3 Project Plan

In this next section, we will discuss various aspects of our project plan.

3.1 Project Management/Tracking Procedures

To maintain our project, our team utilizes an agile style. This methodology ensures we can easily adapt to the challenges we face at each step of the project process. By splitting our work into shorter sprints, we can set short-term deadlines that allow us to better seek feedback and potentially pivot without losing substantial progress.

In order to decompose our tasks and assign them to group members, our team keeps track of progress by utilizing the issue board on our Gitlab repository. To facilitate communication and ensure all team members are making progress and not running into roadblocks, we utilize Discord for regular communication. Through this medium, we are able to discuss the division of topics, schedule meetings, track important information, and keep well-documented meeting minutes.

3.2 Task Decomposition

To delineate and break down the tasks that need to be accomplished to complete the final software deliverable, we broke the project into discrete conceptual tasks. These tasks cover the main sets of functionality that we need to implement. The tasks are broken down into four groups represented by different colors. Yellow represents database and data parsing tasks, red represents documentation, blue represents UI development, and orange represents our SQL querying system. All of this can be seen below in the diagram.

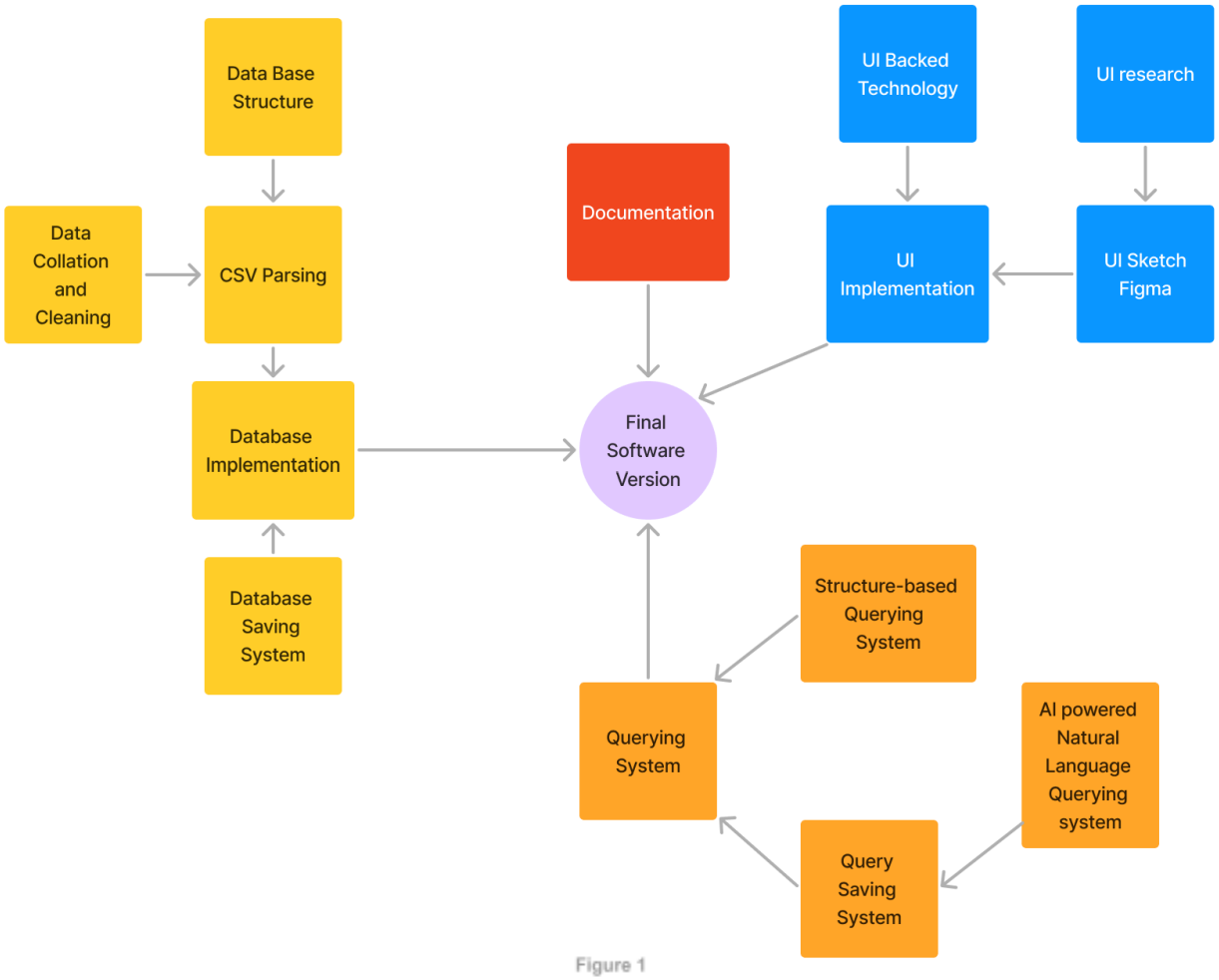

Figure 1

The tasks from the task decomposition are the following:

1. Database and Data Parsing (Yellow)

- Formulate Database Structure
    - Create a schema that easily matches the raw data into the tabular format.
- Create and test Data Collation and Cleaning system
    - Ensure that all processes relating to data loading work as expected.
- Develop CSV Parsing and Data normalization System
    - Create subject scripts to parse data from CSV files
- Integrate Database At Rest Saving System
    - Implement mechanisms to ensure the data integrity when not being accessed.
- Fully Implement and Integrate Database
    - Integrate the database into the larger system, ensuring that data is properly handled.

