2 Project Plan

2.1 Task Decomposition

To solve the problem at hand, it helps to decompose it into multiple tasks and subtasks and to understand interdependence among tasks. This step might be useful even if you adopt agile methodology. If you are agile, you can also provide a linear progression of completed requirements aligned with your sprints for the entire project. At minimum, this section should have a task dependence graph, description of each task, and a justification of your tasks with respect to your requirements. You may optionally also include sub-tasks.

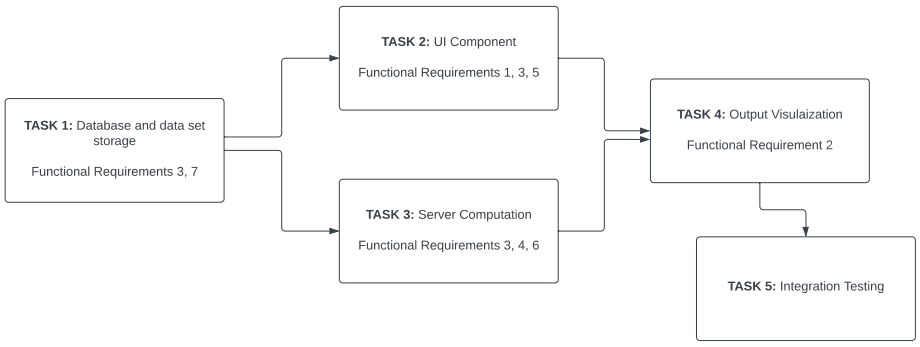


Figure 1. Task Decomposition

For our project we derived 5 major tasks that need to be completed in order to fulfill the functional requirements that we defined in the previous section. From the above tasks we have defined more specific subtasks that will need to be completed in order to fully complete the tasks.

Task 1: Database

1a. Pick a database to store .npz datasets (Could potentially use file system)

1b. Pick relational database to store user information and results

1c. Determine tables needed for relational database

1d. Create ER diagram for relational database

1e. Translate ER diagram to SQL statements

1f. Set up/deploy database server

Task 2: UI

2a. Design Prototype for UI (Figma)

2b. Choose UI framework (React, Angular)

2c. Determine list of detection algorithms and required parameters

2d. Project Setup

2e. Select Database/Algorithm/Parameters Page (Functional Requirement 1)

2f. Import Dataset Page

2g. Login/Create user Functionality

2h. Job Status Page

2i. View Results/Select Convoy to Visualize Page

2j. Connect UI to backend (Axios)

Task 3: Server/API

3a. Choose Backend Framework (Flask, Spring)

3b. Project Setup

3c. Obtain all algorithms being used for the project

3d. Endpoint to take in job parameters and start a convoy detection job

3e. Import Dataset Endpoint

3f. User Login/Creation Endpoints

3g. Retrieve Job Status Endpoint

3h. Retrieve results of convoy detection algorithm endpoint

Task 4: Convoy Visualization

4a. Discuss specific visualization needs with client

4b. Choose visualization framework (Plotly, VMD as failsafe)

4c. Create endpoint that produces a 3D visualization of a convoy

4d. Send convoy visualization to Frontend

4f. Display an interactive visualization to the user in the UI

Task 5: Testing

5a. Ensure that all components of the project can communicate with each other and produce the desired output

5b. Ensure that all component unit tests are correct.

5b. Ensure that all functional requirements are met

2.2 Project Management/Tracking Procedures

Our group will be using an agile project management style. Our tasks/subtasks will be broken down into 2 weeklong sprints (Roughly 8 sprints). The goal of our project is to produce a user friendly and easy to use product which requires constant communication, flexibility, and adaptability. By choosing the agile methodology we can iteratively develop each of our subtasks and get meaningful feedback after each sprint which is crucial when developing a customer focused product.

Our progress and tasks will mainly be tracked using GitLab issues and milestones. GitLab allows issues to be assigned to specific group members and have wights assigned to them ensuring a even workload distribution for our group members. Daily standups and communication will mainly be done through discord.

2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Figure 2. Milestones and Metrics Table

Milestone	**Metrics**
Relational Database is designed, implemented, and deployed (Task 1)	Database captures/maintains the required data for the project. Database response time: Less than 10 milliseconds for 95% of queries. Transaction throughput: Handle up to 10 concurrent users without performance degradation.
Method to store and retrieve element datasets is implemented (Task 1)	Efficiency: Large datasets are stored efficiently with a compression ratio of at least 30% Data retrieval should be reasonably fast, and dataset should take no longer than 1 minute to be loaded into the computation system
Login/Register functions are implemented (Task 2)	Prospective users are able to register new accounts and login with them. Security: all users should have their passwords and data encrypted.
Dataset/Algorithm/Parameter selection is implemented (Task 2)	Ensures that all of functional requirement 1 is met. Users can easily select the algorithm and parameters for convoy detection in an intuitive and easy to use way
Upload Dataset Functionality (Task 2/Task 3)	System should be able datasets as large as 5GB.

