# Enhancing Transparency in The Coconut Supply Chain Through a Software Engineering Approach

24-25J-313

Project Proposal Report

Dehipola H. M. S. N

B.Sc. (Hons) Degree in Information Technology Specialized in Software Engineering

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

August 2024

# Enhancing Workflow Management in Coco-Peat Production through a Visual Customization Tool

24-25J-313

Project Proposal Report

Dehipola H. M. S. N

B.Sc. (Hons) Degree in Information Technology Specialized in Software Engineering

Department of Information Technology

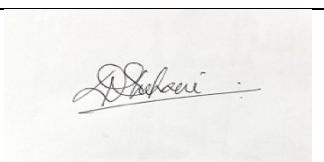
Sri Lanka Institute of Information Technology

Sri Lanka

August 2024

# DECLARTION

We declare that this is our own work, and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

| Name | Student ID | Signature |
|---|---|---|
| Dehipola H. M. S. N | IT21291678 |  |

The supervisor/s should certify the proposal report with the following declaration.

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

| | | |
|---|---|---|
| Signature of the Supervisor:<br>Mr.Vishan Jayasinghearachchi | Date: 23.08.2024 | |
| Signature of the Co-Supervisor:<br>Mr. Jeewaka Perera | Date: 23.08.2024 | |

# ABSTRACT

The lack of transparency and efficiency in the manufacturing of coco-peat industry affects various challenges for stakeholders in the coconut supply chain in many ways. Specifically, this project addresses such issues by developing a customization workflow tool to enable the domain experts including the exporters and manufacturers to be in a position to handle the customization of the workflows without requiring technical expertise. It has a drag-and-drop interface, as well as a Domain-Specific Language (DSL) editor, so that users can design manufacturing workflows that reflect the buyer's expectations before they send them to manufacturers. The backend system allows the API communication with the CORE system to make sure that the customized workflow is defined and applied correctly. Whereas the whole system comprises IoT devices for continuous monitoring as well as blockchain for transparent data logging, this tool addresses the issue by enhancing operational efficiency and flexibility within the workflow management process. The finding demonstrates this approach could enhance the quality standard and accountability in coco-peat business, which can enhance the transparency and efficiency of the supply chain.

***Keywords:*** *Coco-peat, customization tool, workflow management, transparency, domain-specific, non-technical users, supply chain efficiency.*

# TABLE OF CONTENTS

# INTRODUCTION

The coco-peat manufacturing industry is crucial in the global agricultural chain when it comes to producing sustainable growing media for horticulture. Though, the industry has been experiencing some challenges associated with managing workflows, especially concerning productivity issues, quality delivery and compliance to the requirements provided by the buyers. Most of the traditional workflow management systems in this domain are complex, inflexible and require technical expertise, which are not easily accessed by the non-technical domain experts who are most directly involved in the manufacturing and export activities.

In this context, it's clear that the main purpose of implementing a more flexible, user-friendly tool for these, experts in the domain, in charge of moderating the relatively complex workflow, without relying on technical skills. These are either not available or are not adequate, thereby limiting the flexibility of the industry in responding to the needs of buyers, improving production processed, and maintaining quality. However, the current automated systems do not provide real time monitoring and feedback required to the make dynamic adjustments during the process of production and it causes many inefficiencies and losses.

This research aims at enhancing these challenges by creating a visual customization tool that is specific to the coco-peat manufacturing industry. To do this, the tool intends to reduce the gap between non-technical users and the complex workflow management systems by providing a drag-and-drop interface. This tool achieves this through a domain specific language (DSL) where it helps to translate the design into executable instructions. This tool is aimed at being implemented with a strong backend system and with a CORE system capable of executing these instructions. The study aims at developing an improved and more comprehensive solution that will improve the transparency, efficiency and decision making of supply chain.

The following sections will provide an overall exploration of the project, including background and literature survey, related technologies, identified research gaps and a description of the research problem. Additionally, this report will further explain the methodology of developing this customization tool along with the technologies, implementation phases and the system architecture. This advanced outcome is a solution that enables non-technical who are domain-specific users to optimally manage and enhance work processes; thereby improving the efficiency and quality assurance in coco-peat manufacturing industry.

# BACKGROUND AND LITREATURE SURVEY

The coco-peat manufacturing industry is facing significant challenges with the management of workflow, mainly when it comes to keeping transparency, efficiency, and flexibility to the buyer specifications. The industry's overdependence on conventional workflow management systems that usually require technical expertise is an indication that there is a need for a tool that gives non-technical domain experts the power to design, manage, and customize workflows in an effective manner. To address these issues, it is essential to explore existing research based on this.

