AI Powered 3D Process Simulation & Visualization

Group Members

Abdur Rafay (01-131222-005)

Huzaifa Ahmed (01-131222-056)

Supervisor: Dr Tamim Ahmed Khan

A Final Year Project submitted to the Department of Software Engineering, Faculty of Engineering Sciences, Bahria University, Islamabad in the partial fulfillment for the award of degree in Bachelor of Software Engineering

May, 2026

FYP COMPLETION CERTIFICATE

Student Name: __Abdur Rafay_____ Enrolment No: _01-131222-005__

Student Name: __Huzaifa Ahmed_____ Enrolment No: _01-131222-056__

Programme of Study: Bachelor of Software Engineering

Project Title: _AI-Driven 3D Simulation and Organizational Training Platform_

It is to certify that the above students' project has been completed to my satisfaction and to my belief, its standard is appropriate for submission for evaluation. I have also conducted plagiarism test of this thesis using HEC prescribed software and found similarity index at _____ that is within the permissible limit set by the HEC. I have also found the thesis in a format recognized by the department.

Supervisor's Signature: _____

Date: ____16, April 2026_____ Name: _____Dr Tamim Ahmed Khan_____

CERTIFICATE OF ORIGINALITY

This is certify that the intellectual contents of the project AI-Driven 3D Simulation and Organizational Training Platform are the product of my/our own work except, as cited properly and accurately in the acknowledgements and references, the material taken from such sources as research journals, books, internet, etc. solely to support, elaborate, compare, extend and/or implement the earlier work. Further, this work has not been submitted by me/us previously for any degree, nor it shall be submitted by me/us in the future for obtaining any degree from this University, or any other university or institution. The incorrectness of this information, if proved at any stage, shall authorities the University to cancel my/our degree.

Name of the Student: ___Abdur Rafay Minhas_____

Signature: _____ Date: __16, April 2026____

Name of the Student: __Huzaifa Ahmed_____

Signature: _____ Date: __16, April 2026____

PROJECT TITLE (MENTION PROJECT TITLE HERE)

Sustainable Development Goals

(Please tick the relevant SDG(s) linked with FYDP)

SDG No	Description of SDG	SDG No	Description of SDG
SDG 1	No Poverty	SDG 9	Industry, Innovation, and Infrastructure ✓
SDG 2	Zero Hunger	SDG 10	Reduced Inequalities
SDG 3	Good Health and Well Being	SDG 11	Sustainable Cities and Communities
SDG 4	Quality Education	SDG 12	Responsible Consumption and Production
SDG 5	Gender Equality	SDG 13	Climate Change
SDG 6	Clean Water and Sanitation	SDG 14	Life Below Water
SDG 7	Affordable and Clean Energy	SDG 15	Life on Land
SDG 8	Decent Work and Economic Growth	SDG 16	Peace, Justice and Strong Institutions
		SDG 17	Partnerships for the Goals

Range of Complex Problem Solving			
	Attribute	**Complex Problem**	
1	Range of conflicting requirements	Involve wide-ranging or conflicting technical, engineering and other issues.	✓
2	Depth of analysis required	Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models.	
3	Depth of knowledge required	Requires research-based knowledge much of which is at, or informed by, the forefront of the professional discipline and which allows a fundamentals-based, first principles analytical approach.	
4	Familiarity of issues	Involve infrequently encountered issues	
5	Extent of applicable codes	Are outside problems encompassed by standards and codes of practice for professional engineering.	
6	Extent of stakeholder involvement and level of conflicting requirements	Involve diverse groups of stakeholders with widely varying needs.	✓
7	Consequences	Have significant consequences in a range of contexts.	
8	Interdependence	Are high level problems including many component parts or sub-problems	✓
Range of Complex Problem Activities			
	Attribute	**Complex Activities**	
1	Range of resources	Involve the use of diverse resources (and for this purpose, resources include people, money, equipment, materials, information and technologies).	
2	Level of interaction	Require resolution of significant problems arising from interactions between wide ranging and conflicting technical, engineering or other issues.	✓
3	Innovation	Involve creative use of engineering principles and research-based knowledge in novel ways.	✓
4	Consequences to society and the environment	Have significant consequences in a range of contexts, characterized by difficulty of prediction and mitigation.	
5	Familiarity	Can extend beyond previous experiences by applying principles-based approaches.	✓

Abstract

Crystal System CMS is a web-based enterprise management platform designed to centralize organizational, operational, and process-related data within a single digital environment. For all those entities seeking to enhance operational efficiency at all stages, from companies and countries to cities, buildings, levels, rooms, people, positions, employees, roles, occupations, operations, operations management, key performance indicators, etc., crystal system CMS becomes the perfect solution. Crystal System CMS will make sure that all these complex systems become more efficient. All that thanks to the presence of up-to-date React technologies and such API services as authentication, authorization, audit log, data import/export, dashboards, etc., capable of improving decision-making processes. One of the essential features of the software is process optimization, including such tools as BPMN visualization, workflow, what-if analysis, AI functions, which allows measuring and optimizing the efficiency of performance/resource management. This ensures greater productivity and efficiency of management.

Keywords: Crystal System CMS, Content Management System, Enterprise Management, Business Process Management, BPMN, Workflow Automation, KPI Management, Role-Based Access Control, Audit Logs, Asset Management, Task Management, Job Management, Process Optimization, What-If Analysis, React Application, Dashboard Analytics, Organizational Hierarchy, User Management, Facilities Management, Digital Transformation

Dedication

WE DEDICATE THIS FINAL YEAR PROJECT TO OUR PARENTS, FAMILIES, TEACHERS, AND MENTORS WHOSE CONSTANT SUPPORT, ENCOURAGEMENT, AND PRAYERS MADE THIS JOURNEY POSSIBLE. THEIR GUIDANCE, PATIENCE, AND BELIEF IN OUR ABILITIES INSPIRED US TO REMAIN COMMITTED THROUGHOUT THE DEVELOPMENT OF THIS PROJECT.

WE ALSO DEDICATE THIS WORK TO OUR FRIENDS AND CLASSMATES WHO MOTIVATED US DURING DIFFICULT STAGES AND PROVIDED VALUABLE SUPPORT THROUGHOUT OUR ACADEMIC JOURNEY. FINALLY, WE DEDICATE THIS PROJECT TO EVERYONE WHO ENCOURAGED US TO LEARN, GROW, AND STRIVE FOR EXCELLENCE IN THE FIELD OF TECHNOLOGY.

Acknowledgments

Glory and praise be to God who made it possible for me to carry out this research project with patience and perseverance.

I express my deep sense of gratitude to Dr. Tamim Ahmed Khan who acted as my project guide and helped me throughout the project with his valuable inputs and suggestions which have enhanced the quality of my research project.

My special thanks go to all my teachers and faculty members for imparting me the requisite skills and knowledge for completing my studies. They have rendered a great deal of assistance for me in achieving my objectives.

I also wish to express my gratitude to my parents and family members who have been praying for my success and have supported me in every way possible. Their blessings have always motivated me.

I am also thankful to all my friends and colleagues for lending me their assistance during the completion of this project. They have encouraged me in every manner possible during difficult times.

I would like to thank each and everyone who was responsible for completing this project in any way.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

The quick evolution of information technologies has revolutionized the way organizations conduct themselves when dealing with their information requirements and organizational processes. Using inefficient and outdated methods, along with using separate software packages, might lead to inefficiency, redundancy, confusion, and an overall lack of comprehension on how the organization is performing. Organizations in today's environment need a system where they can centralize their information and improve their processes.

The Crystal System CMS is the product that appeared as an answer to the demands outlined above. This system is implemented in the form of internet-based software for company management. It contains some organizational tools that help organize organizational features such as management of organizations or locations, management of processes and tasks, jobs management and other numerous aspects. In addition, it allows completing some tasks connected with user access management, asset task management, KPI tracking, auditing and workflow modeling.

In this chapter, it is possible to trace the logic of the project implementation, as well as the goals pursued by this work and the contribution that has been achieved.

1.1 Motivation

In modern times when everything moves quickly, efficiency, reliability, and speed in accessing information are crucial to gaining an edge over competitors. Many businesses utilize multiple independent systems for managing their organizational processes in the forms of departments, personnel, tangible assets, jobs, work flows, and so on. Poorly organized work causes poor coordination among involved parties, a lack of responsibility, inconsistencies in information, and difficulties in evaluating performance within the organization as a whole.

The second issue related to the necessity of process management lies in the fact that it is rarely formally documented. Therefore, it is difficult to determine interdependencies between work processes and responsibility for particular tasks. Furthermore, it is hard

to diagnose weak links in the process and to control them effectively, which is crucial for optimal resource utilization and goal achievement.

The reason why I chose the Crystal System CMS project is that there exists the possibility of creating one platform that would help with administrative and functional issues related to any type of business organization. In order to do this, it is not necessary to use several different applications such as user management, process models, key performance indicators (KPI), facilities management, among others. The goal of this project is therefore to combine all of these elements within a single solution. This project was chosen because of the need for automated workflows, RBAC security, real time reporting, and process improvement solutions (such as BPMN and what-if analysis).

In terms of education, this project provides an ideal chance to apply the knowledge learned in classes related to software engineering, web development, database management, user interface design, among others.

1.2 Objectives

The main objective of this research project is to design and deploy an integrated web-based management solution that will boost efficiency, transparency, and control in organizational activities.

Towards this end, the major goals of this research study are to:

1. Design a centralized CMS that can control organizational aspects including firm, country, city, structure, floor, room, individual, user, role, job, activity, and procedure in one element.
2. Deploy mechanisms that will help in controlling organizational business procedures via users' capacity to create workflows, represent them using BPMN, link activities to procedures, and optimize the execution of work processes.
3. Ensure security and intelligent management features such as user log-in, role management, audit logs, dashboard reporting, KPIs, and import/export data.
4. Apart from the aforementioned goals, other project goals are meant to improve administration effectiveness and facilitate decision-making.

1.3 Main Contributions

Some of the major contributions made by the Crystal System CMS towards developing an enterprise system are as follows:

First, the project provides a full management system that combines many different areas of study within the area of business management into one platform. The system does not view organizational management, facility management, process management, and process evaluation as separate functions but rather integrates them into one system.

Second, the project enhances the capabilities of organizations regarding their processes through workflow diagramming based on BPMN, process creation, linking processes to tasks, and scenario analysis. It is not only possible for organizations to store information about their processes but also evaluate them and improve them.

Benefits and Relevance of the System

Third, role-based access and auditing make it possible to control users' activities. As a result, the appropriate users only would get access to the sensitive parts of the system, and each user activity would be recorded for auditing and compliance reasons.

Fourth, by providing a dashboard, KPIs management, and reports capabilities, the developed solution would become a useful instrument of analysis of organizational performance. It will help to learn about performance and work of different organizations, business processes, and positions.

Fifth, by applying the proposed project, it will be shown how up-to-date frontend tools could be used in order to create a flexible and maintainable enterprise solution. Being based on a component architecture and SOA principles, the proposed solution was created with future scalability in mind.

Thus, summarizing the discussion above, it should be noted that the considered project is particularly important, as it would contribute to creation of the platform, which would ensure effective data collection, visualized processes, coordinated resources, and improved management performance.

1.4 Report Organisation

The present report has been subdivided into several chapters that focus on various aspects of the project.

Chapter 1 presents an overview of the project in question, which can be done by discussing background, motivation, objectives, contribution of the report, and its structure.

Chapter 2 gives the explanation of literature review and background study. The chapter explains definitions of such concepts as content management system, enterprise management system, business process management, BPMN, workflow automation, access control, dashboard system, and application of advanced technological solutions in today's organizations.

Chapter 3 discusses requirements analysis and design of Crystal System CMS. Requirements both functional and non-functional, design diagrams such as use cases, flows and entity relationships, and description of databases and modules are provided in the chapter.

Chapter 4 discusses the implementation of the system. In this chapter, the development environment and implementation procedures, such as modules and API integration, are discussed.

Chapter 5 addresses issues of testing and evaluation. Chapter five provides information on the testing techniques utilized to evaluate the performance, security, functionality, and usability of the proposed software system. In addition, Chapter five highlights the results that were achieved from testing and the success of the proposed solutions.

Chapter 6 addresses Conclusion and Future Work highlights the conclusion and potential future work areas. This chapter outlines the achievements made by the research study.

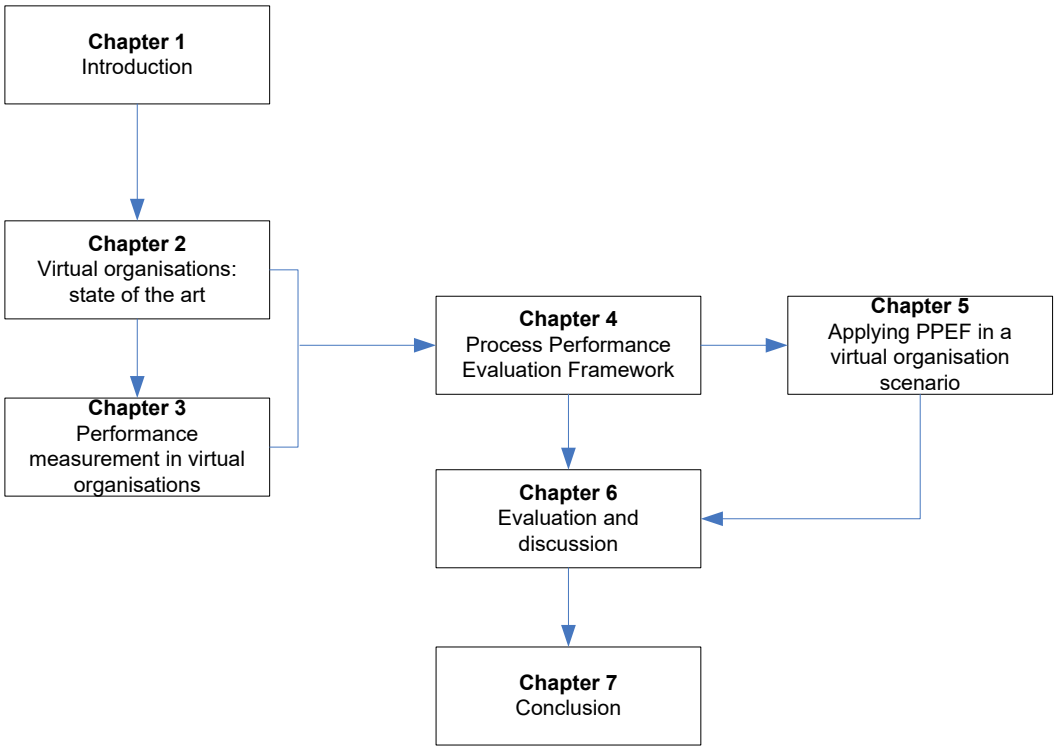


Figure 1.1: Thesis organisation

1.5 Scope of the Project

The scope of the Crystal System CMS includes the development of a web-based management solution for enterprises. The scope covers the management practices inside an organization and includes such modules as organization structure, personnel management, facilities management, process and workflow management, KPIs monitoring, user management, rights management, and audit trails.

The major emphasis made on the project is on the development of frontend solutions for workflow management using backend APIs. The purpose of the system is to provide the users with secured access to the system, convenient navigation within the processes, centralized processing of data, and transparency of processes. Even with all the sophisticated features such as process optimization and management through artificial intelligence technology, the scope is limited only to workflow management.

1.6 Chapter Summary

Crystal System CMS has been described in this chapter, and also the necessity of creating such a project has been discussed here. Objectives to be fulfilled through this

project have been highlighted in this chapter. The contribution of this project has also been discussed in this chapter. The structure of this report has also been described in this chapter.

Chapter 2
Background Study/Literature Review

The following chapter will examine the background, existing software tools, and relevant literature that could be used during the development of the Crystal System CMS. The chapter aims at providing the necessary background for the entire research paper as well as demonstrating the need for an integrated enterprise management system. Given the fact that the Crystal System CMS includes organization management features, workflow management, BPMN-based process visualization, performance management based on KPIs, security, and auditing capabilities, the literature review chapter will focus on the above-mentioned issues.

2.1 Key Concepts

2.1.1 Enterprise Management Systems

Enterprise management systems pertain to computer-based frameworks that help integrate all the different business processes into one system. Nowadays, businesses do not use separate frameworks for HR, business process management, facility management, workflow management, reporting, and user management because they are interested in using one tool that helps manage processes effectively, maintain uniformity, and improve efficiency. While tools such as ERP, BPI, and workflow management help integrate processes, there are many companies that face issues with the integration of their processes and data.

As far as the Crystal System CMS is concerned, enterprise management would encompass the management of companies, sites, buildings, floors, rooms, users, positions, tasks, and processes from the same dashboard. Such integration is necessary since there is some kind of link between different components of a company. Thus, the process must have a company attached to it; it has to have certain tasks and users/positions attached to it as well as appear on dashboards and logs.

2.1.2 Business Process Management and BPMN

BPM (Business Process Management) is an approach focused on designing, executing, controlling and optimizing the business processes. Modern research on BPM reveals

that today process management can not be considered simply as documenting the existing processes. Instead, process management involves continuous work on discovery, modeling, execution, monitoring, analysis and optimization of business workflows. Recent trends in BPM lifecycle research point to the need to concentrate more on the process improvement rather than on merely defining the processes.

One of the major standards in business process management field is BPMN (Business Process Modeling Notation), an open-source standard supported by OMG (Object Management Group). BPMN's key feature is a unified notational system, which includes flow objects (events, activities, gateways), connecting objects (sequence flows, associations, and message flows) and swimlanes, specifying the participants in the process. The greatest strength of BPMN consists in using a common language that is used by both business analysts and IT professionals. This makes BPMN a powerful tool for the Crystal System CMS deployment.

However, using BPMN by itself is not sufficient to solve all the problems within an organization. While the diagram might represent the process, it doesn’t mean that permissions, corporate structure, KPIs, or auditing will take place by themselves. That’s why Crystal System CMS integrates BPMN into a much broader corporate management system.

2.1.3 Role-Based Access Control

The security aspect is one of the essential prerequisites for any enterprise system as each user needs a specific set of permissions. One of the most popular models used for authorization in enterprise systems is Role-Based Access Control (RBAC). The concept of RBAC was first introduced by Ferraiolo and Kuhn. According to their suggestions, RBAC can be used as an alternative to traditional access control mechanisms almost universally in all businesses. Later, the concept of RBAC was further developed in the NIST RBAC model, which categorized the architecture into different types, such as flat, hierarchical, and constrained.

One of the advantages of RBAC is that the permissions are separated from individual users. Instead, they belong to particular roles that have certain privileges. Thus, the

whole system becomes significantly easier to manage, implement, and scale, especially when there is a need to expand the company in the future. In Crystal System CMS, RBAC plays a pivotal role as there are features that should be accessible exclusively for some employees, such as KPI management, auditing, and other settings.

2.1.4 Audit Logging and Accountability

The limitation of access alone does not suffice; it has to be followed up with logging of activity. The need for logs comes from the critical importance of operation monitoring, investigations, accountability, and compliance. According to the guidelines on logging established by NIST, logging in a secure environment needs to be treated not as an additional component, but as one of the basic instruments of operational control of the organization.

Audit capabilities in Crystal System CMS allow the organization to monitor logins, user actions performed, deletions, and other actions made, and general user behavior. In case of multiuser systems, the importance of audit functions grows, as their activity can change the information architecture, workflows, and policies on permissions used. The most valuable aspect of employing audit functions is their contribution to higher accountability levels and ease of performing investigations. However, poor implementation of audit functions can generate excessive background noise, be deprived of filtering capabilities, and violate the user's right to privacy. That is why audit functionality has to be properly structured and related to the user role.

2.1.5 Dashboards, KPI Monitoring, and Decision Support

Another example of an informative tool would be a dashboard, which provides an opportunity to illustrate the performance of the organization through visualization of critical metrics. According to Yigitbasioglu and Velcu-Laitinen, the effective use of dashboards is possible when there is no information overload, there are opportunities for analysis of particular metrics, and there is enough data available to a user. The improperly designed dashboard does not provide any help and only pleases the owner with its design.

As far as KPIs are concerned, they should be connected with the aims set by the organization. Monitoring of KPIs requires using dashboards in business systems, which makes it possible to monitor the performance and resolve arising problems. Three primary elements of the Crystal System CMS include general metrics, managerial metrics, and KPI modules. Importantly, it should be noted that this concept relies on the results obtained and showing that relevance, flexibility, and functionality represent the true nature of dashboards.

2.1.6 Process Analysis, Simulation, and What-If Evaluation

But nowadays process management takes other directions rather than just creating diagrams and tends to concentrate on analysis and optimization of processes. Indeed, according to the Process Mining Manifesto process mining is defined as analysis and optimization of processes according to event data. Besides, according to recent researches that examined process simulation, decisions about the processes become much more effective if processes are modeled in combination with historical and current data.

It concerns the current project devoted to Crystal System CMS, which involves visualization of processes, their optimization, as well as "what if" analysis. Though the current project does not provide all instruments of process mining, there is a scientific foundation proving the necessity of evolution from process description to measurement of the process improvement. That is exactly what takes place in this project since it gives a possibility to optimize processes rather than just describe them.

2.2 Existing Products and Software Applications

2.2.1 Odoo

According to the official description, Odoo can be referred to as an enterprise solution incorporating different applications such as CRM, accounting, inventory management,

project management, etc. The primary strength of Odoo lies in its modularity, where multiple aspects of a firm are connected. Hence, Odoo would be recommended to companies aiming at adopting an enterprise solution.

However, in contrast to Odoo, Crystal System CMS focuses on BPMN process modeling, workflow visualization based on audit, and organization structuring. Besides, Crystal System CMS focuses on processes more than Odoo does. Moreover, Odoo might be regarded as excessive for certain tasks.

2.2.2 SAP Signavio

Signavio Process Intelligence from SAP focuses on process mining, analysis, and optimization. This software comes with features like process data management, process connectors, process pipelines, and process analysis. It is the best choice for those looking for process optimization.

The strength of Signavio Process Intelligence is based on its strong capabilities for process intelligence. However, as compared to this particular project, it lacks a feature which can act as a customizer for organizational management purposes. Without other modules, this software alone will not be able to replace the requirement for building, room, people management, user administration, and operational logging modules. From its documentation, it is evident that it is focused on process intelligence and not all aspects of the organization.

2.2.3 Appian

Appian positions itself as an artificial intelligence-powered platform for automation to facilitate collaboration between people, robotics, software applications, and AI in end-to-end automation processes. The strengths of this organization are enterprise automation, orchestration, integration, and process application development using low-code capabilities.

The primary limitation of this strategy is that all these tools are general-purpose and usually require a lot of configuration and management and are expensive when licensed. While all of these provide immense value, they could be less useful to those companies

that would need to create their custom automation platforms. For instance, Crystal System CMS has overcome this challenge through its unique selling point of a customizable platform.

2.2.4 Kissflow

Kissflow pays attention to no-code workflow design, dashboard design, process auditing, task management, and optimization of process flow. The main strength of Kissflow is its user-friendliness and functionality concerning workflow designing by business users.

Nonetheless, the main problem with applications such as Kissflow is that the level of customization may be lower in some cases. For example, if it is required to work with certain departments, such as facilities, then custom systems can be much better in this situation. Based on the information available on the products' web pages, Kissflow seems to be more suitable for companies focusing on workflow, whereas Crystal System CMS seeks to implement them into the overall administration/management system.

Table 1: Comparison of Related Software with Crystal System CMS

System	Main Strength	Main Limitation	Relevance to Crystal System CMS
Odoo	Broad integrated business applications	More ERP-focused than BPMN-focused	Shows value of integration
SAP Signavio	Advanced process intelligence and mining	Not a full custom organizational management system	Supports process analysis ideas
Appian	Powerful enterprise automation and orchestration	Complex and general-purpose for smaller tailored needs	Supports workflow automation concepts
Kissflow	User-friendly no-code workflow management	Limited domain-specific customization	Supports workflow, audit, and dashboard ideas
Crystal System CMS	Custom integration of organization, facilities, process, KPI, access control, and audit logs	Depends on continued extension and backend maturity	Addresses project-specific requirements directly

2.3Related Work From Literature

Research Findings Summary

From the research, the findings suggest that enterprise systems are getting more integrated, more visible, and improved continually. Research conducted in the area of Business Process Management (BPM) underscores the need to develop and improve processes in a cyclical way. BPMN provides a standard notation of processes, enabling effective collaboration among technical and nontechnical people. From process mining literature, the findings show that event logs can be used to discover and improve business processes.

From the security literature, the findings are clear that RBAC is key in designing business applications. This is particularly true where the system deals with sensitive data and multiple user roles. This also applies to structured logging based on the guidelines issued by NIST on log management.

The use of dashboards was found to assist the management with summaries that were lucid, giving them an overview of the detailed data as well as being well organized. These results are in accordance with the dashboard and KPI features in Crystal System CMS. Further studies carried out on the simulation of workflows indicate the advantages of trying out different scenarios in order to assess the processes involved. This is in line with the intended aim of the project based on what-if scenario analysis.

All the above results show that the application of only one approach alone will not lead to the desired outcome. What is required is the combination of all these features, namely process modeling, security access control, logging, analysis, and organizational information. And this is exactly what Crystal System CMS does.

2.4Critique of Existing Work and Research Gap

The current literature offers useful knowledge regarding certain aspects, but it cannot be considered a universal tool that can be used for creating custom solutions for a specific project. BPMN languages of modeling allow illustrating the processes in organizations very well; however, there is no system that combines data, users, infrastructure, and security roles. The RBAC model represents a good ground for providing permissions to the users, but it lacks a workflow module, key performance indicators module, and dashboards module.

Hence, the major problem with the current state of literature can be defined not as a lack of instruments but as a lack of a customized solution:

- organizational structure management,
- facility and location management,
- process and task management,
- BPMN-based visualization,
- KPI and dashboard monitoring,
- role-based access control, and
- audit accountability.

Crystal System CMS addresses this gap by combining these features into one web-based platform. The core of the system does not lie in the creation of any specific tool, for example, BPMN, RBAC, dashboards, and logs. It lies in its practical application as well as the integration of these tools into a unified enterprise management system.

2.5Chapter Summary

Some Background Concepts and Research Work Related to Crystal System CMS were discussed in this chapter. Some of the subjects discussed in this chapter are enterprise management systems, BPM and BPMN, RBAC, audit logging, dashboards, KPIs monitoring, and process analysis. Some of the software applications like Odoo, SAP Signavio, Appian, and Kissflow have also been discussed along with their pros and cons. It can be observed from the above discussion that though sufficient amount of information and solutions regarding a particular problem exist to some extent in the literature as well as current solutions, yet a partial solution always exists for a particular problem. It is for this reason that the need for an organization management system like Crystal System CMS becomes inevitable.

Chapter 3
System Requirements

Crystal System CMS Requirements have been described in this chapter following the guidelines provided by the Software Requirements Specification document. This chapter focuses on the determination of the use cases of the system, its functional and non-functional requirements, as well as its interfaces and database requirements. The feasibility study of the project and analysis of analysis models have also been covered in this chapter.

3.1 Use Case Diagram

CMS Crystal System is a tool for managing enterprises, utilized by different types of users. Depending on the modules installed, the system assigns certain roles to guarantee that all the users have adequate access and functionalities, the main actors are:

- Super Admin
- Admin
- Normal User
- System / Backend API

The major use cases include authentication, company and location management, people and role management, process and task management, KPI handling, dashboard viewing, audit review, and data import/export.

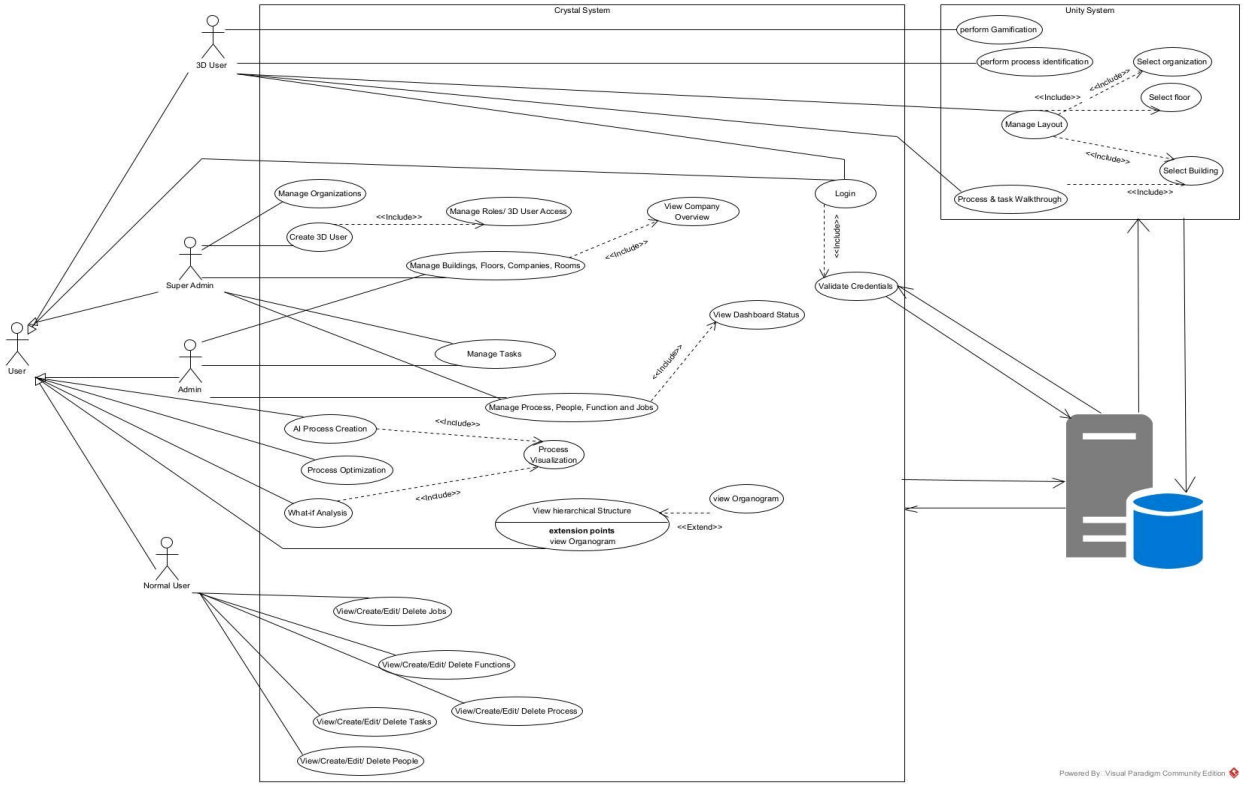


Figure 1: System-Level Use Case Diagram

3.2Functional Requirements

The functional requirements are derived from the use cases.

