INITIAL PROJECT PLAN

Faculty of Engineering, Dalhousie University -TranSECT Project WP5.

Yinuo Yang

Development Director | yn514369@dal.ca

Yingtian Zhang

Technical Director | yingtian@dal.ca

Key Stakeholders

1. Clients
    a. Name: Camilo Botero (Faculty of Engineering)
    b. Responsibilities: Provide feedback, validate project needs and propose requirements that help achieve end goals.
    c. Contact: cmbotero@dal.ca, (902)-3996309
2. Project Manager TA
    a. Name: Ridwan Oladipupo
    b. Responsibilities: Oversees project, monitors progress and provides assistance if necessary.
    c. Contact: ridwan.oladipupo@dal.ca
3. Development Director
    a. Name: Yinuo Yang
    b. Responsibilities: Managing and leading the development team, collaborating with the stakeholders to create tasks and ensure project is on track towards the final goal.
    c. Contact: yn514369@dal.ca
4. Technical Director
    a. Name: Yingtian Zhang
    b. Responsibilities: Analyze and refine requirements with the client, author technical documents, identify risks, design architecture, be responsible for the technical solution
    c. Contact: yingtian@dal.ca
5. Senior Developer
    a. Names: Madhulika Reddivari, Gavin Sharma
    b. Responsibilities: Researching technical solutions, developing and testing the software, mentoring their juniors.
    c. Contact: mreddivari@dal.ca, gavin.sharma@dal.ca
6. Junior Developer
    a. Name: Jaylyn Publicover,Yinning Zhu, Nuoling jin, Ahmed Elbayomi
    b. Responsibilities: Assist in coding, debugging, and testing while supporting the development team to achieve project goals.
    c. Contact: jy298482@dal.ca, yn561087@dal.ca, nl916044@dal.ca, ahmed.elbayomi@dal.ca

Work that Has Been Done Already

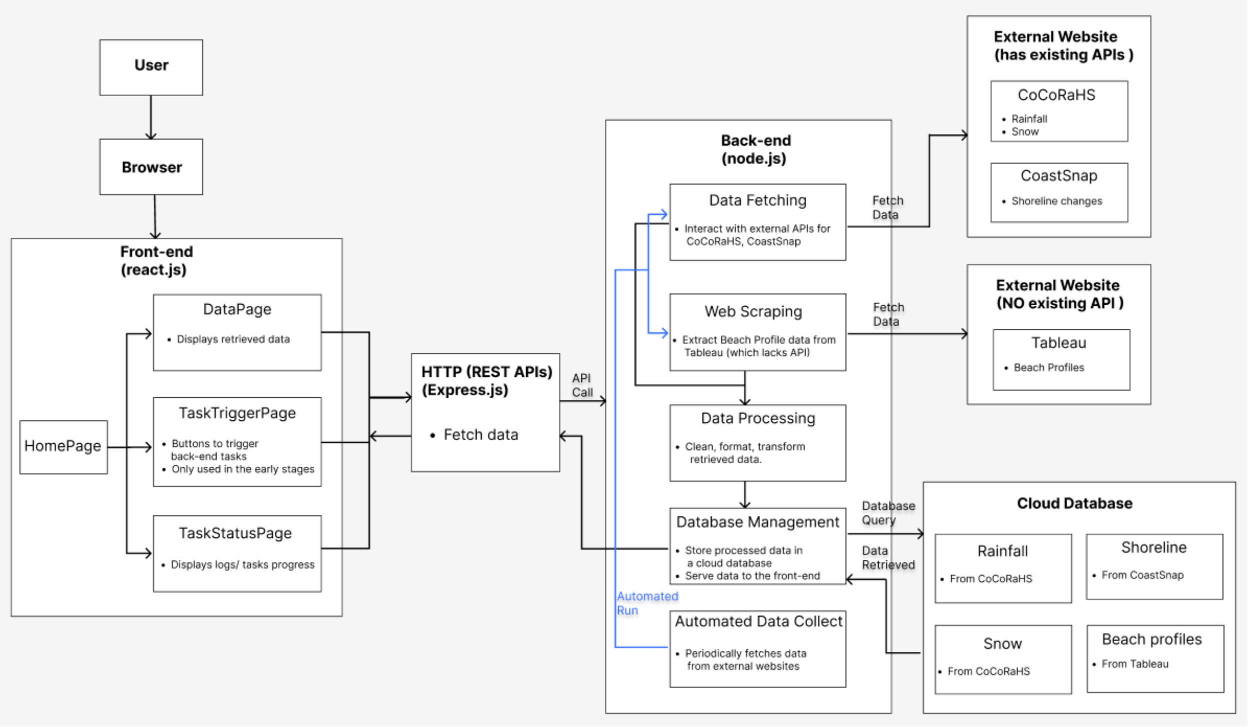
Since this is a new project, there is no existing codebase. However, initial step have been completed to plan for the project.