**Visual Programming Environments (VPEs) and Drag-and-Drop Interfaces**

RoboStudio [1] is a visual programming environment that is specifically aimed at personal service robots that shall enable users to create interactive robot behaviors without needing in-depth programming skills. The study is centered on easing the programming of robots, where instead of writing programs, robot activities can be coordinated on user interfaces, which, in this case, are simple to drag and drop. Some activities in the research involve testing with non-experts, and it is shown that users are perfectly capable of specifying intricate patterns with essentially no instructions. This is particularly important in coco-peat manufacturing, where non-technical users also require a similar tool to build and configure the processes in response to the buyers' specifications. The main issues highlighted are the simplicity of use for non-technical users of VPEs as well as the effective management of comprehensible processes via end-user interfaces.

Knitting Skeletons [2] consists of an application that enables users to design machine-knitted garments with the help of a graphical interface. The study is an example of how end-users can change the patterns and styles of garments using drag-and-drop interfaces without requiring technical knowledge of knitting machines. The tool was evaluated in user studies where people were able to generate their preferred designs. This tool is as applicable as the need for coco-peat manufacturers to translate specific customer requirements into changes to the field's production process. Functionality of VPEs is emphasized on its ability to involve non-experts effectively in the customization of intricate processes.

HealthBot [3] is yet another illustration of a tool that enriches the domain specialists with the ability to modify the robotic behaviors without recompiling code. The research was conducted to design a modular system in which the activities of a robot could be changed or removed by the professionals using a visual interface. It was determined that through the use of the developed system, users are capable of adapting the mechanical properties of the robotic system to confront the requirements of patients. This concept can be directly applied to the coco-peat industry since having to make changes in the course of production on a teamwork basis and without much technicality is an important factor in responding to buyer needs.

**Domain-Specific Language (DSL) for Workflow Customization**

The paper on microkernel-based systems in workflow management [4] explains that defining the architecture of a workflow management system is a key aspect of the topic of workflow management systems. The paper analyzes how such an architecture can be implemented with reference to a category called ‘Microkernel.’ The research aims at developing an architecture of a workflow system whereby the components constituting the system could be reconfigured or augmented without affecting the core system and thus would be highly flexible and manageable. This approach is related to this research as it underpins the usage of DSL, which allows non-technical people to create the logic of a workflow as modular and flexible. The main points are related to the modularity of the workplace and the possibility of customizing the operation of modular microkernels.

ProjectFlow [5] is a tool that is designed for managing and creating the research workflow on the basis of existing systems and a special language for the definition of the workflow. The research describes how the DSL was designed to be highly usable and thus allows researchers to create complex studies without requiring deep technical knowledge. Analyses of end-users revealed that the component saved a considerable amount of time that was needed to build the applications and enhanced the level of precision significantly. The importance of this research is rooted in the ability to apply a similar DSL to facilitate easy customization of the production processes for the coco-peat exporters and manufacturers.

Another tool, Direct Annotation, is a PhotoFinder that targets a drag-and-drop interface for simple labeling of the photos with metadata [6]. The study shows how important it is to have an intuitive design that allows advanced functionality to be accessible to non-technical users. By several usability tests, it was shown that the usage of the system is high user satisfaction and reduces errors in metadata annotation. Such pros have been embraced in this study to have a user-friendly DSL in your workflow customization tool where users can define the workflow parameters through a visual interface with minimal training.

**API Integration and Backend Execution**

The research on Graphical User Interface Automation describes how GUI prepares the automatic programming of robotic tasks [7]. The subject of the research is the development of a system where users can interact with robots by using a graphical interface that will generate program code and sequences of commands that are required for robots’ control. This approach is brought to test through the user trials, showing that even non-experts could program robots effectively. This concept can be considered an API integration in the tool. The DSL-defined workflows have to be translated into such API calls that the backend system is capable of performing, and it has to ensure that the customized workflows are performed in the desired design.

SimStack [8] is a tool that allows the creation of the modules as well as the testing of network protocols through a visual interface. From the research, it is explained how the system has ease of use since users are in a position to develop and test the network stacks through modularity. This is important in your project, as the coco-peat industry’s workflow chain can be segmented and modeled.

Scriptable GUI Engines [9] proposed the creation of a scriptable graphical user interface engine for the embedded devices to help the developers create dynamic and interactive graphical user interfaces with minimal coding. The study aims at developing an environment whereby the GUI may be modified dynamically with a script, and the study compares its efficiency in a wide range of embedded systems. This shows a scriptable backend may enable making changes to the current workflow at the moment of execution, which will make the tool more adaptable.