FR-01 User Authentication

- Users should be able to access the system through their registered usernames and passwords.
- The system should maintain user logins for the entire duration of their sessions.
- Users should be able to logout from the system.
- Unregistered users should not be allowed access to restricted parts of the platform.

FR-02 Role-Based Access Control

- The permission assignment must be made based on the role of the user.
- The access must be limited to certain modules like KPI management, audit log, and asset task module so that only those roles that are authorized can access these modules.
- There should be at least three roles for accessing the system, i.e., SUPER_ADMIN, ADMIN, and NORMAL_USER.

FR-03 Dashboard Management

- The permission assignment must be made based on the role of the user.

- The access must be limited to certain modules like KPI management, audit log, and asset task module so that only those roles that are authorized can access these modules.
- There should be at least three roles for accessing the system, i.e., SUPER_ADMIN, ADMIN, and NORMAL_USER.

FR-04 Organization and Location Management

- The system shall allow management of countries and cities.
- The system shall allow creation and update of company records.
- The system shall support organizational structure management for enterprise entities.

FR-05 Facility Management

- The system shall allow users to manage buildings, floors, and rooms.
- The system shall maintain associations between buildings and their internal structure.
- The system shall support viewing and editing facility details.

FR-06 People, Role, and User Management

- The system shall allow creation, editing, and viewing of people records.
- The system shall manage system users and access credentials.
- The system shall allow role creation and assignment.
- The system shall support password reset and user administration functions.

FR-07 Process Management

- The system shall allow creation, editing, viewing, and deletion of processes.
- The system shall store process code, name, category, status, and overview.
- The system shall support process-task relationships and workflow ordering.
- The system shall support process version requests and review workflows where applicable.

FR-08 Task, Job, and Function Management

- The system shall allow creation, update, and deletion of tasks.
- The system shall allow creation, update, and deletion of jobs and functions.
- The system shall support status and category management for tasks and jobs.
- The system shall allow linking of tasks and jobs within process workflows.

FR-09 BPMN Visualization and Workflow Editing

- The system shall display BPMN diagrams for selected processes.

- The system shall support workflow visualization using tasks, gateways, and transitions.
- The system shall support saving and updating BPMN XML or equivalent workflow definitions.

FR-10 What-If Analysis and Process Optimization

- The system shall allow the user to select a process for analysis.
- The system shall support scenario-based process optimization.
- The system shall display alternative allocations or process outcomes for comparison.

FR-11 KPI and Performance Management

- The system shall allow management of KPIs.
- The system shall support units of measure and KPI attribute management.
- The system shall link KPIs with process-related monitoring where required.

FR-12 Asset Task Management

- The system shall allow creation, editing, and viewing of asset tasks.
- The system shall support linking asset tasks with process tasks and workflows.
- The system shall allow coverage checking for input and output assets within a process chain.

FR-13 Import and Export

- The system shall support Excel or bulk import for dashboard or management data.
- The system shall support export of company or organizational data.
- The system shall validate uploaded data before processing where possible.

FR-14 Audit Logging

- The system shall maintain records of login attempts and user actions.
- The system shall support viewing action logs, failed logins, and deletion logs.
- The system shall restrict audit-log access to authorized users only.

3.2.1 Use Case Descriptions

Table 2: Login

Field	**Description**
Use Case ID	UC01
Use Case Name	Login
Actor(s)	Super Admin, Admin, Normal User, 3D User
Pre-Conditions	User has a registered account and is on the Login page.
Priority	High
Basic Flow	User enters credentials, system validates them, and redirects to the appropriate dashboard.

Actor Action	**System Response**
1. User enters Username and Password.	2. System validates input format.
3. User clicks "Login".	4. System calls **Validate Credentials**. 5. If valid, System redirects user to the role-specific dashboard.

Alternative Course of Action	
Actor Action	**System Response**
3a. User enters incorrect password.	4a. System displays error "Invalid Credentials" and prompts retry.

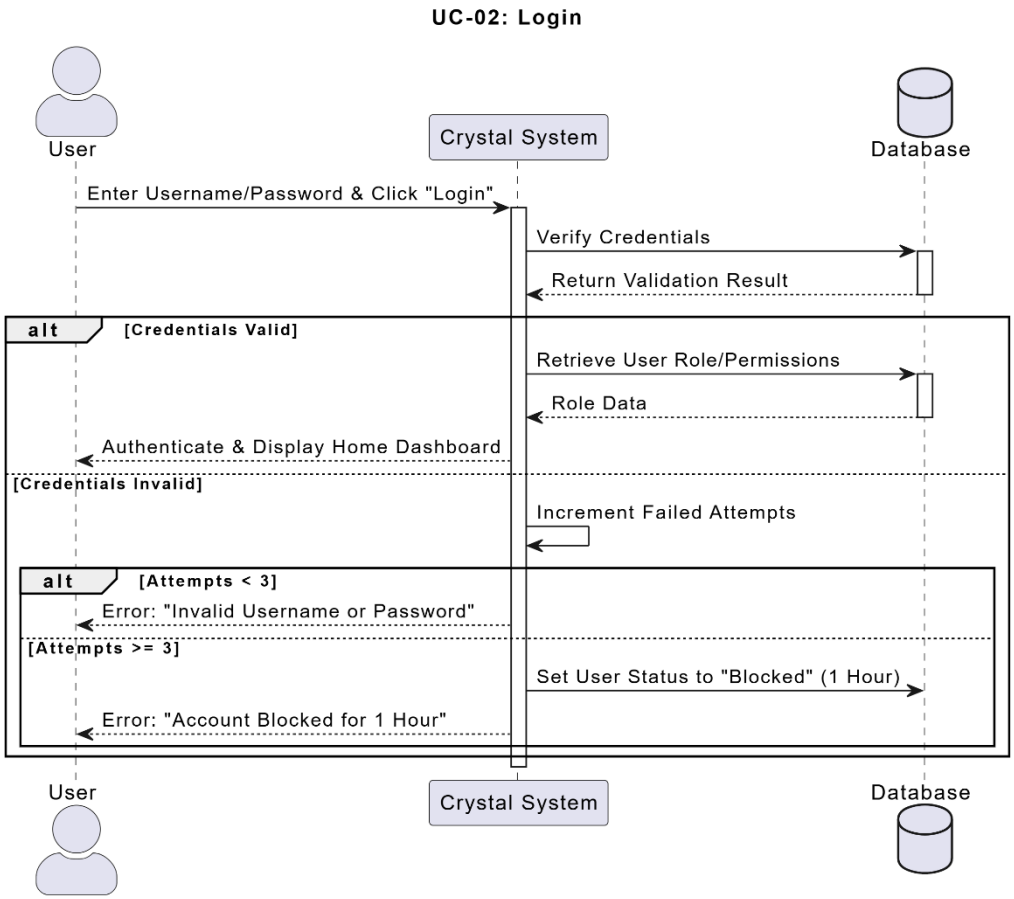


Figure 2: Login

Table 3: Manage Organizations

Field	**Description**
Use Case ID	UC02
Use Case Name	Manage Organizations
Actor(s)	Super Admin
Pre-Conditions	Super Admin is logged in.
Priority	High
Basic Flow	Super Admin creates or edits organization details and configures user access roles.

Actor Action	**System Response**
1. Super Admin clicks "Organization Management".	2. System displays list of existing organizations.
3. Super Admin clicks "Add Organization" or selects one to edit.	4. System displays organization details form.

5. Super Admin enters details and clicks "Configure Roles".	6. System initiates **Manage Roles / 3D User Access** sub-flow.
7. Super Admin clicks "Save".	8. System saves organization data and confirms success.

Table 4: Manage Organization

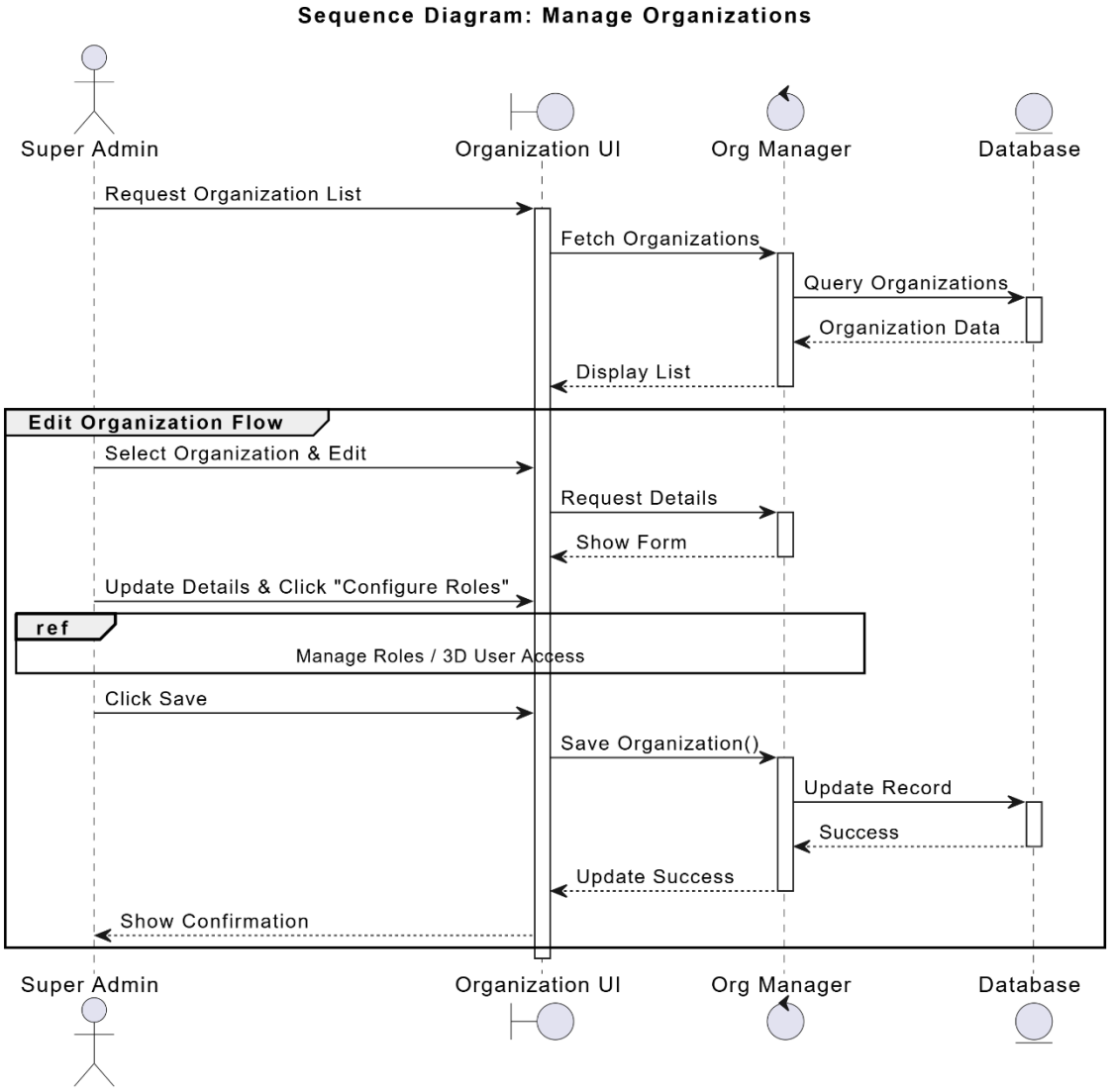


Figure 3: Manage Organization

Table 5: Create 3D User

Field	Description
Use Case ID	UC03
Use Case Name	Create 3D User
Actor(s)	Super Admin
Pre-Conditions	Super Admin is logged in.
Priority	Medium
Basic Flow	Super Admin registers a new user specifically for the Unity 3D environment.

Actor Action	System Response
1. Super Admin navigates to "3D Users".	2. System displays 3D user form.
3. Super Admin enters user details and assigns specific 3D privileges.	4. System validates input.
5. Super Admin clicks "Create".	6. System generates account and sends credentials to the user.

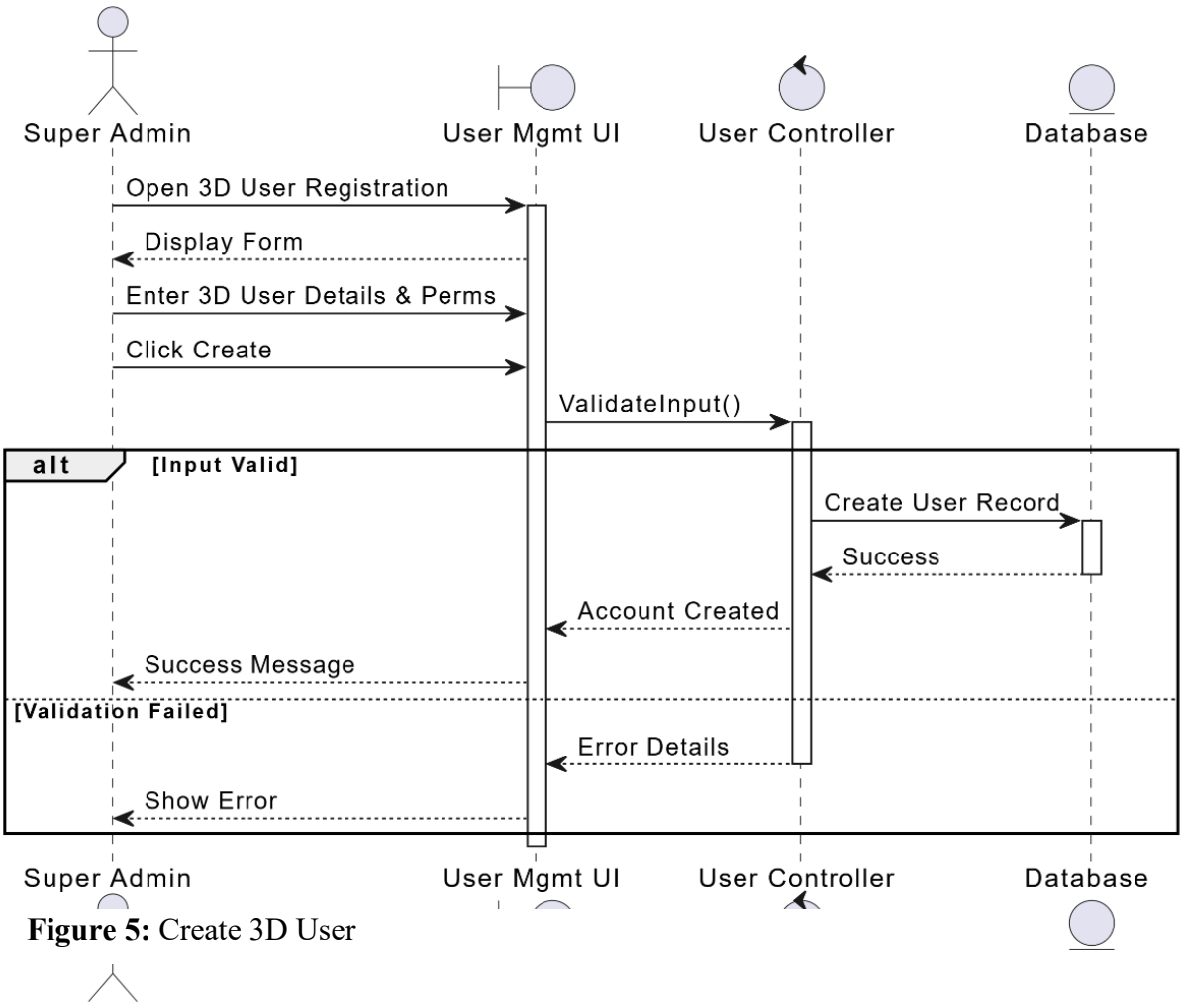


Figure 5: Create 3D User

Table 6: Manage Infrastructure (Buildings, Floors, Companies)

Field	**Description**
Use Case ID	UC04
Use Case Name	Manage Buildings, Floors, Companies, Rooms
Actor(s)	Super Admin
Pre-Conditions	Super Admin is logged in.
Priority	High
Basic Flow	Super Admin sets up the physical hierarchy of the facility.

Actor Action	**System Response**
1. Super Admin selects "Infrastructure Setup".	2. System displays the hierarchy tree.
3. Super Admin adds a Building, Floor, or Room.	4. System updates the tree structure.
5. Super Admin requests overview.	6. System executes **View Company Overview** to show the summary.

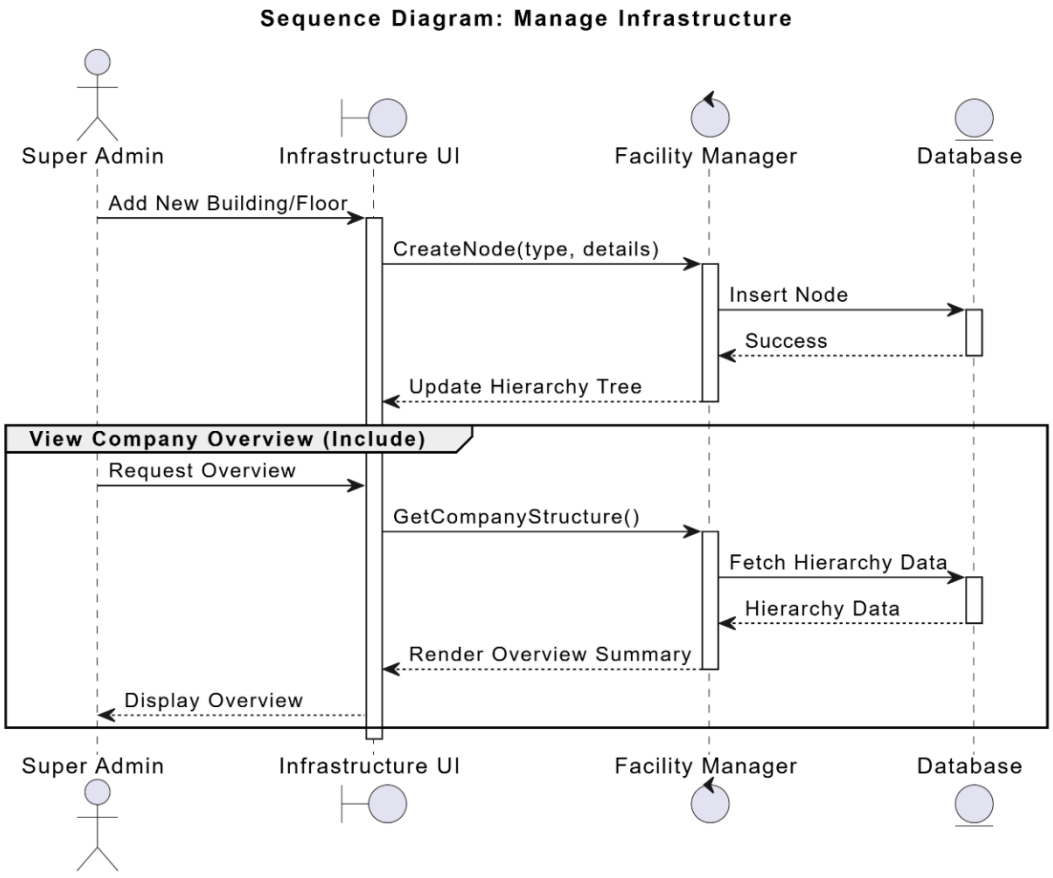


Figure 6: Manage Infrastructure

Table 7: AI Process Creation

Field	**Description**
Use Case ID	UC05
Use Case Name	AI Process Creation
Actor(s)	Admin
Pre-Conditions	Admin is logged in and has AI credits/access.
Priority	Medium
Basic Flow	Admin inputs parameters, and AI generates a process workflow.

Actor Action	**System Response**
1. Admin clicks "AI Process Generator".	2. System displays prompt entry field.
3. Admin enters business requirements and clicks "Generate".	4. System processes input via AI engine.
	5. System executes Process Visualization to display the generated flow diagram.
6. Admin saves the process.	7. System stores the new process definition.

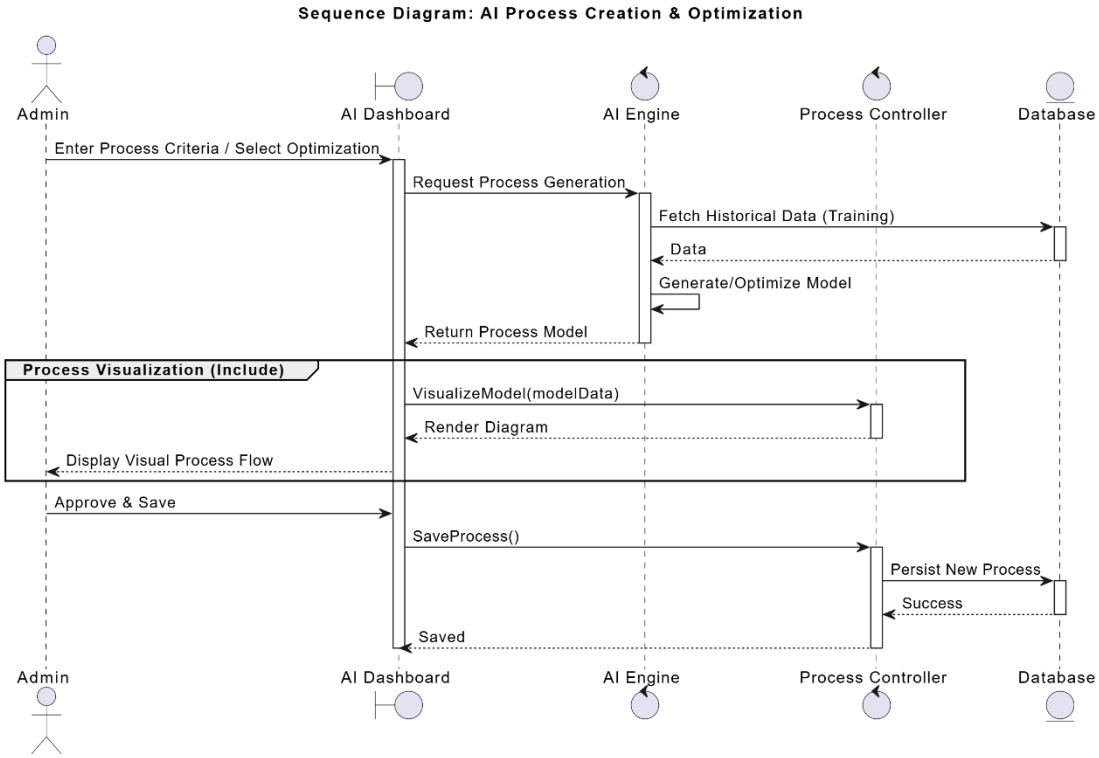


Figure 7: AI Process Creation

Table 8: Process Optimization

Field	Description
Use Case ID	UC06
Use Case Name	Process Optimization
Actor(s)	Admin
Pre-Conditions	Existing processes are defined in the system.
Priority	Medium
Basic Flow	Admin selects a process to analyze for efficiency improvements.

Actor Action	System Response
1. Admin selects a process and clicks "Optimize".	2. System analyzes process data (bottlenecks, time).
3. Admin reviews suggestions.	4. System executes **Process Visualization** to show "Before vs After".
5. Admin applies changes.	6. System updates the process version.

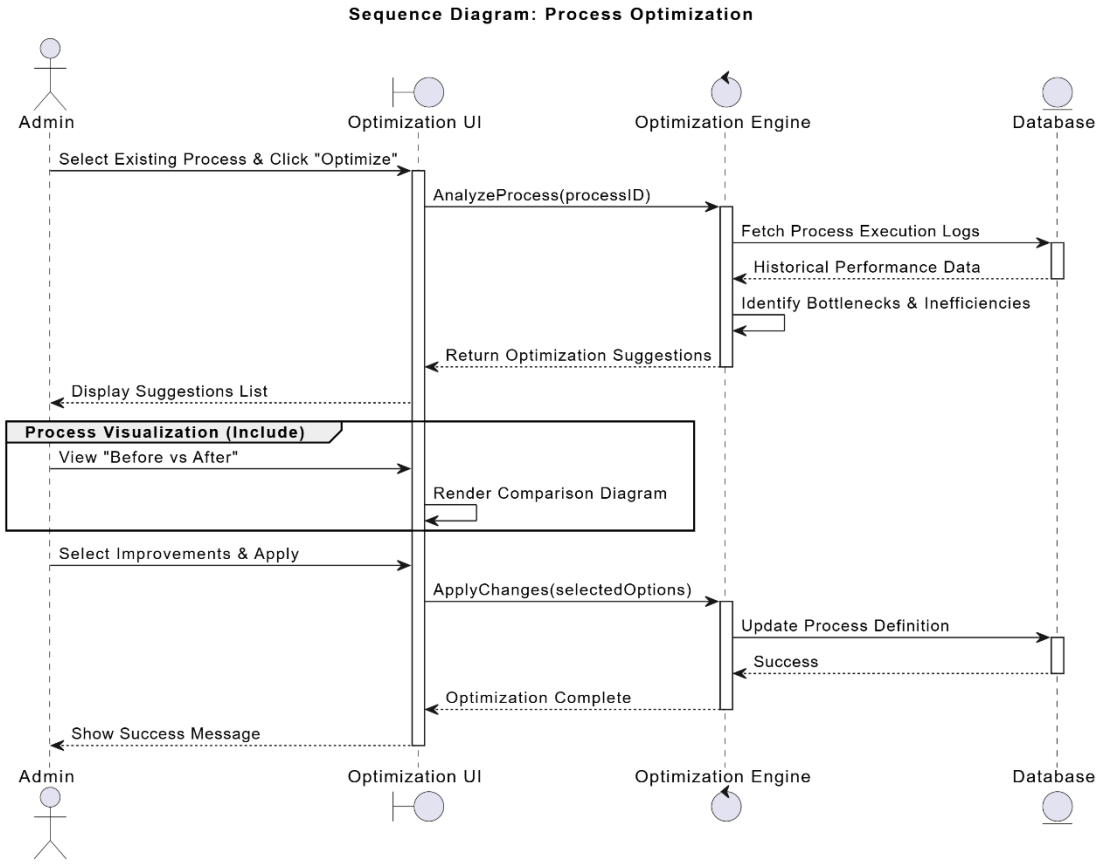


Figure 8: Process Optimization

Table 9: What-if Analysis

Field	**Description**
Use Case ID	UC07
Use Case Name	What-if Analysis
Actor(s)	Admin
Pre-Conditions	Admin is logged in.
Priority	Low
Basic Flow	Admin simulates scenarios to predict outcomes.

Actor Action	**System Response**
1. Admin navigates to "Simulation/What-if".	2. System requests scenario parameters (e.g., "Reduce staff by 2").
3. Admin inputs variables and clicks "Simulate".	4. System calculates impact.
	5. System executes **Process Visualization** to display results.