1. Requirements Gathering
    a. Initial meetings were held with the client and other collaborating groups to gather and clarify project requirements.
    b. The types of data to be collected and stored in the cloud database were determined to understand the client's basic needs for APIs
    c. Already understand the budget of the project
2. Technology Exploration
    a. Initial exploration of the CoCoRaHS and CoastSnap APIs is currently ongoing to understand their integration potential, while awaiting more detailed client requirements and collaborative research with other groups.

Work that Needs to be Done

1. Automated Data Collection:
    a. Automate data collection from the CoCoRaHS and CoastSnap projects.
    b. Develop a new data collection process for Beach Profiles, possibly including a Telegram mini-app or web platform.
2. Data Management:
    a. Centralize all data into a cloud-hosted MySQL database.
    b. Develop extract, transform, load processes to clean and transform data to fit the database structure.
    c. The result is a simple and intuitive way for citizen scientists to submit data.

High-Level Architectural Design



Initial Plan for the Solution

1. Timeline
    a. Phase 1: Project Kickoff (1 Week)
        i. Meetings with client
        ii. Gather project requirements and priorities
    b. Phase 2: Research, Design and Initial Development (1 Week)
        i. Document project requirements and initial solution design
        ii. Sketch user-friendly designs for the web platform and Telegram mini-app.
        iii. Confirm the tech stack (e.g. MySQL, React.js, Python/Node.js, Telegram Bot API).
    c. Phase 3: Development (7 weeks)
        i. Develop and refine core interfaces
        ii. Implement user roles and permissions for citizen scientists and administrators.
        iii. Database setup
            1. Design and initialize the MySQL database schema.
            2. Develop ETL pipelines for integrating data from CoCoRaHS and CoastSnap.
        i. API integration
            1. Build scripts to fetch data from CoCoRaHS and CoastSnap.
            2. Set up endpoints for user-submitted data.
        ii. Data upload and download
            1. Enable data submission through web and Telegram apps.
            2. Store validated data in the cloud-hosted MySQL database
    d. Phase 4: Testing & Quality Assurance (1 week).
        i. Refining smaller features based on feedback.
        ii. Testing the result.
        iii. Bug fixes.
    e. Phase 5: Project Closing (1 week)
        i. Conduct final checks.
        ii. Write closing documents for the project.
2. Tools and Technologies
    a. Back-end:
        i. MySQL: Cloud-hosted relational database for storing and managing all data
        ii. Node.js: For regularly pulling data, parsing and cleaning the data, and finally storing it in the database.
        iii. Postman: For testing API endpoints for data collection and integration workflows, validating data fetched from external APIs (such as

CoCoRaHS and CoastSnap), and debugging and automating API-related development processes.

b. Front-end:
   i. React.js: For create a simple webpage for developers to check whether the data has been successfully collected and stored

c. Development tool:
   i. Version Control: Git and Gitlab for code management.
   ii. Development Environment: Visual Code studio.

d. Project Management:
   i. Jira for track tasks and sprints management.
   ii. Microsoft Teams for team communication.

3. Meeting Time Constraints
   a. Agile Methodology: The team will be adopting an agile approach with iterative development. This allows for flexibility and adaptability when it comes to client feedback and any changes to requirements.
   b. Bi-weekly meetings: The team will meet with the client after each sprint to review progress and discuss feedback.
   c. Prioritize working on core requirements, features and functionalities.

Deliverables and Sprint Plan

1. Frequency

   a. There will be four two-week sprints, the last of which will last 1 week.