	Users should be able to drag and drop datasets to upload them to the server Upload speed should only be affected by client's internet speed.
UI component is able to communicate with backend (Task 2)	Only authenticated users should be able to interact with backend Response time should be less than 100ms
Convoy Detection Algorithms are implemented (Task 3)	Convoy algorithm runtimes should be the same as described in journal articles. Users should have a way of seeing the status of the algorithms
All systems are able to work and communicate with each other (Task 5)	Latency between communicating from one system to another should be less than 100ms Proper https encryption standards are used when communicating over the internet

2.4 Project Timeline/Schedule

Figure 3a. Gantt chart for the Fall 2023 Semester

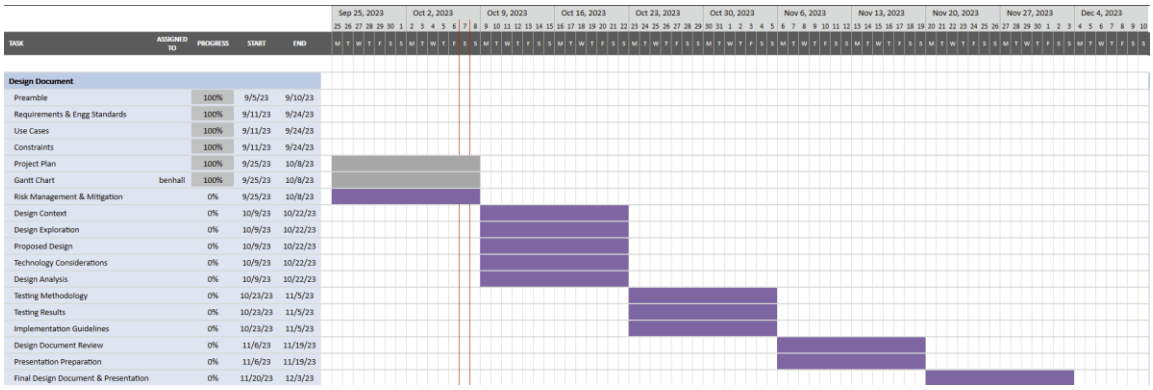
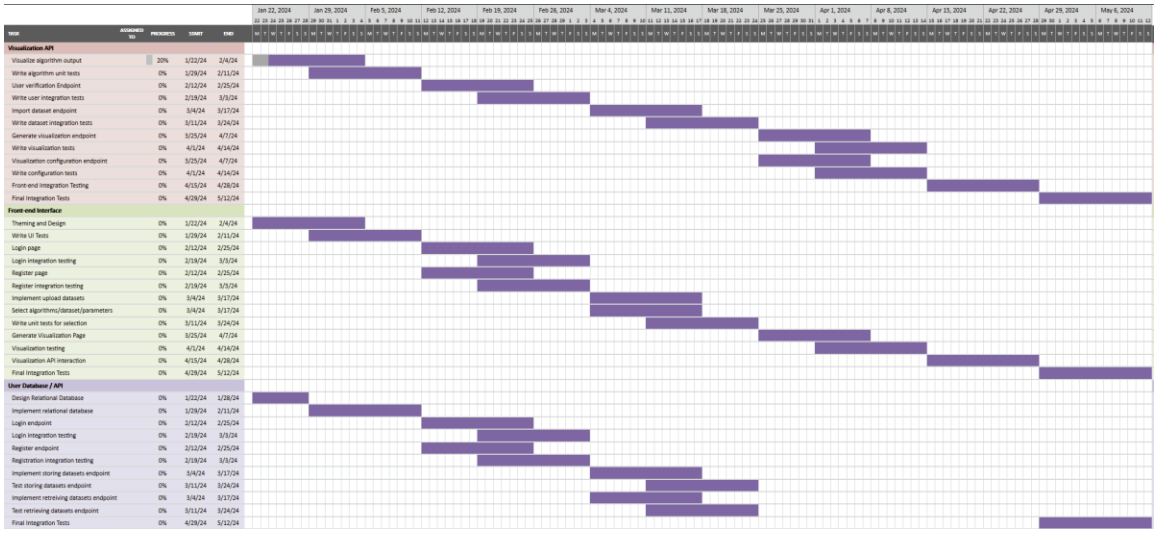


