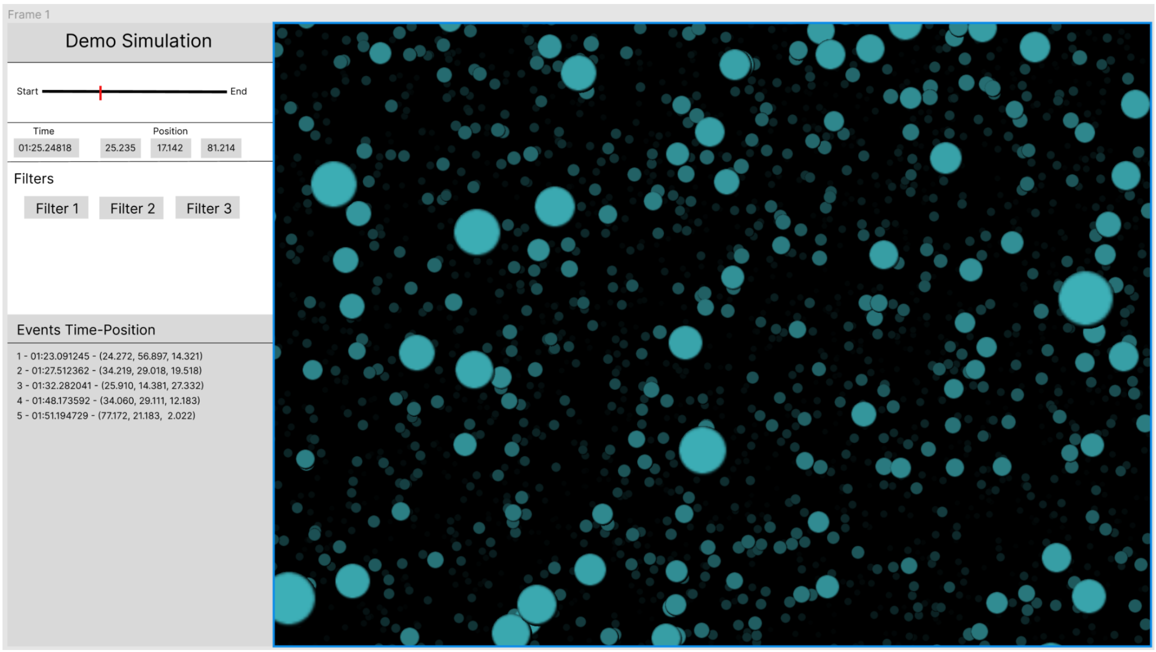

Figure 6: Theoretical UI

Figure 6 displays a potential design for the website after a user has received the results of their query. The user will be able to interact with the graphing software shown on the right and be able to zoom in and out and move in the space. The user would also be able to move by selecting their x, y, z, or time in the simulation. They can also view the events that were found using the event time-position section.

4.3.4 Areas of Concern and Development

Our designed solution should satisfy the needs and wants that our clients have asked for and we have gone back and forth with them to get feedback on our design choices to make sure that it fits the vision that they have for this system. When it comes to concerns we don't really have any major ones as most of our questions are answered by the client but we do have one that we have a plan to manage, that being our unfamiliarity with some of the libraries and technologies that we will be using. To manage this we plan to split up the load of learning these technologies, these will be listed in section 4.4. Splitting up the learning will help us each focus on one thing and effectively learn different parts of the system which can help speed up development with one master of the technology. Another minor issue is developing test cases or scenarios to verify the interaction between sub-components (i.e., integration testing).

4.4 Technology Considerations

The technologies that we plan on using in this design are as follows: React, MySql, C#, and VMD. When it comes to using react the main strength is its pace of development is very fast and easy to get started on which is something that we want for a smaller-scale frontend and the disadvantages are basically non-existent for the scope of this project but an alternative could be Angular if issues arise. The next one is MySql which will be used as a database and this has the advantage of being a very good relational database to store our results of the simulation or queries, the only disadvantage is that it might not be needed for the current scope of this project but it can help if any additional features are needed in the future. As for the graphing technology we are using VMD which will be advantageous as it is meant for displaying molecules and has a lot of those features added in and doesn't seem to have any noticeable drawbacks when compared to other graphing solutions. Alternatively, we could use threes.js or babylon.js as these can do just as well but lack the features added in initially. Our last technology is C#, which will be used on the backend of our system to manage the heavy lifting of the convoy algorithms and return the data that will be displayed on the front end. C# will be easy to deploy and a fast backend for our system and our alternative would be using Node.js instead if after starting it seems like a better fit.