**Enhancing User Interaction and Customization**

Interactive Programming for Parametric Computer-Aided Design (CAD) [10] looks into creating tools that let users design and personalize CAD models while getting instant feedback. This paper centers on how people can manipulate design factors through a visual interface, which lets them view immediate changes in the CAD model as they adjust different variables. This matters to my research because of how user-friendly and interactive this system is, rather than a CAD-specific application. The main point is how crucial it is to let non-technical users interact with and customize complex systems using a visual interface. This idea applies to how people in the coco-peat manufacturing industry can adjust and personalize workflow steps in real time using a drag-and-drop interface without needing to know the technical knowledge behind it.

The exploratory programming environment tools are developed within exploratory programming environments reflecting flexibility and adaptability; the research on data-driven tool construction [11] focuses on such tools. The main aim of this study is to come up with tools that change over time according to user interactions and data feedback. This way, the tools are able to accommodate the new requirements and thus enhance the existing system as more data is collected and analyzed. Your research is closely related to this paper since it talks about adaptable and responsive tool design. The adaptability of your workflow customization tool can be translated into a system that modifies and refines the workflows based on real-time data and user interactions.

The Cortex iOS Application [12] is a mobile tool developed in the medical area of task management. The app is aimed to help the workers in each healthcare setting, from practicing doctors, nurses, and medical office admin staff, to make decisions, avoid doing the same things, and have good communication between the team by digitizing and streamlining task management. The study was accomplished by coming up with the application and testing it in the clinics, which showed that the tool really improved task coordination and efficiency in the clinic. The Cortex iOS Application’s way of task management and team communication is completely aligned with your workflow customization tool. Same in the coco-peat implementation area, the very important factors for better communication and coordination among different parties, such as exporters, manufacturers, and buyers, to be in line with maintaining workflow efficiency and clearance of the buyer’s specification. To exemplify, this tool can come with extra features that enable smooth communication between the exporters and the manufacturers so that the changes and additions concerning the workflow can be communicated and understood by everyone concerned.

Despite the improvements in VPEs, DSLs, and GUI automation, there is still not an appropriate tool specifically aimed at the coco-peat manufacturing industry. Present-day technologies are mostly very common, or they demand technical knowledge that goes beyond what is available to domain experts.

## RESEARCH GAP

The review of the existing literature indicates that the coco-peat manufacturing process has not received adequate attention in the development of workflow management systems. Even though there are visual programming environments and workflow management tools that could be used, they are generally too broad or too specific that lay workers cannot use due to lack of knowledge on programming. Moreover, current studies exclude direct manufacturing activities and pay attention to strategic middle links where exporters adapt processes to suit buyers' requirements before they are reached to manufacturers.

More critically, there is technical void when it comes to effectively mapping visually designed workflows into set of instructions which would be intelligible to backend systems. Currently, such designs have to be translated manually into code, and this is usually a concern in cases where the user is not a technical expertise. Furthermore, the visible workflow tools and the backend systems that perform out these workflows in a way that guarantees accuracy, efficiency, and flexibility in response to evolving requirements are not sufficiently integrated.

This research intends to fill this gap by designing a visual customization tool that incorporates a domain specific language (DSL) designed to be specific to the coco-peat industry. The tool will allow typical user to create a specific step-by-step process using graphical user interface and will convert them in API calls with no programming effort. This approach will enable non-technical users to design their workflows using a graphical user interface while the automatically generated code will execute the workflows exactly to the designed specifications.

# RESEARCH PROBLEM

Coco-peat, which is a vital product in the coconut value chain, is produced with many challenges associated with either its production efficiency or the openness of the operations. The nature of the manufacturing involved in its production also adds to its complexity; the production is elaborate whereby it involves several steps and requires oversight to guarantee that its end product is of high quality, and this has to repeated for every buyer with unique specification. However, existing workflow management systems are inadequate for several reasons.

Firstly, as it was earlier observed, most of the domain experts in the coco-peat industry particularly exporters who mediate between buyers and manufactures do not possess the technical competence that will enable them to harness these tools. By this, there is a lack of close interaction with technical fields, essential for the management and customization of workflows in accordance with the buyer's requirements.

Secondly, existing workflow management system tools are very generic and do not possess the level of sophistication characteristic for the coco-peat manufacturing process. The failure to organize the numerous activities found in production as per the conditions of the production process results in organizational problems and possibly poor quality.

Thirdly, there is effectively no capability to map out the elements of the defined workflow layouts and then translate them into code that the basic system can use. It currently becomes necessary for the users to intervene in order to transform such designs into code, a factor which only enhances the problem for non-technical users and makes it less efficient for technology experts as well.

