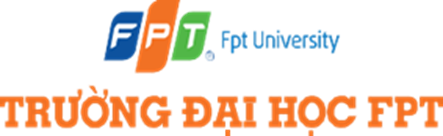
**Capstone Project Report**

**Report 2 – Project Management Plan**

# **II. Project Management Plan**

## **1. Overview**

### **1.1 Scope & Estimation**

| **#** | **WBS Item** | **Complexity** | **Est. Effort**  **(man-days)** |
| --- | --- | --- | --- |
| ***1*** | **Research and collect requirements** |  | ***90*** |
| 1.1 | Research about LCA and how it works | Complex | 30 |
| 1.2 | Research about some existing LCA systems with similar function | Complex | 30 |
| 1.3 | Research the data needed for LCA calculation | Complex | 30 |
| ***2*** | **Set up development environment** |  |  |
| ***2.1*** | Set up Back-end project | Medium |  |
| ***2.2*** | Set up Front-end project | Medium |  |
| ***2.3*** | Set up Back-end API deployment | Medium |  |
| ***2.4*** | Set up CI/CD | Medium |  |
| ***3*** | **Initialise database** |  |  |
| ***3.1*** | Designing database | Complex |  |
| ***3.2*** | Create database | Medium |  |
| ***4*** | **Security** |  |  |
| ***4.1*** | Authentication | Medium |  |
| ***4.2*** | Authorization | Medium |  |
| ***5*** | **System Admin** |  |  |
| ***5.1*** | Configuration administration project | Medium |  |
| ***5.2*** | Create login context | Medium |  |
| ***5.3*** | Create login page | Medium |  |
| ***5.4*** | Create admin dashboard layout | Medium |  |
| ***5.5*** | Create User | Medium |  |
| ***5.6*** | View User | Medium |  |
| ***5.7*** | Update User | Medium |  |
| ***5.8*** | Delete User | Medium |  |
| ***5.9*** | View Impact Category | Medium |  |
| ***5.10*** | Create Impact Category | Medium |  |
| ***5.11*** | Update Impact Category | Medium |  |
| ***5.12*** | Delete Impact Category | Medium |  |
| ***5.13*** | View Emission Factor | Medium |  |
| ***5.14*** | Create Emission Factor | Medium |  |
| ***5.15*** | Update Emission Factor | Medium |  |
| ***5.16*** | Delete Emission Factor | Medium |  |
| ***5.17*** | View Emission Impact | Medium |  |
| ***5.18*** | Create Emission Impact | Medium |  |
| ***5.19*** | Update Emission Impact | Medium |  |
| ***5.20*** | Delete Emission Impact | Medium |  |
| ***5.21*** | View Life Stage Template | Medium |  |
| ***5.22*** | Create Life Stage Template | Medium |  |
| ***5.23*** | Update Life Stage Template | Medium |  |
| ***5.24*** | Delete Life Stage Template | Medium |  |
| ***6*** | **Manager** |  |  |
| ***6.1*** | Create organization manager account | Medium |  |
| ***6.2*** | View organization manager | Medium |  |
| ***6.3*** | Update organization manager | Medium |  |
| ***6.4*** | Delete organization manager | Medium |  |
| ***6.5*** | Create contract | Medium |  |
| ***6.6*** | View contract | Medium |  |
| ***6.7*** | Update contract | Medium |  |
| ***6.8*** | Delete contract | Medium |  |
| ***6.9*** | Create transaction | Medium |  |
| ***6.10*** | View transaction | Medium |  |
| ***6.11*** | Update transaction | Medium |  |
| ***6.12*** | Delete transaction | Medium |  |
| ***7*** | **Organization Manager** |  |  |
| ***7.1*** | Create Team | Medium |  |
| ***7.2*** | View Team | Medium |  |
| ***7.3*** | Update Team | Medium |  |
| ***7.4*** | Delete Team | Medium |  |
| ***7.5*** | Upload Proof of Transaction | Medium |  |
| ***8*** | **LCA Staff** |  |  |
| ***8.1*** | Create Project | Medium |  |
| ***8.2*** | View Project | Medium |  |
| ***8.3*** | Update Project | Medium |  |
| ***8.4*** | Delete Project | Medium |  |
| ***8.5*** | Import Project | Medium |  |
| ***8.6*** | Export Project | Medium |  |
| ***8.7*** | Create Object Library | Medium |  |
| ***8.8*** | View Object Library | Medium |  |
| ***8.9*** | Update Object Library | Medium |  |
| ***8.10*** | Delete Object Library | Medium |  |
| ***8.11*** | View Node | Medium |  |
| ***8.12*** | Create Node | Medium |  |
| ***8.13*** | Update Node | Medium |  |
| ***8.14*** | Delete Node | Medium |  |
| ***8.15*** | Conduct LCA Analysis | Medium |  |
| ***8.16*** | Import Environmental Data | Medium |  |
| ***8.17*** | Update Environmental Data | Medium |  |
| ***8.18*** | Create Report | Medium |  |
| ***8.19*** | View Report | Medium |  |
| ***8.20*** | Update Report | Medium |  |
| ***8.21*** | Delete Report | Medium |  |
| ***8.22*** | View team member | Medium |  |
| Table 3 - Scope & Estimation | | |  |

