SECP3744-01

ENTERPRISE SYSTEMS DESIGN AND MODELING

AK MAJU SYSTEM

<< Project Management >>

Prepared By:

CHAI YU TONG A22EC0145

MUHAMMAD DANIEL HAKIM BIN SYAHRULNIZAM A22EC0207

KOH LI HUI A22EC0059

TANG YAN QING A22EC0109

VINESH A/L VIJAYA KUMAR A22EC0290

Lecturer Name:

DR. ARYATI BINTI BAKRI

TABLE OF CONTENTS

Chapter 1

Introduction

1.1 Overview

The rapid digital transformation in today's business environment demands companies to adopt integrated and scalable solutions to remain competitive. AK Maju Resources Sdn Bhd, a trusted provider of printing services for over 14 years in Segamat, Johor, has recently diversified into advertising and construction services. This expansion has increased operational complexity, creating a pressing need for robust systems to manage customer projects, resource allocation, and financial operations efficiently.

This thesis proposes the integration of AK Maju's business processes into SAP HANA Project Management and SAP Business Technology Platform (BTP) to address these challenges. SAP HANA Project Management provides tools for project planning, scheduling, and monitoring, enabling AK Maju to streamline workflows for both advertising and construction services. Meanwhile, SAP BTP serves as a cloud-based platform for extending and integrating enterprise applications, ensuring flexibility, scalability, and real-time data access across AK Maju’s operational ecosystem. By leveraging these technologies, AK Maju can transition to a centralized, data-driven system capable of managing project lifecycles while fostering collaboration and informed decision-making.

The focus of this thesis is on integrating SAP HANA Project Management with AK Maju’s core business processes. This includes creating and managing project schedules, defining start and end dates, tracking milestones, assigning resources, and visualizing critical paths through Gantt charts. SAP BTP complements these functionalities by enabling seamless integration with other modules, such as inventory and financial management, ensuring a unified and adaptive solution. Together, these tools provide a comprehensive framework to digitize and enhance AK Maju’s operational efficiency.

In this thesis, we outline AK Maju’s current business processes and operational challenges, followed by an architectural design of the proposed system. This introduction also frames the transition of AK Maju to a digital-first organization by aligning its business processes with cutting-edge tools like SAP HANA and SAP BTP. The subsequent chapters will delve deeper into the technical integration, business outcomes, and scalability considerations of this transformation.

1.2 Problem Statement

AK Maju Resources Sdn. Bhd. provides services in the advertisement and construction sectors. Currently, the company relies on manual processing for key business functions such as project-based order management. This reliance on manual methods led to inefficiencies and errors that affected overall productivity. Below are the problems faced by AK Maju's staff with the current system:

1. **Inefficient Project Monitoring**
   The lack of a system for managing project schedules, allocating resources, and monitoring project progress has led to complex workflows. AK Maju's staff currently relies on manual tools like Excel spreadsheets and physical notebooks to track the status of projects. This has increased the probability of delays and mismanagement, ultimately impacting the project delivery times. In addition, typographical errors in the data recording process can lead to inaccurate project records or loss of data.
2. **Manual Cost Tracking and Budget Management**
   AK Maju's staff needs to manually calculate the expenses and sales prices for materials with varying specifications for each project. This process is not only time-consuming but also prone to errors. Furthermore, the lack of real-time budget tracking makes it difficult to identify and address financial overruns in a timely manner, further complicating decision-making.
3. **Scaling Challenges for Expanding Operations**
   As AK Maju expands into the construction sector, the complexity of managing multiple concurrent projects increases. The current system does not have the ability to scale effectively, leading to inefficiencies in productivity as project volumes grow. Therefore, without an integrated project management solution, it will be difficult for the company to maintain operational efficiency and deliver quality services across an expanding portfolio of projects.

1.3 Project Objectives

- Develop an enterprise system to handle tasks related to project management, order management, customer management and supplier management.
- Automate administrative processes such as order price calculation, inventory updates, and customer-supplier interactions.

1.4 Project Scope

The scope of the AK Maju Administrative System (AKMAS) encompasses the development of a comprehensive digital platform to support the company's expansion into advertising and construction services. The system aims to streamline core operations, including order management, resource allocation, and customer interactions, while addressing inefficiencies in their current manual processes. Key functionalities include modules for quotation creation, invoice generation, inventory management, delivery tracking, user management, and cost reporting. These modules will automate crucial processes such as pricing adjustments, material availability checks, and inventory updates, providing real-time data for improved accuracy and decision-making. The system will enable stakeholders to generate and manage digital documents, such as quotations, invoices, and reports, enhancing operational efficiency and ensuring timely customer service.

The AKMAS will integrate features like low-stock alerts, delivery status updates, and comprehensive reporting tools, catering to the specific needs of both advertising and construction divisions. Collaboration with Mr. Azam, the company representative, ensures the system aligns with AK Maju's business goals. The project is limited to automating processes for these two services and managing deliveries solely through AK Maju's logistics team. By replacing manual workflows with an advanced, centralized system, AKMAS will empower AK Maju Resources Sdn Bhd to operate more efficiently, reduce errors, and enhance customer satisfaction.

1.5 Project Importance

The importance of the project management module in the AK Maju Administrative System (AKMAS) cannot be overstated. The success of the advertising and construction system implementation and its alignment with organizational goals largely depend on effective project management, which ensures that all activities are well-planned, executed, and monitored. Customers expect their orders to be delivered accurately and on time, and a poorly system without an effective project management module can lead to order delays, higher delivery costs, and order damage, which can cause dissatisfaction among customers.

For AK Maju, the project management module plays a critical role in streamlining business operations, such as order processing, inventory management, and customer interactions. Without proper project control, the system implementation process may experience delays, cost overruns, or misalignment with business requirements. Furthermore, the project management module facilitates resource allocation, ensuring that resources are utilized efficiently without conflicts or overuse while boosting productivity. By organizing the development process into milestones, such as the creation of quotations, invoices, and inventory tracking, the module keeps the team focused and on track to meet deadlines of completing customer orders.

By focusing on the project management module, AK Maju can ensure that the AKMAS system is implemented effectively and meets the organization's operational and strategic requirements. The module also helps mitigate risks by identifying potential challenges early, enabling proactive solutions to be implemented. This will lead to a robust system that will enhance customer satisfaction, operational efficiency, and profitability, thus positioning AK Maju as a competitive and innovative market leader.

Chapter 2

Literature Review

2.1 Introduction

AK Maju Resources Sdn Bhd is a trusted provider of printing services with over 14 years of experience in Segamat, Johor, has recently expanded its business portfolio to include advertising and construction services. As the company diversifies its offerings, the need for an efficient and centralized enterprise system has become increasingly apparent. The existing system needs to be upgraded to consolidate critical data, including customer information, supplier details, item stock records, and other operational data, into a unified platform. This thesis focuses on documenting the development of an enterprise system for AK Maju Resources utilizing SAP technologies. The proposed solution aims to seamlessly integrate the company's existing systems and data, enabling streamlined operations, enhanced decision-making, and improved scalability as the business continues to grow. Five main modules have been identified which are Project Structure Management, Budgeting and Cost Planning, Scheduling, Resource Allocation and Progress Monitoring. The system will leverage **SAP HANA** for advanced project planning, scheduling, and monitoring, enabling streamlined workflows for both advertising and construction services. Additionally, **SAP Business Technology Platform (BTP)** will serve as a cloud-based foundation to extend and integrate enterprise applications, providing flexibility, scalability, and real-time data access. This comprehensive solution aims to centralize AK Maju's operational ecosystem, optimize resource utilization, and support data-driven decision-making, ensuring the company stays competitive and responsive to customer needs.

2.2 Enterprise Architecture

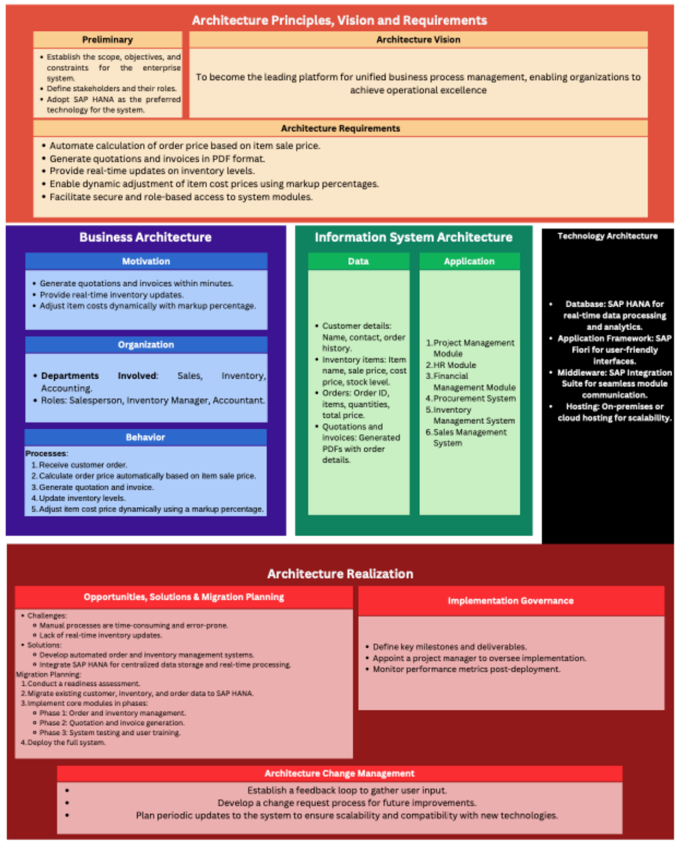


Figure 2.2.1: Enterprise Architecture for AK Maju System

2.2.1 What is Enterprise Architecture?

Enterprise Architecture (EA) is a framework used to structure and manage an organization's IT assets, business processes, and information flows to align with strategic goals. It integrates processes, technology, and data to ensure that the organization operates efficiently and adapts to changing environments.

2.2.2 Previous Studies on Enterprise Architecture

The TOGAF (The Open Group Architecture Framework) is widely studied as a comprehensive method for developing and managing Enterprise Architecture (EA). It provides a structured approach to designing, implementing, and governing EA, ensuring alignment between IT and business goals. The framework is highly regarded for its ability to enhance decision-making processes, reduce operational costs, and improve the overall efficiency of organizational operations. However, implementing EA using TOGAF comes with challenges. These include resistance to change within organizations, the high costs associated with implementation, and the complexity of scaling EA to meet the dynamic needs of modern businesses. Despite these challenges, TOGAF remains a cornerstone in the study and application of EA due to its strategic and systematic approach.

2.2.3 Importance of Enterprise Architecture in Modern Organizations

Enterprise Architecture provides a strategic roadmap for driving digital transformation within organizations, enabling them to adapt to evolving technologies and market demands. It ensures optimal resource utilization by aligning IT investments with business objectives, thereby maximizing efficiency and achieving desired outcomes. Additionally, it plays a crucial role in supporting regulatory compliance by organizing and documenting processes systematically, ensuring that organizations meet legal and industry standards while maintaining operational transparency and accountability.