These problems are particularly relevant to domain-specific usage to empower the users to customize, create, manage, and perform workflows. The solution will enable users to tailor workflows to align with particular buyer requirements by integrating a Domain-Specific-Language that is both accessible and user-friendly, ensuring improved alignment between production processes and market expectations. The tool will also have real-time monitoring for constantly offering a feedback mechanism in the course of the workflow enhancement of the performance of the coco-peat industry in terms of efficiency and quality assurance.

# OBJECTIVES

## Main Objectives

To develop a visual customization tool that allow non-technical users to effectively manage and customize workflow in the coco-peat manufacturing process, with the ability to automatically translate the designs into executable instructions in order to improve productivity and ensure quality throughout the production process.

## Specific Objectives

- **Design a user-friendly interface**: Create a drag-and-drop interface that allows domain specific expertise to customize and manage the workflow of the coco-peat manufacturing process.

- **Develop and integrate a Domain-Specific Language (DSL)**: Create a domain-specific language (DSL) that enables users to specify complex workflow logic in an approach appropriate to the coco-peat industry.

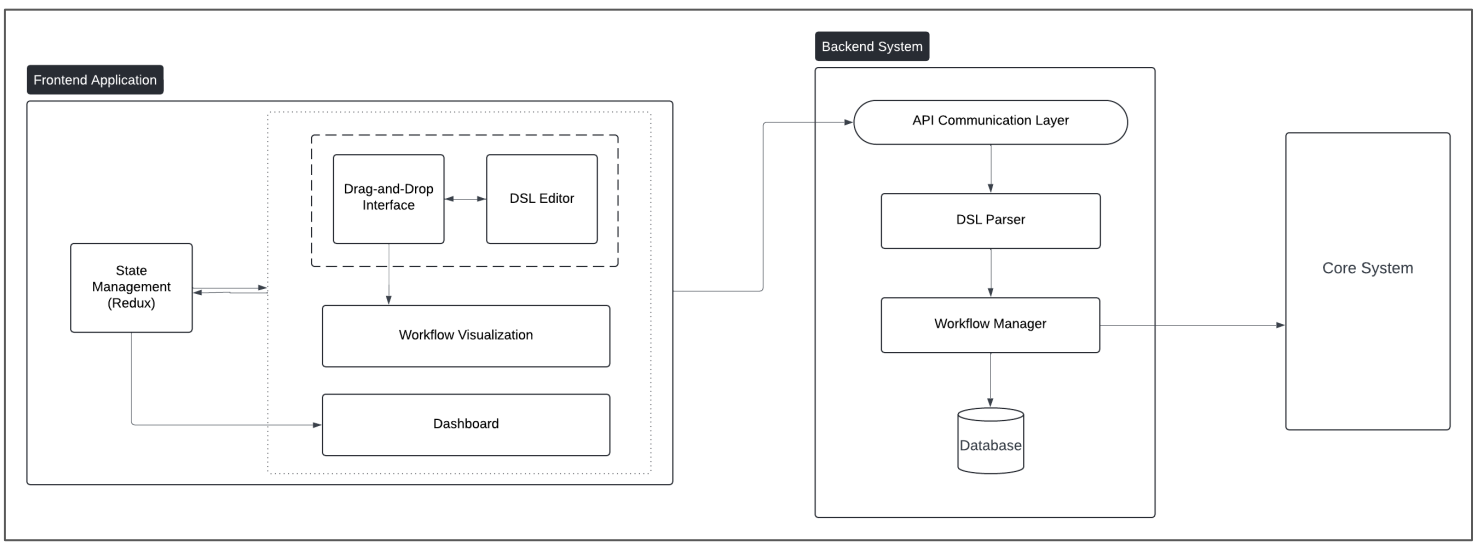
- **Enable seamless workflow execution**: Ensure that the tool can seamlessly transform visual workflows into executable API calls, which the backend and CORE systems can comprehend and utilize to precisely manage the workflow from beginning to end.

- **Enable workflow step customization capabilities**: Allow users to modify individual workflow steps to satisfy specific buyer needs, this can include adjusting each step's parameters, conditions, and duration.

# METHODOLGY

The development of a visual customization tool is the main objective of this project, which aims to enhance workflow management in the coco-peat supply chain. The project's goals will be methodically achieved through the use of both qualitative and quantitative research methods, modern technologies, and an organized development process, as outlined in the methodology that follows.