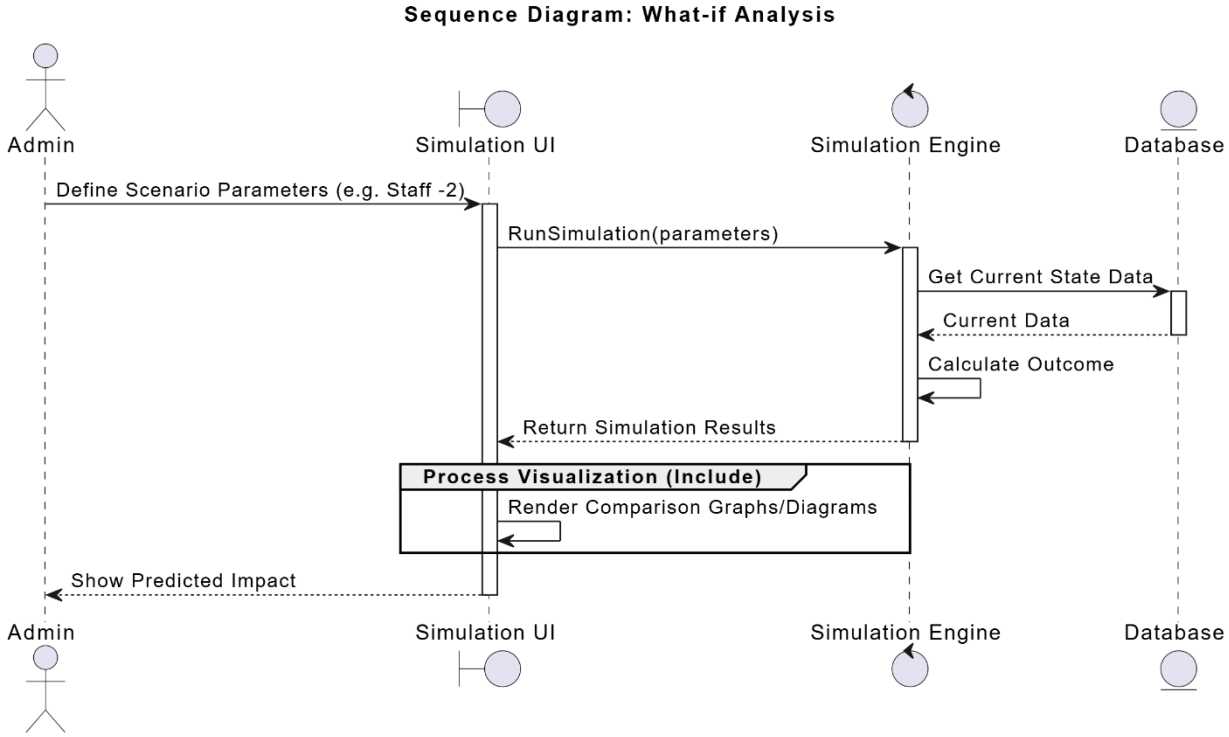


Figure 9: What-if Analysis

Table 10: View Hierarchical Structure / Organogram

Field	**Description**
Use Case ID	UC08
Use Case Name	View Hierarchical Structure
Actor(s)	Admin
Pre-Conditions	Hierarchy data exists.
Priority	Low
Basic Flow	Admin views the reporting lines or structure of the organization.

Actor Action	**System Response**
1. Admin clicks "View Hierarchy".	2. System displays a list/tree view of the organization.
3. Admin clicks "Switch to Organogram View".	4. System executes **View Organogram** (Extension) to render a graphical chart.

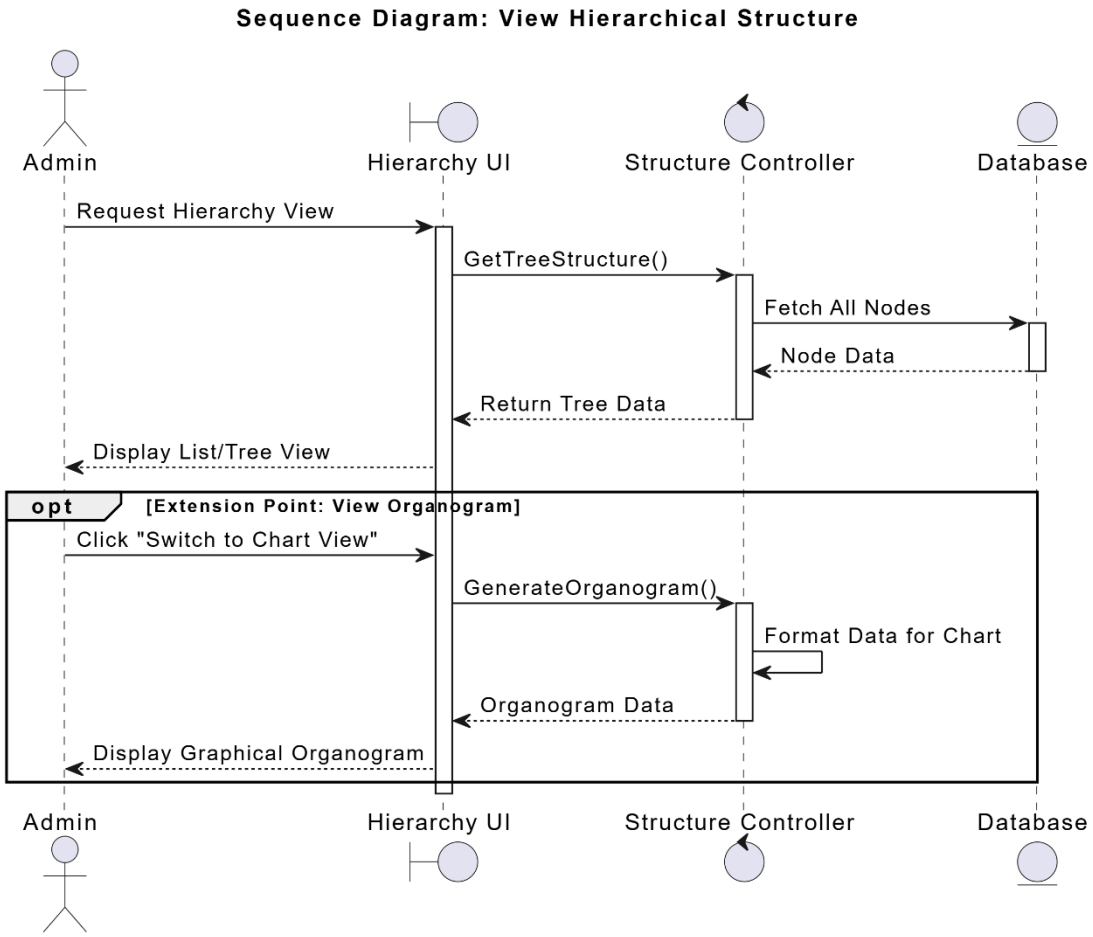


Figure 10: View Hierarchical Structure / Organogram

*Note: This table applies to **Jobs, Functions, Process, Tasks, and People** use cases for the **Normal User**.*

Table 11: Manage Operational Entities (CRUD)

Field	**Description**
Use Case ID	UC09 - UC13
Use Case Name	View/Create/Edit/Delete [Entity]
Actor(s)	Normal User
Pre-Conditions	User is logged in.
Priority	High
Basic Flow	User manages the lifecycle of operational data (Jobs/Functions/Process/Tasks/People).

Actor Action	**System Response**
1. User selects the specific module (e.g., "Jobs").	2. System displays a list of existing items.
3. User clicks "Create New".	4. System displays the creation form.
5. User fills details and clicks "Submit".	6. System saves the new item and updates the list.
Alternative: Edit/Delete	
3b. User clicks "Edit" or "Delete" on an item.	4b. System updates or removes the item from the database.

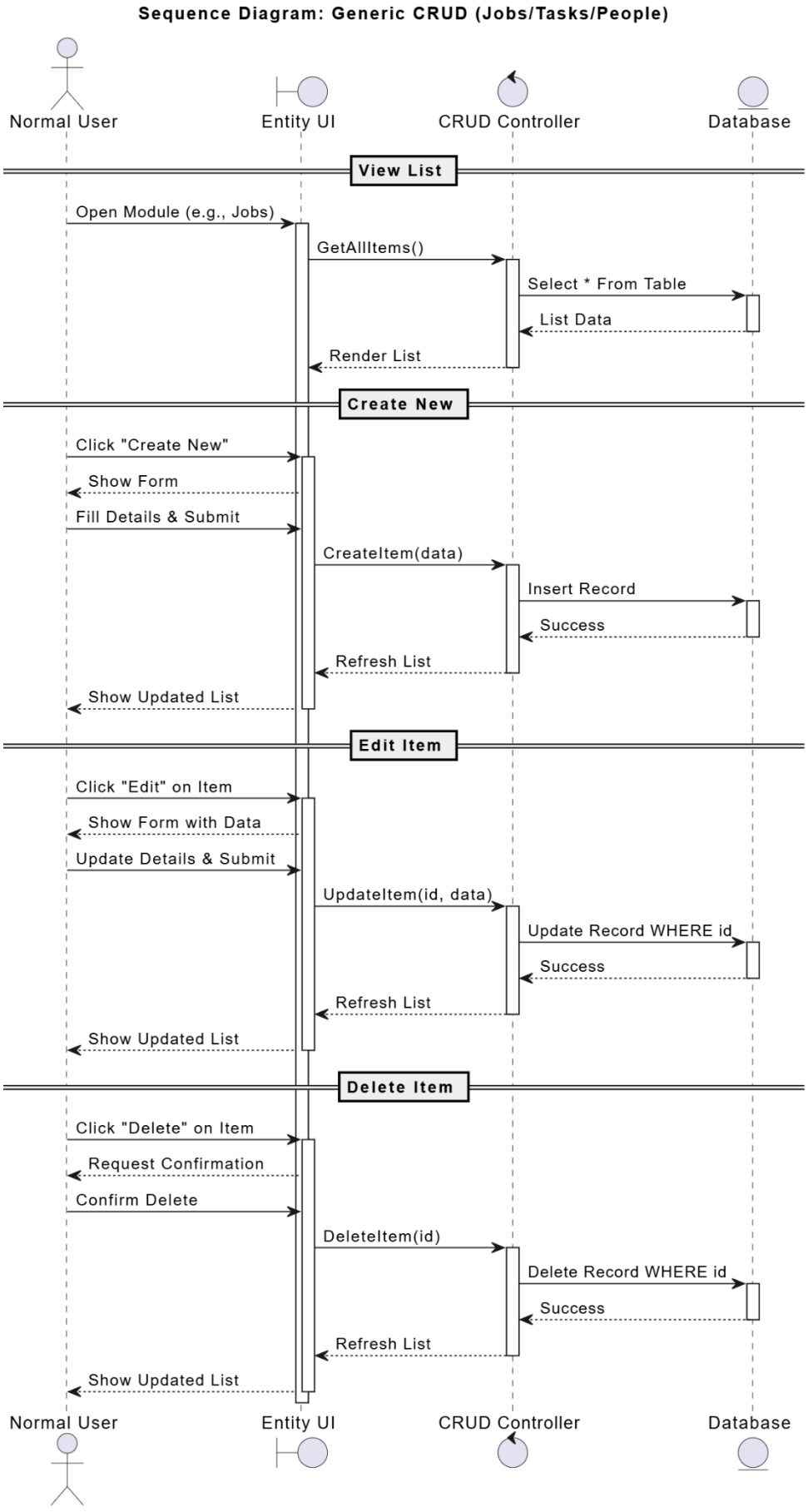


Figure 11: Generic CRUD (Job/Tasks/People)

Table 12: Manage Layout

Field	**Description**
Use Case ID	UC14
Use Case Name	Manage Layout
Actor(s)	3D User
Pre-Conditions	3D Unity Interface is loaded.
Priority	High
Basic Flow	User configures the 3D environment view.

Actor Action	**System Response**
1. User launches Layout Manager.	2. System executes **Select Organization**.
3. User selects Organization.	4. System executes **Select Building**.
5. User selects Building.	6. System executes **Select Floor**.
7. User selects Floor.	8. System renders the specific 3D layout for that location.

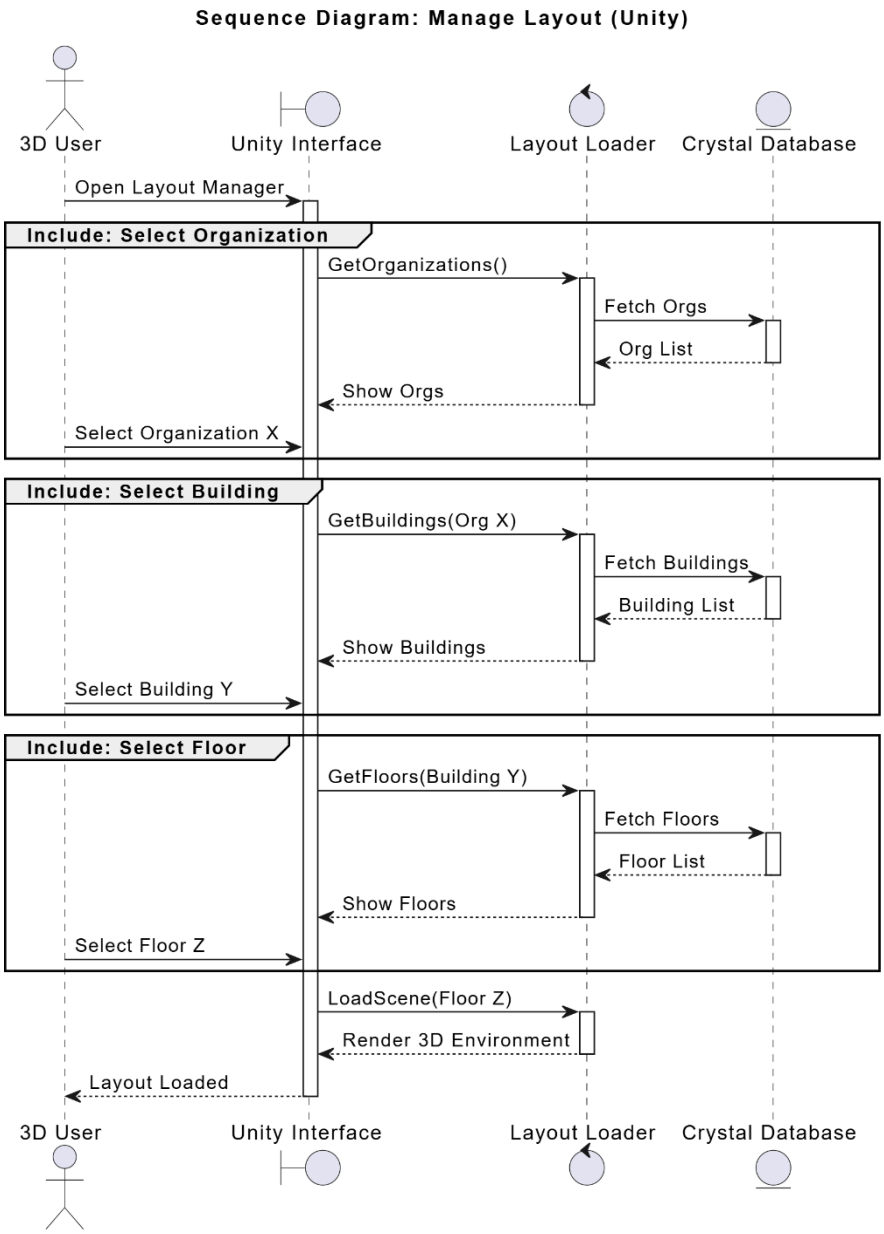


Figure 12: Manage Layout

Table 13: Process & Task Walkthrough

Field	Description
Use Case ID	UC15
Use Case Name	Process & Task Walkthrough
Actor(s)	3D User
Pre-Conditions	A layout is loaded and processes are defined.

Priority	Medium
Basic Flow	User visualizes a process flow within the 3D space.
Actor Action	**System Response**
1. User selects a specific process from the menu.	2. System highlights the start point in the 3D model.
3. User clicks "Start Walkthrough".	4. System animates the path or flow of the task through the building.

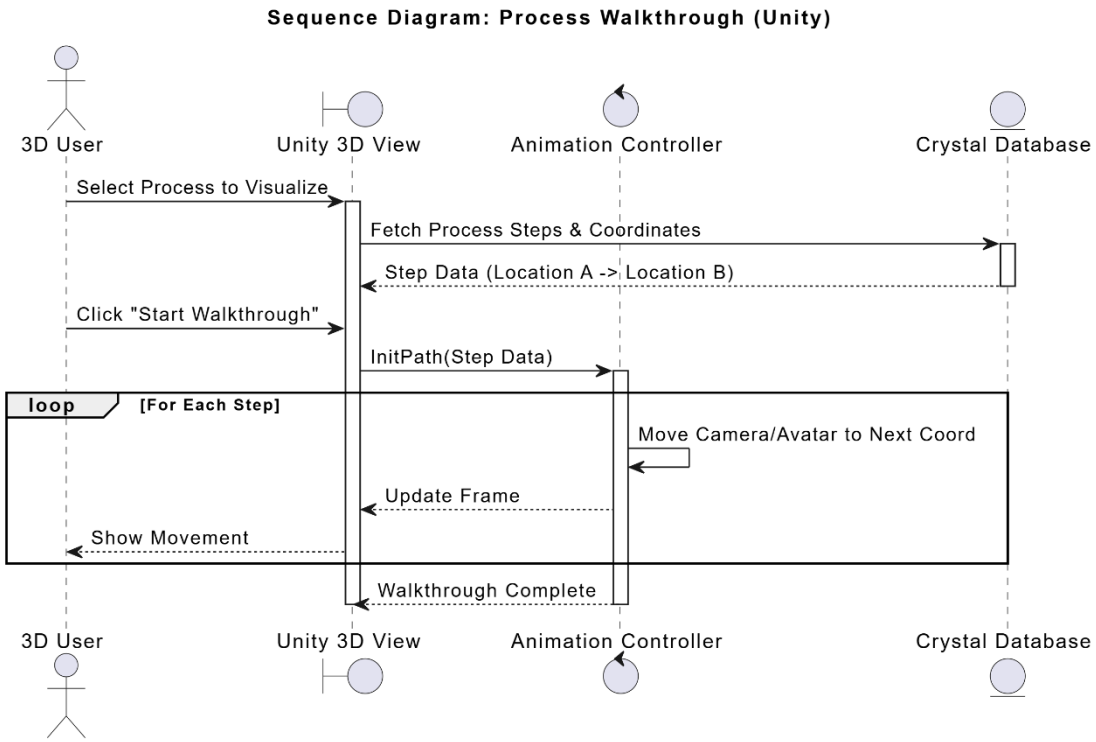


Figure 13: Process & Task Walkthrough

Table 14: Perform Process Identification

Field	**Description**
Use Case ID	UC16
Use Case Name	Perform Process Identification
Actor(s)	3D User
Pre-Conditions	User is in the 3D view.
Priority	Medium
Basic Flow	User identifies and tags objects or zones with process information.

Actor Action	**System Response**
1. User navigates to a specific machine or zone in 3D.	2. System highlights the selectable object.
3. User clicks the object and selects "Identify Process".	4. System displays attached process metadata (e.g., "Assembly Line A").

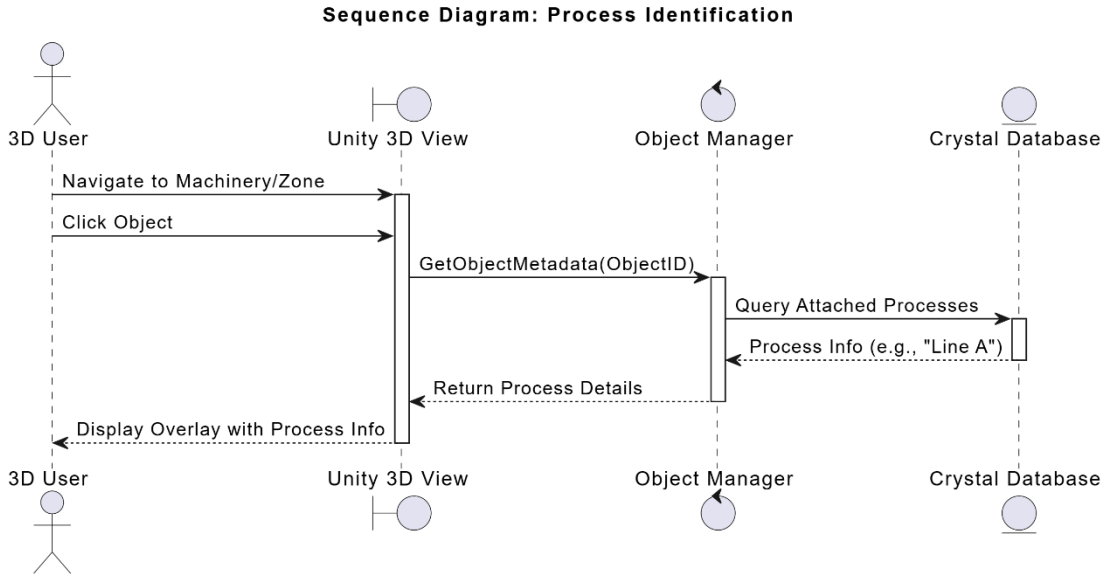


Figure 14: Process Identification

Table 15: Gamification

Field	Description
Use Case ID	UC17
Use Case Name	Gamification (Earn Rewards)
Actor(s)	Normal User
Pre-Conditions	User is logged in and has pending tasks or training modules.
Priority	Low
Basic Flow	User completes an activity (task/process), and the system awards points, checks for badge eligibility, and updates the leaderboard.

Actor Action	System Response
1. User completes a specific task or process and clicks "Complete".	2. System validates task completion and closes the task.
	3. System calls **Gamification Engine** to calculate points based on task difficulty/timeliness.
	4. System adds points to User's profile.
	5. System checks Badge rules (e.g., "Completed 10 tasks").
	6. If rule met: System assigns Badge and generates "Badge Unlocked" notification.
	7. System updates global/team Leaderboard.
8. User views profile or notification.	9. System displays current points, new badges, and rank.