### **1.2 Project Objectives**

| **#** | **Testing Stage** | **Test Coverage** | **No. of Defects** | **% of Defect** | **Notes** |
| --- | --- | --- | --- | --- | --- |
| 1 | Reviewing | 99% | <20 | 1% |  |
| 2 | Unit Test | 99% | <20 | 1% |  |
| 3 | Integration Test | 99% | <20 | 1% |  |
| 4 | System Test | 99% | <20 | 1% |  |
| 5 | Acceptance Test | 99% | <20 | 1% |  |

Table 4 - Project Objectives

### **1.3 Project Risks**

| **#** | **Risk Description** | **Impact** | **Possibility** | **Response Plans** |
| --- | --- | --- | --- | --- |
| 1 | The requirement is not fully understood about the subscription | Critical | High | Hold discussions between the developer team and product owners to better understand the user's requirements. |
| 2 | Inaccurate data input for the LCA process | High | Medium | Implement data validation protocols and periodic reviews of input accuracy. |
| 3 | Difficulty in ensuring security and privacy of sensitive data | High | Medium | Integrate security best practices such as encryption and regular security audits. |
| 4 | A certain member of the team is forced to leave the team or leave the team due to personal reasons. | Critical | Low | + Scale down the scope of the  project.  + Share the workload among the remaining team members to meet the deadline. |
| 5 | Lack of experience or skills in coding and testing the project.  Could not find the  suitable technique or solution to implement a certain feature. | Critical | High | Identify the main techniques and the difficult techniques in advance to build a reasonable study plan. |

Table 5 - Project Risks

## **2. Management Approach**

### **2.1 Project Process**

The Cabonerf project has chosen the Waterfall methodology, for several compelling reasons:

1. **Structured and Sequential Process**: Cabonerf requires meticulous planning and step-by-step analysis of phases like raw material extraction, production, and transportation, etc. Waterfall’s linear progression ensures that each phase (e.g., data collection, environmental impact analysis, reporting) is completed before moving on, reducing the risk of missing critical data or stages.
2. **Clear Documentation and Requirements**: Cabonerf is heavily data-driven and depends on specific environmental criteria and regulations. Waterfall emphasizes early-stage documentation and requirement gathering, which is crucial for setting unambiguous system parameters, ensuring compliance with environmental standards, and aligning with the business's sustainability goals.
3. **Minimized Rework and Iteration**: Since each stage in Waterfall is completed before moving to the next, there’s less need for backtracking. For an LCA system, where environmental impact calculations are complex and costly to reanalyze, a well-executed linear process minimizes the chance of needing to redo large portions of the work.
4. **Well-Suited for Stable Requirements**: The Cabonerf's requirements are well-defined upfront—such as predefined environmental indicators or regulatory standards—Waterfall’s rigid structure is advantageous, as the system is built around these stable parameters without needing frequent changes or adjustments.
5. **Easier Quality Control and Testing**: With Waterfall, testing occurs only after all development phases are complete. This ensures the entire LCA system is evaluated as a whole, which can help in catching errors that might affect multiple interconnected phases (e.g., a data inconsistency in raw material analysis impacting the final environmental assessment).