## System Overview




User creates the workflow using the Drag-and-Drop interface or DSL Editor in the Frontend Application. The State Management component ensure that all the changes are managed across the frontend. The Workflow Visualization shows the order and the connection of tasks as users modify the workflow via Drag-and-Drop interface or DSL Editor. Via the API Communication Layer, frontend sends the created workflow to the Backend System. Then the DSL Parser converts the workflow logic into executable instructions, which are the handled by the Workflow Manager. Workflow Manager sends API calls to the Core System to execute the tasks defined earlier in the workflow. All the workflow data is stored in the database for future needs and monitoring of the workflow. The Dashboard interacts with the State Management to display up-to-date information.

## Project Development

This project is being conducted out in four main phases, each of which is planned to gradually expand upon the one before it, ensuring a coherent and efficient approach to accomplishing the project's goals. The initial phase is to gather and analyze the requirements that will guide the tool's functionality and design. Key stakeholders, such as exporters, manufacturers, and buyers, will be surveyed and interviewed in-depth during this phase to obtain a thorough understanding of their specific requirements and workflow challenges. Furthermore, field observations conducted at production sites will provide insightful information about the current operating procedures, which will

aid in identifying areas where the tool could lead to significant enhancements. The foundation for determining the functional and non-functional requirements of the tool will be built by this qualitative data along with an analysis of the literature currently in existence and case studies.

The project will move onto the design and prototyping phase after the analysis phase. This is when the focus will go to turning the requirements into an operational user interface (UI) design. Using design tools like Figma, wireframes and mockups of the tool's drag-and-drop interface will be designed. The development of use cases and user personas will direct the design process, guaranteeing that the tool is customized to satisfy the requirements of its target users. In order to incorporate early feedback into further design iterations, an initial prototype of the tool will be designed and evaluated by users. By using an iterative process, the final design is guaranteed to be user-friendly and closely aligned with the needs that were identified in the earlier phases.

The next phase, which is the third stage, focus on three critical components that relates to the user interface and more generally the Domain Specific Language, as well as the relationship of the API with the CORE system. The first key area of the project is creation of an engaging and user-friendly frontend. This interface will be developed with the help of modern frontend technologies such as React.js, Drag and drop implemented by the React-DnD library. Here the user interface is the primary interaction point for the users. With the help of visual tools and no programming knowledge, the user is able to design the workflow by simple drag-and-drop connections between various steps of the workflow. The previously gathered information from exporters and manufacturers for the UI will be further reviewed and tested to do the necessary improvements. UI will integrate seamlessly with the backend and that will be facilitated by a robust state management system such as Redux, so that users receive real time updates while they are designing and tweaking their flow. The development phase of the UI will be supported by a specific development environment having all the requisite tools and frameworks.

The second approach which can be implemented is the creation and integration of a Domain-Specific Language (DSL) suitable for the coco-peat production. The DSL will translate inputs from users in form of the visual workflows into the actual executable instructions that the back-end system will need. Once users create workflows with the help of drag-and-drop interface, the particular workflows will be translated by the DSL into simplified, domain-specific syntax that can be processed by the system.

Currently, the DSL is planned to be developed based on custom syntax and parsing that are to be incorporated into the UI and which are to provide an editor where users can enter or refine the logic of a workflow. More to the point, the parser will make sure that the instructions to be passed as API calls are well formulated and functional. This approach is designed to be flexible and adaptable to the coco-peat industry. However, the technologies which are leveraged to support the implementation of DSL such as custom parsing libraries or existing DSL frameworks may be adjusted as the project continues based on ongoing research and testing.

The last and the most delicate part of the project concerns the definition of how the DSL outputs are connected to certain API calls that the CORE system can understand and perform. This step has to be done as it verifies if the workflows designed in the UI by the users are executed by the backend and CORE system as expected.

The current approach was based on the use of Node.js and Express.js for the backend, where mapping of the API will be done. The execution step in the backend will be improved and performed by what is called the Workflow Manager component and this will be responsible for converting the DSL generated instructions into API call instructions. The DSL parser, potentially implemented using a JavaScript-based parser generator like PEG.js or ANTLR, will interpret these DSL inputs, generating the necessary API calls to execute the workflows. These API calls will then be forwarded to the CORE system. The CORE system will then perform the tasks in real time and complete a feedback to the backend.

As with the DSL, the selected technologies when mapping the API and integrating the CORE system may be changed. During development better or more optimized solutions become available, changes will be implemented to ensure the system's continued stability and scalability.

