COMP 4560 Industrial Project Proposal

January 16, 2024

Ajay Nair
Ishraq Md Nazrul
Logan Doran
Tahmina Sadia Mahmud Rodshi

Contents

Abstract

The Canadian Grain Commission manages a vast amount of data every year. This project aims to help visualize this data in an efficient and user-friendly way. The focus is on creating a visual tool that enables users to upload various data sets and generate engaging visualizations.

Core features include presenting scientific data through multiple visualization techniques, such as mapping multiple data points, a dynamic legend system to enhance data interpretation, and incorporating interactive elements to explore trends over time. Emphasizing creativity and adaptability, this project seeks to transform raw data into comprehensive geographical maps for the Canadian Grain Commission.

Background

Courses Completed

- COMP 3010: Distributed Computing
- COMP 3020: Human-Computer Interaction I
- COMP 3030: Automata Theory and Formal Languages
- COMP 3190: Introduction to Artificial Intelligence
- COMP 3350: Software Engineering I
- COMP 3370: Computer Organization
- COMP 3380: Databases Concepts and Usage
- COMP 4180: Intelligent Mobile Robotics
- COMP 4190: Artificial Intelligence 2
- COMP 4360: Machine Learning
- COMP 4380: Database Implementation
- COMP 4820: Bioinformatics
- COMP 4620: Professional Practices in Computer Science

Ajay Nair

- Experience in software development following traditional software engineering principles and agile methodologies.
- Experience in project management and client relations.
- Experience in database management.
- Experience in UX/UI development following the User-Centered Design Cycle.
- Experience in Object Oriented Programming.
- Experience in data cleaning and analysis in an academic research setting.

Ishraq Md Nazrul

- Experience in Linux server setup, maintenance, and development, along with experience in Windows and macOS.
- Experience in PERN stack, HTML, JavaScript, CSS for web development, and Android app development.
- Experience in user-centered design using Figma and in multiple scripting languages, including Bash.
- Experience in AI/ML model development, bioinformatics and biology algorithms, and building AI-powered small-scale robots.
- Experience with the Unity game engine for game projects and developed IoT devices for real-time motion detection with display attachments.
- Experience in network configuration and management (CCNA, A+ courses) and experienced in database technologies.
- Experience in modern languages like Rust for low level system apps and experince in formal languages and automata theory for computabiltiy and complexity analysis.

Logan Doran

- Extensive C# and Java experience in industry and project work.
- Experience with MSSQL, and sqlite.
- Experience designing interfaces under usability principles.
- Experience in data cleaning, data visualization, and data analysis in an industry context.
- Experience in Object Oriented Programming.

Tahmina Sadia Mahmud Rodshi

- Experience in designing user interfaces under usability principles.
- Extensive Python, Java, C and C# experience in project work.
- Experience in implementing front-end web applications using React.
- Experience with Postgresql and database implementations.
- Experience with project planning and management.
- Experience with Agile methodologies in software development.
- Experience in managing and maintaining databases.

Problem Statement

The Canadian Grain Commission (CGC) lacks an efficient tool for visualizing complex agricultural data on geographical maps, resulting in inefficiencies in data presentation. Currently, creating maps is a time-consuming process with difficult reproducibility.

Our project aims to streamline the creation of agricultural visualization tools and provide the CGC with a dedicated platform that assists in the efficient and creative representation of their data. By establishing a robust visual tool, we hope to enhance the interpretation and accessibility of crucial agricultural information.

Methodology and Timeline

The first step of the project is to use a public GitHub repository to set up a collaborative environment. This repository will serve as the main hub for monitoring tasks, managing active dev tasks, and organizing deliverables. The configuration of each development environment will be tailored to the specific needs of the project for managing agricultural data, with an emphasis on database and platform choices.

The implementation of an Agile development technique will be paired with a user-centered design. This strategy makes it possible to make adaptable and flexible adjustments in response to frequent input, ensuring that the tool is in line with the requirements and experiences of non-technical users. To serve a varied user base, important elements like web accessibility standards and user-friendly interfaces will be given priority.

A well-structured relational database on disk and on memory will be the core of our application, enabling seamless user interaction with agricultural data. These tables will support efficient data storage, integration, and retrieval, allowing for efficient query capabilities while ensuring data accuracy and security. As the project progresses, we will adapt the database schema to handle all forms of data that a user could want to upload to the application.

Time-series and geographic data management will be accomplished through the use of libraries that are tailored to handle and process geospatial data efficiently. Preprocessing and normalization of data will also be done as part of backend development. To reduce edge cases and facilitate data management, this step is essential as it entails cleaning, converting and standardizing data formats.