### **2.2 Quality Management**

To ensure the quality of our project, we adhere to the following general rules:

1. **Coding Conventions:**

English Language Usage: All class names, variable names, constant names, functionnames, and file names must be written in clear and meaningful English. Casing Conventions: Follow standard casing conventions for different programming languages, such as Pascal case or Kebab case, as documented for the project. Consistency In Naming conventions is essential for code readability.

1. **Code Formatting:**

Whitespace and Operators: Ensure that there is proper spacing between variables and operations to enhance code readability and maintain uniformity.

1. **Code Comments:**

Unused Code: Unused code should be removed rather than commented out, as it keeps the codebase cleaner.

Comment Warnings: Use comments to indicate potential issues, bugs, or areas of improvement. This helps in identifying and addressing problems during code reviews. Code Clarification: For complex logic or algorithms, provide comments that clarify the meaning and functionality of the code. This aids in understanding and maintaining the codebase

1. **System Testing:**

Adherence to SRS: System testing should verify that the application's entire functionality adheres to the Software Requirements Specification (SRS) document. Ensure that every feature, including core functionalities and edge cases, functions correctly.

Holistic Approach: Test the application as a whole, checking interactions between different modules and ensuring that they collectively fulfill the intended purpose. Regression Testing: Execute regression testing to confirm that new changes or features do not negatively impact existing functionalities.

1. **Documentation:**

Detailed Documentation: Maintain detailed documentation that describes the coding conventions, test cases, testing procedures, and testing results. This documentation is crucial for reference, quality control, and future improvements

1. **Feedback Loop:**

Continuous Improvement: Encourage a culture of continuous improvement and feedback among the development team. Regular retrospectives can help identify areas for enhancement in the quality management process.

1. **Security and Data Privacy:**

Data Security: Prioritize the security of user data and ensure compliance with data privacy regulations. Perform security testing and code reviews to identify and address vulnerabilities

### **2.3 Training Plan**

| Training Area | Participants | When, Duration | Waiver Criteria |
| --- | --- | --- | --- |
| Java Spring Boot | Đặng Minh Thắng  Phan Hữu Hoàng Sơn | Week 1, 3 days | Mandatory |
| Git, Github | Everyone | Week 1, 3 days | Mandatory |
| ReactJS | Bùi Quang Vinh  Trần Quang Minh | Week 1, 3 days | Mandatory |
| 3rd parties' services: AWS | Everyone | Week 1, 3 days | Mandatory |
| Business general procedure | Everyone | Week 1, 3 days | Mandatory |