2. Deliverables by Sprint

   a. Sprint 1 Focus: Project Setup and Initial Design.

      i. 2025/Jan/24 to 2025/Feb/06

      ii. Initial Project Plan.

      iii. Setup of development tools and technologies.

      iv. Design wireframe for the front-end interfaces.

      v. Basic Node.js application structure and initial development of interfaces
      vi. Initial MySQL database

   b. Sprint 2 Focus: User Interface Design and Initial Database Setup.

      i. 2025/Feb/06 to 2025/Feb/20

      ii. Refining the database structure and APIs rules.

      iii. Development of simple interfaces:

1. Public interface
    a. Home
    b. Database display
    c. Trigger Tasks
    d. Task status

iv. Set up the cloud-hosted MySQL database for managing user accounts and collected environmental data.

c. Sprint 3 Focus: Data Integration, API Development, and Authentication
    i. 2025/Feb/20 to 2025/Mar/06
    ii. Integrate APIs to automate data collection from CoCoRaHS and CoastSnap**.**
    iii. Develop a data tracking dashboard on the website to monitor in real-time whether data has been successfully collected and stored in the database.

d. Sprint 4 Focus: Application-Database functionality.
    i. 2025/Mar/06 to 2025/Mar/20
    ii. Develop and test the Telegram Mini-App for beach profile data submissions.
    iii. Implement data upload and download functionality, ensuring seamless storage and retrieval in the Cloud hosted MySQL database.

e. Sprint 5 Focus: Testing, quality assurance.
    i. 2025/Mar/20 to 2025/Apr/03
    ii. Refine smaller features based on feedback.
    iii. Test APIs using Postman, and simple Node.js scripts to simulate front-end requests.
    iv. Perform unit tests, integration tests, and user acceptance testing.
    v. Fix bugs identified during testing.
    vi. Deploy the final application.

Deadlines

The client has indicated that the deadlines are set by the course and the expectations are to see the team achieve the greatest success possible within the time constraints.

Quality Assurance Plan

1. **Manual Testing**

The team will conduct manual testing 2 days before the end of each Sprint, including the display of the front-end page, the call of the API endpoint, and the data retrieval process to ensure that the tasks completed in this Sprint meet the requirements and perform well during operation.

2. **User Acceptance Testing (UAT)**
   At the customer meeting after each Sprint, we will show the current results to users and collect their feedback. We will ensure that the product meets user needs in terms of functionality and usability and make improvements based on feedback.
3. **Data Validation**
   Data verification will be performed through front-end page display and database checking. Test the back-end process to ensure that the data obtained from external websites is accurate, in the correct format, and correctly stored in the cloud database.
4. **Performance Testing**
   Performance testing will ensure that the backend can reliably obtain data from external websites and store it correctly in the database. Use Postman to test API response and run simulated crawling to test the stability of Web Scraping. At the same time, by inserting large-scale data into the database and monitoring query performance, we ensure data integrity and the system's error handling capabilities under high load.

Project Risks and Risk Management Strategies

Project Risks

1. Team members do not have experience using APIs to get data, which may cause delays or mistakes in development.
2. Website Tableau doesn’t have API to get data, we need to build new tools like data scraping or automation scripts, but the team lacks experience in these areas.

Project Risks Mitigation:

1. Learn the API structure in advance, study the API request method on the official websites of CoCoRaHS and CoastSnap, and use Postman for testing to ensure that you understand the API response format.
2. Analyze Tableau's data storage methods and develop small test scripts to verify the feasibility of partial data extraction.

Product Risks

1. Since this is a new project, if the programming language (react.js, express.js, node.js) we choose does not work well for the project, we may need to spend extra time and effort to redo the work.

2. The API response time might be too slow, which can hurt the performance of the automated tools.
3. The API may have request limits (rate limits) that are too low, which could stop us from getting all the data we need on time.

Product Risks Mitigation:

1. When encountering technical limitations, prioritize optimization rather than changing the technology stack, and improve performance through code optimization, caching, or concurrent processing.
2. Optimize API calls, implement asynchronous requests and batch data processing, reduce waiting time, and set timeout processing to prevent the system from freezing.
3. Study API rate limits, optimize code to reduce invalid requests, and use request queues (such as Bull) to execute API requests in batches to avoid exceeding quotas.

Business Risks

1. CoCoRaHS and CoastSnap might change how they let people access their data, for example, by switching from free to paid access. This could affect the project.
2. The project depends a lot on CoCoRaHS and CoastSnap for data. If they stop their service or block access, the project will face big problems.

Business Risks Mitigation:

1. Pay attention to changes in API access rules and check official announcements and API documents regularly. Study other available data sources, establish data redundancy solutions, and reduce dependency risks.
2. Establish a data backup mechanism and regularly store historical data to prevent the impact of sudden changes or loss of data sources.