2.3 Related on Sub System

Successful project management is crucial for organizations to ensure timely delivery, manage expenses effectively, and enhance resource efficiency. The SAP S/4HANA platform offers a robust framework for managing projects, featuring advanced capabilities and smooth integration with various business processes. This system enhances visibility, operational efficiency, and collaborative efforts, positioning it as an optimal choice for organizations seeking to refine their business project workflows.

2.3.1 What is Project Management Subsystem?

The Project Management Subsystem is a component of SAP S/4 HANA that focuses on managing project lifecycles, from initiation to execution and closure. It serves as a platform for organizing projects, allocating resources, and tracking progress effectively. By aligning project activities with organizational goals, it ensures that projects are executed efficiently while adhering to established constraints.One of the key features of the Project Management Subsystem is the Work Breakdown Structure (WBS), which enables users to break projects into smaller, manageable components. The WBS establishes a crucial framework for the planning and oversight of expenses, resources, and schedules, giving project managers a systematic overview of their tasks.Furthermore, the subsystem provides extensive budgeting and cost control features that allow organizations to set budgets, monitor expenditures, and perform real-time variation analysis. This subsystem's scheduling and milestone management are essential components since it provides tools to create project timelines and keep track of important deliverables. Visual aids such as Gantt charts are used to identify critical paths and ensure deadlines are met.Additionally, the subsystem facilitates effective resource management by allocating human resources, tools, and materials to project tasks while monitoring utilization to prevent conflicts. All of these characteristics work together to promote accountability, streamline project execution, and enhance collaboration across departments.

2.3.2 SAP S/4 Integration?

SAP S/4 HANA integrates the Project Management Subsystem seamlessly into broader business processes, such as finance, procurement, and logistics.Through this integration, real-time visibility is enabled to give decision-makers and project managers access to current, accurate information for improved planning and execution. The interconnectivity feature of the system allows project-related tasks to be linked with important operational areas, which enhances overall efficiency. One of the significant integration points is that project budgets and actual costs are directly connected to financial systems. Similar to this, the subsystem works with procurement processes to track schedules and material purchases, ensuring alignment with project specifications. It also supports logistics by combining resource allocation with inventory control and transportation schedules. By streamlining processes like scheduling, resource allocation, and reporting, SAP S/4 HANA significantly improves the user experience for project planners. Real-time dashboards and other tools offer practical insights into resource usage, key performance indicators (KPIs), and project performance. Through these insights, businesses may fulfill project deadlines, maximize resources, and improve decision-making processes, which makes SAP S/4 HANA a useful tool for handling challenging projects. By integrating project management with other essential business processes, SAP S/4 HANA offers a unified platform that ensures smooth operations and helps projects in a variety of industries succeed.

2.3.3 Previous System Architecture for Project Management Subsystem

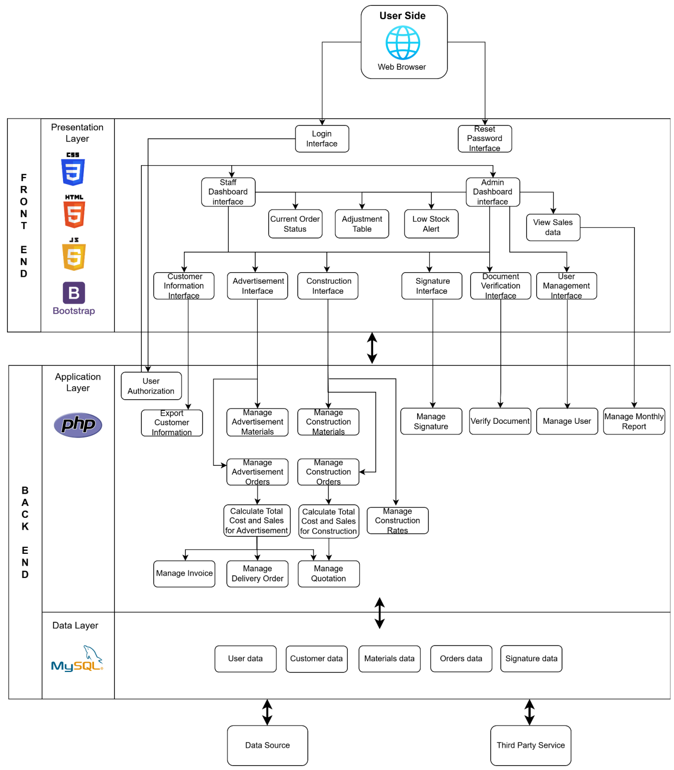


Figure 2.3.1: System Architecture for AK Maju System without using SAP

The previous system, as illustrated in Figure 2.3.1, was built on a basic architecture that used HTML, CSS, JavaScript and Bootstrap for the user interface, a relational database for storage, and PHP for backend logic. Such architecture is sufficient to manage simple orders but does not provide scalability, integration, and automation for more complex projects. Processes such as progress tracking, resource allocation and budget planning were either missing or highly dependent on manual updating, leading to inefficiencies and frequent errors.

The architecture of SAP-based systems has undergone a major transformation. SAP technology includes in-memory computing, which offers centralized data management and real-time processing. Unlike the rigid relational structure of the previous system, SAP supports advanced data models that allow seamless integration between modules like project management, finance, and inventory. Furthermore, the SAP architecture integrates powerful automation tools that streamline workflows and reduce reliance on manual methods.

This architectural upgrade ensures scalability, faster data access, and comprehensive integration, enabling AK Maju to efficiently and accurately handle complex project requirements. The transition highlights a move towards a modern, unified platform that supports real-time updates and decision-making.

2.3.4 Relevance to Current Project

The Project Management Subsystem is designed to handle the challenges faced by AK Maju in managing complex projects. It introduces functionalities like real-time tracking, resource allocation, and budget management. This subsystem integrates project data with organizational operations, automates notification processes, and reduces the dependence on manual tools through the use of SAP.

There are 5 submodules to enhance specific aspects of project management. Firstly, the ***Project Structure Management*** submodule will help in organizing tasks through a Work Breakdown Structure (WBS), defining project phases, tasks, and milestones, ensuring clear timelines for complex orders. Secondly, the ***Budgeting and Cost Planning*** submodule allows staff to allocate budgets, track financial performance, and perform cost estimation. It aims to help in maintaining projects within budget. Thirdly, the ***Scheduling*** submodule will help in streamlining project timelines by creating schedules, setting task deadlines, and monitoring critical paths using tools like Gantt charts. Fourthly, the ***Resource Allocation*** submodule optimizes the allocation of human resources, equipment, and materials, preventing resource conflicts and improving efficiency. Lastly, the ***Progress Monitoring*** submodule will help in tracking task completion, measures project performance using KPIs, and updates project progress reports, enabling timely adjustments and informed decision-making. Together, these features, developed using SAP BTP, ensure that the subsystem effectively supports the management of both simple and complex projects.

2.4 Technology Used

To develop the enterprise system for AK Maju Resources, we will fully utilize **SAP HANA** for robust project planning, scheduling and monitoring, ensuring high-performance data processing and real-time features. **SAP Business Technology Platform (SAP BTP)** will serve as the central cloud-based platform for extending and integrating enterprise applications, enabling scalability, flexibility, and seamless connectivity across various systems. Together, SAP HANA and SAP BTP will provide the tools required for data storage, advanced analytics, and system integration, empowering AK Maju Resources to streamline workflows and enhance decision-making. Additionally, the cloud capabilities of SAP BTP will ensure a secure deployment process, real-time data access and give access for all users.

2.5 Summary

The document outlines the system architecture, enterprise system design, and technologies used for AK Maju Resources. It describes a centralized, multi-layered enterprise system framework leveraging SAP tools, consisting of a presentation layer using SAP Fiori for user-friendly interfaces, an application layer managing core functionalities like financial management, project planning, and resource monitoring, and a data layer enabling cloud and on-premise data integration. The architecture supports real-time data management and cross-departmental integration, emphasizing the transition from the company's previous PHP-based system to an advanced SAP platform. Built on SAP Business Technology Platform (BTP) and SAP HANA, the system delivers scalability, flexibility, and data-driven decision-making capabilities. To form the backbone of the system, five modules will be integrated which are Project Structure Management, Budgeting and Cost Planning, Scheduling, Resource Allocation, and Progress Monitoring. A phased implementation approach, starting with data migration and module integration followed by system testing, ensures smooth deployment. Governance measures, including performance monitoring and feedback loops, align the system with emerging technologies, positioning AK Maju to manage complex projects effectively and maintain its competitive edge.

Chapter 3

Methodology

3.1 Introduction

The SAP Activate Methodology was selected for this project due to its structured approach and adaptability in managing complex SAP solutions. Consisting of six phases which are discover, prepare, explore, realize, deploy, and run. It emphasizes integration, flexibility, and adherence to SAP best practices. It supports hybrid development scenarios, enabling seamless connectivity between custom applications, such as PHP interfaces, and SAP modules through the SAP Business Technology Platform (BTP) and APIs. The methodology accelerates implementation with pre-configured content for standard modules like Work Breakdown Structure (WBS) and Budgeting while allowing iterative development for early prototyping and validation. Additionally, its cloud integration capabilities ensure real-time access, scalability, and alignment with SAP HANA standards, providing an efficient and effective framework for project success.

3.2 The Chosen Methodology

The chosen methodology for this project is the SAP Activate Methodology, originally designed for planning and executing complex SAP solutions. This methodology consists of six distinct phases that guide the delivery of SAP updates and solutions effectively. The phases are, discover, prepare, explore, realize, deploy, and run. These phases will be elaborated on further in the subsequent section.

The SAP Activate Methodology was chosen for several reasons. Firstly, it emphasizes integration by supporting hybrid development scenarios. This is particularly beneficial for integrating custom applications, such as PHP interfaces, with SAP modules using SAP Business Technology Platform (BTP) and APIs for seamless connectivity. Secondly, it offers flexibility in development, accommodating both out-of-the-box SAP implementations and customized or external systems that require integration with SAP.

Another advantage is its emphasis on best practices, ensuring the custom system aligns with SAP's enterprise standards for data exchange, APIs, and integration points. The methodology also leverages pre-configured content to accelerate implementation for standard modules like Work Breakdown Structure (WBS) and Budgeting. Furthermore, its iterative development approach allows for early prototyping and validation, minimizing risks.