The development of the user interface will prioritize usability for non-technical users with an added option for an Advanced Mode for users needing extra statistics and information. The intuitiveness of the interface will be verified through usability testing and prototyping. There will be plenty of customization choices, including dynamic style options that enable users to change visualization elements like color schemes and intensity and careful considerations will be made to comply with the Web Content Accessibility Guidelines.

The system's architecture will have security and privacy in mind when in development to ensure a robust toolset in production. The system will be continuously monitored, and tests will be implemented to ensure proper safety of user data and proper integration of the system. Maintenance will be done throughout the term to ensure everything is updated and complies with proper licensing when using external libraries.

There will be clear and comprehensive user manuals available, aimed at non-technical consumers. The architecture and data processing of the system will be described in detail in the technical documentation. The tool's general usability will be improved by providing users with assistance in addressing frequent issues through a troubleshooting guide and FAQ area.

The project will be polished at the last stage, making sure every feature functions as it should. The system will be more versatile and portable thanks to containerization, which will be accomplished via Docker. Frequent check-ins with Sean from Canadian Grain Commission and Professor Robert Guderian will guarantee ongoing development in response to input. The product will be ready for use, guaranteeing simple incorporation into current systems and usability for the intended audience.

The following is the detail of the timeline:

Timeline:

Combined Project Tasks	
Tasks	Tentative Completion Dates
Infrastructure and Environment Setup: • Discuss architecture. • Setup individual environments. • Setup CI/CD and Containerization. • Generate Issues for user stories and features. • Split up dev tasks. • Discuss potential platforms to develop application. • Set up Development database.	January 25, 2024
Check in with Sean and Robert Guderian	January 27, 2024
Milestone 1: • Set up the map interface, including geographical regions (provinces and agricultural zones). • Develop data upload functionality. • Implement basic data visualization with initial color customization. • Add download functionality for visualized data maps in a single file format as photo format.	Febuary 6, 2024
Check in with Sean and Robert Guderian	Febuary 7, 2024

Milestone 2: • Preprocessing of data and Outlier Detection. – Outlier data detection in terms of GIS. • Implement feature to upload multiple data sets. • Implement Observational View for multiple data sets. – Two main options: Overlay or Side by side comparison • Searching and modifying Map: – Implementing a feature to allow users to view changes occurring in cities or municipalities by focusing on a subset of a city. – Implement a feature to allow users to modify colors to preferred color choices. • Implement advanced numerical filtering options for combined datasets (this is dependent on dataset that is uploaded) – Ability to filter results based on fields that are present in the dataset.	Febuary 25, 2024
Check in with Sean and Robert Guderian	March 10, 2024
Milestone 3: • Enhance data export options, supporting multiple file formats. • Implement history feature to track changes made to files. – Incorporate undo/redo feature for region coloring • Matrix/Album of images of how the trend would change: – Add feature to view the data with clips – use sliders to show the change over time.	March 10 , 2024
Check in with Sean and Robert Guderian	March 11, 2024

• Conduct Unit, Integration Tests for the application with more complicated datasets to ensure proper functionality	March 20, 2024
Check in with Sean and Robert Guderian	March 22, 2024
• Create a guide that walks the user through the process of interacting with the application for users without a technical background. • Outline an index with a description of the project and provide brief summaries of each feature that the application offers. • Additionally, assemble a comprehensive instruction manual detailing the steps necessary to configure the environment for running the application.	March 30, 2024
Check in with Sean and Robert Guderian	April 1, 2024
• A final document outlining final tasks completed over the course of the project. • Prepare and compile final presentation slides and discussing potential improvement to be made to the application. • Decide who will present which parts of the projects.	TBD

Infrastructure, facilities and expert personnel requirements

To ensure the success of this project, we would need access to the data collected and held by the Canadian Grain Commission. Additionally, we would require insights and support from an HCI professional, specifically Dr. Patrick Dubois, to ensure a proper user-designed approach. These insights will prove beneficial when we require guidance on decisions outside of our areas of expertise. We have also discussed potential development-focused support from Sean through a colleague from the Canadian Grain Commission.

We are planning to communicate and establish a connection with Sean's co-worker over email. We are expected to require additional resources such as the Linux lab, and potential Geographic Information System (GIS) experts such as Dr. David Walker from the Riddell Faculty of Environment, Earth, and Resources within the university. However, should we encounter issues in accessing any of these resources we will investigate alternatives such as graduate students in GIS-focused areas of research.