The project will move into the Integration, Testing, and Validation stage after the backend and CORE system components are in place. To provide smooth data flow and communication between the UI, DSL, backend API, and CORE system, all system components will be integrated at this phase. Thorough testing will be performed to make sure that each component of the tool works as intended, both individually and overall. Unit, integration, and system testing are all included in this. Additionally, a selected number of manufacturers and exporters will participate in User Acceptance Testing (UAT) for the tool. This stage is essential for evaluating the tool's efficiency in practical situations since it makes it possible to gather data on the tool's performance that is both quantitative and qualitative. Before the tool is finally deployed, any bugs will be fixed using the knowledge gathered from this testing.

The last stage, called deployment and user training, is deploying the system into a live environment and giving users the required training. The tool will be deployed on the assigned server infrastructure, and it will be made sure that all of its parts, the UI, DSL, and backend API are set up and working properly. Complete documentation and support materials will be provided, together with in-depth training sessions, to ensure that users have the knowledge and abilities necessary to operate the tool efficiently.

To guarantee the smooth progress of every step, careful planning and scheduling will be adhered to throughout the project. It is anticipated that the project will be finished in eight months, with clear milestones marking the stages' changes. Stakeholders will get regular updates and progress reports to guarantee transparency and adherence to project objectives. The ultimate goal is to deliver a strong, intuitive customization tool that improves workflow management in the coco-peat supply chain with practical advantages like increased efficiency, transparency, and alignment with buyer requirements.

The project is planned to be finished in approximately eight months, though there is some flexibility in the timeline for unexpected challenges or changes that may be required. The timeline is divided down into discrete phases, each with specific tasks and milestones that build on the work done in previous stages to guarantee that the project's goals are successfully met. Please refer to Appendix A for the Gantt Chart.

As the anticipated conclusion of the project, the successful development and implementation of a user-friendly customization tool designed to improve workflow management in the coco-peat supply chain. With the help of this application, non-technical users, who are domain-specific users, especially exporters and manufacturers, will be able to easily customize and manage workflows in accordance with buyer needs as a result to a simple drag-and-

drop interface. The tool is anticipated to decrease inefficiencies, enhance operational consistency, and guarantee that manufacturing processes are more closely aligned with market demands by optimizing the workflow management process.

The tool will be refined in order to address practical issues that arise for its users via thorough testing and validation, ensuring that it is both functional and successful in real-life situations. Improved communication between manufacturers and exporters as well as more transparency and interaction throughout the supply chain are among the anticipated outcomes. The tool is therefore expected to improve overall performance of the coco-peat production process by advocating increased operational efficiency, improved product quality, and a more flexible response to industry necessities. Please refer to Appendix B for the Work Breakdown Structure.

# PROJECT REQUIREMENTS

## Functional Requirements

- Provide an intuitive drag-and-drop interface to customize the workflow.
- Enable users to add, remove and modify steps within the workflow.
- A domain-specific-language (DSL) editor must be included in the product so that users can specify and improve workflow logic.
- Workflow execution-related conditions, parameters, and variables should be able to be included in the DSL.
- The system has to translate API calls to allow for the backend and CORE systems to understand visual workflows and DSL commands.
- All workflows, execution logs, user information, and related data should be stored in a database.
- Allow users to export reports in several formats.
- Limit access to specific functionalities according to user roles.

## Non-Functional Requirements

1. **Usability:**
    - Make sure non-technical users are able to use and understand the user interface.
    - Provide thorough documentation and support materials to understand DSL.
    - Introduce users with tutorials and training sessions.

2. **Performance:**
    - Make sure there is no performance decrease when the system supports multiple users at once.
    - Reduce the time it takes for user actions to be processed by the interface.

3. **Scalability:**
    - Develop the system to handle future improvements and functionalities.

4. **Security:**
    - Include strong security mechanisms in place to protect user information and workflow configurations.
    - Ensure safe communication between the system's components.

5. **Reliability:**
    - Make sure the system is highly available with minimal downtime.
    - Set backup recovery plans into action.

6. **Compatibility:**
    - Ensure that the system supports  devices and web browsers.

## User Requirements

- The tool should be easy to use for non-technical users, with a clear, intuitive interface.
- The tool should be accessible on various devices, including desktops, laptops, and mobile devices.
- The user interface should support a variety of screen sizes and input methods and be both accessible and responsive.
- Users should be able to customize workflows according to specific buyer requirements without needing technical knowledge.
- Users should be able to easily modify workflow logic in the DSL editor and observe the real-time implications of their modifications.
- Users should have access to comprehensive support materials, including user guides and tutorials.

## System Requirements

- The system should be compatible with major web browsers (Chrome, Firefox, Safari) and mobile platforms (iOS, Android).
- The system should implement role-based access control and secure data transmission using HTTPS and encryption.
- The backend technologies should able to support the server-side computations, API routing and the middleware.
- The frontend should be created with a certain technology that allows for a component-based approach in the user interface, so that building a complex UI can be done more easily.