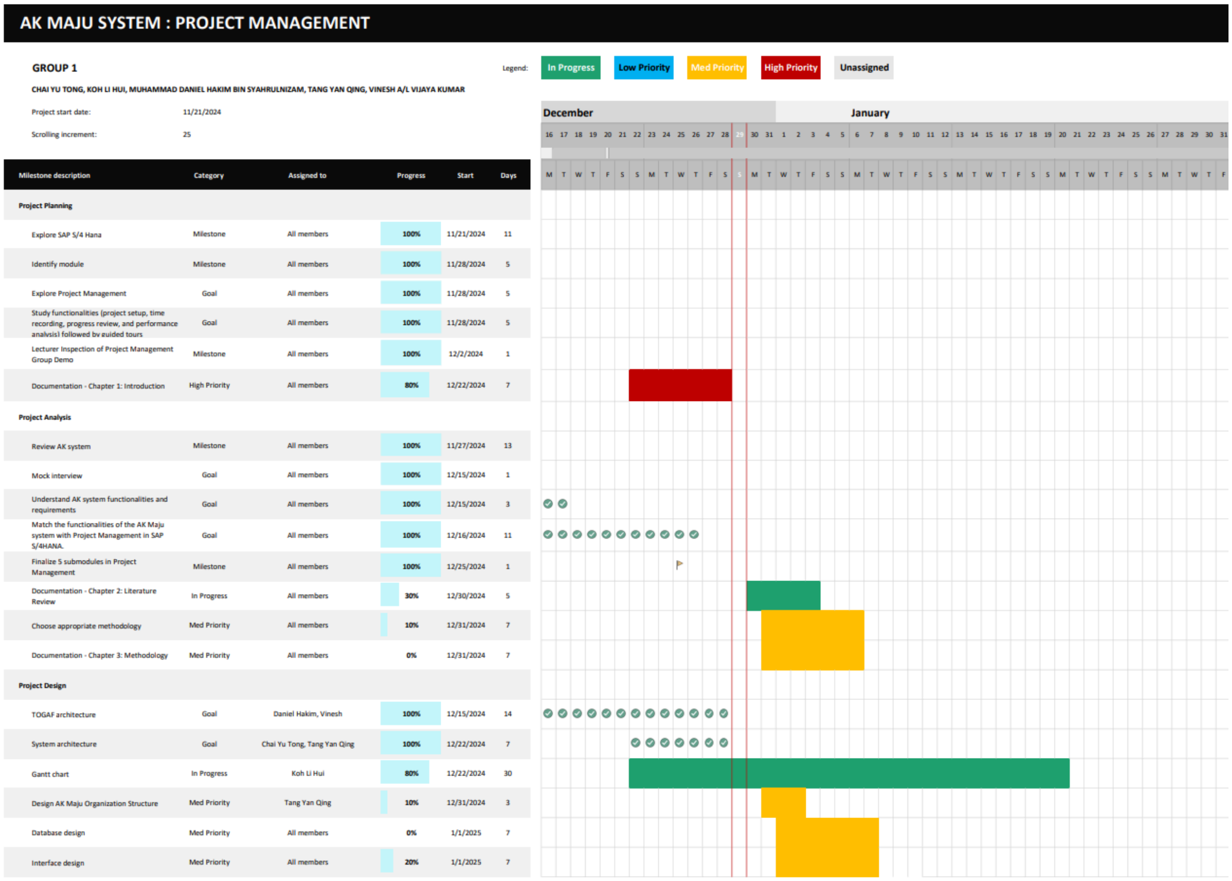
Additionally, cloud integration via SAP BTP ensures real-time access and scalability, while the methodology's design is tailored for SAP. By leveraging SAP best practices and ensuring smooth integration with SAP HANA, SAP Activate provides a structured, efficient, and effective framework for successful project execution.

3.3 Phases of the chosen methodology

The SAP Activate Methodology follows a structured six-phase approach to ensure the successful implementation of SAP solutions. It begins with the Discover phase, where business needs are identified, project objectives are defined, and feasibility is assessed. This is followed by the Prepare phase, where the project team is established, a high-level plan is created, and the necessary tools and infrastructure are set up. In the Explore phase, a fit-gap analysis is conducted to align business processes with SAP best practices, the system architecture is designed, and project requirements are refined. Next, the Realize phase focuses on configuring the SAP system based on agreed processes, developing necessary custom extensions and integrations, and performing iterative testing to validate the solution. The Deploy phase ensures final system readiness by conducting comprehensive testing, training end-users, and executing data migration before the system goes live. Finally, in the Run phase, ongoing support and optimization are provided, system performance is monitored, and continuous improvements are implemented to align with evolving business needs. This structured methodology ensures a seamless, efficient, and flexible implementation process, enabling successful project execution.

3.4 Project Planning Schedule

This section outlines the timeline and milestones for the project using a Gantt chart. The chart provides a clear visualization of the project phases, tasks, priorities, and deadlines to ensure effective planning and timely delivery of objectives.



AK MAJU SYSTEM : PROJECT MANAGEMENT

GROUP 1

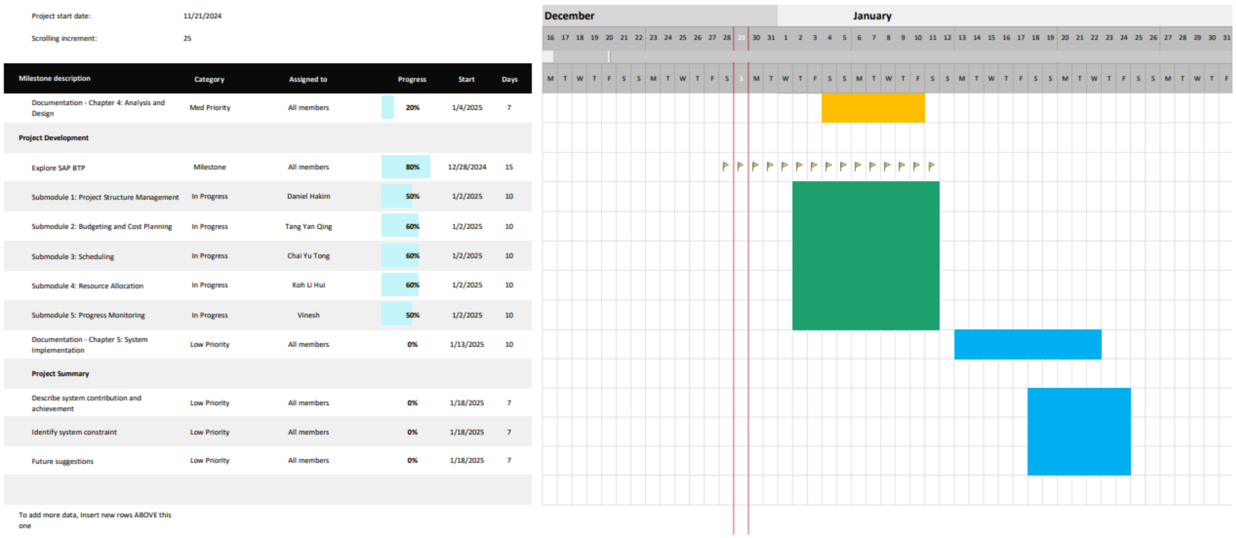
CHAI YU TONG, KOH LI HUI, MUHAMMAD DANIEL HAKIM BIN SYAHRULNIZAM, TANG YAN QING, VINESH A/L VIJAYA KUMAR

Project start date: 11/21/2024

Scrolling increment: 25

Legend: In Progress, Low Priority, Med Priority, High Priority, Unassigned

Milestone description	Category	Assigned to	Progress	Start	Days
Project Planning					
Explore SAP S/4 Hana	Milestone	All members	100%	11/21/2024	11
Identify module	Milestone	All members	100%	11/28/2024	5
Explore Project Management	Goal	All members	100%	11/28/2024	5
Study functionalities (project setup, time recording, progress review, and performance analysis) followed by guided tours	Goal	All members	100%	11/28/2024	5
Lecturer Inspection of Project Management Group Demo	Milestone	All members	100%	12/2/2024	1
Documentation - Chapter 1: Introduction	High Priority	All members	80%	12/22/2024	7
Project Analysis					
Review AK system	Milestone	All members	100%	11/27/2024	13
Mock interview	Goal	All members	100%	12/15/2024	1
Understand AK system functionalities and requirements	Goal	All members	100%	12/15/2024	3
Match the functionalities of the AK Maju system with Project Management in SAP S/4HANA.	Goal	All members	100%	12/16/2024	11
Finalize 5 submodules in Project Management	Milestone	All members	100%	12/25/2024	1
Documentation - Chapter 2: Literature Review	In Progress	All members	30%	12/30/2024	5
Choose appropriate methodology	Med Priority	All members	10%	12/31/2024	7
Documentation - Chapter 3: Methodology	Med Priority	All members	0%	12/31/2024	7
Project Design					
TOGAF architecture	Goal	Daniel Hakim, Vinesh	100%	12/15/2024	14
System architecture	Goal	Chai Yu Tong, Tang Yan Qing	100%	12/22/2024	7
Gantt chart	In Progress	Koh Li Hui	80%	12/22/2024	30
Design AK Maju Organization Structure	Med Priority	Tang Yan Qing	10%	12/31/2024	3
Database design	Med Priority	All members	0%	1/1/2025	7
Interface design	Med Priority	All members	20%	1/1/2025	7



Project start date: 11/21/2024

Scrolling increment: 25

Milestone description	Category	Assigned to	Progress	Start	Days
Documentation - Chapter 4: Analysis and Design	Med Priority	All members	20%	1/4/2025	7
Project Development					
Explore SAP BTP	Milestone	All members	80%	12/28/2024	15
Submodule 1: Project Structure Management	In Progress	Daniel Hakim	50%	1/2/2025	10
Submodule 2: Budgeting and Cost Planning	In Progress	Tang Yan Qing	60%	1/2/2025	10
Submodule 3: Scheduling	In Progress	Chai Yu Tong	60%	1/2/2025	10
Submodule 4: Resource Allocation	In Progress	Koh Li Hui	60%	1/2/2025	10
Submodule 5: Progress Monitoring	In Progress	Vinesh	50%	1/2/2025	10
Documentation - Chapter 5: System Implementation	Low Priority	All members	0%	1/13/2025	10
Project Summary					
Describe system contribution and achievement	Low Priority	All members	0%	1/18/2025	7
Identify system constraint	Low Priority	All members	0%	1/18/2025	7
Future suggestions	Low Priority	All members	0%	1/18/2025	7

To add more data, Insert new rows ABOVE this one

3.5 Summary

This project follows the SAP Activate Methodology, which is a structured framework designed for efficient and seamless implementation of SAP solutions. This methodology includes six clear phases: Discover, Prepare, Explore, Realize, Deploy, and Run. All these phases help in guiding the project from initiation to post-implementation support. The main benefits of this methodology involve its integration capability, including enabling SAP Business Technology Platform (SAP BTP) for hybrid development scenarios and also opening smooth connectivity of custom applications to SAP modules. The flexibility of this methodology supports the implementation of standard and customized systems while aligning to SAP best practices. The milestones, priorities and deadlines for the planning schedule have been detailed in the Gantt chart to ensure timely delivery. Together, these elements form a coherent approach to achieving project goals while maintaining efficiency and minimizing risk.

Chapter 4

Analysis and Design

4.1 Introduction

This topic discusses the process of designing an enterprise system for AK Maju, based on its existing system. The previous system was built on a basic architecture that relied on technologies such as PHP, HTML, CSS, JavaScript, Bootstrap, and a relational database. While functional, this architecture resulted in the formation of data silos, necessitating manual processes to consolidate and manage information across various departments.