Table 6 - Training Plan

## **3. Project Deliverables**

| **#** | **Deliverable** | **Due Date** | **Notes** |
| --- | --- | --- | --- |
| 1 | ProjectCode | 15 weeks |  |
| 2 | Project Start |  |  |
| 3 | **Initiation (Stage 1)** | 3 days | Week 1 |
| 4 | Prepare Report 1 |  |  |
| 5 | Deliver Report 1 (Project Introduction) |  |  |
| 6 | **Planning & Requirement (Stage 2)** | 7 days | Week 2 |
| 7 | **Project Planning** |  |  |
| 8 | Prepare Project Plan |  |  |
| 9 | Deliver Report 2 (Project Plan) |  |  |
| 10 | **Project Preparing** |  |  |
| 11 | Create Technical Prototype |  |  |
| 12 | Technical Training/Self Study |  |  |
| 13 | **Requirement Analyzing** |  |  |
| 14 | Write SRS |  |  |
| 15 | Update Project Plan |  |  |
| 16 | Deliver Report 3 (SRS + Plan) |  |  |
| 17 | **Database Designing (Stage 3)** | 7 days | Week 3 |
| 18 | Draw ERD |  |  |
| 19 | Update Project Plan |  |  |
| 20 | Deliver Report4 (SDS + Plan) |  |  |
| 21 | **Implementation (Stage 4)** | 70 days | Week 4 -> 12 |
| 22 | Update SRS |  |  |
| 23 | Create Test Cases |  |  |
| 24 | Code for customer use cases |  |  |
| 25 | **Verification (Stage 5)** | 7 days | Week 13 |
| 26 | Fix Bugs |  |  |
| 27 | Prepare for Deployment |  |  |
| 28 | Update Project Documents |  |  |
| 29 | Complete Full Software Package |  |  |
| 30 | **Transition (Stage 6)** | 7 days | Week 14 |
| 31 | **Software Optimization** |  |  |
| 32 | Optimize/Refactor Codes |  |  |
| 33 | Fix Bugs |  |  |
| 34 | Deploy Final Code |  |  |
| 35 | **User Guides** | 7 days | Week 15 |
| 36 | Prepare User Guides |  |  |
| 37 | Deliver Report 6 (User Guides) |  |  |
| 38 | Prepare Final Project Report |  |  |
| 39 | Prepare Thesis Presentation |  |  |
| 40 | Deliver Report 7 (Final Project Report) |  |  |

Table 7 - Project Deliverables

## **4. Responsibility Assignments**

D~Do; R~Review; S~Support; I~Informed; <blank>- Omitted

| **Responsibility** | **MINHTQ161122** | **THANGDMSE**  **161126** | **VINHSE160534** | **SONPHHSE171001** |
| --- | --- | --- | --- | --- |
| Project Planning & Tracking | S | D | R | R |
| Prepare Project Introduction Document | S | S | D | R |
| Prepare SRS Document (Overview Part) | R | D | S | S |
| Prepare SRS Document (User Requirements) | D | R | S | S |
| Implementation | D | R | S | S |
| Verification Type | D | R | S | S |
| Transition Stage | D | R | S | S |
| Prepare User Guides Report | D | R | S | S |
| Prepare Final Project Report | D | R | S | S |
| Prepare Thesis Presentation | D | R | S | S |

Table 8 - Responsibility Assignments

## **5. Project Communications**

| **Communication Item** | **Who/ Target** | **Purpose** | **When, Frequency** | **Type, Tool, Method(s)** |
| --- | --- | --- | --- | --- |
| Working in group | Team member | Report code  progress and  documentation  progress. Discuss issues  about requirements  from customers, and technical issues when implementing  code. Prepare demos and  questions for the next mentor and client meetings. Division of work. | 7 days/week | Online (Jira, Google Meeting),  Offline |
| Mentoring and Support | Team members and supervisor | Review work progress, including code and documentation. Answer questions related to requirements and techniques. Control project deadlines, and ensure the project runs on schedule. | 1 days/week | Online (Email),  Offline |
| Project review | Team members and stakeholders | Review work progress | 3 week | Online  Offline |

Table 9 - Project Communications

## **6. Configuration Management**

### **6.1 Document Management**

Using Atlassian-Jira and Google for storing document:

* Jira provides a Page Project that contains a topic, content, comment features to discuss problems so that we can review thereafter
* Use Jira to create tasks, bugs, reports, etc., and then we apply dynamic time for the project.