4.5 Design Analysis

Nothing as of this moment has been completed for this project and is just in the planning and designing phase. With that being said our plans for our future design are to just design everything we can to the best of our ability and implement all of the features we are planning on using in the system to the client's specifications. We also haven't experienced any issues since we haven't begun any work on development.

5 Testing

In this section we will discuss our plan for conducting and implementing various testing procedures to validate the correctness of the project. We would like to note that the requirements for the project, both functional and non-functional, were already translated into architectural components discussed in Section 3 of this document. We will refer to corresponding figures from that section here.

Given that the project involves large and simulation based datasets, there are certain distinct characteristics in our testing that are noteworthy. Primarily, we will be testing the execution of algorithms and visualization of the output. We also want to note that the main objective of this project is to develop a proof-of-concept prototype and not a complete commercially available implementation. As such, the cost component is not a major factor in our testing.

5.1 Unit Testing

As shown in figure 3, in section 4.3.2, there are multiple units that will need to be tested but the primary focus of our unit testing will be on the backend. This will include testing all cases for the algorithms with the many variations of parameters that the user will be able to query. To do this we plan to manually write test cases for each algorithm case that we deem necessary such as edge cases, cases with invalid parameters, and realistic parameters, and compare them to the expected output. This will be done with C# unit testing library through .Net. The main goal is just to make sure that everything is functioning as expected on the backend when it comes to computations.

5.2 Interface Testing

When it comes to interface testing, what we plan to do is test each component of the frontend using tools such as JEST and react test libraries since the react framework breaks down every piece of the frontend into separate components that can be individually tested. We will also want to run some tests on our graphical interface portion, this will be testing using JEST just to make sure that we are providing it the right data based on responses from the backend and that the components are rendering properly and being responsive to user inputs when needed. In doing these tests, it will help to test the response handlers from our queries as we are testing that the data is handled properly and put into the graphical display. We will also test that queries are sent out from the frontend and the data is correctly packaged. We also need to make sure that our save feature is working for the responses that we receive but just checking that the file system saved.

5.3 Integration Testing

For our project, we have identified several critical integration paths. Key among those we have identified is sending input querying data from the frontend to the backend where it can be run through the appropriate algorithms, and for the algorithmically generated resulting data to be sent back to the frontend and visualized via VMD.

Once unit testing has been completed on the backend and interface testing completed on the frontend to ensure that each component operates properly on an individual level, integration testing is then performed on the integrated- connected- system. In order to ensure that our major system components- backend and frontend- work as expected when put together, we plan to test our integration paths using the library of C# integration testing tools from .Net. These tests can also be performed concurrently as each component is developed to ensure that the components continue operating properly as they advance.

5.4 System Testing

For the process of system testing, all previous testing tools will again be deployed. At this level of our project, we want to ensure the quality of our product. System testing is done to verify completeness. Unit tests and integration tests- via .Net libraries- as well as JEST and react libraries for interface tests will need to run together to ensure the whole system works properly from end-to-end.

In addition to ensuring that our system is complete and operating in accordance with all functional requirements, we want to optimize our product in terms of robustness and performance. As we don’t anticipate our system to be very large, our hope is to provide quality in our components. This includes testing for non-functional requirements such as speed and efficiency. We plan to test the timing of requests from input to output in order to make sure that they run at an acceptable level. Finally, as our product concludes system testing, we will continue applying the tools we have to verify that the quality of our system remains in proper shape.

5.5 Regression Testing

As alluded to in the previous sections, we plan to continue the process of testing outlined above throughout the entire lifecycle of our project. While development continues we need to ensure that our critical features: querying, data visualization, and front-to-back communication between components do not break. Tests that were performed previously are applied again at every iteration of the project to ensure that new additions to the system do not cause problems with the functionality implemented prior. Utilizing .Net testing frameworks and JEST or react libraries, features are to be tested as they are added, both to verify that they operate as intended, as well as not cause problems when integrated into the previous design.

5.6 Acceptance Testing

Critical to the conclusion of project development is the presentation of the final product to stakeholders and end-users. Acceptance testing will be done with the involvement of our client. By inviting clients to take part in our acceptance testing process we hope to finally guarantee with certainty that our product fulfills the requirements expected of it. Tools used in previous iterations of the project will be available if necessary, in favor of performing varieties of hands-on tests both in the team and with the user, displaying the features requested by them and receiving final feedback from them on any requirements that may yet need to be fulfilled. Once accepted by our client, final touches may be all that are needed before the end product is then provided to the client and deployed.