The new enterprise system, leveraging SAP S/4HANA for enterprise-grade project management and SAP Business Technology Platform (BTP) for seamless integration with PHP-based interfaces, introduces a unified and scalable solution. It features integrated sub-modules designed for Project Structure Management, Budgeting, Scheduling, Resource Allocation, and Progress Monitoring. Centralized data management will enable real-time updates and comprehensive visibility into operations.

The limitations of the previous system are significant. It lacked essential tools such as real-time tracking, resource allocation capabilities, and progress monitoring features. Visual tools like Gantt charts for scheduling or KPI dashboards for performance tracking were also absent. These shortcomings created inefficiencies and hindered decision-making.

The new system will address these gaps with advanced features, including Work Breakdown Structure (WBS), real-time project tracking, budgeting, and cost control functionalities. Gantt charts will facilitate critical path monitoring, while KPI-based dashboards will enhance performance tracking and analysis. Fully integrated modules will deepen project management capabilities, improve resource utilization, and provide a holistic view of operations.

In addition, this topic provides details of the enterprise architecture and system architecture of AK Maju, which offers a comprehensive understanding of the technical and functional improvements brought by the new design.

4.2 Company Organization Structure

Figure 4.2.1: Logo of AK Maju Resources Sendirian Berhad.

AK Maju Resources Sendirian Berhad, a private limited company was founded in 2008 by its CEO, Noor Azam. It was officially incorporated in 2014 in Segamat, Johor. The company's logo of AK Maju is displayed in Figure 4.2.1.

AK Maju provides construction and advertising services. The construction division offers a wide range of building materials to its clients involved in construction and renovation projects. Meanwhile, the advertising division focuses on graphic design for print advertisements. They work closely with clients to create visually appealing ads based on their target audience and campaign objectives.

The company operates with a team of seven staff members in a simple hierarchical structure, as shown in Figure 4.2.2. At the top, the Manager oversees all organizational activities. Below the Manager is the Admin/Account Officer who is responsible for financial management and operational tasks. The Customer Service Representative maintains the interaction with clients and customers. The design team comprises one Senior Designer, a second Senior Designer, and a Junior Designer. They are in charge of designing solutions for clients. Additionally, a Marketing Specialist responsible for managing promotional initiatives and marketing campaigns.

This streamlined organizational setup will allow AK Maju to deliver high-quality services efficiently. The clear reporting lines, where most employees report directly to the Manager, ensure effectiveness in communication and operational clarity.

Figure 4.2.2: Organizational chart of AK Maju Resources Sdn Bhd.

4.3 Current AK Maju System Analysis

AK Maju Resources Sdn. Bhd. currently relies on the manual processes for their major business operations such as project-based order management. These manual methods may work on a small scale but have become highly ineffective and prone to errors in recent times due to the growth in business. All project tracking is performed with very basic tools such as Excel spreadsheets and notebooks, none of which are suitably designed to efficiently track schedules, resource allocations, and progress. This has resulted in the complexity of workflows, frequent delays, and probable mismanagement in project timelines. Additionally, any mistakes related to data entry could result in the inaccuracy or loss of important project records, further complicating operations.

In terms of cost tracking and budget management, the reliance on manual calculations for material expenses and sales prices introduces delays and raises possibilities of errors. Without the real-time budget monitoring system, it is much easier to run into financial difficulties, and thus making it difficult to detect and address cost overruns promptly. Decisions cannot be taken effectively due to an absence of correct and timely financial data.

As AK Maju continues to expand in the construction sector, the limitations of the current system have become more pronounced. The management of multiple projects at the same time particularly exposes the shortcomings of the current system in handling scalability. Since there is no integrated project management system, the company struggles to maintain productivity and operational efficiency, impacting its ability to deliver quality services. This analysis underlines the urgent need for improvements in the system to overcome these challenges and support future growth.

4.4 Comparison Between Existing System And Proposed System

The existing system at AK Maju focuses on managing basic operations like order processing, quotation creation, and inventory tracking using a PHP-based architecture. However, it lacks advanced features such as real-time project tracking, data integration, and scalability, which are critical for managing complex projects efficiently.

The proposed system introduces an integration with SAP S/4HANA and SAP BTP to address these limitations. This integration streamlines operations, enhances project management depth, and provides advanced automation, real-time data analytics, and scalability to support AK Maju's growth. Below is a detailed comparison between the two systems.

Features	Existing System	Proposed System
Architecture and Design	Built on a basic architecture using PHP, HTML, CSS, JavaScript, Bootstrap, and a relational database. Data silos exist, requiring manual processes to consolidate information.	Utilizes SAP S/4HANA for enterprise-grade project management and SAP BTP for seamless integration with PHP interfaces. Features a unified architecture with integrated sub-modules for Project Structure Management, Budgeting, Scheduling, Resource Allocation, and Progress Monitoring. Centralized data management allows real-time updates and comprehensive visibility.
Features and Functionalities	No real-time tracking, resource allocation, or progress monitoring tools. Lacks visual tools like Gantt charts for scheduling or KPI dashboards for performance tracking.	Advanced features like Work Breakdown Structure (WBS), real-time project tracking, budgeting, and cost control. Includes Gantt charts for critical

		path monitoring and KPI-based dashboards. Fully integrated modules improve project management depth and resource utilization.
Data Handling	Relies on manual data entry, prone to errors and delays. No real-time data processing or analytics. Data silos exist between operational units.	SAP HANA's in-memory computing ensures real-time data processing and analytics. A centralized data repository synchronizes updates across all sub-modules, supporting seamless data integration and flow.
Scalability and Flexibility	Limited scalability, struggling to handle growing data volumes or complex workflows. Modifications require manual effort and code changes.	Cloud-based scalability through SAP BTP supports AK Maju's business growth. Modular design allows flexibility in adding or modifying sub-modules without disrupting workflows.
Automation and Efficiency	Processes like budgeting, resource allocation, and progress monitoring are highly manual, leading to inefficiencies and delays.	Automates these processes using SAP tools. Enhances workflow efficiency and collaboration through integrated modules and APIs, reducing reliance on manual intervention.

4.5 System Requirements Gathering Techniques

The enterprise system at AK Maju needs to go through a gathering process for system requirements effectively such as interviews with stakeholders (e.g., project managers, IT staff, finance teams, and end-users) are essential. Below is a categorized list of interview questions to capture functional, non-functional, and technical requirements.

1. General Requirements

- What challenges do you face in the current system?
- What key features should the new system have?
- Who will use the system, and what are their roles?

2. Functional Needs

- What tasks should the system automate or improve?
- How do you currently manage budgeting, scheduling, and resource allocation?
- What reports or dashboards do you need?

3. Technical & Non-Functional Needs

- Should the system integrate with existing tools?
- What are your security and access control requirements?
- Do you need mobile access and real-time updates?

4. User Experience & Training

- How should the system be designed for ease of use?
- What training or support will users need?
- How should we gather feedback for improvements?

4.6 System Requirements

In this section, we will present the detailed system requirements necessary to meet the project objectives. It is divided into two main categories which are Functional Requirements and Non-Functional Requirements.

4.6.1. Functional Requirements

1. **Project Structure Management:** The system must allow users to define a Work Breakdown Structure (WBS) to organize the project hierarchy into phases, tasks, and sub-tasks. It should support setting and managing project milestones.
2. **Budgeting and Cost Planning:** The system must allow users to define budgets and allocate them to specific WBS elements. Users should be able to track financial performance against budgets and perform cost estimation and forecasting.
3. **Scheduling:** The system must enable users to create detailed project schedules, including start and end dates for tasks and milestones. It should support monitoring critical paths using visual tools like Gantt charts.
4. **Resource Allocation:** The system must facilitate the assignment of materials to various orders in project. It should provide real-time tracking of resource availability and utilization to optimize assignments and avoid conflicts.
5. **Progress Monitoring:** The system should enable users to track task completion and milestone achievements. It must provide key performance indicators (KPIs) and generate progress reports to measure project performance.

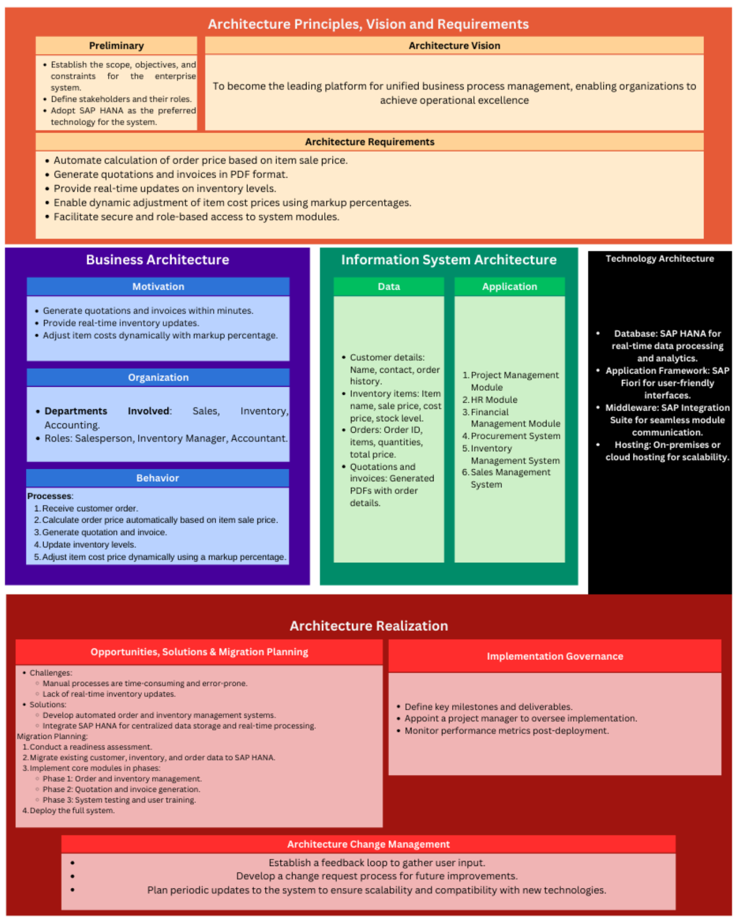
4.6.2. Non Functional Requirements

1. **Performance:** The system must handle up to 500 concurrent users without significant delays, with a response time of less than 2 seconds for most operations.
2. **Usability:** The user interface should be intuitive and user-friendly, ensuring ease of navigation for both technical and non-technical staff.
3. **Scalability:** The system should support future scalability, allowing integration of additional features or modules without major architectural changes.

4. **Security:** The system should implement security measures to ensure authorized access. Only authenticated admins should be able to log in and access project-related information.
5. **Reliability:** The system should be available to handle order operations in various projects and provide seamless operation without frequent crashes or downtime.

4.7. System Design

4.7.1 Enterprise architecture



4.7.2 Explain each component in enterprise architecture

The enterprise architecture uses the TOGAF framework to develop this enterprise system for AK Maju. The **Preliminary Phase** lays the foundation for the architecture by defining the scope, objectives, and constraints of the system. It also identifies stakeholders and their roles and selects

SAP HANA as the preferred technology to support real-time data processing and analytics. This phase sets the stage for all subsequent architectural activities.