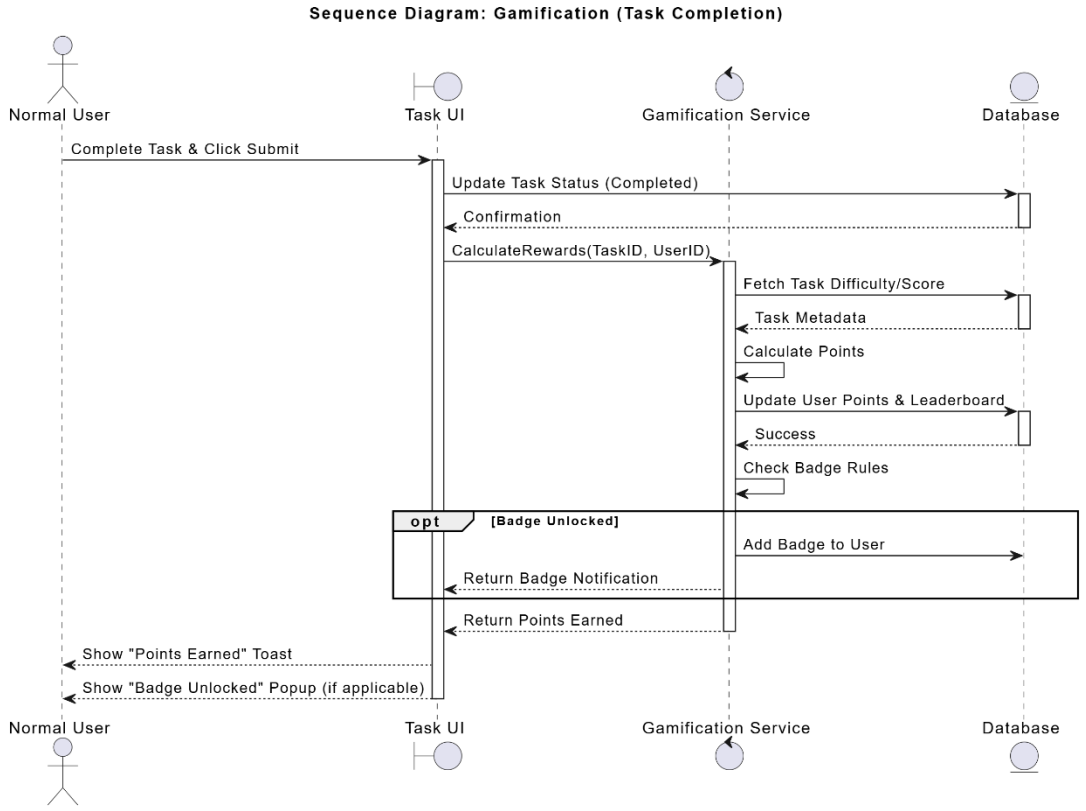


Figure 15: Gamification

3.2.2 Use case Explanation

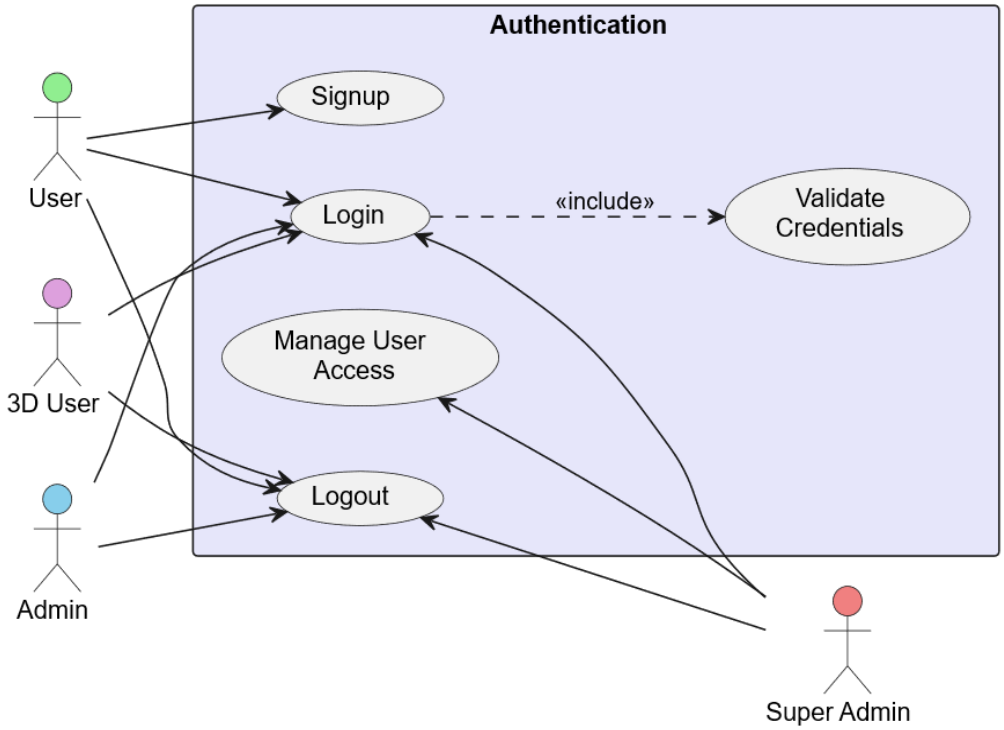


Figure 16: Authentication Usecase

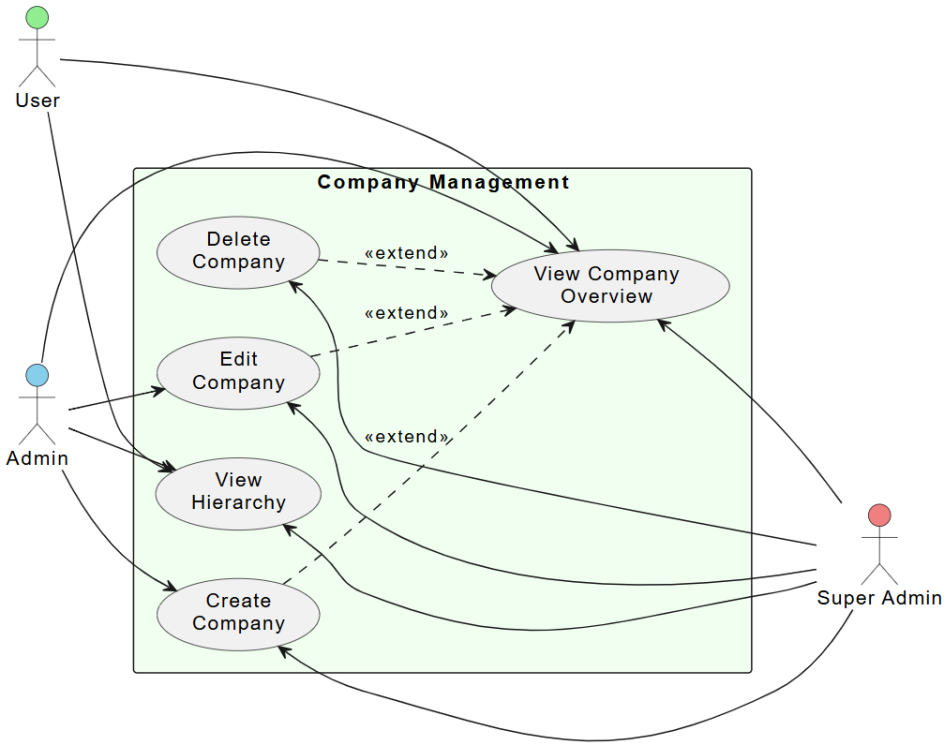


Figure 17: Company Management Module

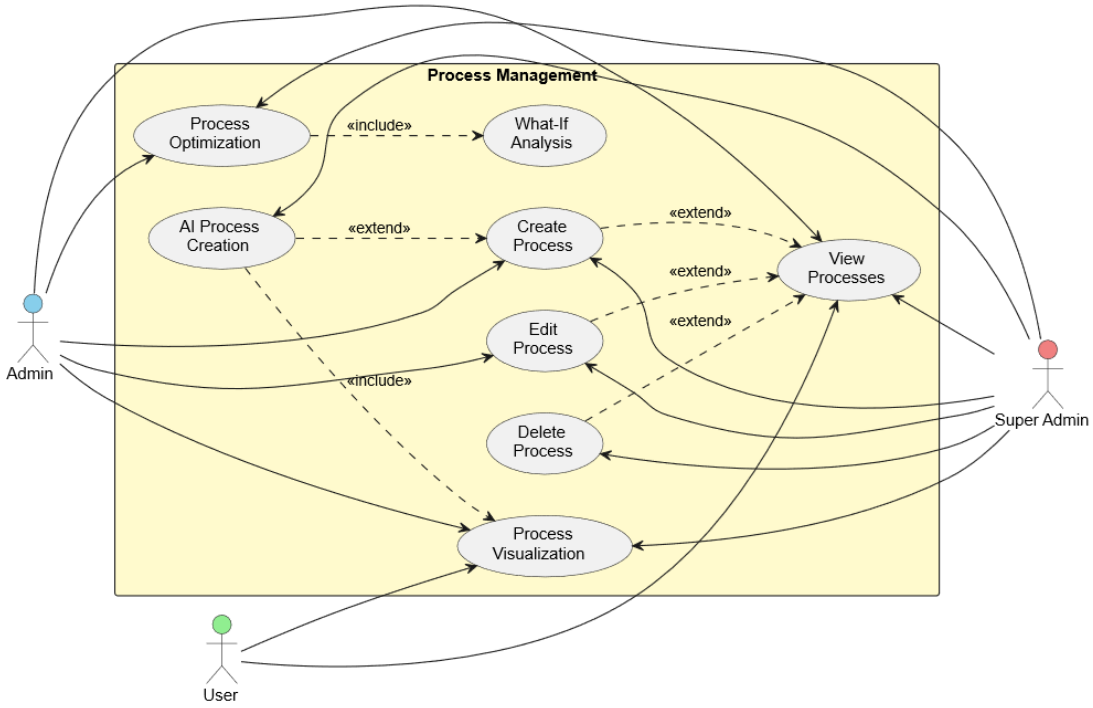


Figure 19: Process Management Module

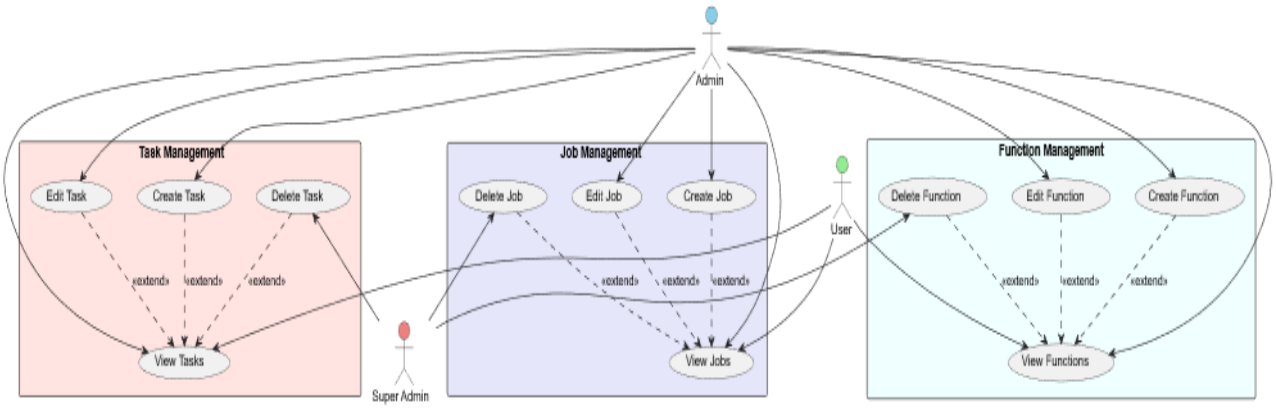


Figure 18: Job and Function Mangement

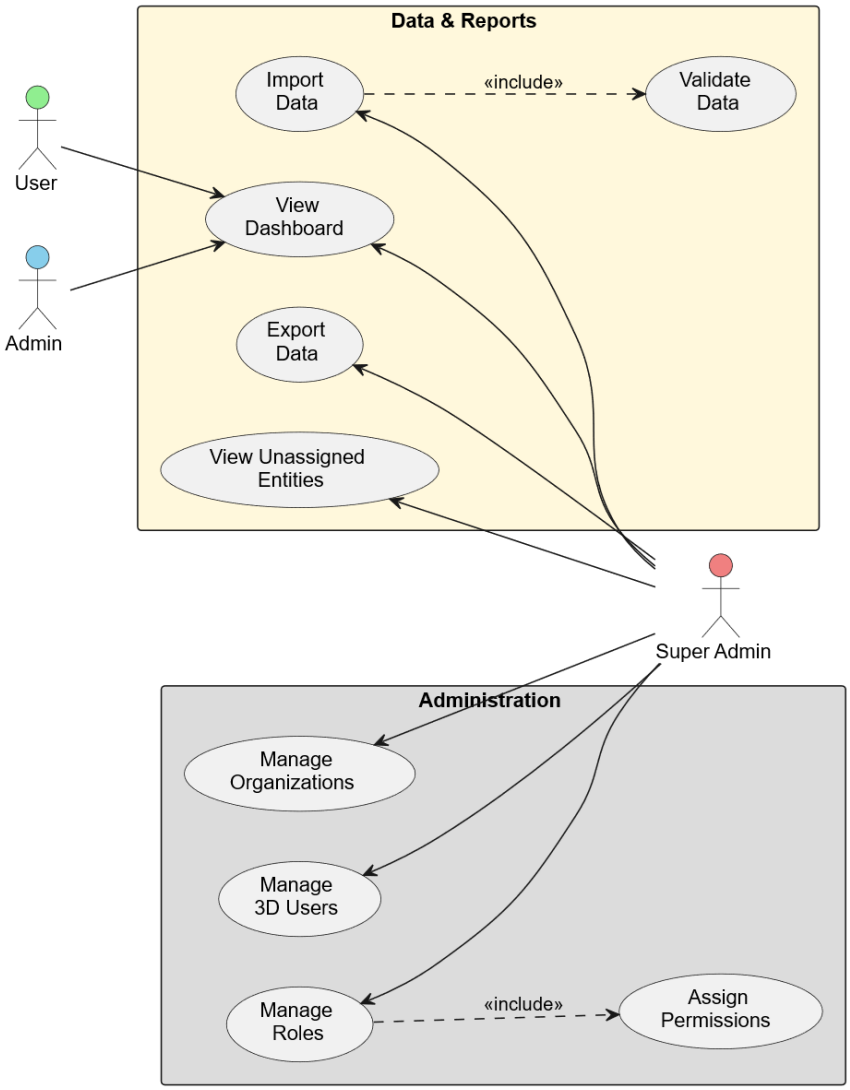


Figure 20: Data, Reports & Administration (Super Admin Features)

3.3Interface Requirements

3.3.1 User Interface Requirements

- A user-friendly GUI interface would be incorporated into the software system for easy navigation.
- In addition, there is a dedicated login page from where one can login to the system. The dedicated dashboard allows you to see statistical analysis at once.
- There would be a proper design of the interface to allow access to other different modules like Process Management, Task Management, User Management, KPI Management, and Audit Logs.

- The interface shall display error pages or alerts for unauthorized access and HTTP errors.
- The interface shall support responsive interaction for common desktop browser use.

3.3.2 Software / Component Interface Requirements

- The frontend shall communicate with backend services through API endpoints.
- The system shall support authentication through session-based or cookie-based mechanisms.
- The data transfer process for regular API requests uses the JSON method.
- In order to send files, we are going to use multipart form-data.
- We created some specific components that allow visualizing and editing BPMN diagrams directly within the application.
- The system is an internal client-side navigation system, allowing users to switch between modules without any difficulties.

3.3.3 Physical Interface Requirements

- This software is able to run on any computer that can execute a modern web browser.
- In order to use the graphical user interface, you will need a mouse, keyboard, and screen.
- You do not have to use any other hardware for the successful operation of the system.
- All you have to do is ensure that your computer is linked to the internet or LAN.

3.4 Database Requirements

In Crystal System CMS, the requirement of storing data persistently will provide the enterprise with the necessary data that it requires to run. The data base must include the following elements:

- Users
- Roles
- Permissions
- Companies
- Countries
- Cities

- Buildings
- Floors
- Rooms
- People
- Processes
- Process Categories
- Process Statuses
- Tasks
- Task Categories
- Task Statuses
- Jobs
- Job Categories
- Job Statuses
- Functions
- KPIs
- Units of Measure
- Asset Tasks
- Audit Logs

3.4.1 Database Requirements

- It should be noted that this specific database has the ability to employ a primary key to guarantee that each record that exists in this database will be unique.
- It handles the relationship between various organizational entities by using one-to-many and many-to-many relationships where appropriate.
- It considers the relationships that exist between different processes, tasks, and jobs.
- It handles the relationships between users and roles.
- In cases where there is an addition or modification of a record, it captures the time stamps of each such instance.
- It retains audit logs to facilitate authentication and behavior monitoring.

3.4.2 Entity Relationship Diagram

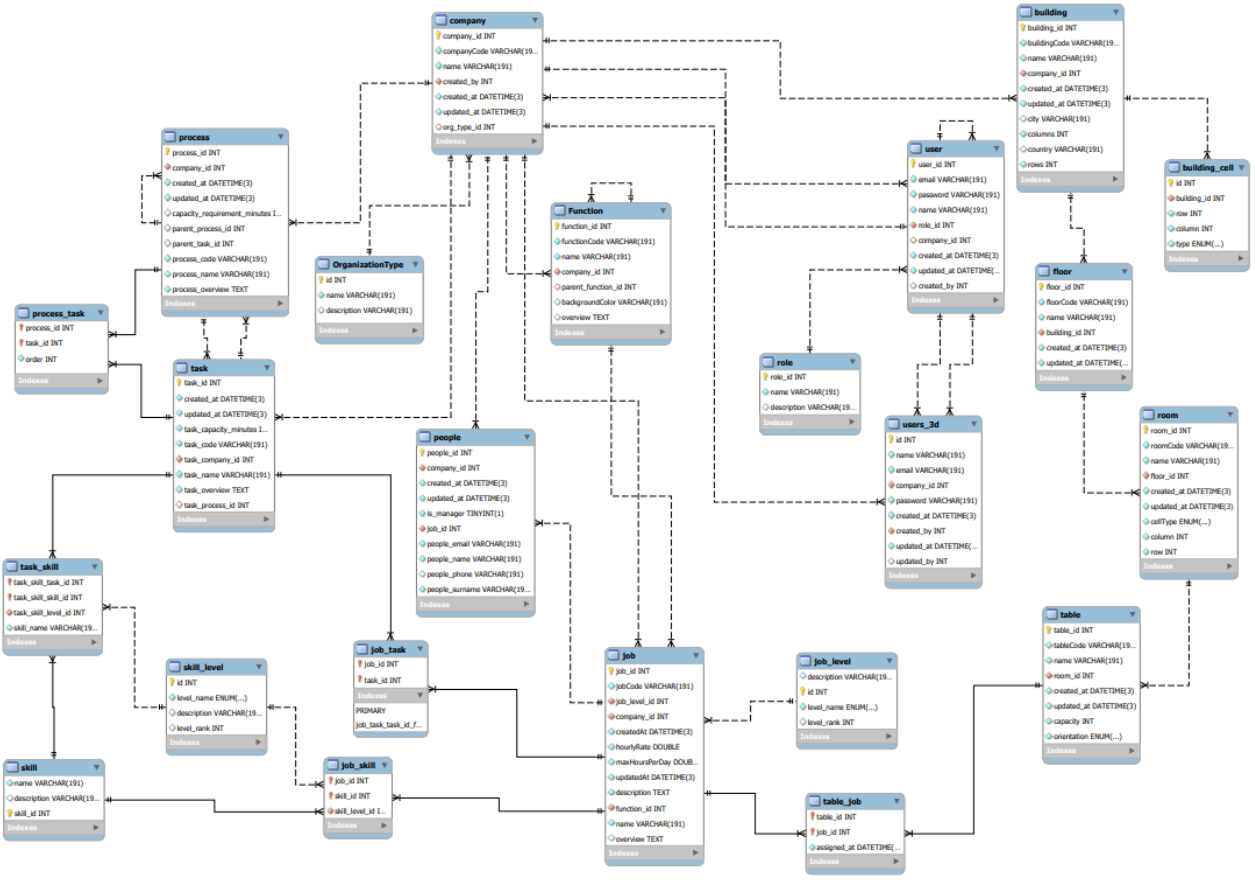


Figure 21: Entity Relationship Diagram

3.5 Non-Functional Requirements

3.5.1 Security

- The system must verify your identity before granting you access.
- Access control is implemented via role-based limitations in viewing and carrying out certain actions.
- This type of system cannot be breached by any other individual attempting to carry out certain actions.
- Logging is performed for all major operations that are carried out by either the user or the system due to security purposes.
- The system also controls login sessions securely.

3.5.2 Performance

- The system makes it easy to reach frequently used management web pages.
- Where there is a large amount of data that needs to be sorted out, the system uses pagination.
- The system is designed to facilitate fast data processing and filtration.
- It is possible to move between different modules of the system without waiting.

3.5.3 Usability

- There is an interface that is easy to use and understand.
- The system contains all the needed buttons and navigation bars that have names that make it clear what each button or navigation bar does.
- If there is a mistake in submitting the form, the system informs the user about it.
- All these considerations were made based on the assumption that users with no technical background would be able to use the system easily.

3.5.4 Reliability and Availability

- This ensures that all your stored data will always be accessible.
- In case of any issues occurring in any API, this system does not crash at all.
- Error messages and error pages will be shown should there be any issue in the back end.
- This system maintains consistency in your data in all CRUD operations.

3.5.5 Maintainability and Modifiability

- The platform is built on modular architecture, which makes maintenance easier.
- This allows for adding new modules or services whenever needed.
- It also separates the code between the interface and the logic of the service itself.
- By doing this, it makes process, KPI, and report generation scalable.

3.5.6 Interoperability

- Information sharing occurs using regular web technologies.
- To interface with the back-end, RESTful APIs may be utilized, and file creation is required for import/export operations.
- The software works seamlessly in all popular web browsers.

3.5.7 Constraints

- Our constraints will be the time available for us to design the software and also the period of our final year project.

- Availability of precise APIs will be crucial in determining our success.
- Due to the nature of the software as an online application, the web stack should always be considered during the process.
- Our budget might affect the scalability of the project.

3.6 Project Feasibility

3.6.1 Technical Feasibility

This project can be developed due to the application of modern and acknowledged technologies available on the Web. In particular, the front-end part will be made on the basis of the React framework along with other things, including routing, APIs, and BPMN. All of the mentioned technologies are quite reliable and adequate to create business-related applications on the Web. Moreover, the modularity of the software guarantees easy extension in the future and adding more functions to the system.

3.6.2 Operational Feasibility

Feasibility of the Project

It is operational feasible because the implemented project focuses on resolving the issues that occur in any enterprise. These issues include information fragmentation, ineffective visualization of the process, insufficient access control, and lack of administrative monitoring. It facilitates more efficient processing of daily procedures due to the integration of information and provision of management modules of high quality. Taking into account the fact that users will interact with the system through an intuitive web-based interface, which complies with office process standards, it is fair for professionals to use it.

3.6.3 Legal and Ethical Feasibility

It will be feasible in terms of legality and ethics provided that there is appropriate handling of the data within the firm. It will be essential to protect the system against any form of access to the user account information, private data, audit trail, and activity log. It will be necessary to guarantee that there will always be observance of the principles of data protection in the institution. Ethically, there will be the need to respect the users' privacy as well as prevent any form of access to the private data.

3.7 Analysis Models

3.7.1 Sequence Diagram

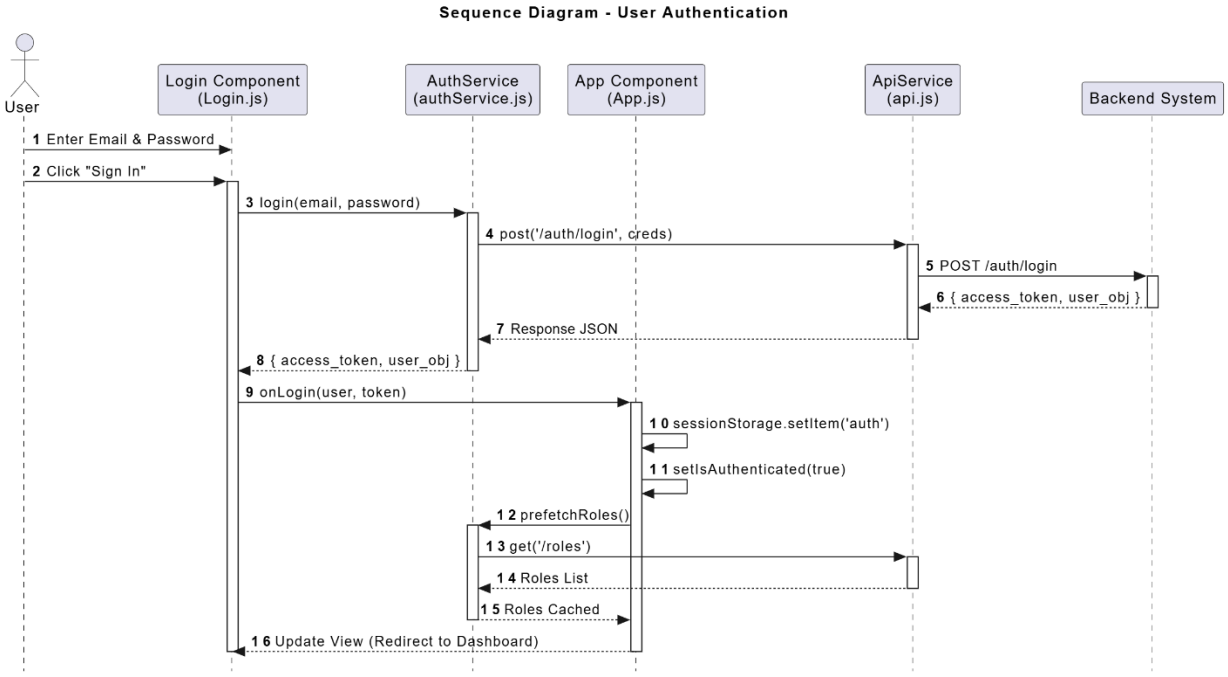


Figure 22: User Authentication Sequence Diagram

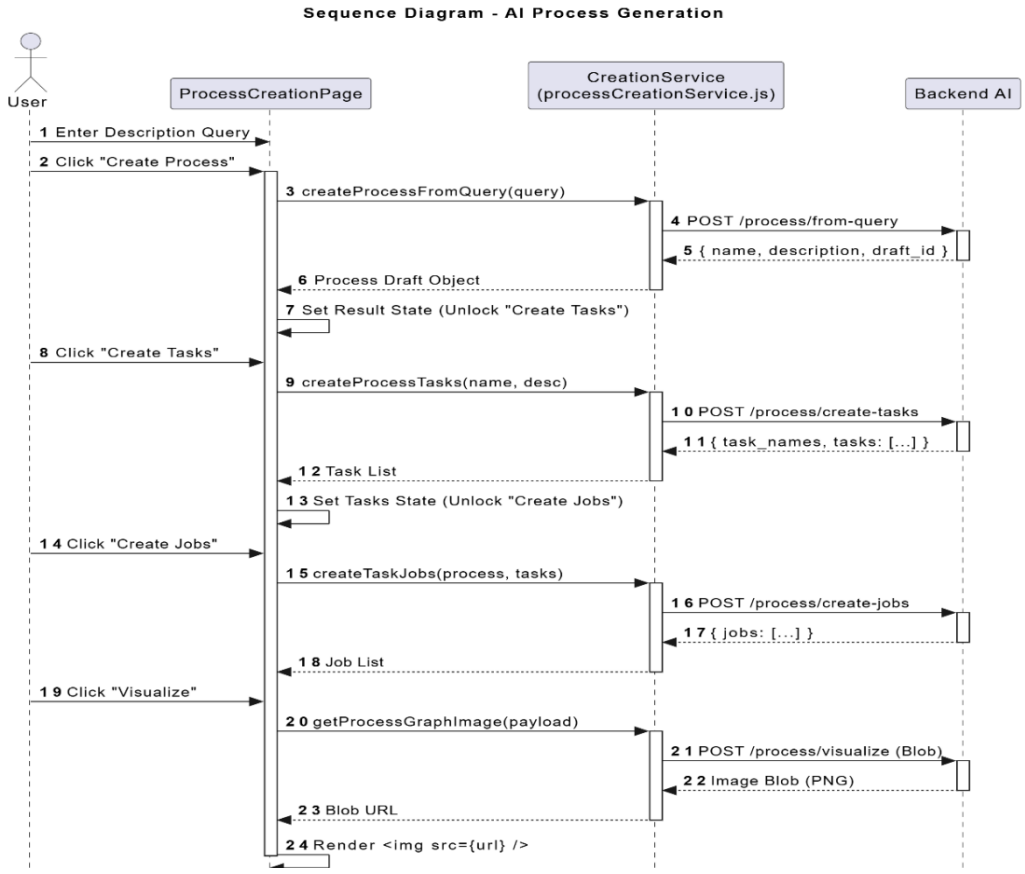


Figure 23: AI Process Generation Sequence Diagram

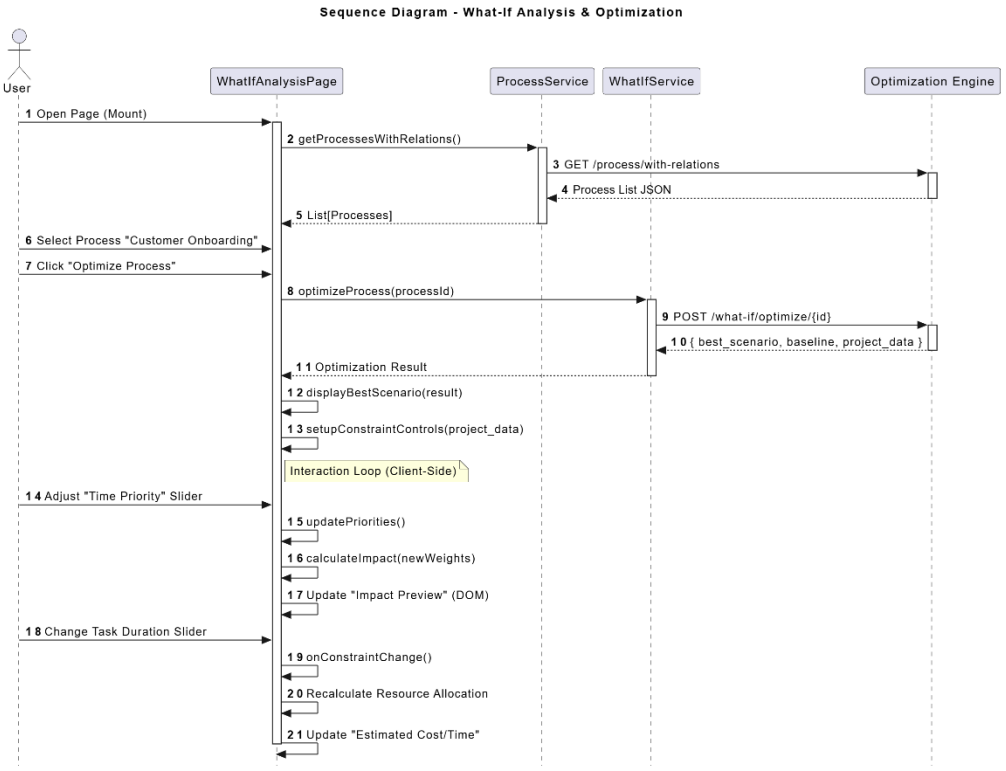


Figure 24: What-If Analysis Optimization

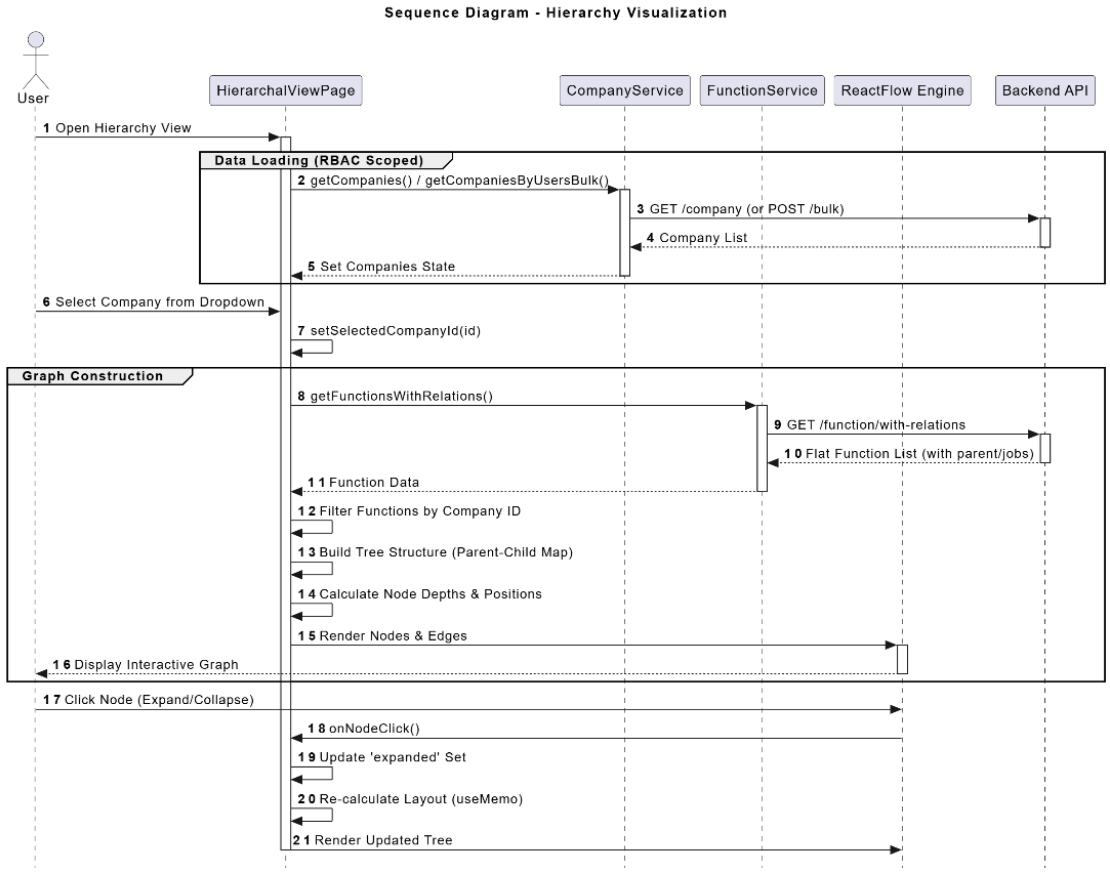
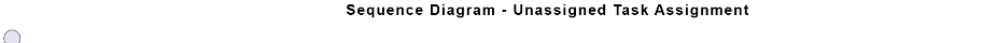