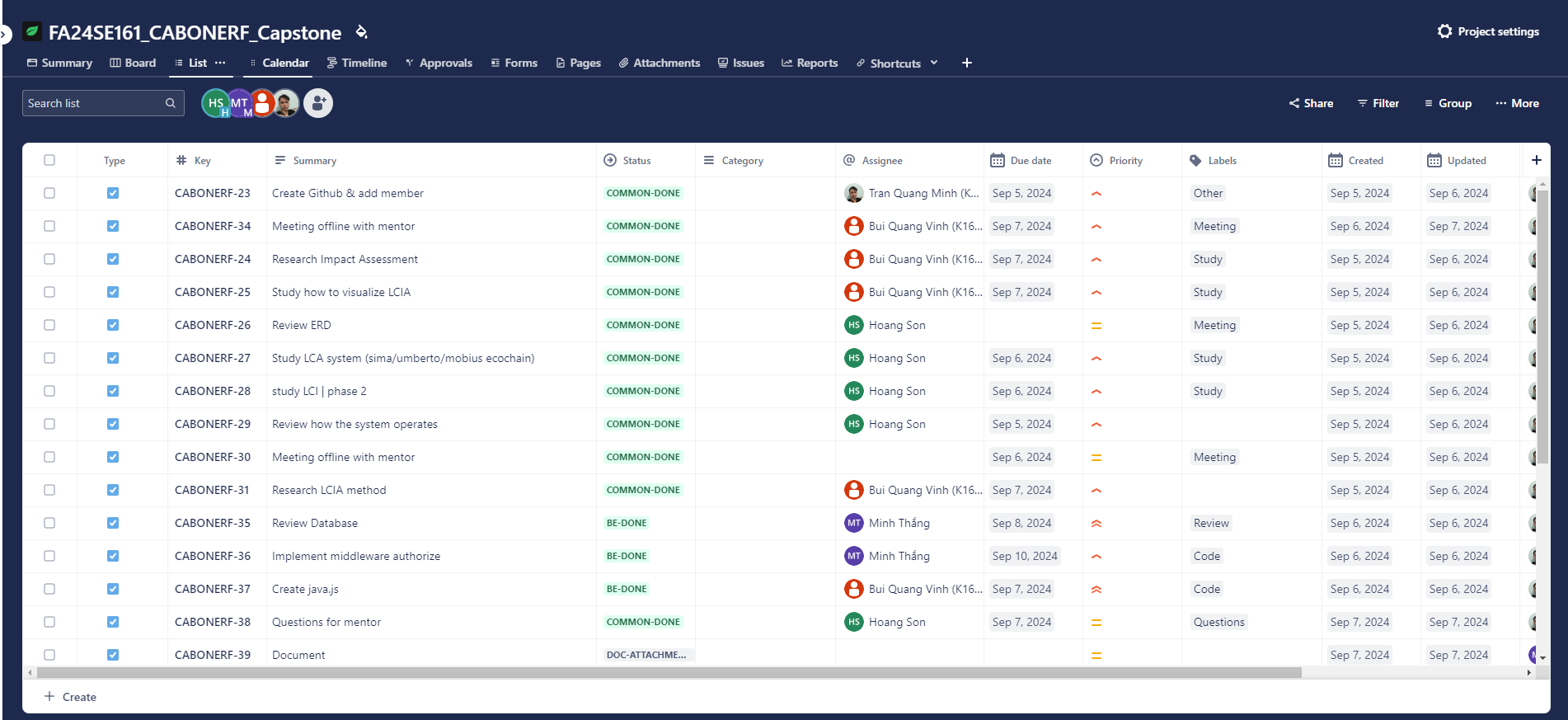


Image 1 - Jira Task

### **6.2 Source Code Management**

Using GitHub to Create an organization that covered 3 source projects for Front-end include front-end saas and front-end admin, Back-end and one for document . We can define many rules in the organization that will apply to all projects.

* In each source project, we created feature branches that will be cloned from other branches for building features. After that, we made a pull request from the features to the main branch that is deployed into the server.
* During pull request, merged codes will be sent to the server for passing pipelines and accept merge thereafter.