The **Architecture Vision** outlines the overarching goal of the system, which is to become a unified platform for business process management. This vision emphasizes enhancing operational excellence by streamlining processes and leveraging advanced technology.

The **Architecture Requirements** define the core capabilities of the system. These include automating order price calculations, generating quotations and invoices in PDF format, providing real-time inventory updates, and dynamically adjusting item costs using markup percentages. Additionally, the system is designed to ensure secure and role-based access to critical modules.

The **Business Architecture** focuses on organizational roles, processes, and motivations. It aims to improve efficiency by automating quotation generation, enabling real-time inventory updates, and supporting dynamic pricing. The architecture involves key departments like sales, inventory, and accounting, with roles such as salespersons, inventory managers, and accountants. The business processes include receiving customer orders, calculating prices, generating invoices, updating inventory, and adjusting item costs.

The **Information System Architecture** is divided into two subcomponents: data and application. The data segment specifies the essential information managed by the system, such as customer details, inventory items, and generated documents like invoices. The application segment lists the system modules, including project management, financial management, procurement, sales management, and others, which collectively enable seamless operations across various business functions.

The **Technology Architecture** describes the technical infrastructure of the system. It employs SAP HANA for real-time data processing and analytics, SAP Fiori for user-friendly interfaces, and the SAP Integration Suite for seamless communication between modules. Hosting options, whether on-premises or cloud-based, provide flexibility and scalability to meet organizational needs.

The **Architecture Realization** addresses implementation and change management. It identifies challenges, such as time-consuming manual processes and the lack of real-time data, and proposes solutions like automating order and inventory management. A phased migration plan ensures smooth implementation, starting with readiness assessments and ending with full deployment. Implementation governance involves defining milestones, appointing a project manager, and monitoring performance post-deployment. To adapt to evolving requirements, architecture change management facilitates user feedback, handles change requests, and updates the system to incorporate new technologies.

4.7.3 System Architecture

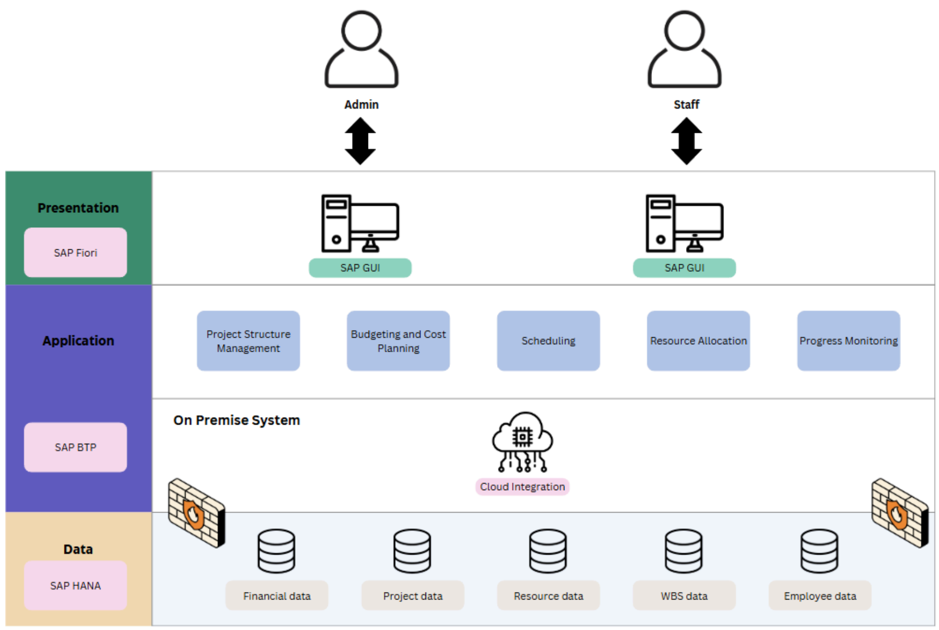


Figure 4.7.3.1: System Architecture Diagram for Project Management Module

4.7.4 Explain each component in system architecture

The system architecture is designed to streamline operations through a structured and integrated framework. It supports two primary user roles: Admin and Staff. In the project management sector, they can perform functionalities like project structure management, budgeting and cost planning, scheduling, resource allocation, and progress monitoring.

The **Presentation Layer** utilizes SAP Fiori to enhance user experience by providing intuitive applications. This modern interface simplifies workflows, improves accessibility, and ensures that both Admin and Staff can effectively perform their tasks within the system.

The **Application Layer** acts as the functional core, supporting operations such as project structure management, budgeting and cost planning, scheduling, resource allocation, and progress monitoring. These functionalities ensure efficient project execution and proper utilization of resources. Besides, the On-premise System has integration with **SAP Business Technology Platform (BTP)** to achieve hybrid development for seamless connectivity with on-premise to cloud environments. Cloud integration enhances the system by allowing real-time data exchange and ensuring that it is scalable and accessible.

The **Data Layer** is powered by SAP HANA, a high-performance in-memory database. including financial data on budgets and costs, project data related to tasks and milestones, resource data such as personnel and equipment, WBS data, and employee data. All these datasets are well-protected through a powerful firewall, preventing them from unauthorized access and cyber threats. Together, this architecture ensures a secure, and efficient system for managing projects and resources.