Outcome and Deliverables

Shrink Goals

- Develop an upload feature using Node.js and Express, with drag-and-drop functionality and file type validation (CSV, JSON, XML) using JavaScript and HTML5 APIs.
- Implement a PostgreSQL database with PostGIS extension for storing and managing raw spatial data, ensuring normalization and efficient indexing for optimized query performance.
- Create Python scripts using Pandas and NumPy for preprocessing, including outlier detection, normalization, and handling missing values, wrapped as importable modules.
- Design the interface in React, ensuring compliance with WCAG (Web Content Accessibility Guidelines) standards, including keyboard navigation and screen reader support.
- Utilize D3.js and Leaflet.js for creating interactive heatmaps, applying data binning and color gradient techniques to represent data density and variations.
- Implement a timelapse feature in JavaScript with an interactive slider control using React and CSS3, enabling users to view changes in data over time.
- Provide file export options using Node.js, supporting formats like CSV, PNG for images, and JSON, with options to choose data range and parameters.
- Develop a dynamic legend in the UI using React and JavaScript, allowing users to toggle data categories, adjust visibility, and personalize display settings.
- Create comprehensive documentation using Sphinx for Python components and JSDoc for JavaScript code, covering system setup, configuration, and feature guides.
- Develop Python libraries for time-series analysis, integrating Pandas for data manipulation, and use libraries like GeoPandas for geographic data processing.
- Implement basic security measures including self-signed HTTPS, secured API-endpoints, file validation, middleware to handle user states/settings, and encryption for sensitive data in the database.
- Integrate a color filtering feature in the frontend using JavaScript and CSS, allowing automatic adaptation based on data values or manual adjustments by users through a UI control panel.

Expected Goals

- Implement Agile methodologies using tools like JIRA or Trello for project management, ensuring regular sprints and user feedback integration. Utilize design thinking approaches for UI/UX design, focusing on React-based frontend development.

- Integrate additional testing frameworks like Jest for React components and Mocha for Node.js backend. Optimize PostgreSQL database queries using EXPLAIN ANALYZE, and index tuning for enhanced performance.

- Develop Python scripts using Scikit-learn and Statsmodels for statistical analysis to identify outliers. Use Pandas for data imputation strategies (like mean, median, or mode substitution) for handling missing data.

- Enhance Leaflet.js map interactivity to support multiple dataset uploads, enabling users to overlay different datasets or view them side-by-side. Implement advanced visualization features like matrix views and album-style image galleries for comparative analysis.

- Expand the Node.js/Express backend to support additional data formats like Excel (XLSX), Parquet, and GeoJSON. Implement file parsers and converters to ensure seamless data integration and preprocessing.

- Strengthen security by preventing SQL injection in PostgreSQL. Incorporate data anonymization techniques and regular security audits to enhance privacy.

- Develop functionality in the backend to export visual data in formats like SVG, PDF for static visuals, and implement WebM or MP4 formats for dynamic representations using FFmpeg or similar libraries.

- Implement advanced filtering and data slicing features in the React frontend, using Redux for state management to allow users to select, filter, and explore specific data subsets dynamically.

- Utilize Docker for containerizing the entire application, including Node.js/Express backend and React frontend. Implement Docker Compose for managing multi-container setups, ensuring system portability and consistency across different environments.

Stretch Goals

- Redesign the user interface using React and CSS frameworks like Bootstrap or Material-UI for responsive design, ensuring compatibility across devices. Incorporate user experience best practices, such as A/B testing and user journey mapping.

- Integrate a logging system in the frontend using Redux or a similar state management library to track and display a history of user interactions and data changes. Use backend logging frameworks like Winston in Node.js for server-side tracking.

- Expand download options to include additional map file formats like KML, GPX, or Shapefile. Implement backend functionalities using libraries such as GDAL (Geospatial Data Abstraction Library) for file conversion and handling.

- Develop a dual-mode interface in React, offering ’Easy’ mode with simplified controls and visualization for novice users, and ’Advanced’ mode with detailed customization options for more experienced users.

- Investigate and prototype AI technologies, such as machine learning algorithms in Python using TensorFlow or PyTorch, for advanced data processing and predictive analytics. Consider NLP techniques for extracting insights from textual data.

- Create a dynamic, customizable dashboard using React and JavaScript, allowing users to configure and save their preferred observational views, including widget layouts and data sources.

- Implement user-centric visualization features, allowing individual users to personalize visual aspects like color schemes, graph types, and filters based on their preferences and usage patterns.

- Enhance map interactivity in Leaflet.js, providing users with options to customize map features such as overlays, layer visibility, and thematic styling, catering to specific user requirements and enhancing the overall experience.