Figure 25: Hierarchy Visualization Sequence Diagram



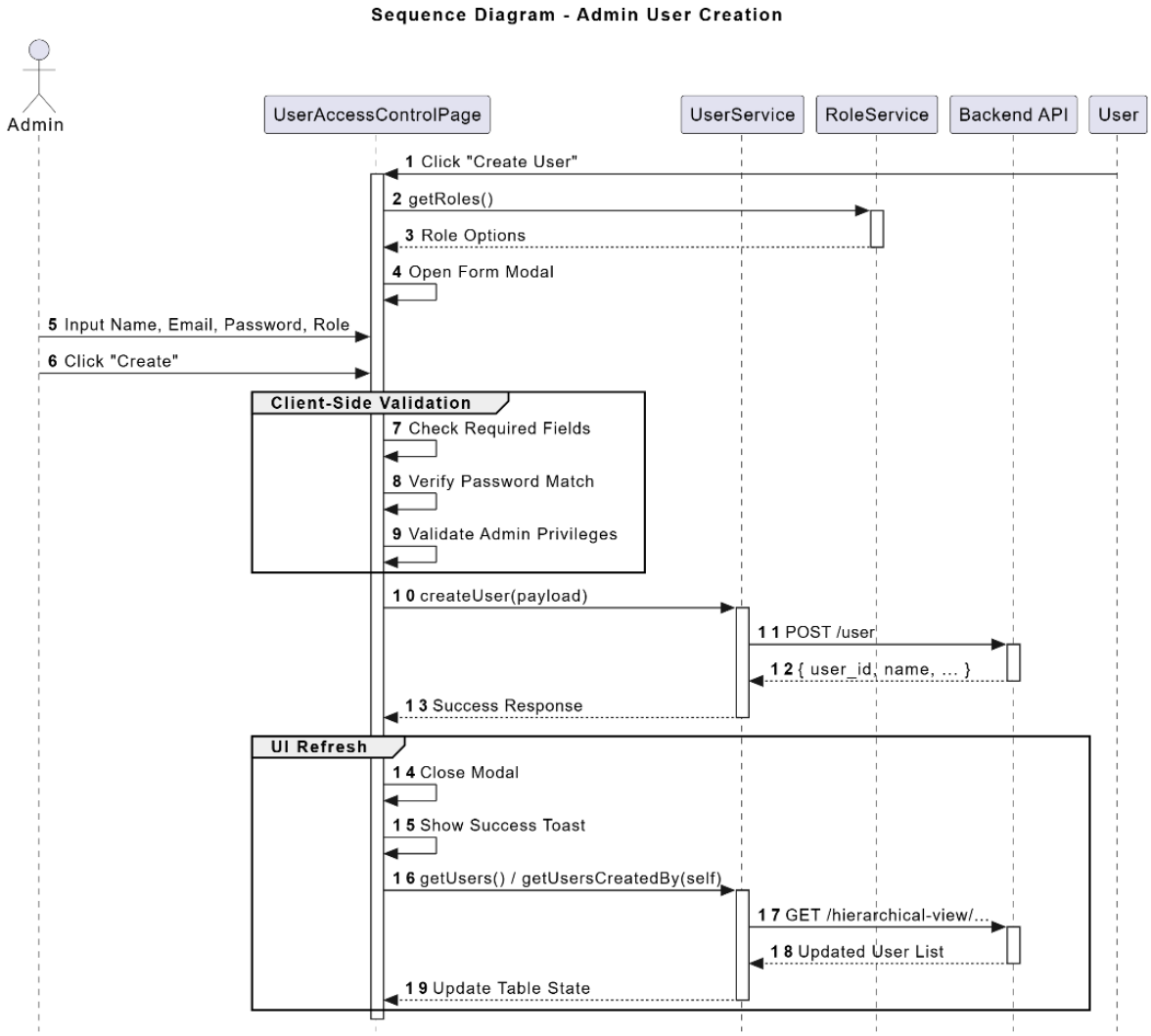


Figure 27: Admin User Creation Sequence Diagram

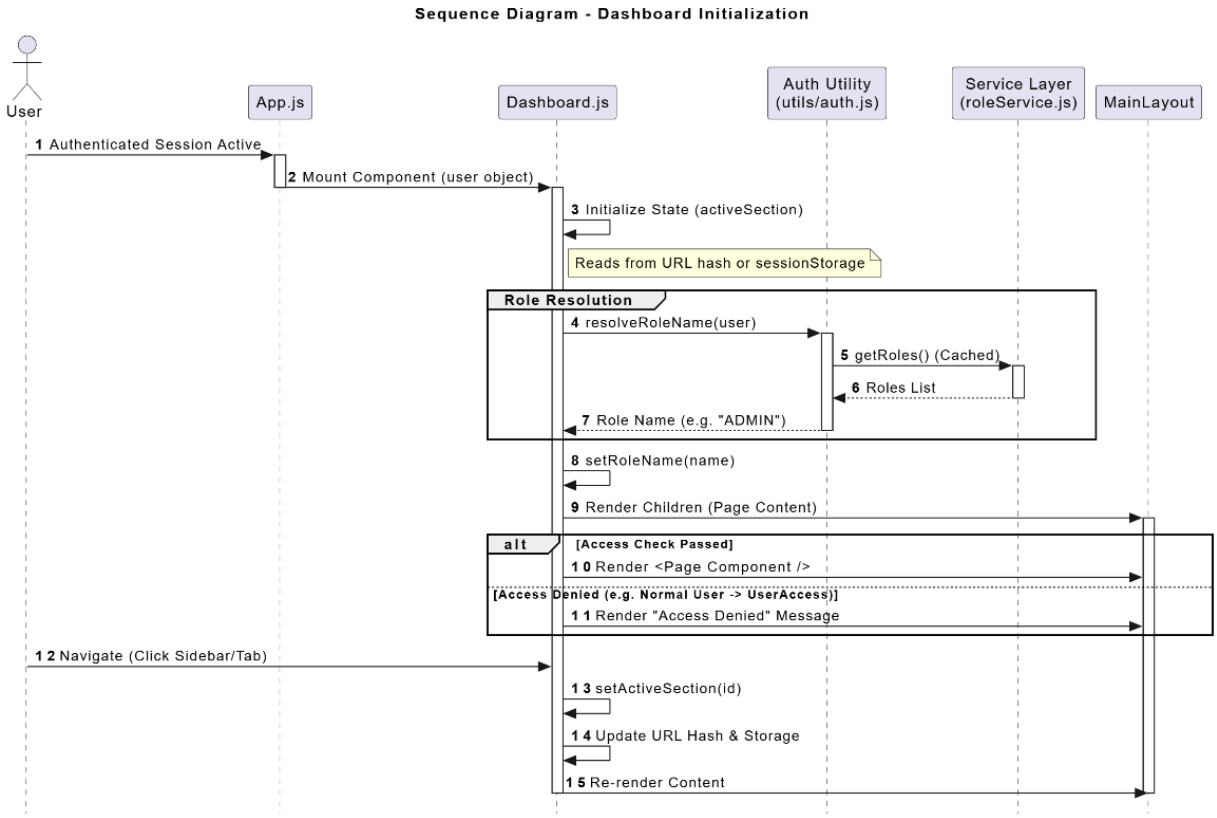


Figure 28: Dashboard Initialization & Routing Sequence Diagram

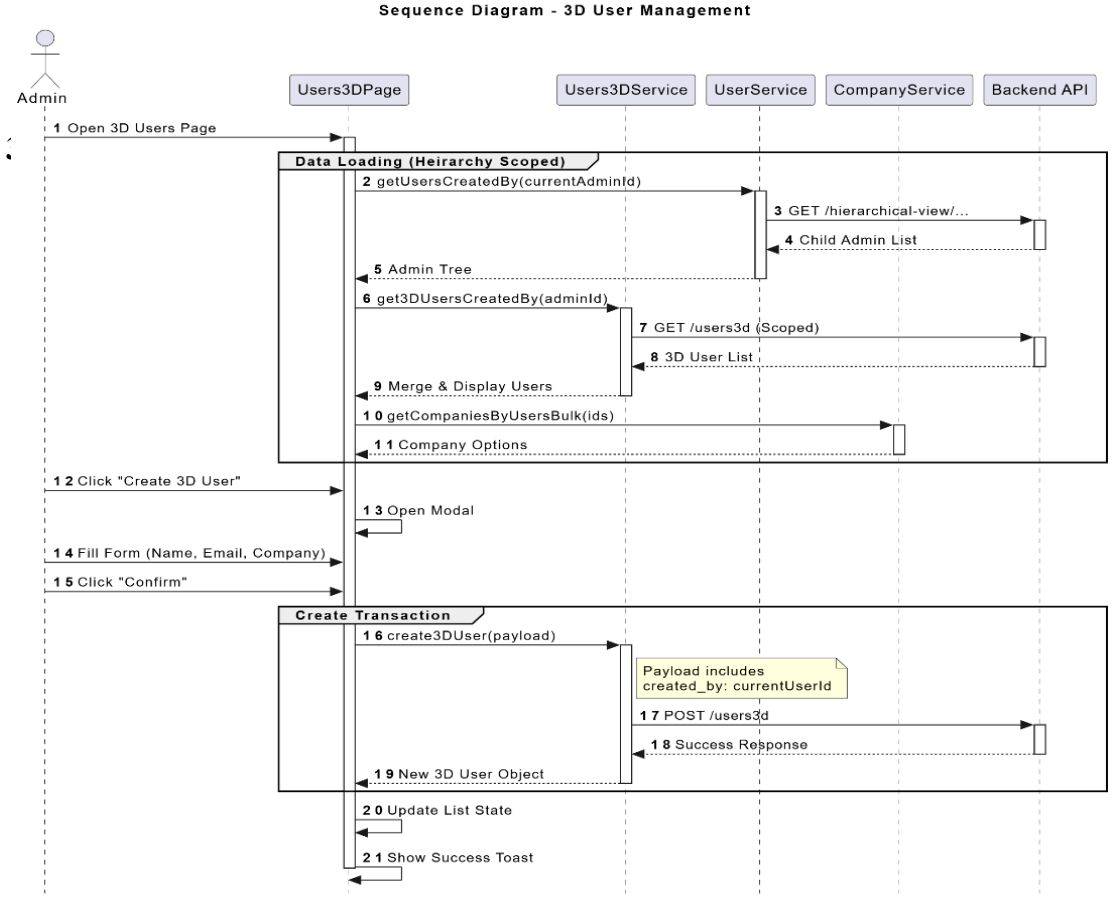
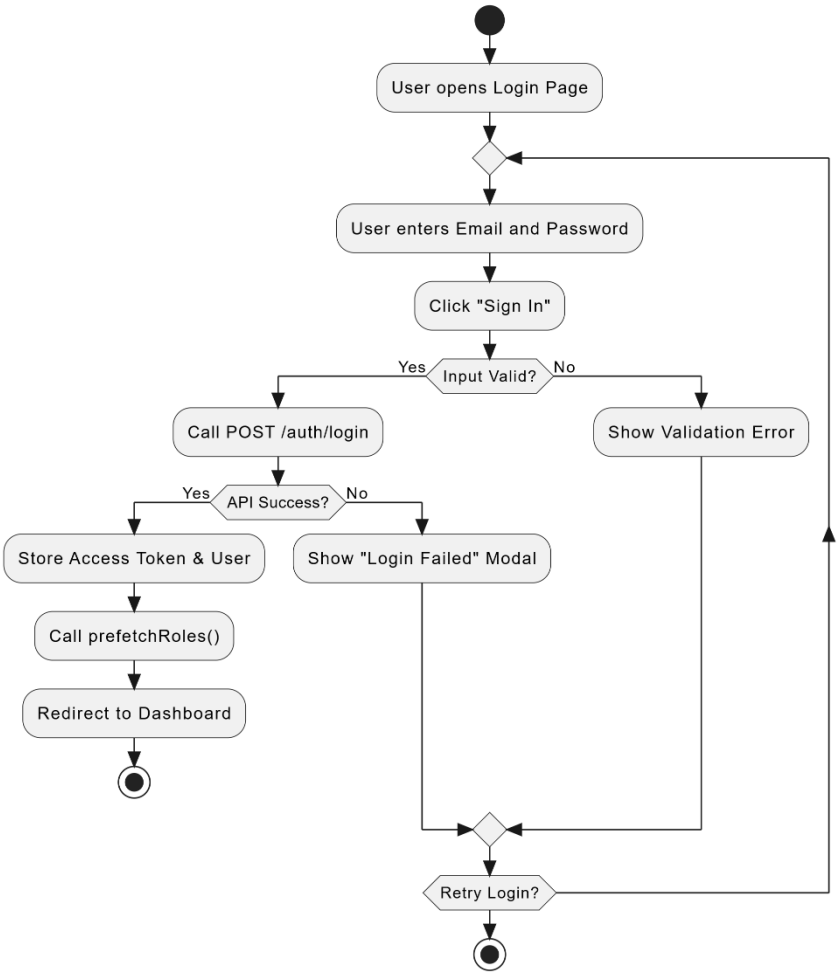


Figure 29: 3D User Management Sequence Diagram

Activity Diagram - User Login



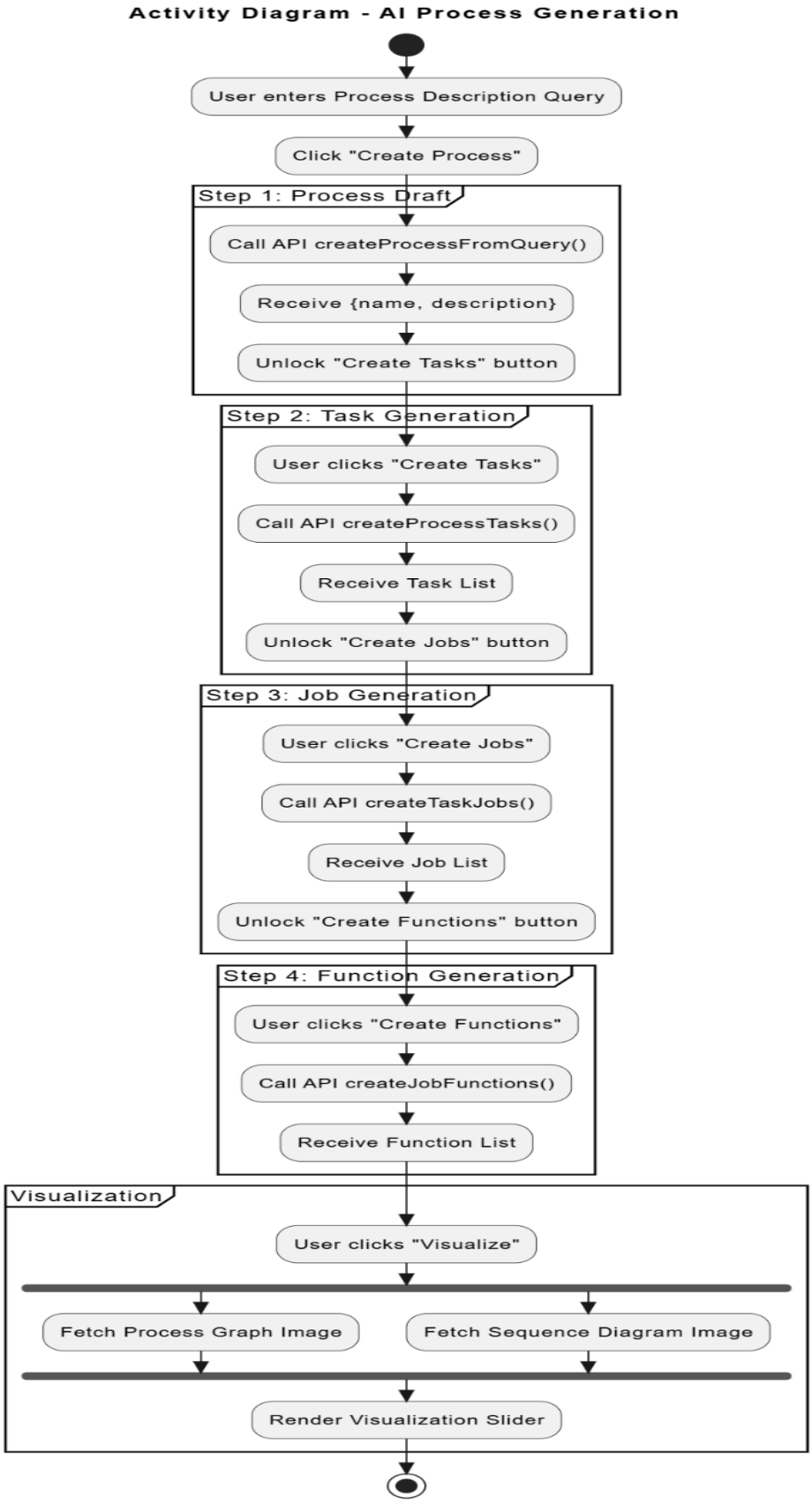


Figure 31: AI Process Creation Wizard

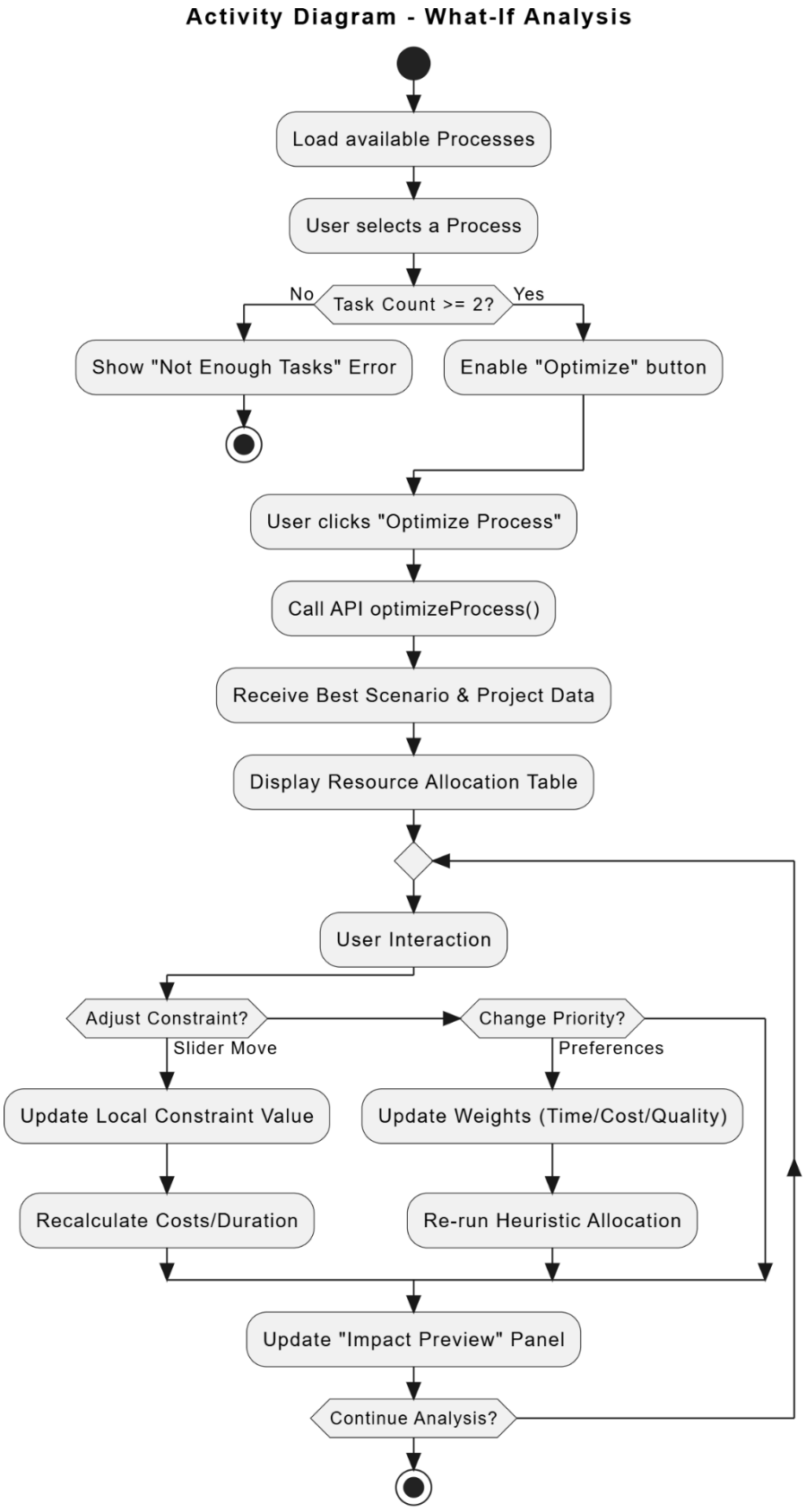


Figure 32: What-If Analysis & Optimization Activity Diagram

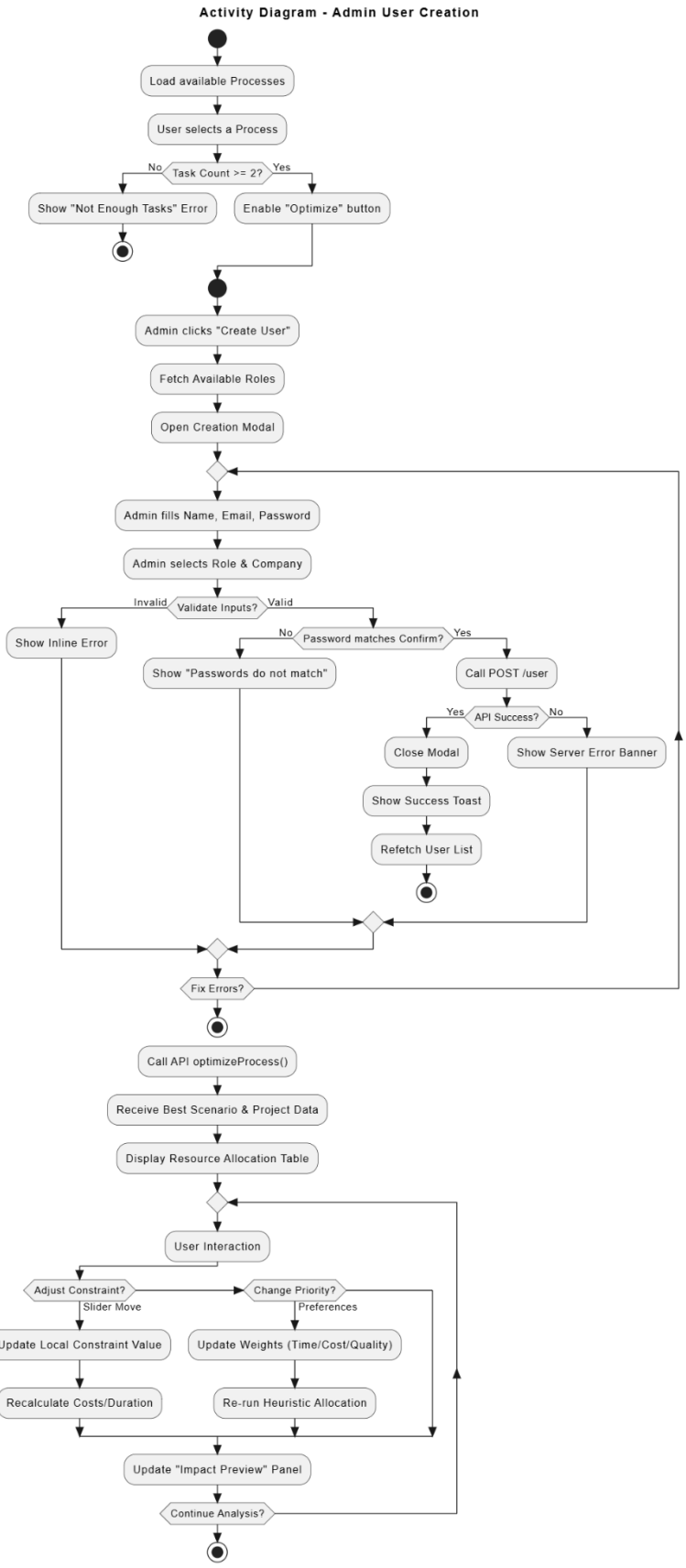


Figure 33: Admin User Creation Activity Diagram

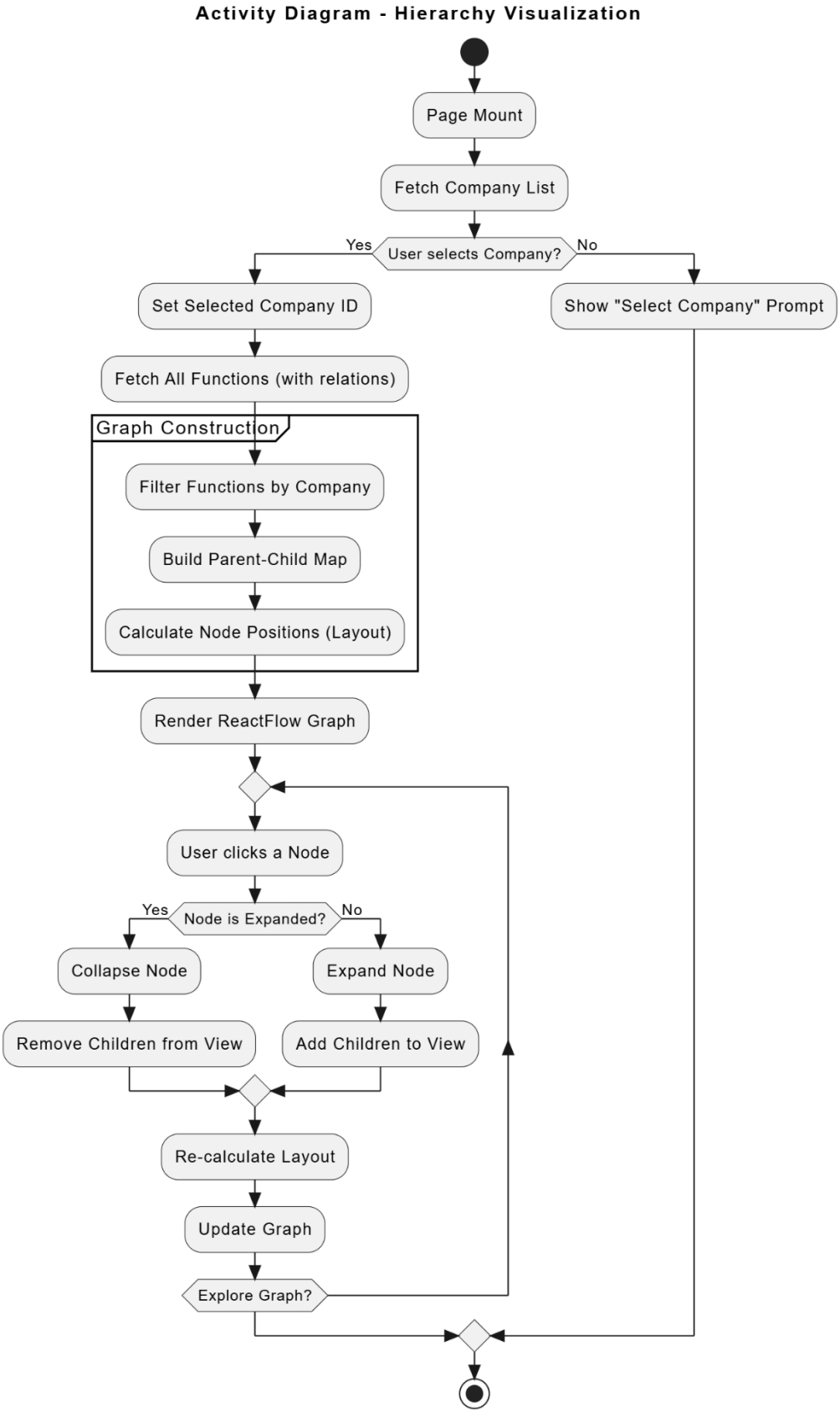


Figure 34: Hierarchy Visualization Interaction Activity Diagram

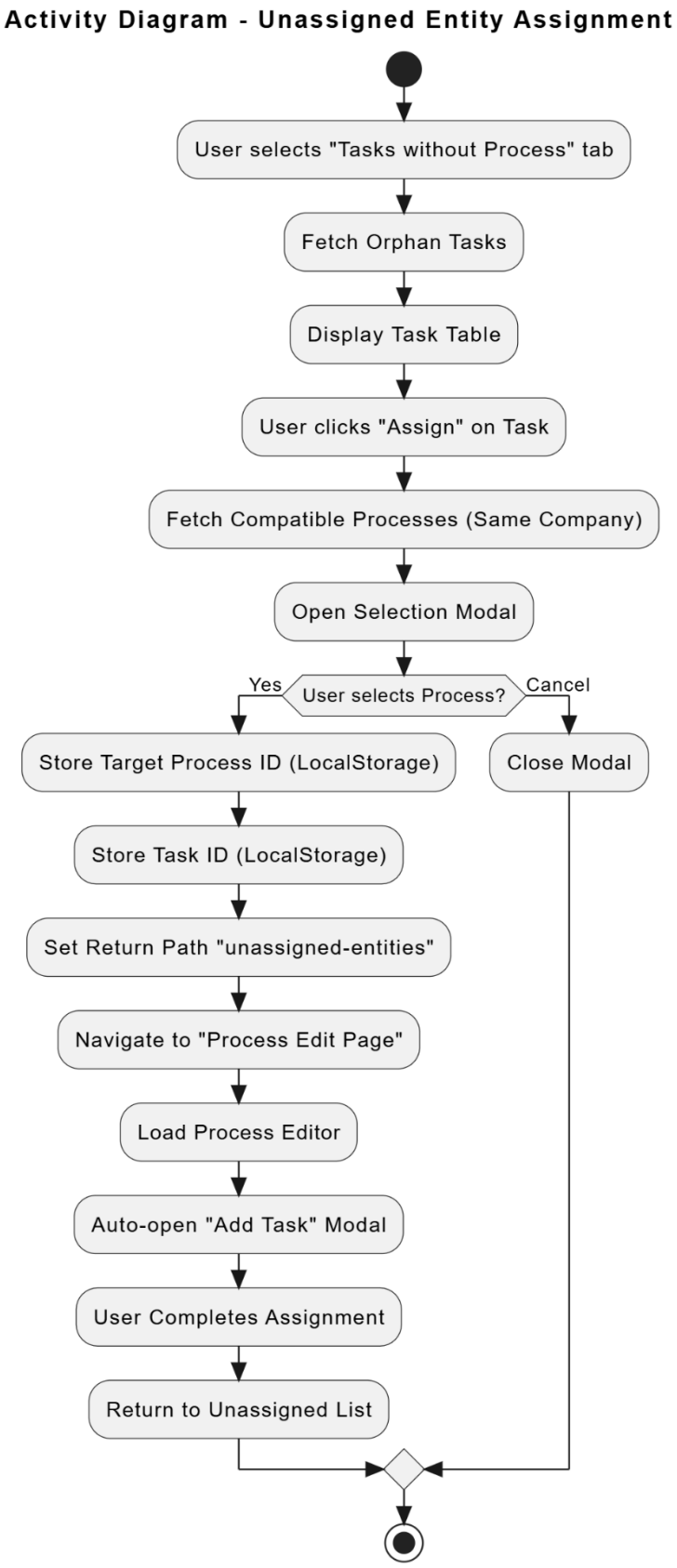


Figure 35: Unassigned Entity Resolution Activity Diagram

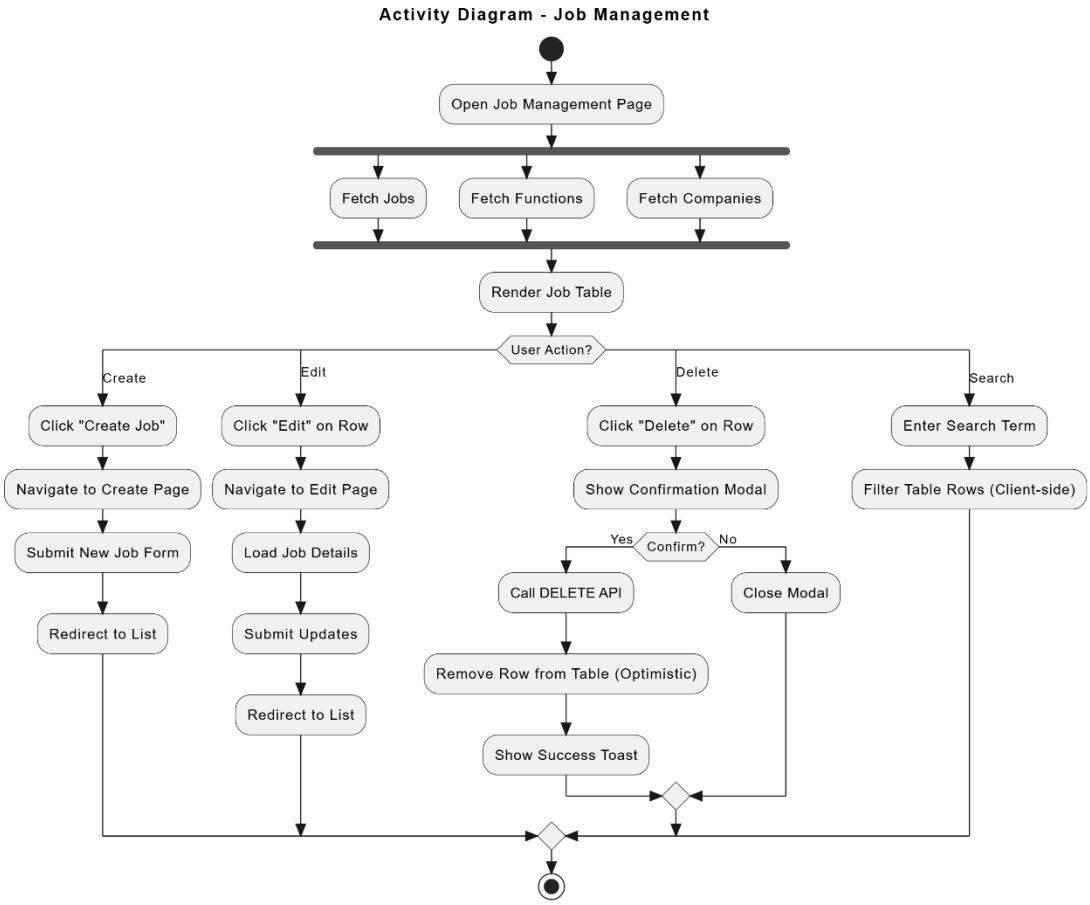


Figure 36: Job Management Lifecycle Activity Diagram

3.8 Conclusion

The current chapter is all about the analysis and requirements for the Crystal System CMS. It includes the actors and use cases, as well as information about the functional and non-functional requirements, and details regarding the need for interface, database, and feasibility. Analysis models have been included within this chapter for better understanding of workflows and interactions. The requirements presented in the chapter serve as the foundation for developing the system that will be introduced in the next chapter.