4.7.5 Project Design

4.7.5.1 Use Case Diagram for Enterprise AK Maju System

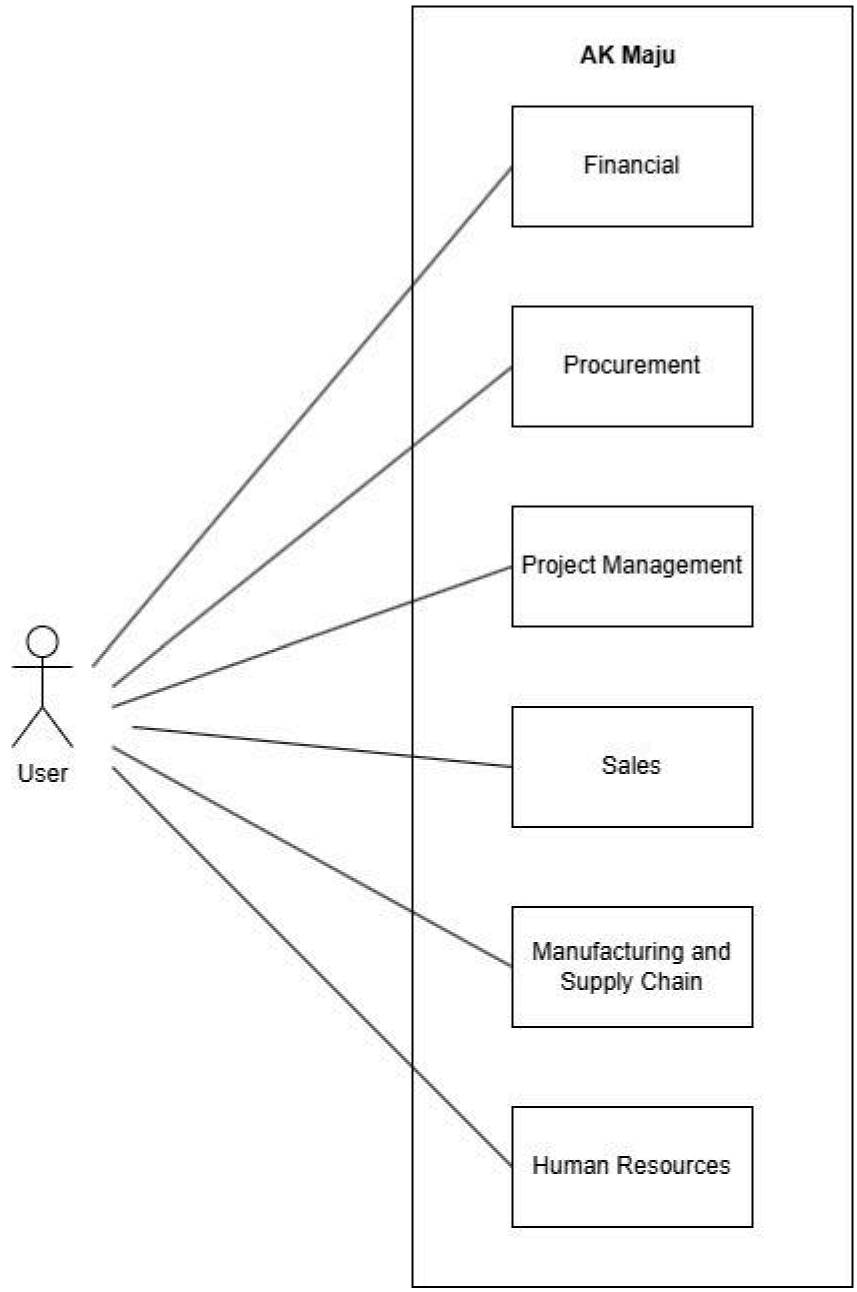


Figure 4.7.5.1.1: Use Case Diagram for Enterprise AK Maju System

4.7.5.2 Use Case Diagram for AK Maju Project Management Subsystem

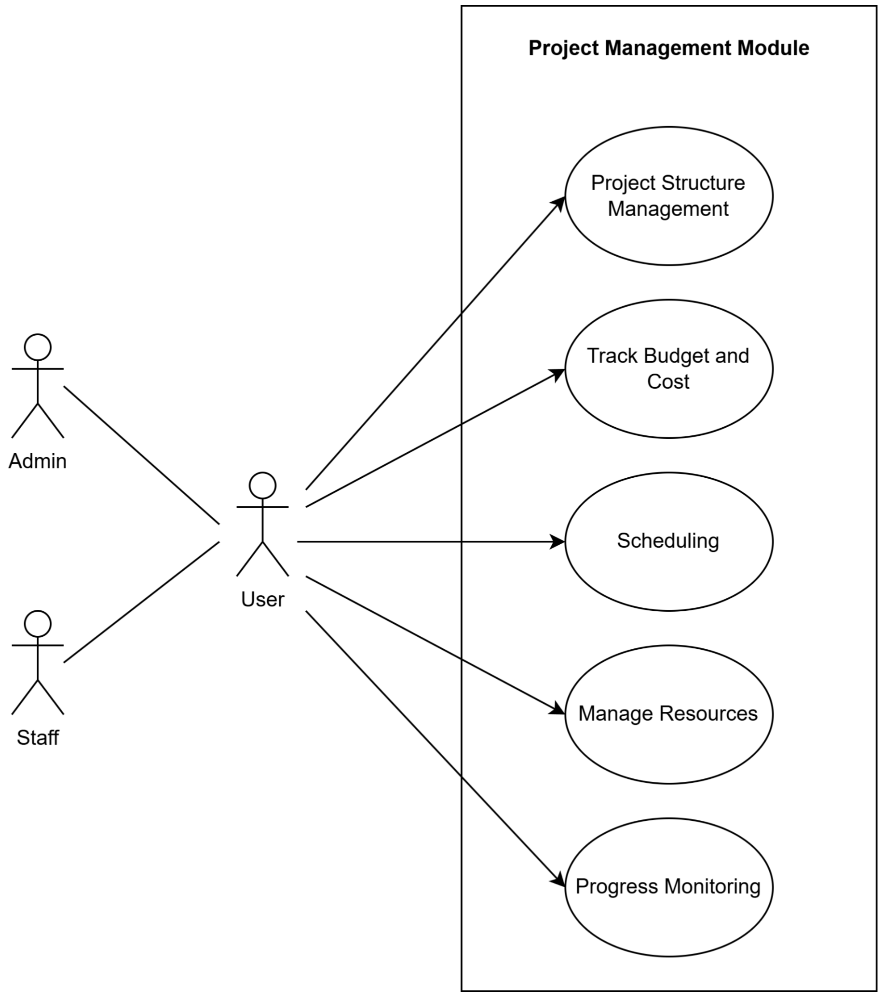


Figure 4.7.5.2.1: Use Case Diagram for Project Management Module

4.7.5.3 Submodule 1: Project Structure Management

Use Case Description

Table 4.7.5.3.1: Use Case Description for <Project Structure Management>

Use case: <Project Structure Management>
ID: UC01
Actors: Admin, Staff
Preconditions: 1. The user must be logged into the system with appropriate permissions to manage project structures. 2. The SAP HANA database must be available and operational.
Flow of events: 1. If the user selects the "Add/Update Structure" option. 2. The system prompts the user to input project details such as title, descriptions, date and assignees. 3. The user enters the details and submits them for validation. 4. The system validates the input: a. Ensures that all required fields are filled. b. Checks for duplicate or conflicting entries. 5. If validation is successful: a. The system stores the project data in the SAP HANA database. b. The system notifies relevant stakeholders about the new or updated project structure. 6. The process ends successfully. 7. If the user selects the "Delete Structure" option. 8. The system prompts the user to confirm the deletion action. 9. The system validates the deletion request: a. Ensures the project is not linked to active dependencies (e.g., ongoing tasks or other modules).

10. If validation is successful:
  a. The system deletes the project structure from the SAP HANA database.
  b. The system notifies stakeholders about the deletion.
11. The process ends successfully.

Postconditions: 1. For Add/Update: The project structure is stored or updated in the database, and stakeholders are notified. 2. For Delete: The project structure is removed from the database, and stakeholders are notified.
Alternative flow 1 (Add/Update): 1. The system displays an error message to the user, indicating the issues with the input data. 2. The user corrects the input and resubmits the details.
Alternative flow 2 (Delete): 1. The system displays an error message to the user, indicating the reasons for the failure (e.g., active dependencies). 2. The user resolves the issue and retries the deletion process.

Exception flow (if any):
1. SAP HANA database is unavailable:
  a. The system displays a message to the user and logs the error for system administrators.
2. User permissions are insufficient:
  a. The system restricts the action and informs the user.

Activity Diagram

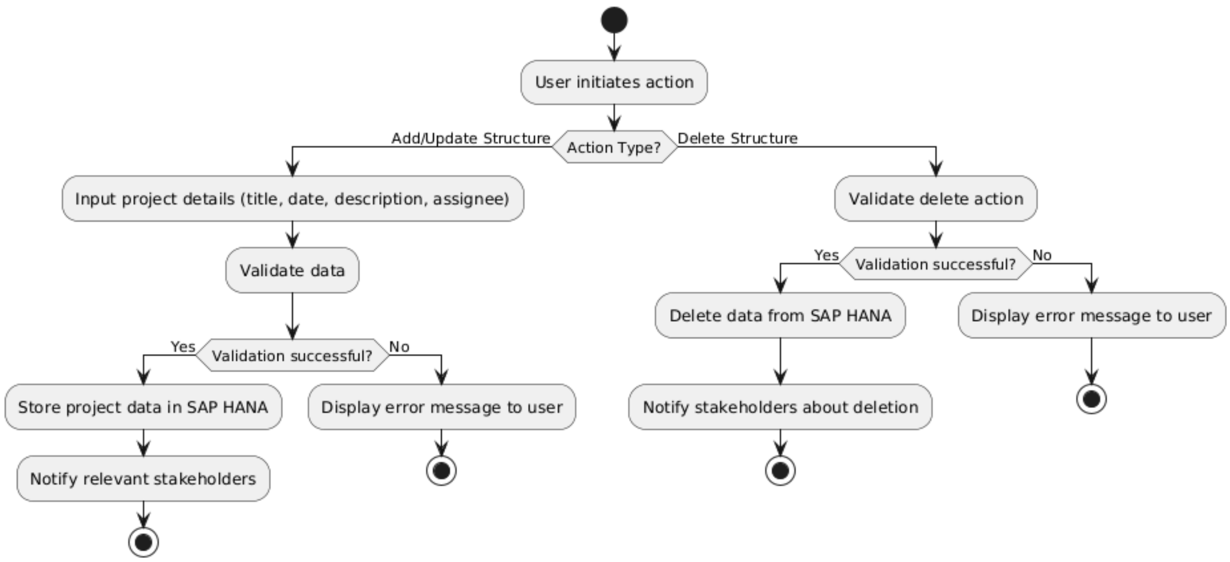


Figure 4.7.5.3.1: Activity Diagram for <Project Structure Management>

Sequence Diagram

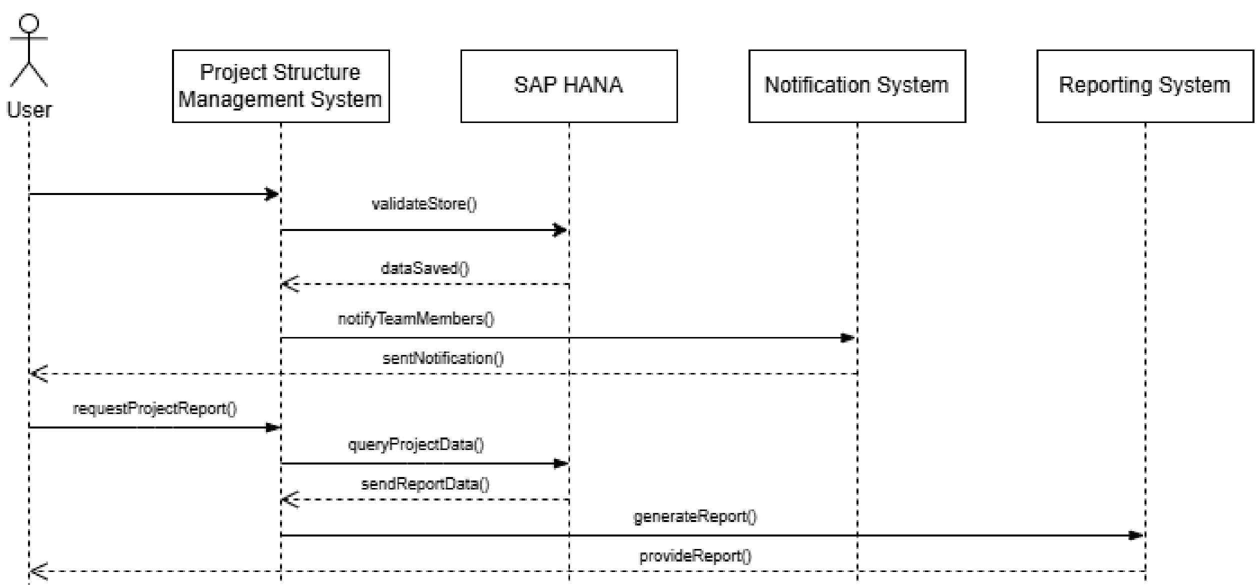


Figure 4.7.5.3.2: Sequence Diagram for <Project Structure Management>

4.7.5.4 Submodule 2: Budget and Cost Tracking

Use Case Description

Table 4.7.5.4.1: Use Case Description for <Track Budget and Cost>

Use case: <Track Budget and Cost>
ID: UC02
Actors: Admin, Staff
Preconditions: 1. The user has defined the tasks in Work Breakdown Structure (WBS). 2. The user has allocated resources to each task. 3. Resource cost data is available in the system.
Flow of events: 1. The user redirects to the Budget and Cost Planning page after allocating the resources. 2. The system calculates the budget based on the allocated resources for each WBS element. 3. The user enters actual costs incurred for specific tasks. 4. The system compares the actual costs against the calculated budget for each WBS element by calculating variances (positive or negative). 5. The system flags any budget overruns or underspending. 6. The user clicks the “Continue” button to the Financial Performance Report page. 7. The system generates a financial performance report, showing Total budget, Budget Remaining, Overall variances, Percentage variances, Overall project financial status, Budget distribution by WBS elements and Comparison of budget and actual costs.

Postconditions: 1. Financial performance reports are generated. 2. Any budget discrepancies are identified.
Alternative flow 1: 1. Before redirecting to the Financial Performance Report page, the system will check for any budget overrun. 2. If the system detects a budget overrun for any WBS element, the system displays a pop-up alert stating the overrun. 3. The pop-up includes options for the user to either: 3.1. Readjust the resources allocation 3.1.1. The system redirects user to Resource Allocation page 3.2. Ignore the overrun
Postconditions: 1. If resources are edited, the system automatically updates the budget. 2. If no resources are edited, the system displays the financial performance report.
Alternative flow 2: 1. If the user enters incorrect or invalid data like negative costs, the system prompts them with an error message. 2. The user corrects the data and re-enters it.
Postconditions: 1. Only valid data is stored in the system.
Exception flow (if any): -