2. Documentation (Red)

- Create Project Documentation
- Create User Documentation
    - Ensure the user knows the features of and how to use the program

3. User Interface (Blue)

- UI research
    - Research potential libraries/frameworks we want to utilize for developing our project. We can ensure consistent design and easier development by establishing our desired tech stack.
- Develop UI Sketch in Figma for all UI pages
    - Create Figma mockups of what we want our screens to look like. By creating mockups, we can work towards a goal and establish what we want our design to look like prior to implementation, resulting in less wasted time.
- Develop UI without full logic
    - By creating a basic UI without all of the logic, we can determine any visual changes that need to be made. We will also have a better understanding of how the Figma mockups translate to reality.
- Develop and Finalize Full UI Implementation
    - Implement natural language processing API, all logic, and database connection. At this point, the website should be fully functional.

4. SQL Querying System (Orange)

- Develop AI-Powered Natural Language Query Formatting System for our Database and Schema
    - Build a system that converts user-generated natural language queries into structured queries for the implemented database.
- Develop and Integrate the Query Saving System
    - Implement a system to store, save, and retrieve custom queries for easy access and reuse.
- Develop and Integrate the Structure-based Query Formatting System
    - Ensure that queries are checked for correctness syntactically.
- Finalize and Integrate the Whole Querying System

- Combine the natural language, saved query, and structure-based query system into one system.

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Milestones:

1. Set up the development environment for all members
2. Create Figma mockups for front-end and consult with the client
3. Create UI components in React
    a. Query Page
    b. Query History Page
    c. Data Upload Page
    d. Settings Page
4. Create database schema
5. Set up CI/CD for the project
6. Implement data parsing solution
7. Implement the back-end server for data hosting
8. Implement API for NLP
9. Attempt to query data

3.4 Project Timeline/Schedule

Task Category	Task	10/17 - 24 1	10/24 - 31 2	10/31 - 11/7 3	11/7 - 14 4	11/14 - 21 5	11/21 - 28 6	11/28 - 12/5 7	12/5 - 12 8	12/12 - 19 9
Research	UI Tools									
Research	NLP Tools									
Design Elements	Figma Implementation									
Develop Software	Software Implementation									
Documentation	Documentation									

Figure 2 - Fall

Task Category	Task	1/20 - 27 1	1/27 - 2/3 2	2/3 - 10 3	2/10 - 17 4	2/17 - 24 5	2/24 - 3/3 6	3/3 - 10 7	3/10 - 17 8	3/17 - 24 9	3/24 - 31 10	3/31 - 4/7 11	4/7- 14 12	4/14 - 21 13	4/21 - 28 14	4/28 - 5/5 15
Research	SQL + Database															
Design Elements	Database Implementation															
Develop Software	Software Implementation															
Develop Software	Database Implementation															
Develop Software	Connection of Database & UI															
Debug	UI and NLP Debug															
Debug	Database Debug															
Documentation	Documentation															

Figure 3 - Spring

Our development cycle will be broken down into multiple sprints with events in chronological order from the top down.

- Sprint 1: October through December.
    - Project Planning
    - Product Research
    - Documentation

    - UI Creation in Figma and with REACT
    - Begin creation of NLP implementation methods
    - Test UI's ability to handle NLP
- Sprint 2: January
    - Finalize design of database structure
    - Finalize design and list of required database tools
    - Begin creation of database structure
    - Begin creation of database tools
    - Test backend communication with the database
- Sprint 3: February
    - Continue creation of database structure
    - Continue creation of database tools
    - Continue creation of NLP implementation methods
    - Begin mapping frontend buttons to backend tools
    - Test and implement currently created tools using frontend UI
- Sprint 4: March
    - Finalize database structure
    - Finalize database tools
    - Finalize NLP implementation
    - Finalize mapping frontend buttons to backend tools
    - Begin optimization of database tools
    - Begin optimization of NLP responses
- Sprint 5: April
    - Finalize optimization of database tools
    - Finalize optimization of NLP responses
    - Finalize documentation for the final deliverable

3.5 Risks And Risk Management/Mitigation

Risks	Risk Probability	Mitigation
Missing the desired deadline	0.2	Our team has established a solid communication channel and regularly meets to determine the progress of our project. If a team member is struggling, our group will work together to ensure the deadline is met.
Software issues	0.3	We are using frameworks and libraries that are already on the market. If we encounter a limitation, we need to prepare to switch technologies.
Database taking too long to query	0.4	When dealing with large datasets, there is some uncertainty regarding queries as it could take longer than expected to process. One mitigation strategy would be to provide an index of certain columns. Another potential mitigation strategy would be to partition the database (theoretically). When it comes to hardware-specific limitations, mitigations go beyond the scope of this project.

Natural language processing is not working correctly	0.4	ChatGPT works well for SQL queries based on a simple schema. However, with multiple terabytes of data and parameters that are not as easy to understand, we might encounter scenarios where generated queries contain invalid syntax. We must thoroughly test queries generated by the AI model to ensure our expected results make sense and attempt to interact with and guid chatGPT towards improving the syntax.

3.6 Personnel Effort Requirements

Task	Projected Effort
Develop a user interface	60 hours
Determine the required tables for the database	5 hours
Develop a tool to transfer data to the database	20 hours
Handle natural language processing via OpenAI	30 hours
Handle communication between the frontend application, the backend application, and the database	20 hours
Develop software to analyze queries and determine compliance with the schema and accuracy of the original request	15 hours

3.7 Other Resource Requirements

We will need a local server with large amounts of storage to host our testing database and run our backend application. In addition, this requires a continuous connection between itself and the device running the frontend application. Beyond this, we require OpenAI API input tokens to ensure we correctly retrieve queries given natural language. As a largely software-based project, most of our resources depend on developers owning personal computers. In sum, We do not expect the development efforts throughout the project to cause any unforeseen financial strain.