Chapter 4
System Design

This particular chapter talks about the design aspect of the Crystal System CMS. The methodology that was followed during the designing process of the system along with its architecture and views will be discussed in this chapter. In addition to the above discussion, there will be an examination of the composition of components, the data model, and the user interface design. In view of the fact that the application is being utilized as the web solution for the management of the enterprise, our approach to design was based upon modularity and scalability.

4.1 Design Approach

CMS Crystal System is designed according to a modular design approach, which means that the system is component-based and API-oriented. This approach suggests developing the system as a modern web application when the frontend is responsible for users' interaction and visualization, while the backend takes care of the business logic and data storage through APIs.

This solution suits our case since there are many different modules in the system which have some connections to each other, like dashboard management, process management, user access management, facilities management, KPI monitoring, and auditing. Thus, the modular approach allows us to develop each module individually.

The decision about the development approach was taken according to the following criteria:

- The separation of concerns, which ensures that the UI component, routing, state, and API calls stay independent of each other.
- The reusability of the components, including modals, input fields, navigation menus, and tables, to incorporate them into different modules.
- Scalability to ensure that further modules and services can be incorporated later on without changing anything about the entire architecture.
- Security concerns through role-based security, meaning only the right individuals are allowed access to each module.

- The user-centric approach, which led to the development of a relatively straightforward interface for the management of the entire process by the administrative staff.

It is worth mentioning that we have adopted an iterative development approach in our project, which means that only the basic setup is designed initially, followed by adding more and more modules and processes.

4.2Design Constraints

Indeed, there have been certain restrictions that affected the development of the system.

First of all, due to the fact that the system operates with backend APIs for the authentication of users, CRUD functionality, analytics, and workflows, it became necessary to make the system reliant on them.

Secondly, given the restrictions in terms of time due to the fact that we had to develop the system in the context of a Final Year Project, it became necessary for us to provide the system with the ability to work in terms of web-based technologies.

Thirdly, some functionalities of the system can only be developed through independent services.

Finally, while developing the frontend, we needed to make it as simple as possible from the viewpoint of academics.

4.3System Architecture

The Crystal System CMS system is built using the client-server architecture, which implies that there is a clear segregation between the presentation layer, the application layer, and the database layer.
There are three main elements:

4.3.1 Architectural Overview

The system consists of three main layers:

- **Presentation Layer**

  This is the React.js frontend that takes care of user interfaces, pages, forms, dashboards, and modules.
- **Application / Service Layer**

  Such services include authentication services, process management services, and dashboarding services, amongst others. Such services also assist with functions like user management, KPI management, and auditing.
- **Data Layer**

  Permanent data in this layer consists of permanent data entities including users, roles, organizations, and processes. Some examples of other permanent data entities are tasks, buildings, asset tasks, KPIs, and auditing.

4.3.2 Architectural Characteristics

These are some of the features which can be included with this design:

- Development of decoupled front-end modules.
- Service file-based API communication.
- Authorized route protection.
- Visibility of modules according to user role.
- Business modules scalability.
- Reusable interface components.

This kind of architecture works well for applications which require different modules to be used via a single interface.

4.4Logical Design

The Logic Diagram illustrates the internal structure of the system. The Crystal System CMS can be divided into multiple domains logically.

4.4.1 Main Functional Domains

- Authentication and Session Management
- Dashboard and Reporting
- Organization and Location Management
- Facility Management
- People, Users, and Role Management
- Process, Task, Job, and Function Management
- BPMN Visualization and Workflow Editing

- KPI and Unit Management
- Asset Task Management
- Audit Logging and Monitoring

Each domain has its own set of pages, services, and UI elements.

4.4.2 Domain Model

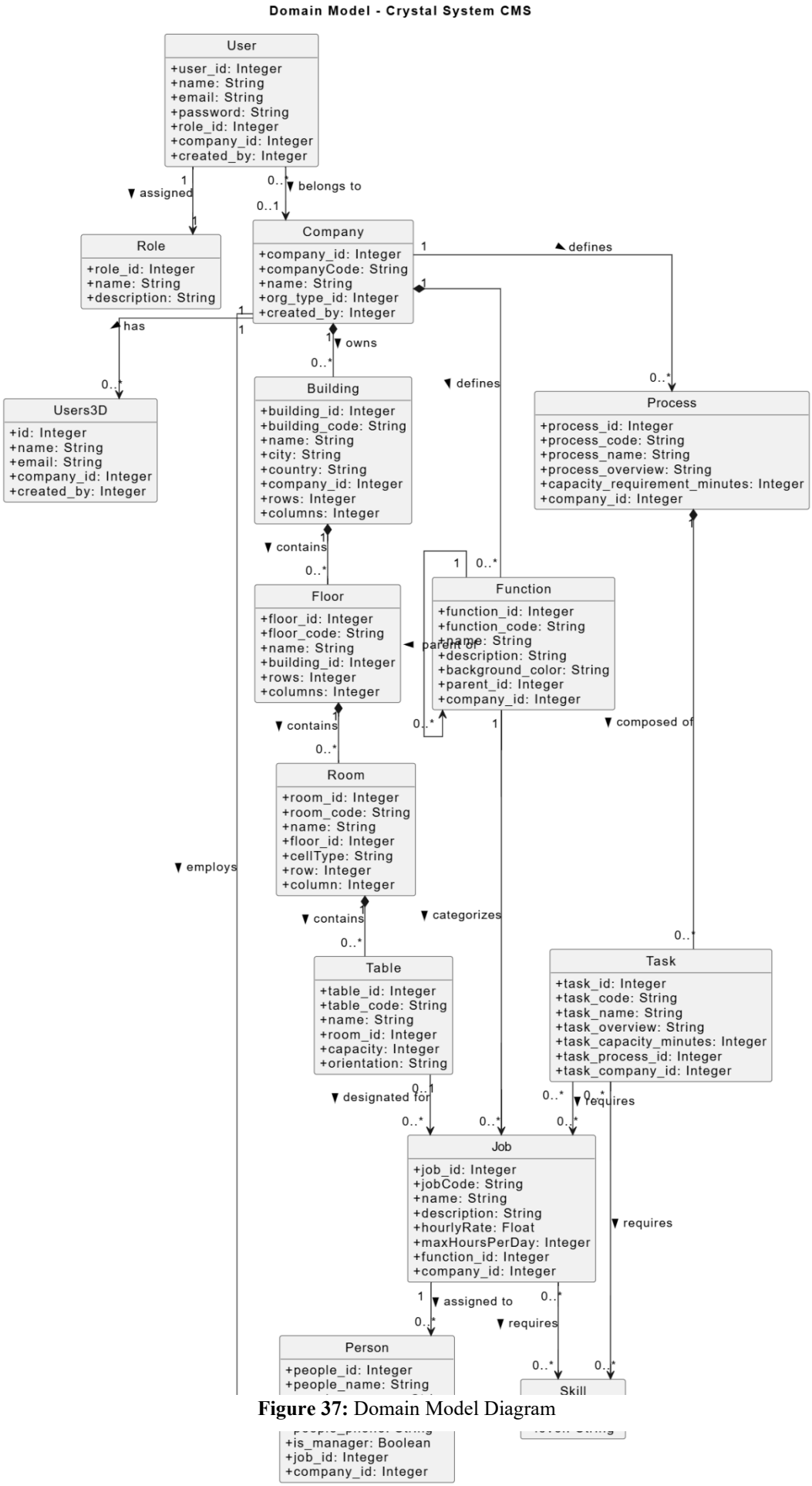


Figure 37: Domain Model Diagram

4.4.3 Logical Design Discussion

Taking into account the logical structure of the solution from the point of view of its composition, we should consider that the solution depends on the organizational and operational aspects. The company becomes the unit of organization, and operations, actions, activities, and resource activities are the operational parts. Control capabilities are provided by the users and their roles, while audit trails allow tracking events.

4.5Dynamic View

The dynamic aspect shows interaction between elements.

4.5.1 Login Interaction Sequence

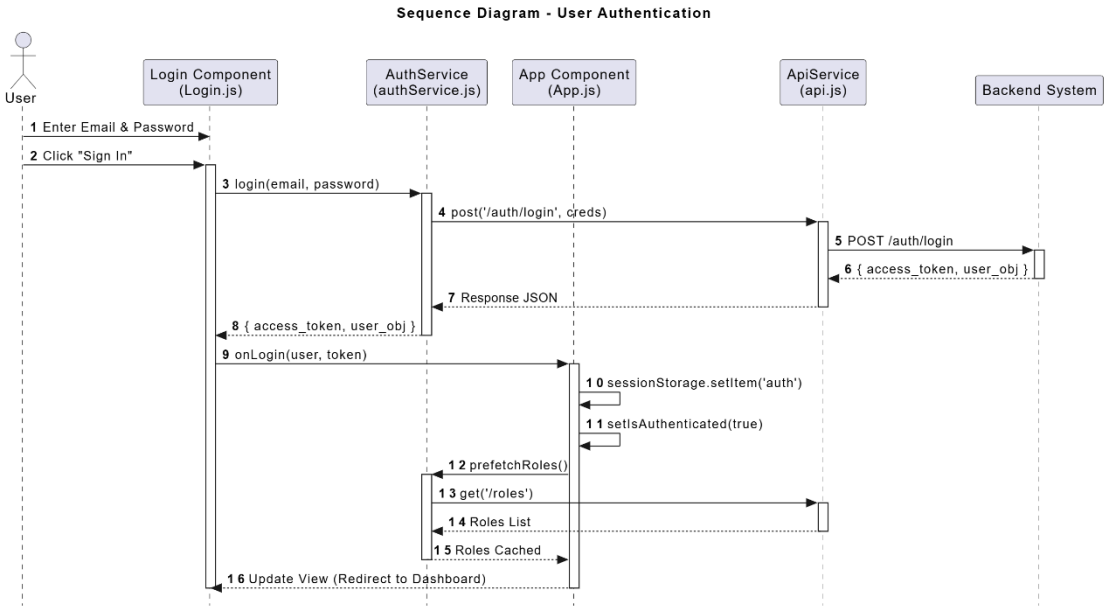


Figure 38 : User Authentication Sequence

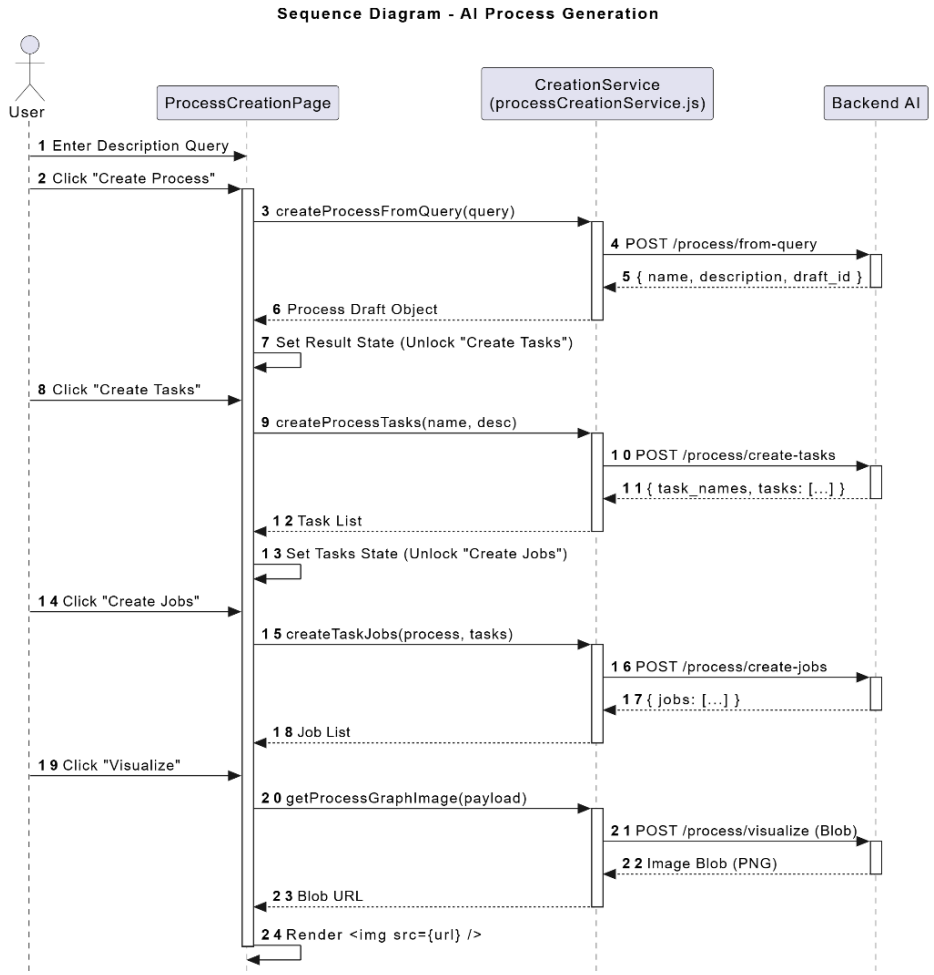


Figure 39: AI Process Creation Workflow

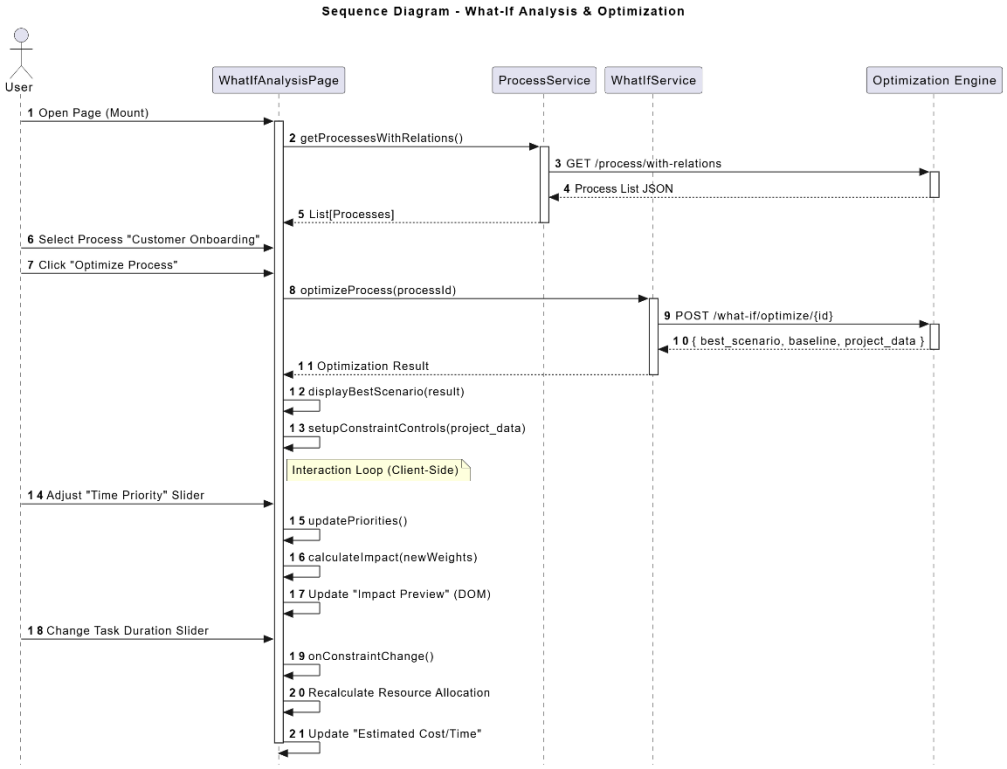


Figure 40: What-If Analysis Optimization

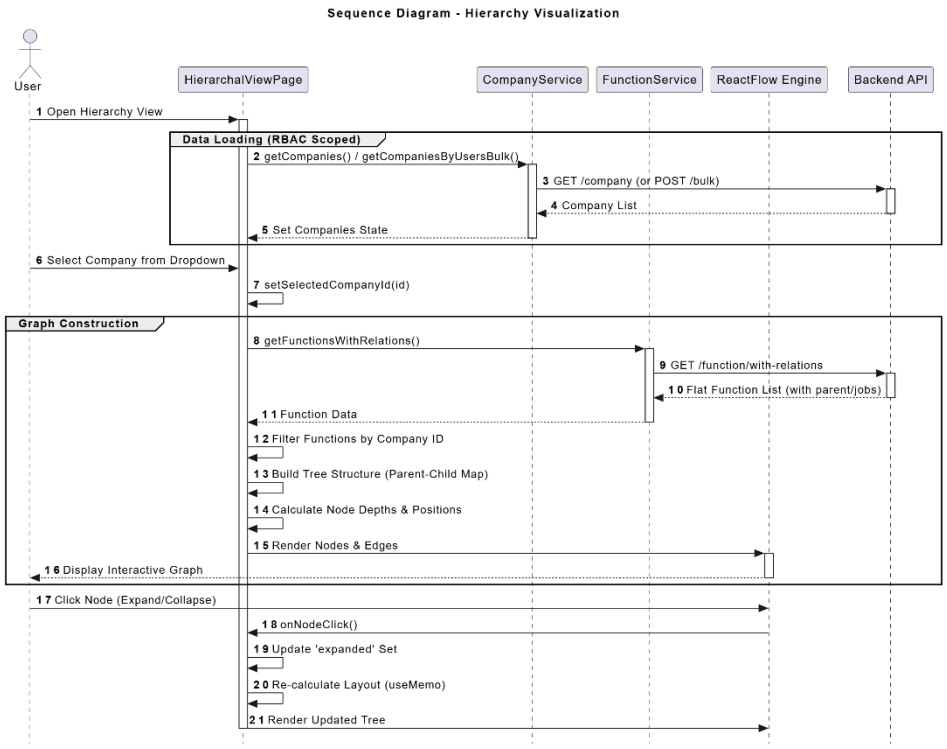


Figure 42: Hierarchy Visualization Sequence

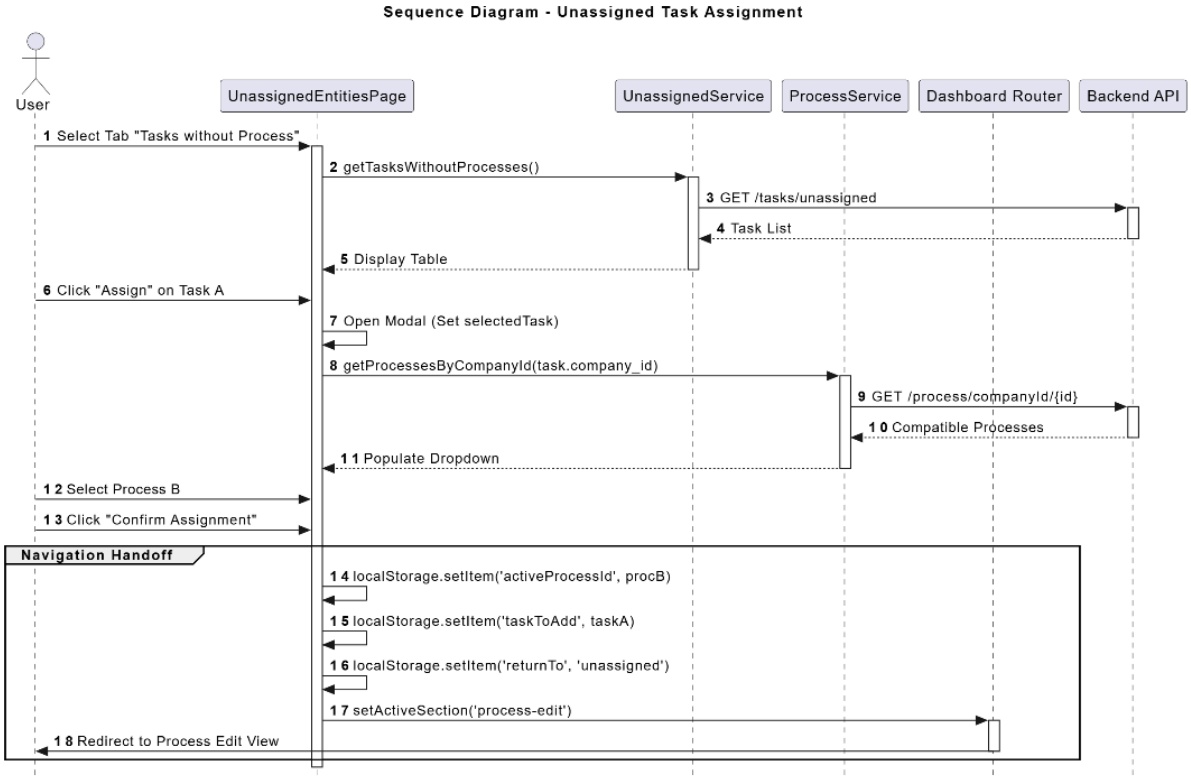


Figure 41: Unassigned Task Assignment Sequence

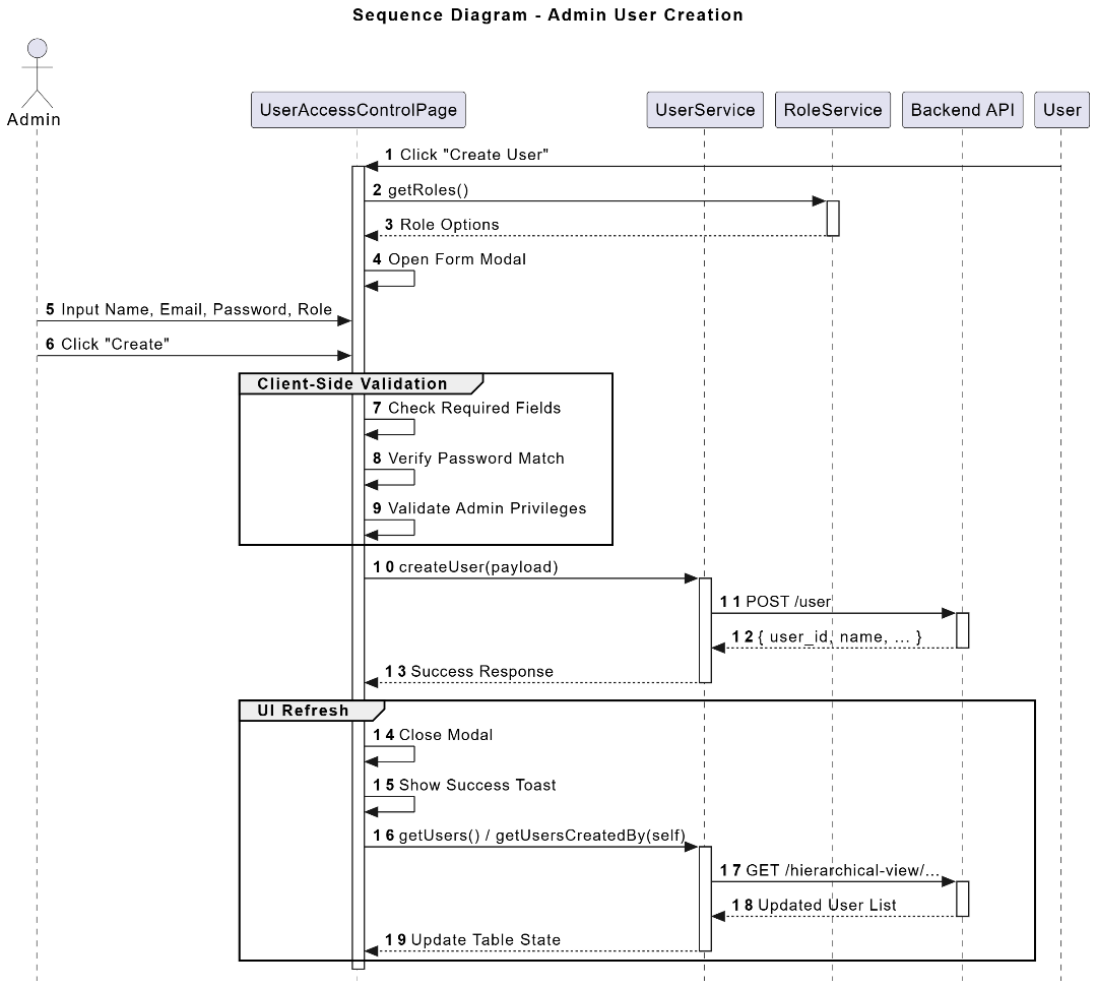


Figure 44: Admin User Creation Sequence

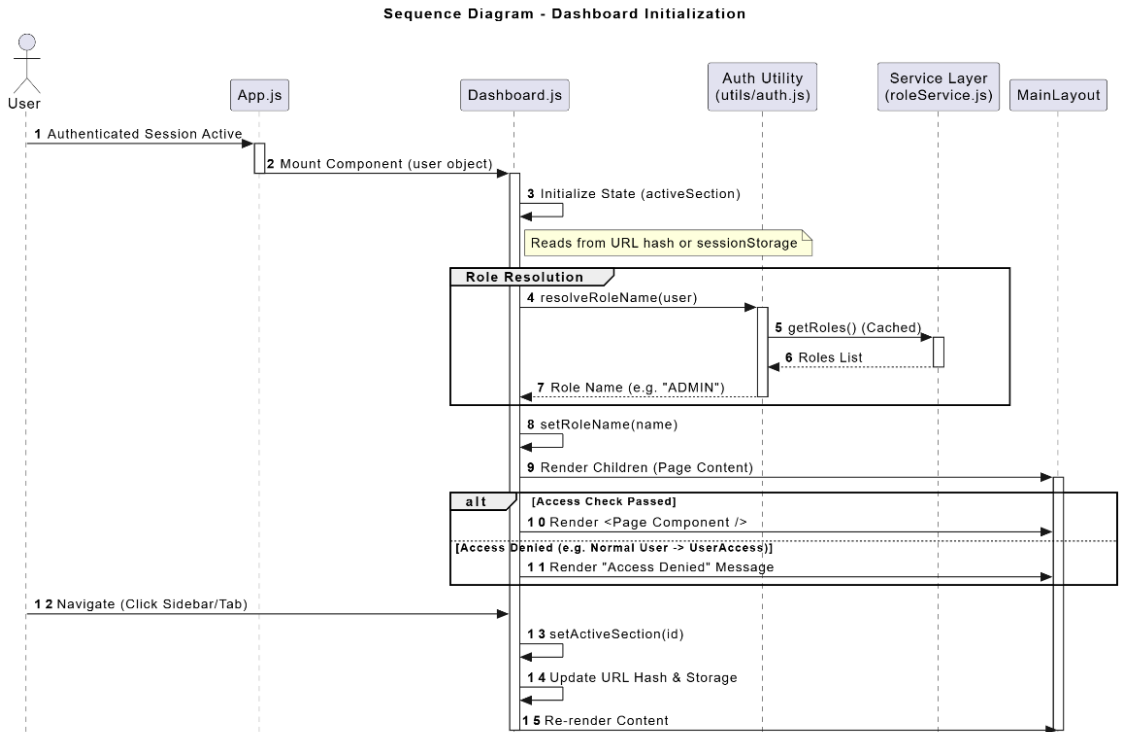


Figure 43: Dashboard Initialization & Routing Sequence

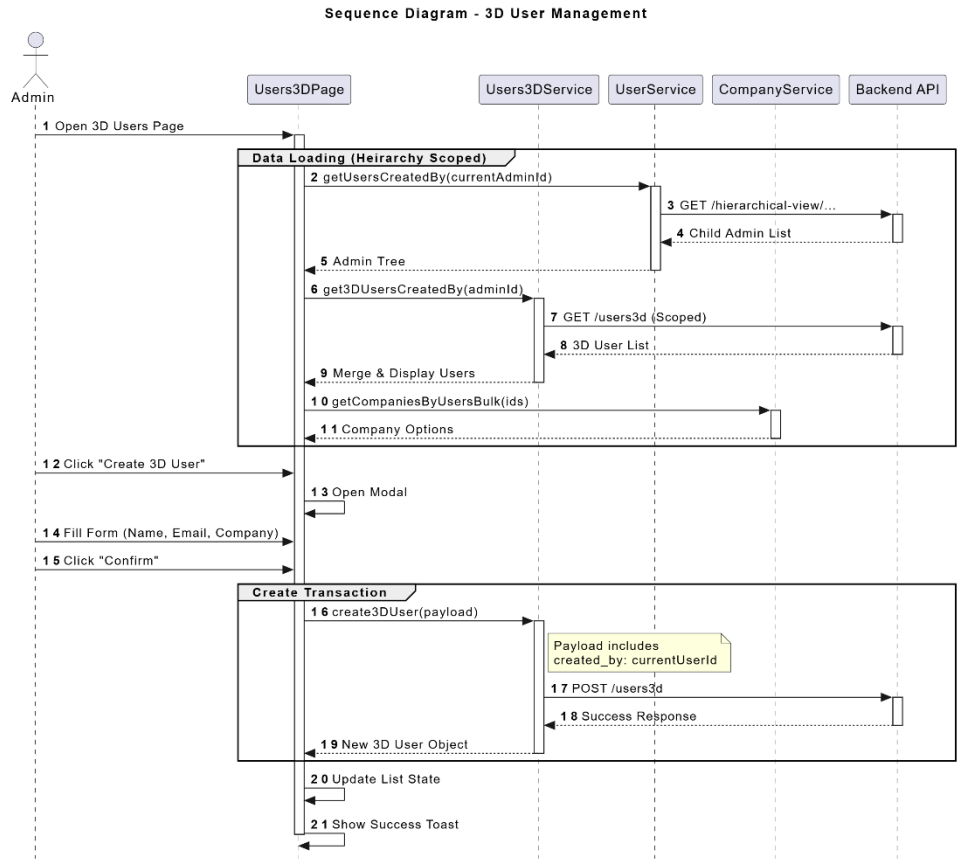


Figure 45: 3D User Management Sequence

4.5.2 **Dashboard Runtime Behavior**

Once the whole process kicks off, the dashboard retrieves data according to the user type. For instance, Super Admin would have access to the total number of counts, but for Admin or Normal Users, it will be determined through different filters or hierarchies. All security and visibility issues will be resolved through this process.