5.7 Security Testing (if applicable)

We recognize that security is important, but we didn't find it applicable to our project at the current moment. Based on the scope currently, only a small number of people will be using this system, and local security isn’t the biggest concern, however, we will continue to analyze the need for security testing as our project develops.

5.8 Results

When it comes to the results of the testing, these tests will help to ensure that all of our smaller components are working separately and together to the client's expectations. It will ensure that the requirements are met since they are based on the functional requirements and some will test around nonfunctional ones such as timing the request processing times. We will have a better description and examples once the system is actually made and tests are produced to see the exact results.

6 Implementation

The following are plans we have put in place or started performing to help with implementation next semester:

- Ensure that all team members have the appropriate software and applications required to start the implementation of the project at the start of the semester.
- Ensure that any data or servers that the team needs access to are working properly and can be accessed remotely so that the team may work with them as needed.
- Communicate with the client about any possible changes that may have happened and confirm any details about that plan.
- Implement features as described in table 2.
- Creation of a template system that will allow us to hit the ground running next semester and start on development next semester.

7 Professional Responsibility

This discussion is with respect to the paper titled " Contextualizing Professionalism in Capstone Projects Using the IDEALS Professional Responsibility Assessment", *International Journal of Engineering Education* Vol. 28, No. 2, pp. 416–424, 2012

7.1 Areas of Responsibility

Table 5. Analysis of responsibilities as informed by professional standards.

Area of responsibility	Definition	NSPE Canon		Q3
Work Competence	Perform work of high quality, integrity, timeliness, and professional competence.	Perform services only in areas of their competence; Avoid deceptive acts.	Product - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.	High
Financial responsibility	Deliver products and services of realizable value and at reasonable costs.	Act for each employer or client as faithful agents or trustees.	Management - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.	Medium
Communication honesty	Report work truthfully, without deception, and understandably to stakeholders.	Issue public statements only in an objective and truthful manner; Avoid deceptive acts.	Judgment - Software engineers shall maintain integrity and independence in their professional judgment.	High
Health, safety, wellbeing	Minimize risks to safety, health, and well-being of stakeholders.	Hold paramount the safety, health, and welfare of the public.	Public - Software engineers shall act consistently with the public interest.	Medium
Property ownership	Respect property, ideas, and information of clients and others.	Act for each employer or client as faithful agents or trustees	Client and Employer - Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest	Medium
Sustainability	Protect environment and natural resources locally and globally			Low
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.	Profession - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.	Medium

7.2 Project Specific Professional Responsibility Areas

The applicability of each responsibility is assessed based on the impact it has on the project.

- Work Competence
  - High
    - Quality of work is always the utmost priority when working on a project as it affects every aspect from the top to the bottom. Our team is performing well in this regard, having gathered the necessary information and technology for this point in time, meaning we have everything we need to move forward.
- Financial Responsibility
  - Medium
    - Thus far our project has not required us to spend any money, we have funding for our project and don't expect to expend more than our allotted budget during the development of our project.
- Communication Honesty
  - High
    - Communication honesty has a high priority due to its impact on the project as a whole. Structure of a project tends to reflect the structure of communication of the team, so to ensure a well made project requires good communication between the team members.
- Health, Safety, and Wellbeing
  - Medium
    - This is an aspect of our project that is lower down in priority due to not being very applicable to our project, we do not expect to encounter any difficulties in this area of responsibility.
- Property Ownership
  - Medium
    - We want to respect the ownership of the property- algorithms and data- we used that are provided to us for the project.
- Sustainability
  - Low
    - Sustainability is low on the responsibility list due to the project having a low impact on the environment.
- Social Responsibility
  - Medium
    - We expect this responsibility to have an impact, but not an outsized impact on our project, considering the small scope of intended users resulting in the project not being a broadly social product.

7.3 Most Applicable Professional Responsibility Area

The most applicable professional responsibility area of our project is work competence. We plan to deliver work of the utmost quality as we can to our stakeholders and end users. This includes doing so in a timely and professional manner, upholding the integrity of the team and the project.

8 Closing Material

8.1 Discussion

The main result of our project has been the system design that has come out of it. We have communicated and worked with our clients to come up with a design that fits their requirements and vision for a tool that will make part of their work easier to perform. When it comes to any other results nothing else has really been made since our project is meant to start development next semester but it is in a good state to develop.