Activity Diagram

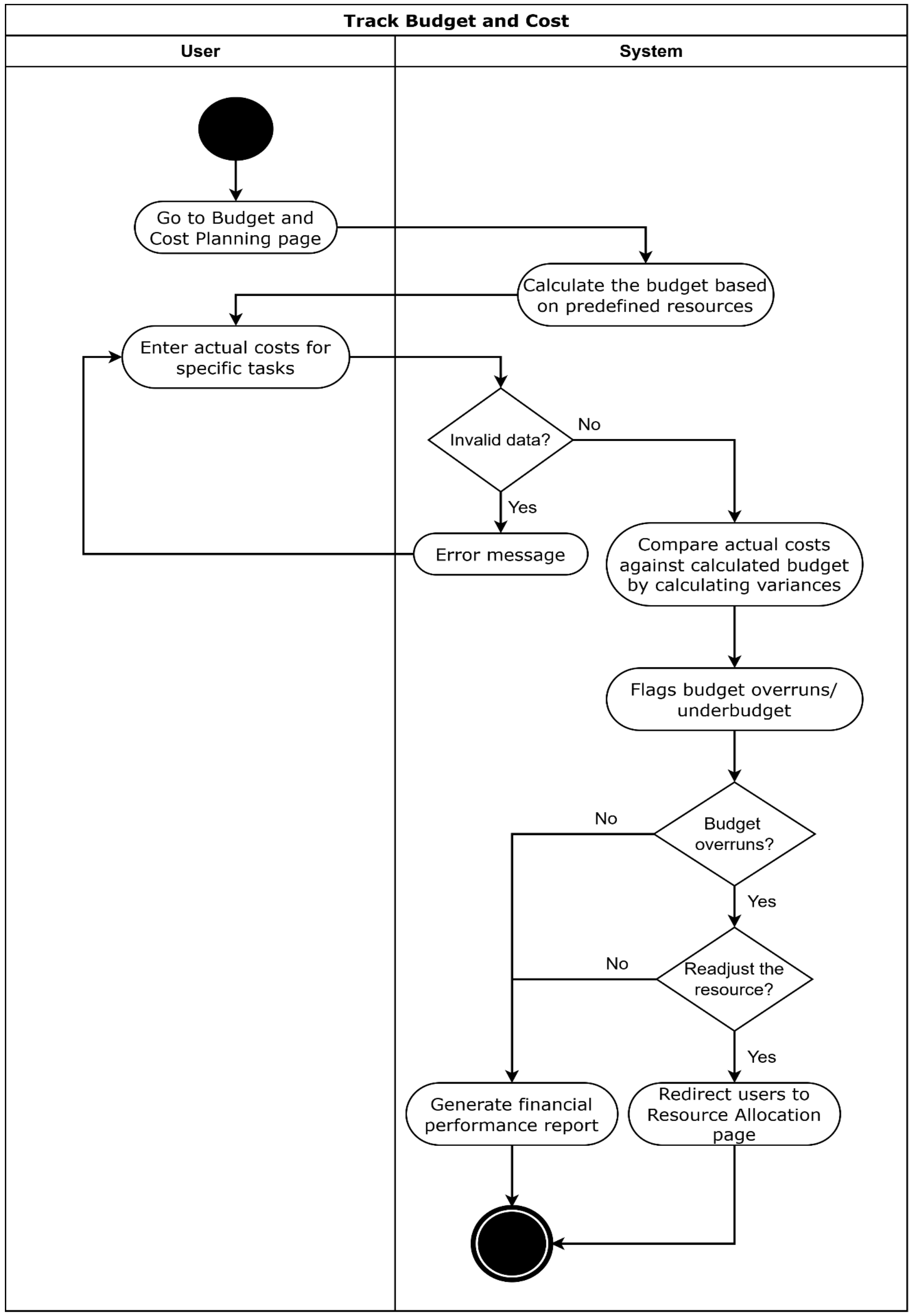


Figure 4.7.5.4.1: Activity Diagram for <Track Budget and Cost>

Sequence Diagram

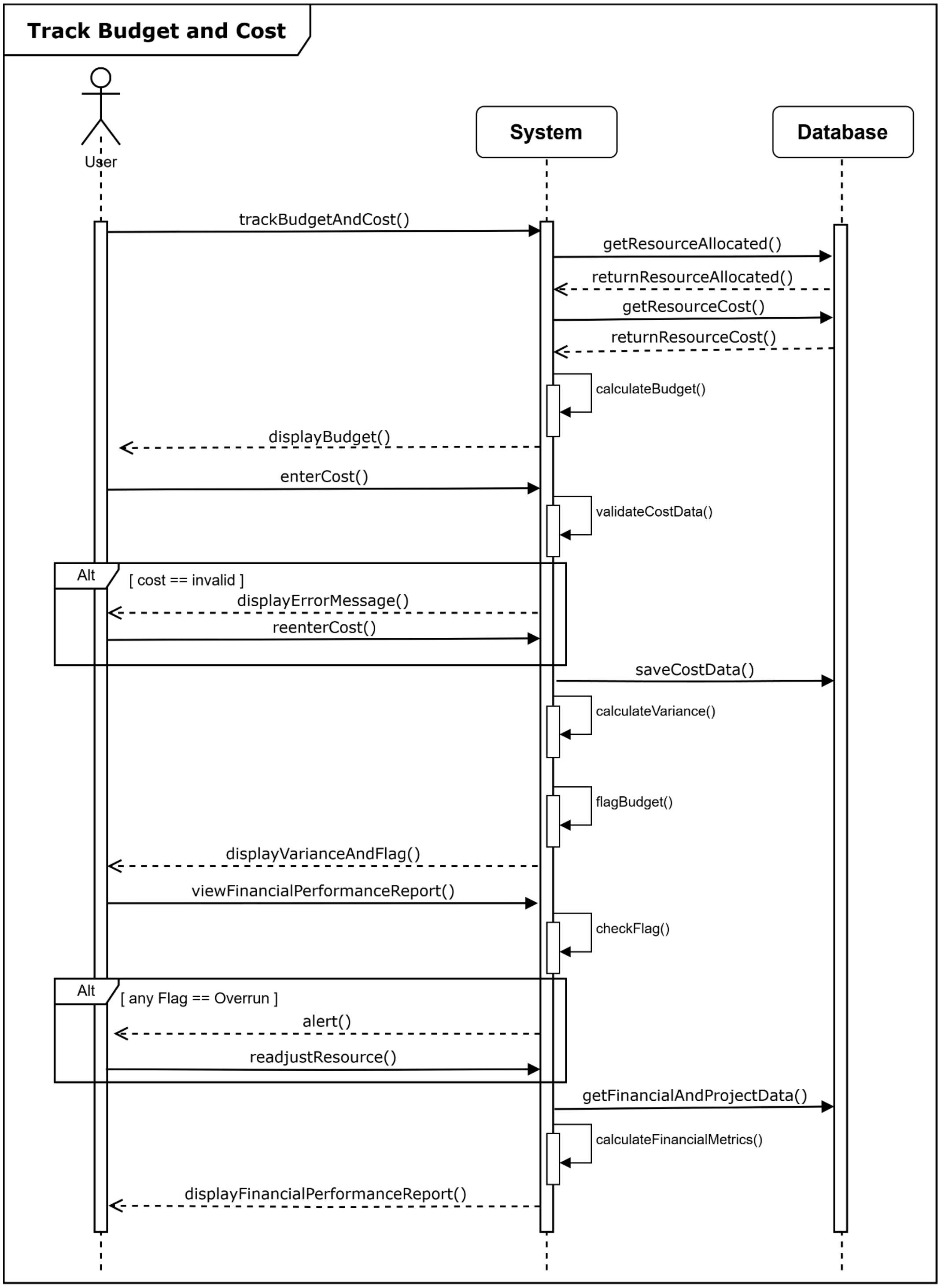


Figure 4.7.5.4.2: Sequence Diagram for <Track Budget and Cost>

4.7.5.5 Submodule 3: Scheduling

Use Case Description

Table 4.7.5.5.1: Use Case Description for <Scheduling>

Use case: <Scheduling>
ID: UC03
Actors:Admin, Database System, Staff
Preconditions: The admin is logged into the system. The project must already be created from the system.
Flow of events: 1. Admin opens the scheduling interface. 2. Admin navigates to the project scheduling module. 3. Admin selects the option to create a new task. 4. Admin enters the following task details: a. TaskID b. Description c. Start Date d. End Date e. Status f. Assigned To 5. Admin submits the task to the system. 6. The system validates the input and stores the task in the database. 7. The assigned staff member receives a notification about the new task. 8. If desired, Admin exports the task list to a CSV file.
Postconditions:

1. Tasks are successfully created, updated, or deleted in the system. 2. The assigned staff member is notified of new or updated tasks. 3. If exported, the CSV file is generated and downloaded by the Admin.
Alternative flow *n*: 9. If the Admin wants to update or delete an existing task: ● They select the task from the task list. ● Perform the desired operation (update or delete).
Postconditions: 1. The task list is updated in the system. 2. Modifications are reflected in real-time for both Admin and assigned staff. 3. Staff members receive notifications of updates or deletions.
Exception flow (if any):

Activity Diagram

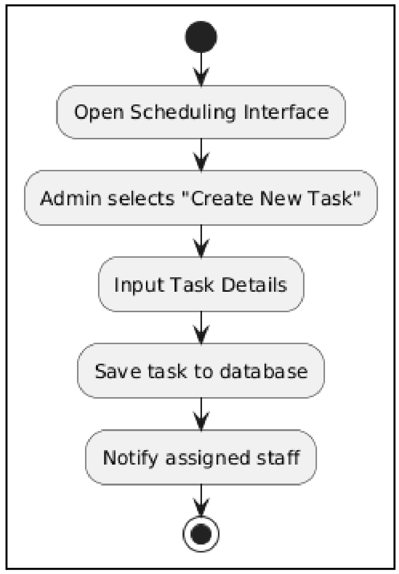


Figure 4.7.5.5.1 Activity Diagram for <Scheduling>

Sequence Diagram

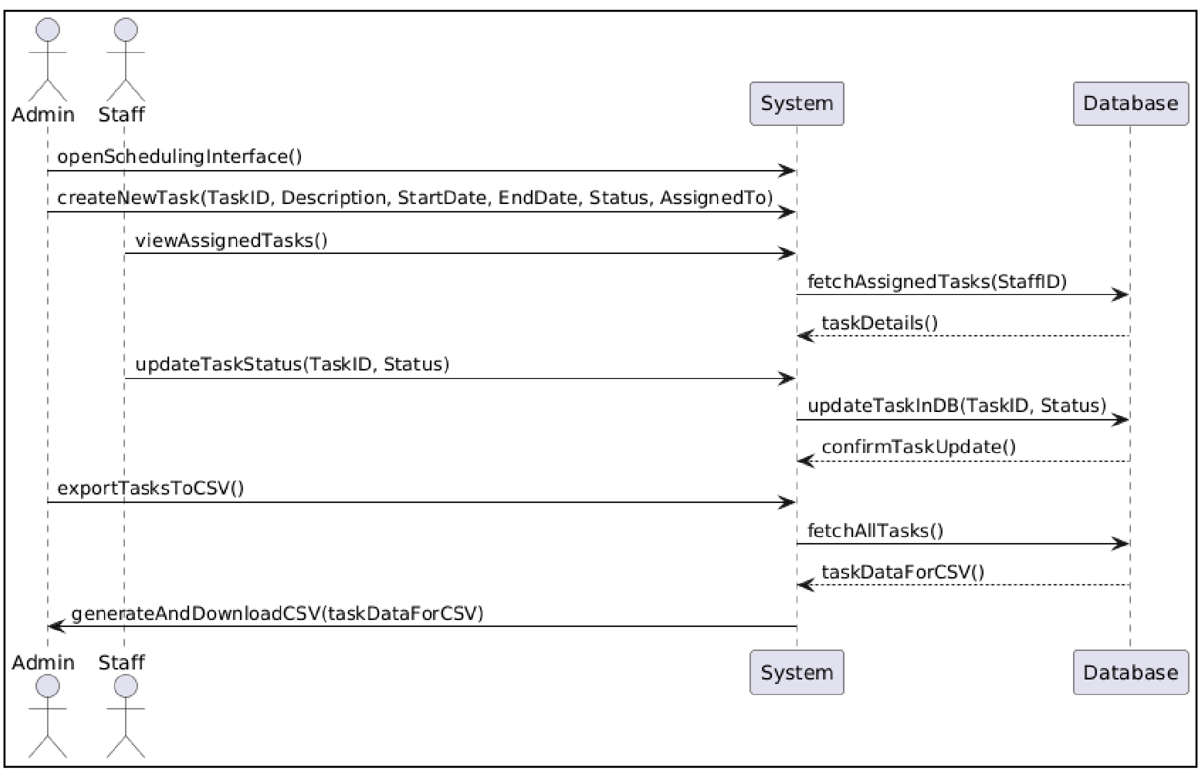


Figure 4.7.5.5.2 Sequence Diagram for <Scheduling>

4.7.5.6 Submodule 4: Resource Allocation

Use Case Name: Manages Resources

Use Case Description: This use case describes the allocation of materials to specific tasks and dynamically updates availability statuses to ensure efficient allocation of materials to project tasks while providing usage optimization insights.