4.5.3 **BPMN and Process Visualization Flow**

BPMN element interacts with the workflow data through importing the process data and BPMN XML files in order to render them in the browser. It is now possible for you to visualize and modify the process without having to implement any workflow data within the user interface.

4.6Component Design

The system architecture is defined as a composition of modules and packages.

4.6.1 Package Structure

The packages involves packages such as:

- components
- pages
- services
- context
- routes
- styles
- utils

The system structure enables easy maintenance because each package serves a specific purpose.

4.6.2 Interface Design Between Components

The major communication interfaces between the components are:

- **Page-to-Service Interface**
  Information from each page is provided or sent through the use of a service file on each page
- **Layout-to-Page Interface**
  The layout module handles navigation, highlights the active part, and handles the common context.
- **Context Interface**
  Common context allows anything that is commonly shared like the section, ID selection, and form mode to be passed through.
- **UI Component Interface**
  Some components can be reused in modals, forms, editors, and tables to get their info and commands from the pages.

4.6.3 Deployment Scenario

The deployment process is achievable by employing a browser-based frontend which accesses the backend APIs and databases using the webserver.

1. **Deployement Diagram**

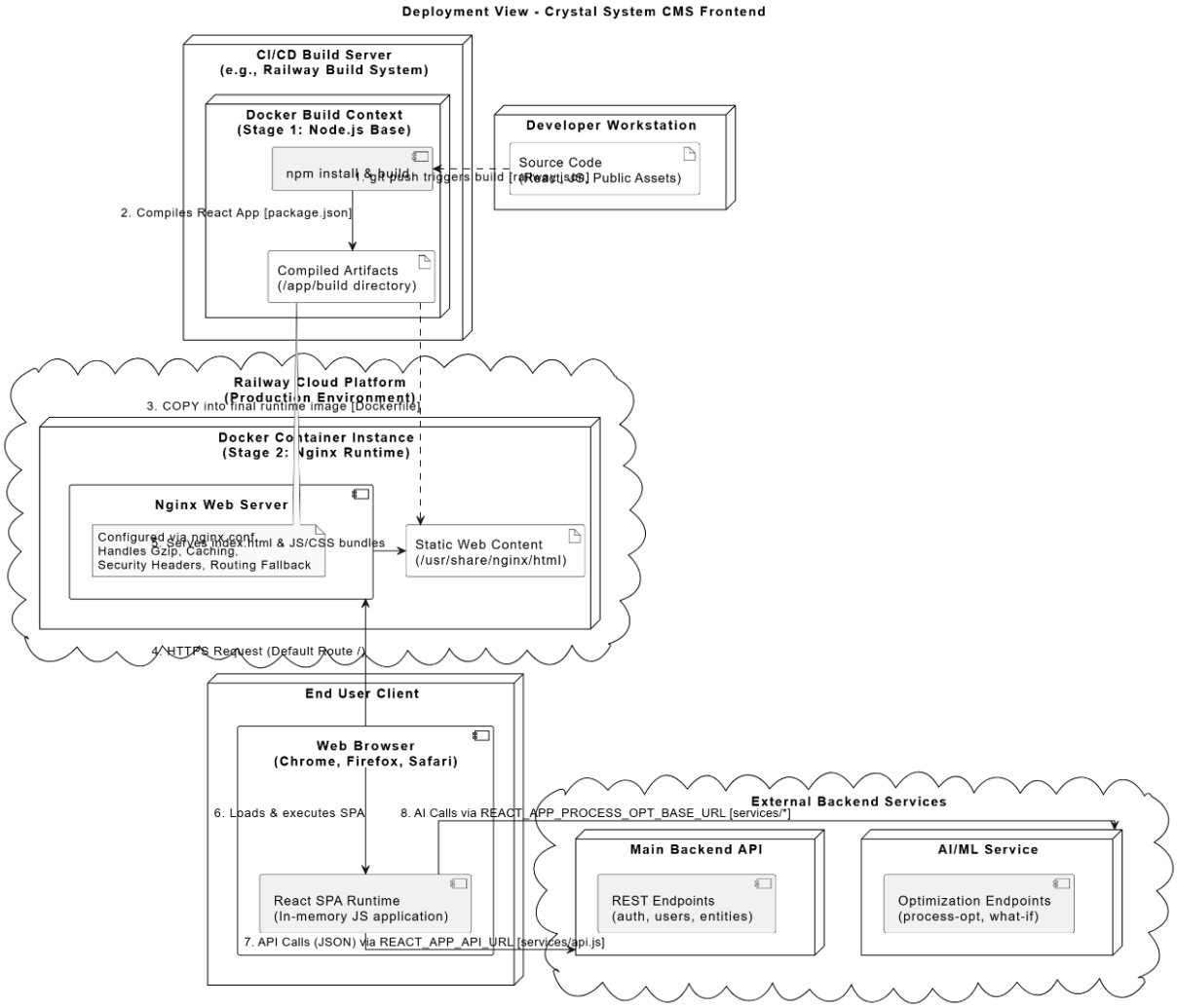


Figure 46: Deployment Diagram

4.7Data Models

This refers to how data will be structured in storage.

4.7.1 Conceptual Data Model

At the conceptual level, the system contains five major categories of data:

- User and Security Data
- Organizational and Geographical Data
- Facility Data
- Process and Workflow Data
- Monitoring and Logging Data

4.7.2 Logical Data Model

Entities and their associations, which are crucial for determining the logical data model, are as follows:

- One role can be allocated to many users.
- One company can have many buildings.
- One building can have many floors.
- One floor can have many rooms.
- One company can have many processes.
- One process can have many tasks.
- One task can have many jobs.
- One process can have KPIs and Asset Tasks.
- One user can create many Audit Logs.

4.7.3 Physical Data Considerations

With regard to the physical model:

- indexed primary keys,
- foreign key constraints,
- timestamp fields,
- normalized tables for controlled redundancy,
- efficient query support for dashboards and filtered views.

When discussing the physical model, another factor that must be taken into account is the expandability of this model.

4.8 User Interface Design

The interface design of Crystal System CMS emphasizes ease of navigation and quick access to various modules.

4.8.1 Interface Structure

The user interface design includes the following components:

- **Login Screen**
  Authentication module for logging into the system.
- **Dashboard**
  Statistics and module selection page displaying different modules.

- **Sidebar / Navigation**

  Navigation into the different modules like process, user, KPI, facility, and audit trail.
- **Module Pages**

  These modules have list view, detail view, and form view for entering details
- **Reusable Dialogs and Modals**

  Pop up windows for confirming any action, filling forms, importing and exporting data, and editing through specialized editors.

4.8.2 UI Design Goals

The design of the user interface was guided by the following goals:

- simplicity of navigation,
- consistency across pages,
- reduced learning effort,
- clarity of forms and management actions,
- quick access to critical administrative tasks.

4.8.3 Wireframe-Oriented Layout Description

A typical page will have:

- header with title and user controls,
- left-side navigation area,
- main content section,
- action buttons such as add, edit, delete, import, export,
- data table or card-based content display.

4.9System Prototype

From the developmental perspective, the process involved developing a number of different models rather than developing a single model and keeping it constant throughout.

4.9.1 Low-Fidelity Prototype

By this stage, we had managed to employ the fundamentals of page flow, grouping of modules, and navigation techniques to express our understanding of the system. From these basics, we could identify the modules required for the system, namely the dashboard, processes, users, and facilities.

4.9.2 High-Fidelity Prototype

The front end created with React technology is regarded as a high-fidelity prototype since it consists of genuine page designs, navigation buttons, and forms. Moreover, this prototype includes services, limited access routes, BPMN diagrams, and analytics. With the help of this advanced prototype, we managed to evaluate how easy it would be for users to navigate the platform, understand interactions between different modules, and execute workflow activities.

4.9.3 Prototype Benefits

The prototype-based method helped in the following ways:

- Organization of the modules,
- Flow of pages,
- Checking if the website is user-friendly,
- Checking whether the navigation menu works efficiently,
- Interface design validation in relation to the project requirement.

4.10 Conclusion

In this chapter, the designing of Crystal System CMS has been explained in different dimensions like architectural dimension, logical dimension, dynamic dimension, and many other dimensions like component perspective, data perspective, and user interface perspective. This particular system is designed based on an architecture that is modular in nature and API driven, wherein the front end will be separate from the back end and modules can work together with each other in the enterprise. Each and every dimension of this particular architecture contributes to the success of the project in different ways. In the next chapter, implementation and testing of this particular system will be explained.

Chapter 5
System Implementation

This chapter provides a technical description of the development process of the Crystal System CMS. In this section, the technical approach towards developing Crystal System CMS is explained along with its functions and their development process. Since the Crystal System CMS is a web application that helps businesses manage their operations effectively, the development process focuses on the development of the frontend module along with the use of APIs.

5.1 Implementation Approach

The implementation of Crystal System CMS follows a **modular frontend architecture** integrated with backend APIs. The system was developed as a React-based single-page application in which each major business domain is implemented as a separate page or module. This approach was selected to keep the system manageable and scalable as the number of modules increased.

The implementation strategy focused on:

- dividing the system into reusable components,
- separating business communication into service files,
- organizing module navigation through routing,
- controlling access through authentication and role-based checks,
- supporting dynamic workflows such as BPMN visualization and what-if analysis.

This approach made it possible to implement a large number of enterprise features without tightly coupling the user interface with backend logic.

5.2 Tools and Technologies Used

Modern technology has been used for developing the dashboards.

5.2.1 Programming Languages and Core Technologies

- **JavaScript** was used as the main programming language.
- **React** was used to build the frontend user interface.
- **HTML and CSS** have been utilized for designing the dashboard.
- **Fetch API** was utilized for communicating with the server etc.

5.2.2 Libraries and Frameworks

The libraries used in building the application included:

- **React 19** to build the front end through component usage.
- **React Router DOM** to manage routing within various pages.
- **bpmn-js** for BPMN workflow visualization and manipulation.
- **React Flow** for process visualization via flows.
- **D3-Geo and TopoJSON** Client for map visualization on the dashboard.
- **Testing Library** for testing purposes on the front end.

5.2.3 Development and Deployment Tools

- **Node.js and npm** to manage dependencies and configure the development environment.
- **Create React App / React Scripts** for build and development workflow.
- **Docker** for containerization support.
- **Nginx** for serving the production frontend build.

This combination of technologies has been selected because it is reliable, well-documented, and appropriate for developing an enterprise-level web application.

5.3 Development Process

Development happened using the incremental approach. We did not implement all modules in the project in one go but rather gradually, one after another. The main stages of development were as follows:

Firstly, authentication and routing were implemented in order to secure access to the system.

Secondly, we designed the dashboard and layout to ensure proper navigation.

Thirdly, we provided the implementation of various core management modules which included the following:

- companies module,
- buildings module,
- people module,
- users module,

- roles module,
- processes module,
- tasks module,
- jobs module.

Fourthly, further development included implementation of advanced modules such as the following:

- BPMN viewer,
- process optimization,
- KPI management,
- audit log management,
- asset task management.

Finally, reusable components were developed which included the following:

- forms,
- modals,
- editors,
- confirmation dialogs.

5.4Frontend Implementation Structure

The frontend part of the project was implemented according to the structure of folders. There are two main categories of implementation:

- **Pages**

  There are full modules such as Dashboard, Processes Management, User Management, KPIs Management, Buildings and Audit Log.
- **Components**

  These are reusable components like Layout, Sidebar, Header and modals. Also there are BPMN Editor and Gateways Configurator here.
- **Services**

  This implements all interactions with API that happen between several parts such as Login, Dashboard, Processes, Users, Buildings, KPIs and Audit Log.
- **Context**

  This uses React context to store state-wide data such as active section/selection.

- **Routes**

  Routes that bind certain section with URL and guarantee its security are created here.
- **Styles**

  Every module includes own css files

This helped to make maintenance simpler as well as to separate code by its purpose.

5.5Implementation of Core Features

5.5.1 Authentication and Session Handling

For authentication, the process involved using a login form that would subsequently connect to the backend service. For a successful log-in, the information about the session would be saved to monitor the access control process, while cookies ensure the safety of transactions. There is also security for the secured routes, which ensures no one else gains access except the user who has the right authorization.

The login feature also includes:

- Validation,
- Error handling,
- Feedback when login fails,
- Timeout after many failed login attempts.

5.5.2 Routing and Protected Navigation

The solution uses client side route mechanism which is responsible for navigating from one module to the other. The route mechanism is implemented using routing configuration developed specifically to make sure each individual route is linked to various areas within the application including dashboard, process management, task management, kpi management, auditing and buildings management.

This guarantees that:

1. each module can be accessed directly using URL,
2. route normalization,
3. selection of section stays constant throughout navigation,
4. security redirect to login page if required.

5.5.3 Dashboard Implementation

- The dashboard is set up as the main access point when the user logs into the system. It obtains the numbers of companies, processes, tasks, and jobs

through the backend. The capabilities of the dashboard change with the kind of user. The top-level users will see information about the whole system, but others will only see filtered information.

Capabilities offered by the dashboard include:

- summary card navigation
- import/export functionality
- location-based map view

5.5.4 Organization and Facility Management

Modules have been created for countries, cities, firms, buildings, floors, and rooms that consist of data pages with features such as create view, edit view, detail view, and list view. Each page communicates with its own service layer and refreshes itself based on the response generated.

This structure helps in creating an abstract representation of the business organization in reality.

5.5.5 People, Users, and Role Management

The modules of the application shall include the following:

- Personnel management,
- Access management for users,
- Role management,
- 3D User management/administration.

These modules shall manage the forms, validation, and lists such that only those items which the roles of the users permit them to use can be accessed. Passwords, user creation, and roles shall be managed by authorized users alone.

5.5.6 Process, Task, Job, and Function Management

One major part of the implementation is understanding how operational workflows work. The software enables:

- creation and editing of processes
- management of tasks/jobs
- management of functions
- management of categories/statuses
- association of process tasks with workflow elements

5.5.7 BPMN Viewer and Workflow Design

The BPMN capability was achieved using bpmn-js and services. The application can perform operations like obtaining process details, creating or loading BPMN XML, displaying process diagrams, and creating or saving BPMN process.

This phase in developing our application is very important because it will allow us to convert data about the business process into visual workflows. Moreover, there are additional features like gateways handling and chaining tasks that contribute greatly to the workflow component.

5.5.8 What-If Analysis and Process Optimization

Implementation of the What-If Analysis module enables you to choose processes and evaluate options related to optimizing resources. The importance of the module lies in its ability to help improve processes by evaluating scenarios of alternative resource allocation. Even though the optimization algorithms work backstage, it enables you to:

- process selection
- scenario display
- result comparison
- interface controls for adjustment and review.

This module gives additional analytical value beyond the usual managerial tasks.

5.5.9 KPI and Performance Modules

KPI-related modules were implemented to manage:

- KPI records,
- units of measure,
- KPI attributes,
- process KPI relationships.

Only the approved user is able to access the modules. This helps in keeping confidential performance standards secure, yet enabling organizational level metrics.

5.5.10 Audit Log Implementation

The audit module was implemented through the use of the service layer and management page. It enables the ability to view:

- login logs,
- action logs,
- failed logins,
- deletion logs,

- user-specific audit history.

5.5.11 Import and Export Features

The import/export functions within the system's dashboard and modules also have to be considered. Data import is done with file imports using multipart form data, while exports let you take data from the system to analyze elsewhere.

In this way, the system is more useful when dealing with administrative work involving large quantities of data.

5.6Implementation of Important Workflows

5.6.1 Login Workflow

Here is where the process starts. Your credentials get submitted through the login form. The credentials will then be sent to the backend by the authentication service. If your credentials happen to be correct, then your session will be stored, your status will be updated, and you will get redirected to the dashboard. If not, an error will be displayed.

5.6.2 Process Management Workflow

While managing processes, one may have the opportunity to open the process list, where he/she can add, modify, or view the process information. Once he/she adds or modifies the process, the information filled out in the form is put together into a package that contains process information, workflow information, and gateways. The package is then sent by the process service to the backend.

5.6.3 BPMN Workflow

When the BPMN component is loaded by the user, the system retrieves the flow meta-data and BPMN data. The diagram visualization is then displayed on the browser window. Upon enabling edit mode, the user can modify the flow meta-data and then save it into a BPMN XML file.

5.6.4 Audit Review Workflow

The administrator will have permission to visit the audit log page and choose the type of log he wishes to see and also filter the information based on different parameters like users, duration, or types of actions taken. The system will bring all logs that meet the desired criteria and show them to the administrator.

5.7 Build and Deployment Support

It also includes facilities for build/deployment, allowing for deployment in a production environment. As an example, the frontend can be made production-ready by using traditional React build scripts. Besides, we have also included some Docker and Nginx scripting files for deploying the system on a server.

In this way, its usefulness improves because it indicates the suitability of the solution in real-world application.

5.8 Challenges During Implementation

Problems encountered during the implementation:

- The integration of several modules while maintaining consistency of navigation,
- Maintaining role-based limitations across several pages,
- Dealing with several back-end responses by means of normalization,
- Combining process management with BPMN diagrams,
- Reusability of components for working with several modules.

All these problems were solved thanks to the modular approach, use of standard services, routing, and continual improvement of the user interface.