8.2 Conclusion

As for work that has been completed, a design has been made and a very basic version of the project that will be built upon has been created. Our goal for this semester was to make a design for our system and work with our client to make sure that it can fit all of their requirements so we can lay out all the features that need to be made. Our plan of action for next semester is to follow our sprints as defined in section 3.2. We feel that we didn't have any big constraint on our plan due to the ease of access to all information needed to plan this project and it is a highly software-based project that made it so no supply issues or anything affected us. As for the future, we could be more detailed oriented when it comes to planning so that it makes it easier to translate our plan into smaller more tangible deliverables/tasks such as user stories in our sprints.

8.3 References

List technical references and related work/market survey references. Do professional citation style (ex. IEEE).

IEEE/EIA 12207

IEEE/ISO/IEC 90003-2018

ISO/IEC 27001

V. Phoha, "A standard for software documentation," in Computer, vol. 30, no. 10, pp. 97-98, Oct. 1997, doi: 10.1109/2.625327.

8.4 Appendices

Discovery of Convoys in Trajectory Databases:

https://arxiv.org/pdf/1002.0963.pdf

This document was provided by the advisor and the client and it pertains to the main algorithms that we will be using in our system and how the math behind them works. This provides all the information on convoys that will be used as an “interesting event” in our system.

8.4.1 Team Contract

Team Members:

1) Jeremy Lewis 2) Claira Springer

3) Jason Guo 4) Daryl Kay

Team Procedures

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

Tuesday 2-3PM.

2. Preferred method of communication updates, reminders, issues, and scheduling:

Discord.

3. Decision-making policy:

Consensus preferred, but majority vote when necessary.

4. Procedures for record keeping:

Meeting minutes

Participation Expectations

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

All team members should attend all meetings whenever possible. If you can not attend a meeting, let the team know before the end of the day before the meeting, but preferably as soon as possible. Try not to cancel on the day of, unless it's an emergency. Be at the meeting no later than five minutes after the scheduled start time, and don't leave until it's over. All team members should be able to demonstrate understanding of what was talked about during the meeting, and to ask questions and clarify whenever necessary.

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

All team members are expected to follow deadlines/timelines set by the team, and to participate in setting deadlines and expectations regarding assignments. Team assignments are expected to be collaborative and all team members must contribute in a meaningful way to every group assignment.

3. Expected level of communication with other team members:

Communication should be frequent. Team members should update the team with regards to the status of their contributions, and all are expected to communicate with the team whenever difficulties or misunderstandings arise. Team members should also be readily available to respond to the team as soon as possible.

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

Team members should follow through with agreed upon tasks, and decisions made by the team. All team members are expected to be involved in making team decisions, and voicing concerns or opposition to decisions before they are made. If a team member needs help with completing tasks given to them, they should communicate with and consult the team for help, and if serious difficulties arise, adjustments to the plan can be made.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction,

individual component design, testing, etc.):

Claira- Administration, Client interaction, Project management, Documentation

Jeremy- Individual component design

Daryl- Testing

Jason- Overall design

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

Team members will collaborate and help each other out when needed, and ask for help when required. Roles can be changed to suit individual strengths.

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

Each team member will provide individual contributions to the project, and all contributions will be documented weekly during meetings on the team's google drive.

Collaboration and Inclusion

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

Claira- Personal/Social skills, Documentation, Testing

Daryl- Professional experience, Software creation

Jeremy- Coding language fluency, UI/UX, Problem solving

Jason- Frontend, Backend, Design

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

All team members should contribute their ideas to important decisions, and listen to ideas from the rest of the team before making decisions so that everyone's contribution is involved. Team members are encouraged to voice any thoughts or ideas they have whenever they decide to.

3. Procedures for identifying and resolving collaboration or inclusion issues

All team members are to communicate with their team whenever necessary. Team members should try to resolve any problems within the team.

Goal-Setting, Planning, and Execution

1. Team goals for this semester:

By the end of the semester we want to have a documented plan for our project, and potentially a prototype for it. All assignments should be completed and turned in on time.

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

We plan on having weekly meetings where we discuss as a team how to assign work based on the team member's individual skill and abilities, and to collaborate on finishing assignments.

3. Strategies for keeping on task:

Weekly meetings will be held where we give progress reports and hold each other accountable to any individual work assigned.

Consequences for Not Adhering to Team Contract

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

We will discuss infractions during team meetings and let the person know that the team doesn't feel they are keeping up to the expectations set by the team. If it continues to be a problem, we will create a document as a paper trail that can be written to to refer back to during team evaluations.

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

If a team member has multiple infractions of the team contract, and does not show signs of improvement, the TA or professor will become involved.

****

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) Jeremy Lewis DATE 12/2/2022

2) Jason Guo DATE 12/2/2022

3) Daryl Kay DATE 12/2/2022

4) Claira Springer DATE 12/2/2022