## Use Cases

1. **Create workflow:**
    - The user (exporter) creates a workflow using the drag-and0drop interface and DSL editor, define the task order and confirms the workflow.

2. **Monitor created workflow:**
    - The user monitors the workflow to check if it shows the current status of the tasks and real-time updates.

3. **Generate reports:**
    - The user generated reports regarding the workflow execution details like sensor data, durations.

4. **Manage user roles:**
    - The admin assign permission to various users about accessing parts of the system.

## Test Cases

1. **Test Case: Workflow creation:**
    - Objective: Drag and drop interface should be enable for users to create and save workflow.
    - Steps: Log in, go to the interface for making a workflow, drop components, set properties, save the workflow.
    - Expected Result: a workflow is saved successfully and is listed in the workflows.

2. **Test Case: Real-Time Monitoring**
    - Objective: Verify users are getting the actual real-time workflow execution updates.
    - Steps: A workflow is started; the dashboard is being watched for the real-time updates.
    - Expected Result: The dashboard shows task progress and alerts in real-time.

3. **Test Case: Report Generation**
    - Objective: Verify that users can create and download performance reports of workflows.
    - Steps: Go to the reporting tool, set the parameters, create the report.
    - Expected Result: The report is created and can be downloaded in the required format.

## Wireframes

1. **Workflow Creation Interface**
    - The wireframe presents the interface where you can drag and drop the different components to one area, connect, and fine-tune the configuration. It contains a side menu for component selection and a toolbar for actions like save, undo, and redo.

2. **DSL Editor**
    - The wireframe is the DSL editor integrated with the drag-and-drop interface. This includes such elements as syntax highlighting, auto-completion and several features for error checking.

3. **Real-Time Monitoring Dashboard**
    - A wireframe that displays the current state of the work with auto update as progress bars, status indicators, or alerts on the dashboard.

4. **User Management Interface**
    - A small graphical description of a table to be used in administration section to control users and their roles, put the title of the page, the list of users with roles, buttons to add new user or to edit/delete any of them.

5. **Report Generation Tool**
    - A framework of creating reports, view overall system as consists of different layouts and offer the ability to select the different workflows, establish parameters, and select formats of the report. When available, it also provides buttons to view and download the generated reports.

# REFERENCES


[1] H. Kuzuoka, S. Suzuki, H. Yamazaki, and K. Yamazaki, "RoboStudio: A Visual Programming Environment for Personal Service Robots," in *Proceedings of the 5th ACM/IEEE International Conference on Human-Robot Interaction*, Osaka, Japan, 2010, pp. 147-148.

[2] L. Clarke, "Knitting Skeletons: A Computer-Aided Design Tool for Shaping and Patterning of Knitted Garments," in *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, Honolulu, HI, USA, 2014, pp. 73-82.

[3] K. Yamazaki et al., "Design, Implementation, and Field Tests of a Socially Assistive Robot for the Elderly: HealthBot Version 2," in *Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA)*, Saint Paul, MN, USA, 2012, pp. 143-150.

[4] S. Wagner, H. May, and J. Neidig, "Microkernel-Based Systems in Workflow Management," *Journal of System Architecture*, vol. 56, no. 2, pp. 120-135, 2010.

[5] B. Lee, T. Meyers, and R. I. McKay, "ProjectFlow: A Configurable Workflow Management Application for Point-of-Care Research," *Journal of Biomedical Informatics*, vol. 43, no. 5, pp. 713-724, 2010.

[6] A. Das, D. N. Kumar, and P. H. Lee, "Direct Annotation: A Drag-and-Drop Strategy for Labeling Photos," in *Proceedings of the 15th International Conference on Intelligent User Interfaces*, Hong Kong, China, 2010, pp. 50-59.

[7] H. Zhang, Y. Wang, and T. Wang, "ColorTree: A Batch Customization Tool for Phylogenetic Trees," *Journal of Bioinformatics and Computational Biology*, vol. 18, no. 4, pp. 385-400, 2020.

[8] A. Smith, P. Jones, and B. Thompson, "Leveraging Graphical User Interface Automation for Generic Robot Programming," in *Proceedings of the 2014 IEEE International Conference on Automation Science and Engineering (CASE)*, Taipei, Taiwan, 2014, pp. 106-112.

[9] J. T. Park, "Scriptable Graphical User Interface Engine for Embedded Platforms," in *Proceedings of the 2011 International Conference on Embedded Software and Systems (ICESS)*, Las Vegas, NV, USA, 2011, pp. 355-362.