5.9User Interface

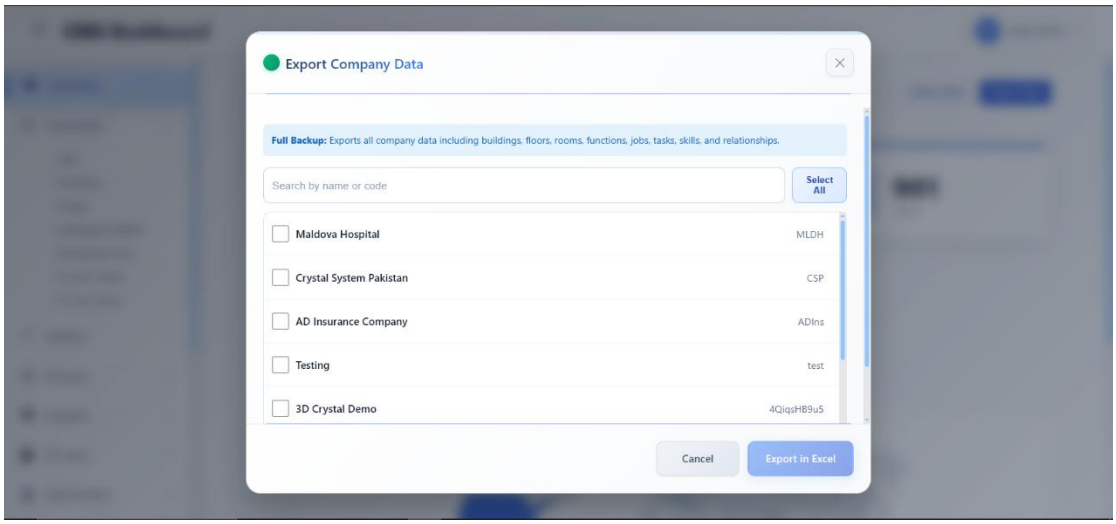


Figure 47: Export comapany data page

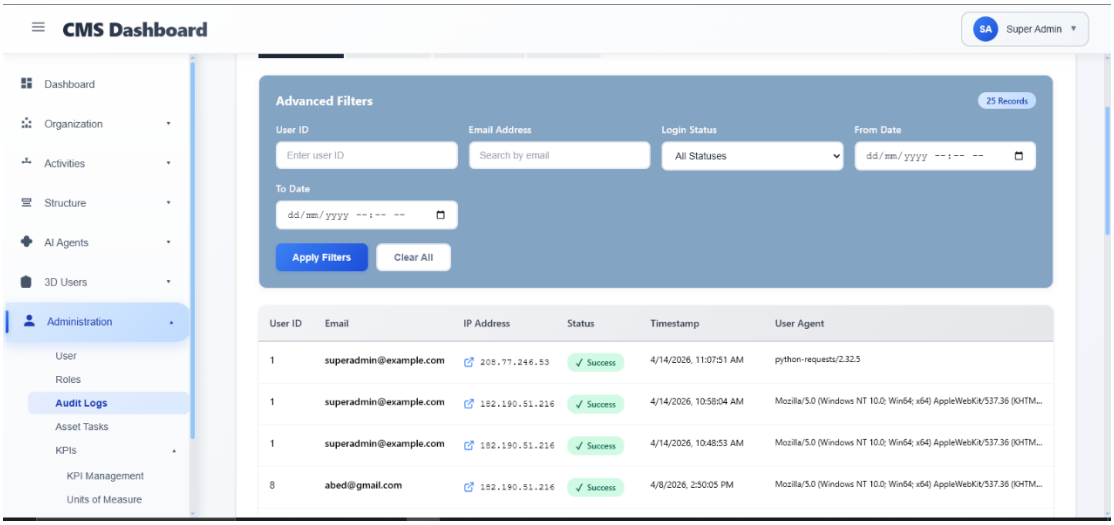


Figure 48: CMS Dashboard

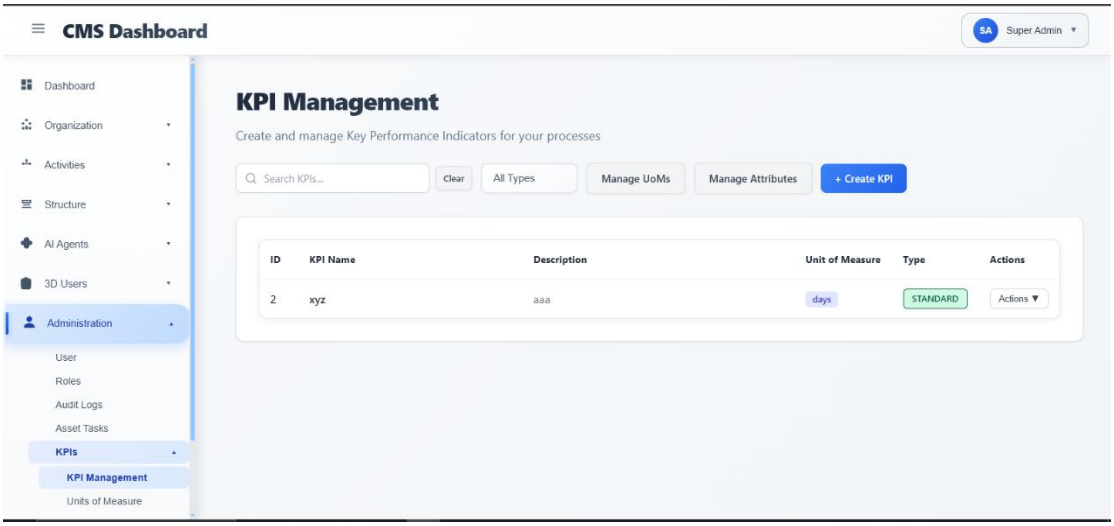


Figure 49: KPI Management page

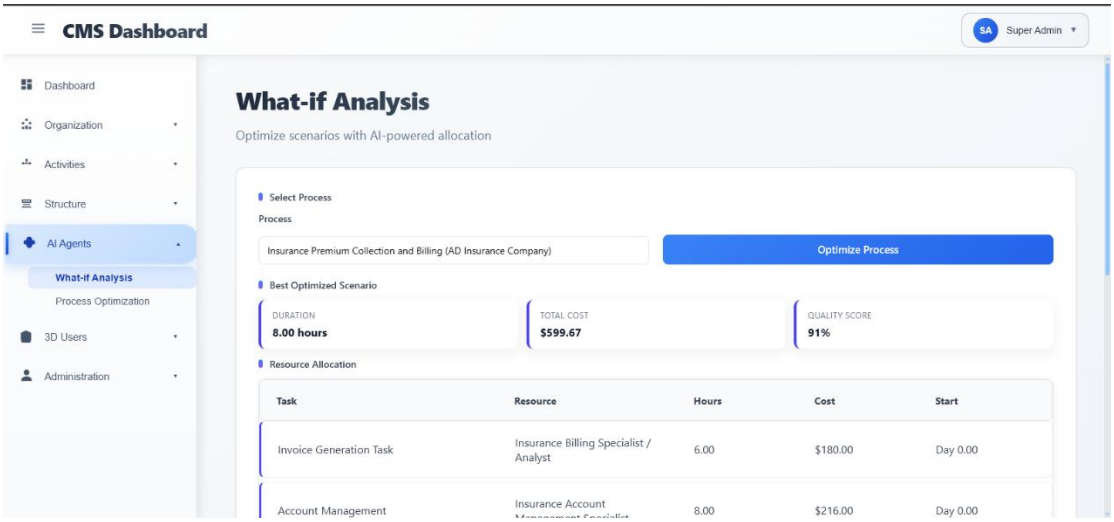


Figure 50: What if Analysis Page

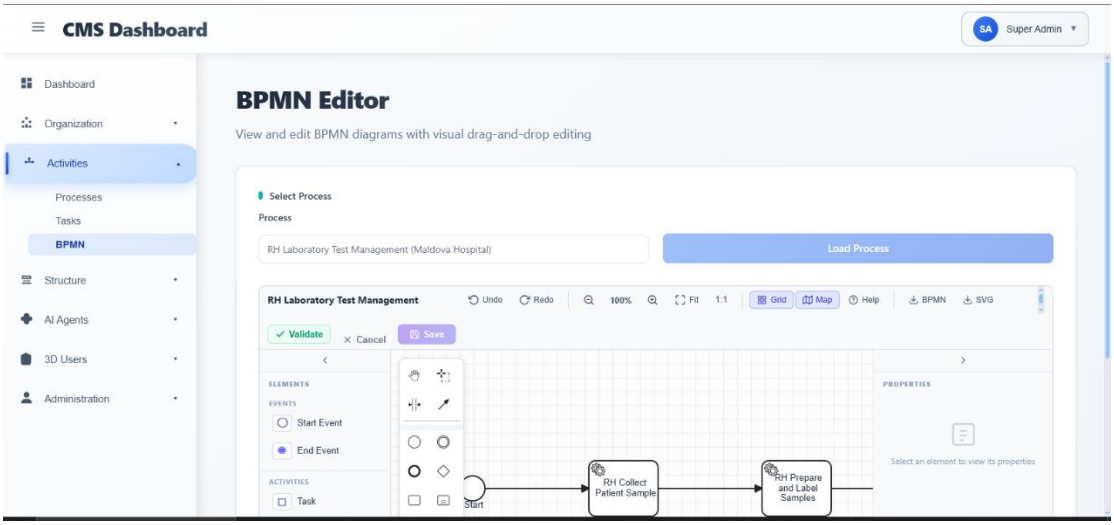


Figure 51: BPMN Module

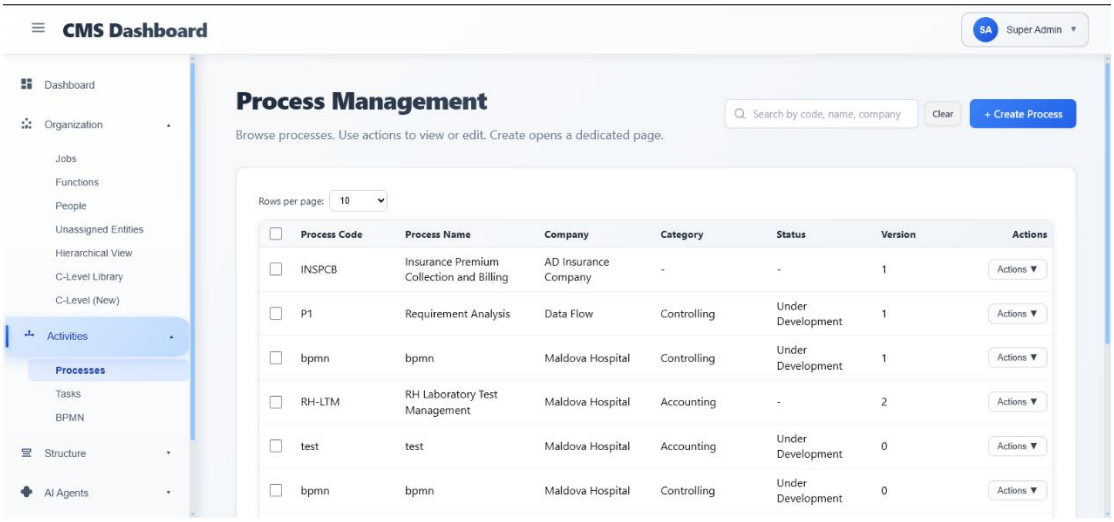


Figure 52: Process Management Page

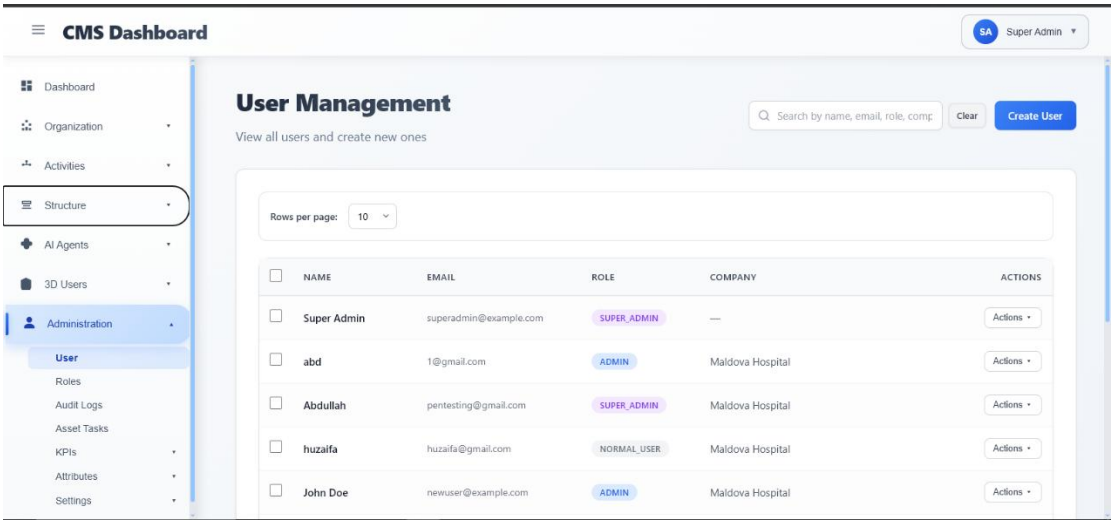


Figure 53: User Management Page

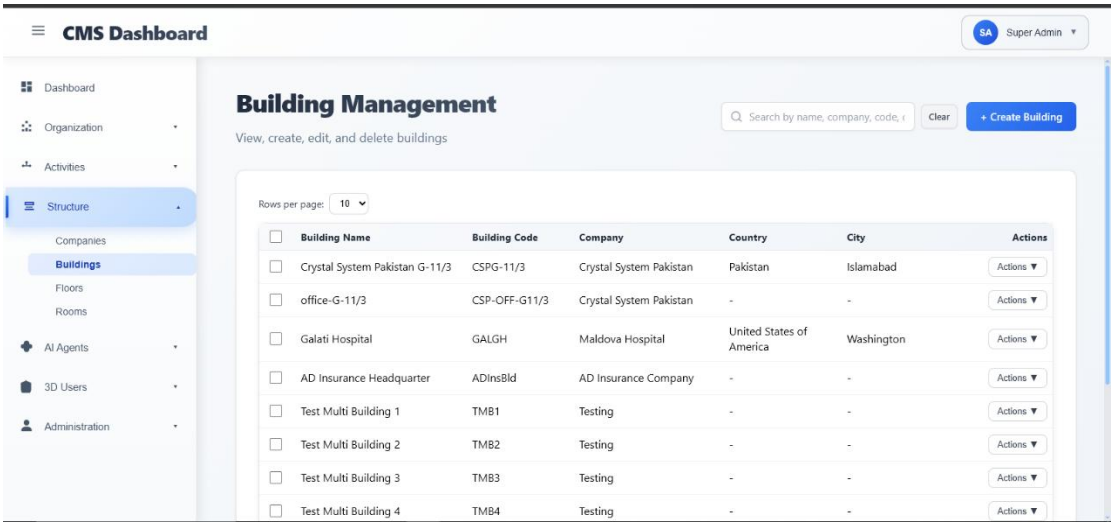


Figure 54: Building Mangement Page

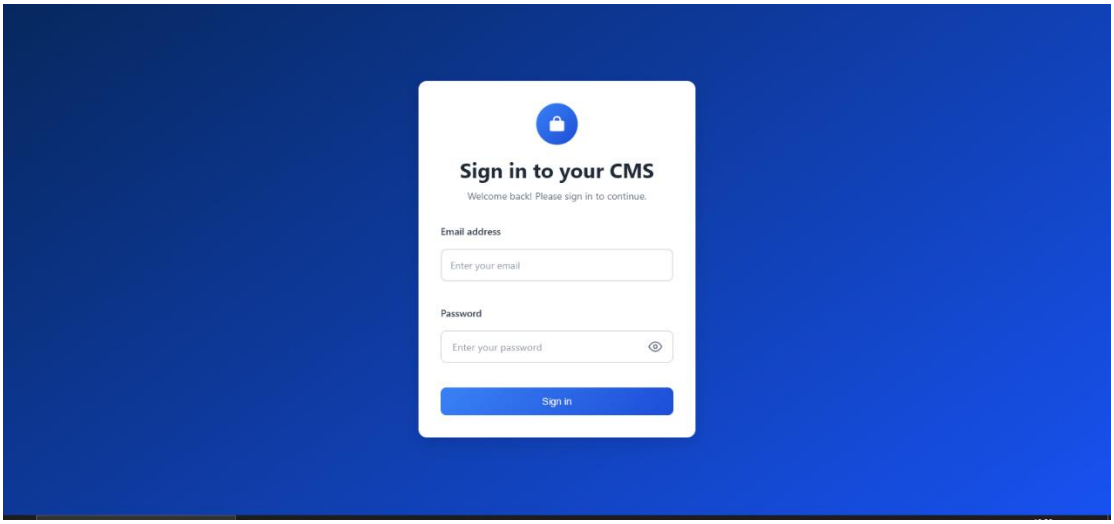


Figure 55: Login Page of CMS

5.10 Conclusion

This chapter has described how the Crystal System CMS has been created. In particular, this chapter provided an overview of the technology stack used in developing this product, including its development approach and implementation of functionality. The project has been created as a modular React application to handle processes in businesses. It is implemented using a number of techniques such as service-oriented APIs, access control and workflow visualization, and management modules.

Chapter 6
System Testing & Evaluation

Testing and evaluating the Crystal System CMS is part of the chapter. The purpose of testing was to confirm that the software performed necessary tasks and responded properly to different error conditions. Moreover, it was essential to test the security mechanism and the friendly user interface of the software for all modules. As the software under development is a web-based system having many integrated components, it was essential to perform different types of testing, such as component testing, unit testing, integration testing, and system testing.

Evaluating the implemented project involves testing its actual performance and validating form build logic and workflow. Besides, route protection was also evaluated, along with various services. Moreover, functional workflows were tested for various modules, including authentication, dashboard, process management, access controls, BPMN visualization, and audit logs.

6.1 Test Strategy

Multi-level testing methodology was used while building Crystal System CMS. The main aim of the testing process that we have undertaken is to validate and verify the developed modules.

The tests that were done by us include the following:

Component testing

Checking the correctness of the working of individual pages, forms and UI elements.

Unit testing

Verifying functions, services, helpers and input validations.

Integration testing

Testing the interaction between frontend pages and backend APIs and application data.

System testing

Validating the application as a whole such as login capabilities, using modules, doing CRUD actions, navigating through the dashboard and visualizing workflows.

Build verification

Making sure that the application can be built into a deployable production artefact.

As this project highly relies upon the application interface and its APIs, we did a lot of validation within the framework of functional and scenario based testing.

6.2 Component Testing

Testing was performed at the level of components, which includes key user interface components and module pages, to ensure that each component performs well in isolation without any problems.

These key components include:

- **Login component:** We made sure that email and password fields are mandatory, as well as verifying the presence of errors when using invalid credentials and when the cooldown happens due to failed login attempts. Redirecting to another page upon successful login was also verified**.**

- **Protected route component:** We made sure that unauthenticated users are directed to the login page, whereas authenticated users can access protected routes.

- **Dashboard component:** We checked if the dashboard renders correct numbers, shows summary cards, and guides the user to appropriate locations when cards are clicked.

- **User & Access Control pages:** We tested the ability of the following actions: creation, updating, deletion, and resetting passwords, depending on the visibility according to role access.

- **Audit log page:** We made sure that tab changing, filtering, and pagination works accordingly. In addition, we ensured that some user roles are authorized to access this module.

- **BPMN Editor & Viewer components:** We tested the functionality of rendering BPMN process diagrams, showing validation errors, and saving and previewing processes.

- **Forms: Building, Room, Task, KPI, Asset Task:** We made sure that these validation messages pop up when the corresponding fields are left empty.

Component testing ensured that each user interface component functioned properly by itself before proceeding with testing of the whole workflow of the system.

6.3 Unit Testing

The following unit testing components are implemented in Jest and React Testing Library. However, having evaluated the project, it was observed that currently, there is no sufficient test automation in use.

In terms of testing, one can observe such unit-based validation in the following cases:

- testing of service functions that have requirements for unique IDs or payload,
- configuration and calculation of the correct paths in route configuration,
- testing of the absence of empty fields in input fields for login and interval time for retries,
- validating forms to make sure that fields like name, email, role, company, and status are present,
- services that analyze and organize data coming from back-end and building frontend data structures.

While the above processes increase accuracy of testing, more work on creating formal tests needs to be done. Considering that there were errors when using Jest in terms of testing, it became clear that such errors were caused by importing ESM packages from App.test.js.

Based on all the above facts about testing, one describes the status of the unit testing in this project as follows:

- Testing Framework: Present
- Automated Unit Test Coverage: Limited
- Unit-Level Validation Logic in Code: Present
- Future Enhancement Need: High

6.4 Integrated Testing

Integration tests have been used for checking the interaction between pages, services, routes, state sharing, and backend API.

Among the most crucial integration points we can single out the following:

- **Login page and authentication service:** Guarantees that input credentials are correctly sent via the backend so that the session created as a result would let the user enter all required routes.

- **Protected routes and state of sessions:** Guaranties that no unauthentic users can access internal modules.

- **Dashboard and dashboard service:** Guarantees that system collects counts needed depending on user roles and that clicking on cards within dashboard will redirect to correct destinations.

- **Management pages and service layer:** Guarantees that operations of creating, updating, deleting and showing entities are performed in coordination with backend API and expected behavior.

- **BPMN viewer and Process/BPMN services:** Guarantees that workflow content could be loaded, previewed, validated and saved.

- **Audit logs and audit service:** Guarantees that some calls made to API would provide table data according to expectations and requirements.

- **Context layer and navigation:** Guarantees consistency of information related to currently active sections/selected objects.

Integration testing shows how well the most crucial parts of our application interact both independently and as a part of business solution.

6.5System Testing

System testing was performed from an end-user perspective to evaluate complete workflows of Crystal System CMS. The system was tested as a whole in a browser environment.
The main workflows covered included:

- user login and logout,
- dashboard loading after authentication,
- navigation between major modules,
- role-based access control for restricted pages,
- CRUD operations in management modules,
- process creation and editing,
- BPMN visualization and validation,
- audit log access and filtering,
- import and export operations,
- error display for unauthorized or failed requests.

A useful verification result was obtained from the production build process. The application successfully generated a deployable production build using npm run build, which confirms that the system compiles and packages correctly. However, a stricter CI-style build failed because ESLint warnings were treated as errors. This indicates that the system is deployable, but code-quality cleanup is still needed for a warning-free production pipeline.

6.6 Test Cases

Representative test cases were designed to verify critical system behavior.

Table 16: Test Case #1: User Login with Valid Credentials

Test Scenario ID	**TS-001**	**Test Case ID**	**TC-01**
Test Case Description	Verify successful login and dashboard redirection using valid credentials.		
Test Priority	High		
Prerequisite	1. A valid user account exists. 2.Backend authentication service is available.		
Post-Requisite	User is authenticated and redirected to the dashboard.		
S. No	**Action**	**Inputs**	**Expected Outcome**
1	Open login page, enter valid email and password, and click "Sign In".	Valid Email, Valid Password	User is authenticated; session created; redirected to dashboard with role-based access.

Table 17: Test Case #2: User Login with Invalid Credentials

Test Scenario ID	**TS-002**	**Test Case ID**	**TC-02**
Test Case Description	Verify error handling and cooldown for invalid login attempts.		

Test Priority	High		
Prerequisite	Login page is accessible.		
Post-Requisite	Access denied; error message displayed.		
S. No	**Action**	**Inputs**	**Expected Outcome**
1	Enter an invalid email or password and submit the form.	Invalid Credentials	System rejects auth; error message shown; cooldown applied on repeats.

Table 18: Test Case #3: Protected Route Access Without Authentication

Test Scenario ID	**TS-003**	**Test Case ID**	**TC-03**
Test Case Description	Verify that unauthenticated users cannot access protected pages.		
Test Priority	High		
Prerequisite	No active authenticated session exists.		
Post-Requisite	User redirected to the login page.		
S. No	**Action**	**Inputs**	**Expected Outcome**
1	Attempt to open a protected module URL directly.	Protected URL (e.g., /dashboard)	The user should be redirected to the login page.

Table 19: Test Case #5: Create or Update a Building Record

Test Scenario ID	**TS-005**	**Test Case ID**	**TC-05**

Test Case Description	Verify form validation and data submission in facility management.		
Test Priority	Medium		
Prerequisite	User has permission to manage buildings.		
Post-Requisite	Building record is successfully created or updated in the database.		
S. No	**Action**	**Inputs**	**Expected Outcome**
1	Open building form and attempt submission with empty required fields.	Null/Empty Fields	Required field errors should appear for empty inputs.
2	Provide valid values and resubmit the form.	Valid Building Details	Valid data should be accepted and submitted successfully.

Table 20: Test Case #6: Load BPMN Workflow for a Process

Test Scenario ID	**TS-006**	**Test Case ID**	**TC-06**
Test Case Description	Verify process visualization and validation within the BPMN Viewer.		
Test Priority	High		
Prerequisite	A process exists with BPMN-related data.		
Post-Requisite	BPMN diagram is displayed and updates can be saved.		
S. No	**Action**	**Inputs**	**Expected Outcome**
1	Open the BPMN Viewer page, select a process, and load the preview.	Process Selection	BPMN diagram loads correctly; validation messages appear if applicable.

6.7 Results and Evaluation

The assessment of the Crystal System CMS proves that the system is rather equipped both functionally and structurally, but there remains room for improvement in automation.

6.7.1 Positive Results

- It offers numerous business management capabilities through a unified interface.

- Role-based authentication is applied across all the sensitive areas within the application.
- Validation of fields is performed in most forms requiring data input.
- API calls have been standardized by making use of services.
- In normal operations, an executable form of the application is generated.

There is structured state handling of loading, error, and denied access at the interface.

6.7.2 Observed Limitations

- Unit tests cannot be automated beyond a certain point currently.
- The default Jest smoke test will not work because of the absence of any setup for the dependencies which have been implemented using ESM in the testing environment.
- The strict build will not work because of the linting warnings.

The dependencies heavily rely upon the backend server being up and running.

6.7.3 Evaluation Summary

Speaking of evaluation criteria, the project could be considered as a functional one since it consists of all required modules and processes.

As to software quality, it can be said that the product:

- has a well-designed module structure
- can be effectively used in business
- has a good integration of features
- can be easily built

To improve the software quality, it would be reasonable to make some changes like:

- making unit/integration tests better
- proper Jest configuration according to new dependencies
- fixing all lint issues
- increasing end-to-end testing

6.8 Conclusion

The method employed for testing and evaluation of the Crystal System CMS has been elaborated on in the above chapter. Testing of the Crystal System CMS was done using a hierarchical approach, which included component tests, unit tests, integration tests, and system tests. From the above discussion, one may conclude that the Crystal System CMS works efficiently and does what it is meant to do, i.e., it is effective in business management. However, despite its proficiency, more testing of the Crystal System CMS is needed in the future.

Chapter 7
Conclusion

The goal of this project was to develop a new Crystal System CMS. This task was done by implementing a web enterprise management system which would allow all the aspects of organization operations to be united in one single system. As the outcome, we have managed to implement a solution which is able to manage the organizations, location, facilities, persons, users, roles, processes, tasks, job positions, KPIs, asset tasks, and audits. In addition to the capabilities of management, we have implemented additional features regarding process management including BPMN visualization, workflow processing, what-if analysis, and process optimization.

This outcome shows how successful the development of Crystal System CMS was and can serve as a solution to our problem. Thus, as an outcome, we have managed to implement a solution which includes management capabilities, monitoring, access management, and visualization of the processes. It is evident that it is very helpful to use up-to-date frontend technologies and services in solving organizational problems.

7.1 Contributions

The key findings and achievements that have been gained during the execution of this project can be easily associated with the initially set goals.

The main goal aimed at the creation of the integrated system for management of organizational entities was fully reached through implementation of the corresponding modules including companies, countries, cities, buildings, floors, rooms, people, users, roles, functions, tasks, jobs, and processes. Instead of having several different applications dedicated to particular segments of the company, there is currently only one application that can handle all necessary information.

The other goal focused on the introduction of business process management functionality which involved the possibility of process creation, visualization, and optimization. The mentioned goal has also been reached through successful implementation of such functionalities as process management modules, viewing/editing using BPMN, connecting tasks/jobs, and what-if analysis.

Lastly, the last goal of having secure and intelligent management included authentication, role-based access control (RBAC), auditing, dashboards, KPI management, and data importing/exporting features.

The overall contributions of this project are listed below:

- The development of a platform to manage enterprises,
- The introduction of a security mechanism for user access according to their roles,
- Process visualization and workflow-oriented management,
- Introduction of analytical features like dashboard and what-if analyses,
- Auditing and process transparency, and

- Showing how a modular and scalable architecture is used for front-end applications' development for enterprises.

All of the above contributions are significant both from academic and practical viewpoints. First of all, from the academic viewpoint, this project proves how software engineering and web interface can address a real-life management issue. Secondly, from a practical standpoint, this project proves how structured the approach must be to support better management decisions.

7.2 Reflections

In addition, one of the accomplishments in this project was gained in developing knowledge in regard to designing a web application that would be big and complex. Integration of functions should be noted among the major characteristics of the software. Process management, user administration, facilities management, and KPI monitoring become a part of the same package in the discussed project. It means that an integration of various features into one package becomes the advantage of the Crystal System CMS.

The possibility to visualize processes as well as make what-if analysis according to BPMN standards is another feature that should be mentioned as advantageous for the developed solution. Of course, by integrating the process management function to the software, it is possible to provide the opportunity to improve processes apart from managing them. Two other functions, namely role-based restrictions and auditing, contribute to security of the solution.

In case we analyze the project from the point of view of design, it is worth mentioning its modularity. Such features as pages, components, services, routes, and utilities have been separated to allow for further development of the app. Besides, building the application in production shows the practicality of the discussed solution.

There are also a few constraints associated with this project. The primary constraint relates to overdependence on the implementation of backend services. As a consequence, some advanced features were impossible to validate until the backend was tested properly. Automated tests are also incomplete, even though they include necessary validation, access control, and error handling. In addition, the automated test suite needs to expand since automated testing is an important part of software maintenance.

Moreover, the development level of particular modules varies due to the fact that it is normal for iterative projects to vary within the last year project time limit. In addition, because of the specific nature of enterprise systems, some possible improvements like advanced reporting, analysis options, testing all along the chain, and additional implementation possibilities have been missed.

As far as contribution is concerned, this research project makes certain contributions to the area since it illustrates the potential value of using an integrated approach to digital transformation. This work adds to the existing knowledge about the management of organizational data, its safety, business process, and analysis. In terms of practical

value, these systems reduce the number of routine activities and support the decision-making process.

7.3 Future Work

Some ideas for improvement for Crystal System CMS in further projects can be suggested.

Firstly, one of them may be related to the development of automatic testing. Further development should imply creating new unit, integration, and end-to-end tests to increase the reliability of the created software and simplify its future maintenance. The creation of a better testing framework should involve using up-to-date dependencies and modules.

Another extension may be associated with improved analytic opportunities. In further development, we could implement better visualizations, different types of data trends, download functions, and comparison of KPIs.

Moreover, it is necessary to make some modifications to process management and BPMN modules. For example, it would be great to develop workflow simulation, various validation rules, connection with process mining software, and recommendations based on system log data.

Finally, another suggestion can be improving the function of user management through adding new permissions, approval procedures, two-factor authentication, etc.

Future development can aid scalability and improvements by incorporating cloud hosting, better monitoring capabilities, and multi-organization tenants. Moreover, performance optimization may be considered by taking advantage of large data sets and high user loads.

On the other hand, future research directions include exploring topics like:

- workflow optimization with the application of AI-based recommendations,
- predictive analytics using audit and performance data,
- organizational performance effects from enterprise integration solutions.

In summary, although Crystal System CMS represents a good starting point in terms of being an enterprise management system, there remains much to be explored technologically, empirically, and in terms of future research.

REFERENCES

1. Object Management Group, “Business Process Model and Notation (BPMN), Version 2.0.2,” Dec. 2013. [Online]. Available: https://www.omg.org/spec/BPMN/2.0.2. [Accessed: Apr. 13, 2026].
2. D. F. Ferraiolo and D. R. Kuhn, “Role-based access controls,” in *Proc. 15th National Computer Security Conference*, Baltimore, MD, USA, 1992, pp. 554–563. [Online]. Available: https://csrc.nist.gov/pubs/conference/1992/10/13/rolebased-access-controls/final. [Accessed: Apr. 13, 2026].
3. R. Sandhu, D. F. Ferraiolo, and D. R. Kuhn, “The NIST model for role-based access control: Towards a unified standard,” in *Proc. 5th ACM Workshop on Role-Based Access Control*, Berlin, Germany, 2000, pp. 47–63, doi: 10.1145/344287.344301.
4. K. Kent and M. Souppaya, *Guide to Computer Security Log Management*, NIST Special Publication 800-92. Gaithersburg, MD, USA: National Institute of Standards and Technology, Sep. 2006, doi: 10.6028/NIST.SP.800-92.
5. W. M. P. van der Aalst *et al*., “Process mining manifesto,” in *Business Process Management Workshops*, Lecture Notes in Business Information Processing, vol. 99. Berlin, Germany: Springer, 2012, pp. 169–194, doi: 10.1007/978-3-642-28108-2_19.
6. M. Dumas, M. La Rosa, J. Mendling, and H. A. Reijers, *Fundamentals of Business Process Management*, 2nd ed. Berlin, Germany: Springer, 2018, doi: 10.1007/978-3-662-56509-4.
7. O. M. Yigitbasioglu and O. Velcu-Laitinen, “A review of dashboards in performance management: Implications for design and research,” *Int. J. Accounting Information Systems*, vol. 13, no. 1, pp. 41–59, 2012, doi: 10.1016/j.accinf.2011.08.002.
8. Rozinat, M. T. Wynn, W. M. P. van der Aalst, A. H. M. ter Hofstede, and C. J. Fidge, “Workflow simulation for operational decision support,” *Data & Knowledge Engineering*, vol. 68, no. 9, pp. 834–850, 2009, doi: 10.1016/j.datak.2009.02.014.
9. Odoo S.A., “More than an ERP software: One platform to run your business,” *Odoo*. [Online]. Available: https://www.odoo.com/erp. [Accessed: Apr. 13, 2026].
10. SAP, “SAP Signavio Process Intelligence,” *SAP Help Portal*. [Online]. Available: https://help.sap.com/docs/signavio-process-intelligence/feature-scope-description/sap-signavio-process-intelligence. [Accessed: Apr. 13, 2026].
11. Appian, “Process automation platform and business process automation software,” *Appian*. [Online]. Available: https://appian.com/products/platform/process-automation/overview. [Accessed: Apr. 13, 2026].
12. Kissflow, “Features: Workflow management software,” *Kissflow*. [Online]. Available: https://kissflow.com/process/features/. [Accessed: Apr. 13, 2026].

APPENDICES

The following examples of deployment, among others, are presented in this appendix.

Routing Configuration

The routing mechanism connects the application URLs with its internal modules, including dashboard, processes, tasks, key performance indicators (KPIs), and audit logging. This allows users to access modules through client-side routing.

Protected Route Logic

The protected route will be used to verify whether the person has logged in or not. If the user is not authenticated, he is then redirected to the login page.

Authentication Service

The login authentication service is the module that provides functionalities regarding logging out, logging in, and validating through the usage of API calls.

API Service Layer

Communication with the backend services is managed by the API service, which sends HTTP requests, processes responses, handles errors, and works with JSON or file requests.

Deployment Files

The following are deployment files which are a part of this project:

- Dockerfile
- docker-entrypoint.sh
- nginx.conf

These files assist in deploying the frontend in production.

Table D.1: Important project configuration files.

File Name	Purpose
package.json	Defines dependencies and scripts
Dockerfile	Supports containerized deployment
nginx.conf	Configures frontend hosting
src/services/api.js	Central API request handling
src/components/routing/ProtectedRoute.js	Route protection for authenticated access

User Guide

This appendix contains a brief tutorial on working with the Crystal System CMS.

Sign in to the System

- Launch the system in your web browser.
- Type your email and password.
- Press the Sign In button.

After logging in, you will see the dashboard.

Using the Dashboard

The dashboard consists of summary statistics and navigation icons. The user can navigate to various sections of the software including companies, process management, tasks, and jobs by using the corresponding card icons.

Managing Records

For users who have access rights:

- create new records,
- edit existing records,
- delete records,
- search and filter records,
- view record details.

The following operations apply to the mentioned modules:

Viewing BPMN Diagrams

For creating the workflow diagram visualization:

- Navigate to the module BPMN Viewer.
- Choose any process from the list of processes provided.
- Open the selected process for visualization of its BPMN model diagram.

Reviewing Audit Logs

Users authorized in the system may use the module Audit Logs to:

- login activity,
- action logs,

- failed logins,
- deletion records.

Filters and tabs can be used to cut down the amount of displayed data.

Logging Out

To securely log out of the system, users can use the logout button of the UI.

Tools and Technologies Used

In the following section, some of the major tools/technologies used for development of the Crystal System CMS are discussed.

Table F.1: Development tools and technologies.

Technology / Tool	**Purpose**
React	Frontend development
React Router DOM	Navigation and routing
JavaScript	Programming language
CSS	Styling and layout
Fetch API	Backend communication
BPMN-JS	BPMN diagram visualization
React Flow	Interactive flow representation
D3-Geo	Map visualization
Node.js / npm	Package and build management
Docker	Containerization
Nginx	Deployment and hosting

APPENDIX B