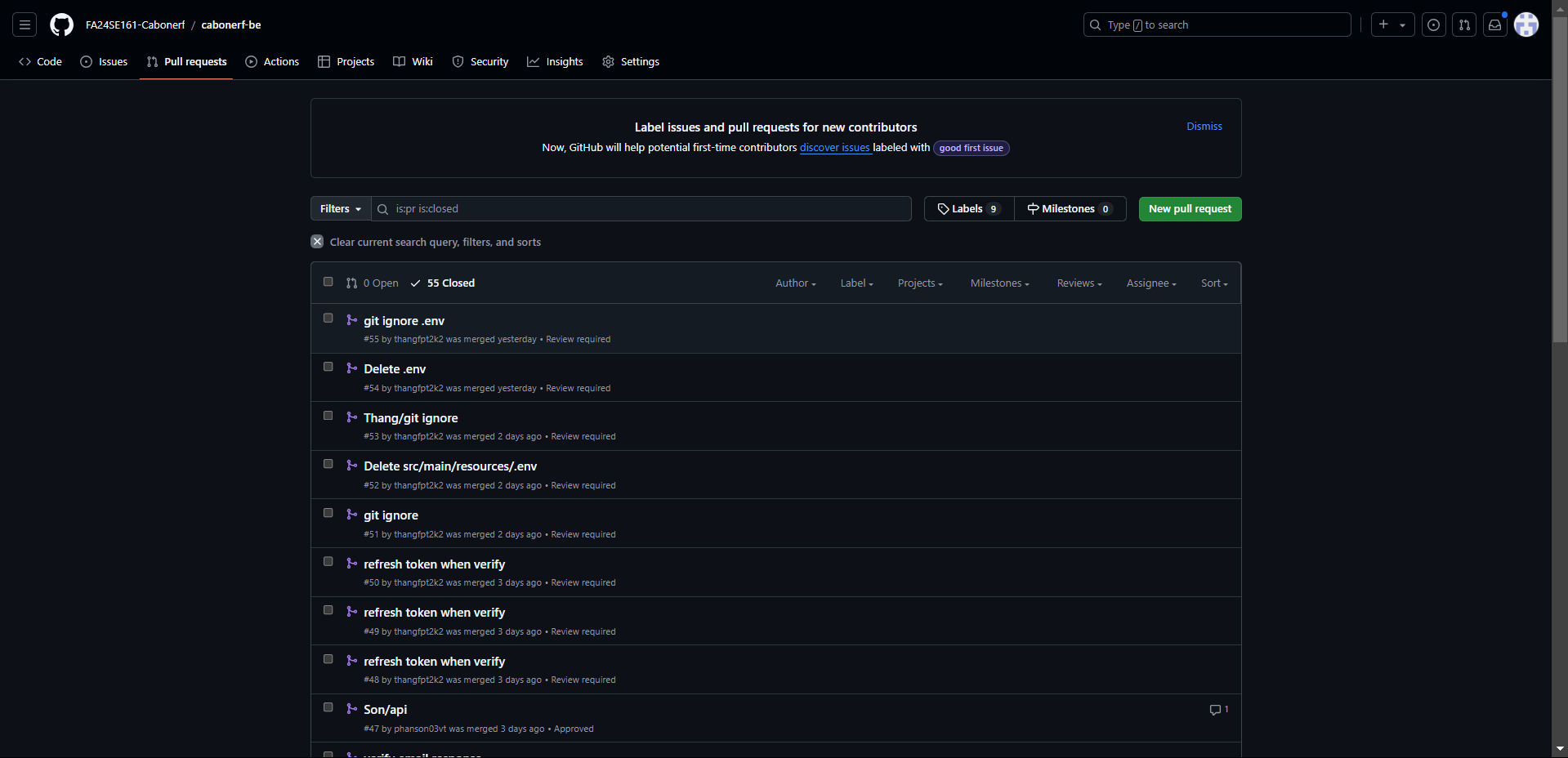


Image 2 - Github Repository

### **6.3 Tools & Infrastructures**

| **Category** | **Tools / Infrastructure** |
| --- | --- |
| **Technology** | Reactjs (FrontEnd), Tailwind Css (Front-end) shadcn (Front-end), Java/SpringBoot (BackEnd) |
| **Database** | Postgres |
| **IDEs/Editors** | Visual Studio Code, IntelliJ IDEA |
| **Diagramming** | StarUML, DrawIO |
| **Documentation** | Ms Office, Google Docs/Sheets/Slides |
| **Version Control** | Github (Source Codes, document) |
| **Deployment server** | Amazon Web Service |
| **Project management** | Atlassian-Jira (Tasks, Schedule), Github (Tasks, Defects) |

Table 10 - Tool & Infrastructures