Figure 3b. Gantt chart for the Spring 2024 Semester



2.5 Risks And Risk Management/Mitigation

Figure 4. Risk Decomposition Table

Task	Risk	Mitigation	Probability
Task 1: Database	Data becomes compromised.	Follow ISO/IEC 27001 standards	0.1
Task 1: Database	Data is deleted.	Keep a backup of the data so that it can't be deleted or become unavailable	0.1
Task 2: UI	Framework does not meet our requirements	Define a backup frontend framework in case our first choice does not work	0.2
Task 2: UI	Unable to communicate with backend	Fallback to a different http request library	0.1
Task 4: Visualization	Visualization Framework unable to handle the amount of data	Reduce resolution of data being displayed, only show some time slices, render elements as simple dots.	0.8
Task 4: Visualization	Visualization API cannot handle several connections and requests	Implement queue system for visualization requests	0.3

2.6 Personnel Effort Requirements

Figure 5. Personnel Effort Breakdown Table

Task	Setup /Research Hours	Implementation Hours	Explanation
1a. Pick a database to store .npz datasets (Could potentially use file system)	3	1	There may only be a few options available
1b. Pick relational database to store user information and results	2	0	Will mostly likely use MySQL but need to ensure that it will meet our requirements.
1c. Determine tables needed for relational database	5	0	Will take some time to figure out what tables are needed for the project
1d. Create ER diagram for relational database	2	4	Some setup time to learn Lucid Charts or some other diagram tool
1e. Translate ER diagram to SQL statements	2	4	Just some research and testing for simple SQL statements.
1f. Set up/deploy database server	1	4	Getting access to Iowa State servers and setting up the database on it.

2a. Design Prototype for UI (Figma)	2	10	Learn Figma if the team has not learned it yet and create program flow and design themes.
2b. Choose UI framework (React, Javascript)	1	0	Decide on which framework would be best suited for our team.
2c. Determine list of detection algorithms and required parameters	2	0	Research the algorithm to determine all the parameters needed from the user.
2d. Project Setup	0	1	
2e. Select Dataset/Algorithm/Parameters Page (Functional Requirement 1)	6	12	Might have to pull available datasets and algorithms from the backend for this.
2f. Import Dataset Page	6	12	Developing an upload function with security concerns and writing tests in conjunction with the backend to test functionality.
2g. Login/Create user Functionality	6	12	Developing a user account system with security concerns in mind. Writing tests to confirm security and functionality.
2h. Job Status Page	8	10	
2i. View Results/Select Convoy to Visualize Page	12	16	Might take some research with the visualization aspect as the output is an HTML file.
2j. Connect UI to backend (Axios)	12	24	Integration with backend for user and dataset storage. Will need to test login, register, upload, and retrieval functions.
3a. Choose Backend Framework (Flask, Spring)	2	0	Research what framework meets our requirements.
3b. Project Setup	1	0	Setting up the project in GitLab.
3c. Obtain all algorithms being used for the project	1	0	Receive all algorithms from client.
3d. Endpoint to take in job parameters and start a convoy detection job	9	19	Will also have to develop or research queue framework for jobs.
3e. Import Dataset Endpoint	9	18	Requires integration with the user database and API.
3f. User Login/Creation Endpoints	9	18	Integration of User API and database.
3g. Retrieve Job Status Endpoint	6	12	Research the best way to implement and retrieve job status.
3h. Retrieve results of convoy detection algorithm endpoint	4	8	Requires integration testing from visualization API and frontend.
4a. Discuss specific visualization needs with client	4	0	Meeting with client.
4b. Choose visualization framework (Plotly, VMD as failsafe)	4	0	Research and decide which framework fits our client's needs.
4c. Create endpoint that produces a 3D visualization of a convoy	9	18	Need to optimize and refine usage of visualization library.

4d. Send convoy visualization to Frontend	2	4	Need to research ways to optimally send data securely and efficiently.
4f. Display an interactive visualization to the user in the UI	9	18	Researching and designing an elegant way to display the results of the algorithm.
5a. Ensure that all components of the project are able to communicate with each other and produce the desired output	9	18	Writing and running integration tests.
5b. Ensure that all functional requirements are met	9	18	Comprehensive testing and client feedback.

2.7 Other Resource Requirements

Algorithms and datasets will need to be obtained from the clients.