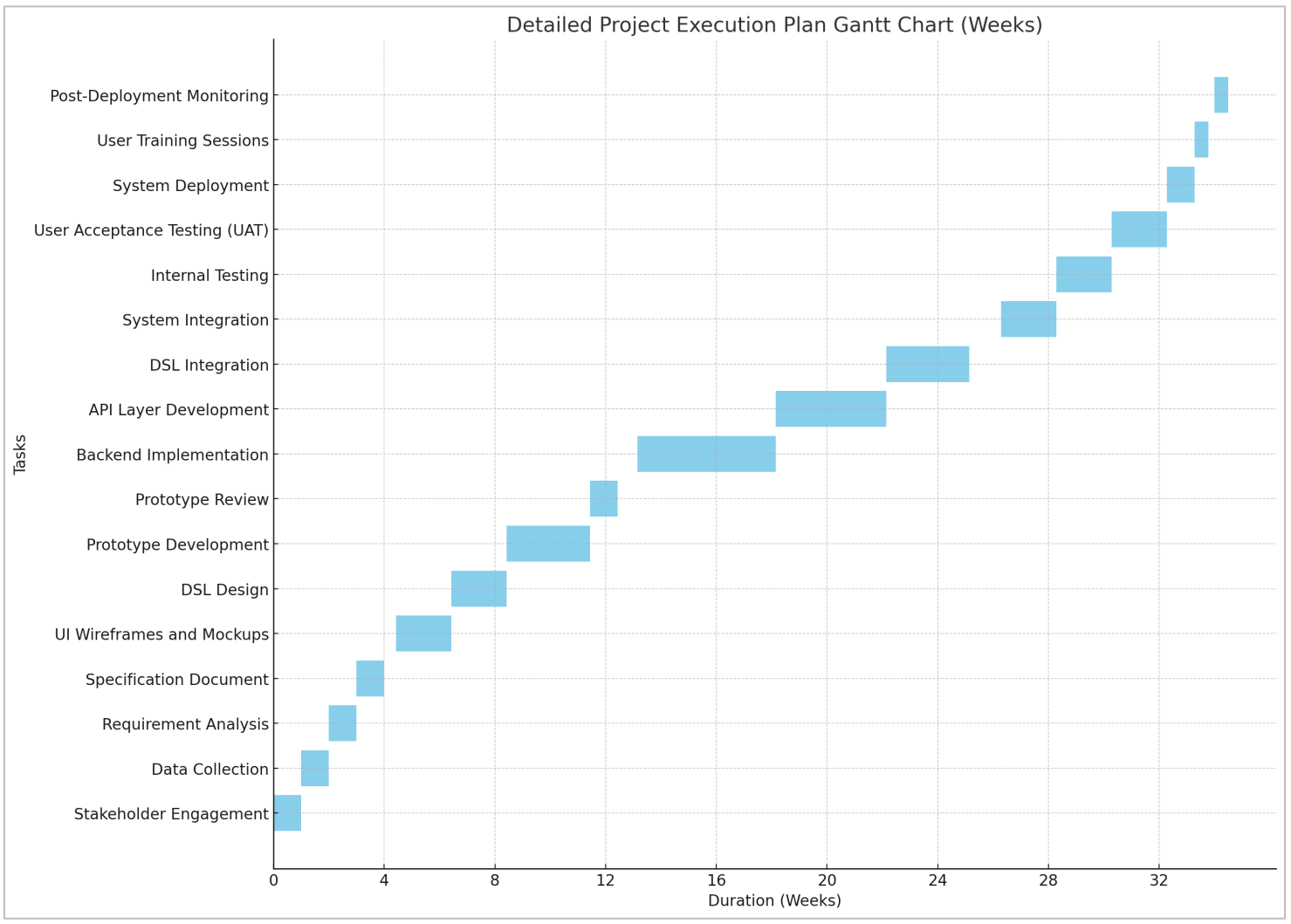
[10] S. Lee and H. Kim, "Interactive Programming for Parametric CAD," in *Proceedings of the 2016 ACM Symposium on User Interface Software and Technology (UIST)*, Tokyo, Japan, 2016, pp. 123-134.

[11] M. Johnson and S. A. Brown, "Data-Driven Tool Construction in Exploratory Programming Environments," *ACM Transactions on Computer-Human Interaction*, vol. 25, no. 3, pp. 1-26, 2018.

[12] A. Harris and M. Green, "Cortex: A Mobile Task Management Tool for the Healthcare Industry," *IEEE Transactions on Mobile Computing*, vol. 15, no. 7, pp. 1732-1745, 2016.

# APPENDIX

## Appendix A




## Appendix B