Actor: Project Manager

Preconditions:

1. Inventory database is updated with current availability and capacities.
2. The project details must be recorded in the system.

Main Flow:

1. Project Manager logs into the system.
2. The system displays the project details on dashboard.
3. Project Manager selects the project for resource allocation.
4. System displays available resources based on real-time inventory availability.
5. Project Manager allocates resources by selecting the amount of specific materials from the list.
6. System checks for conflicts or shortages and suggests alternatives if necessary.
7. System updates resource utilization and displays remaining availability.

Postconditions:

1. Quotation is successfully processed.
2. Materials are allocated to project tasks without conflicts.
3. Notifications for low inventory levels are triggered if specific materials reach the minimum amount level.

Alternative flow 1: Insufficient amount of required materials

At step 6, if the materials selected exceed available stock:

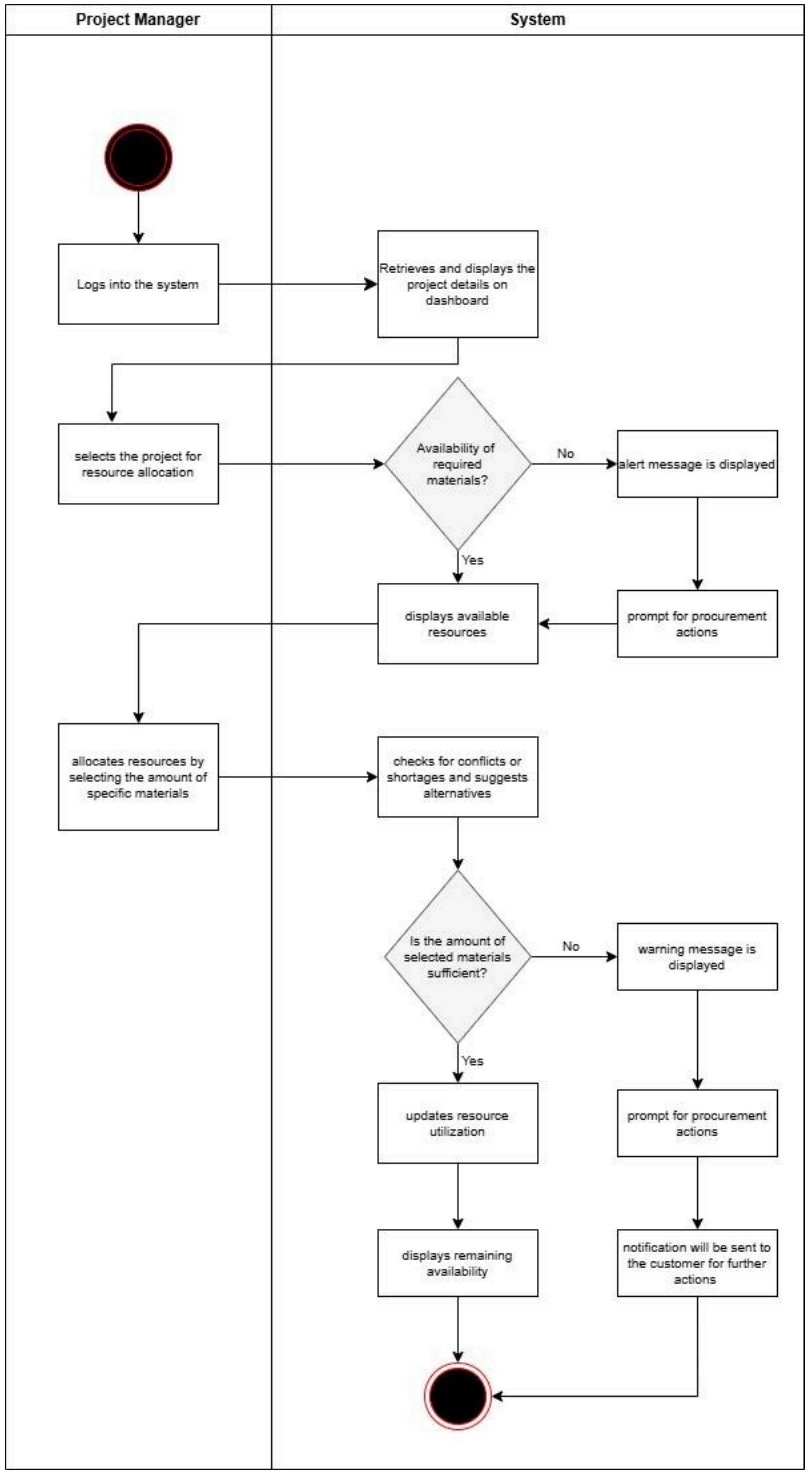
1. Allocation process is forced to stop, and a warning message is displayed.
2. Project Manager is issued a prompt for procurement actions.
3. A notification is sent to the customer regarding insufficient material availability for further actions.

Exception flow 1: Insufficient required materials

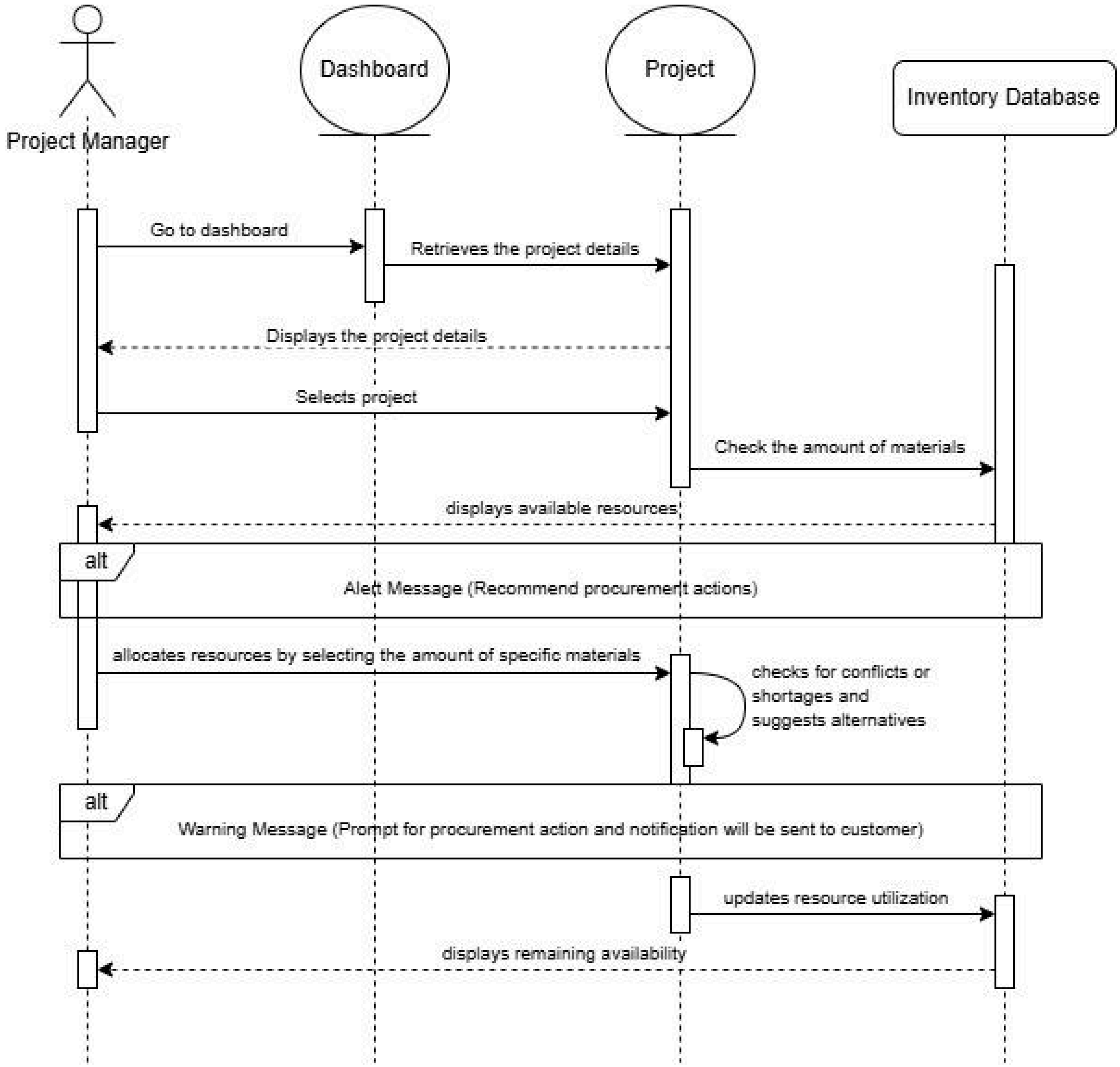
At the step 4, if insufficient resources are available:

1. The system generates an alert to project manager and recommends procurement actions.

Activity Diagram



Sequence Diagram



4.7.5.7 Submodule 5 : Progress Monitoring

Use Case Description

Table 4.7.5.7.1: Use Case Description for <Progress Monitoring>Activity Diagram

Use case: <Progress Monitoring>
ID: UC001
Actors: Project Manager
Preconditions: 1. The project must be created in the system with assigned resources, tasks, and milestones. 2.The user must have appropriate access rights.
1. **System Initialization:**The system is started, and the database is updated with the latest project data. 2.**User Login:**The Project Manager logs into the system with valid credentials.The system verifies user access and displays the dashboard. 3. **Project Selection:**The Project Manager views a list of ongoing projects.They select a specific project to monitor progress. 4. **Dashboard Display:**The system shows the project dashboard, including:Project timeline (via Gantt chart).Task completion percentages.Status of milestones.Resource allocation and utilization metrics.Budget tracking and financial overview. 5. **Issue Identification:**The Project Manager spots a delay or issue in task progress. 6. **Action Taken:**The Project Manager selects the delayed task to view details.They adjust the task deadlines or reassign resources to resolve the delay.The system updates the project plan. 7. **Notifications:**The system sends notifications to affected team members about the updates. 8.**Progress Update:**The system updates the project progress status in real-time.

Postconditions:
The project progress status is updated in the system, reflecting accurate task completion percentages, milestone achievements, and any timeline adjustments made during the monitoring process.

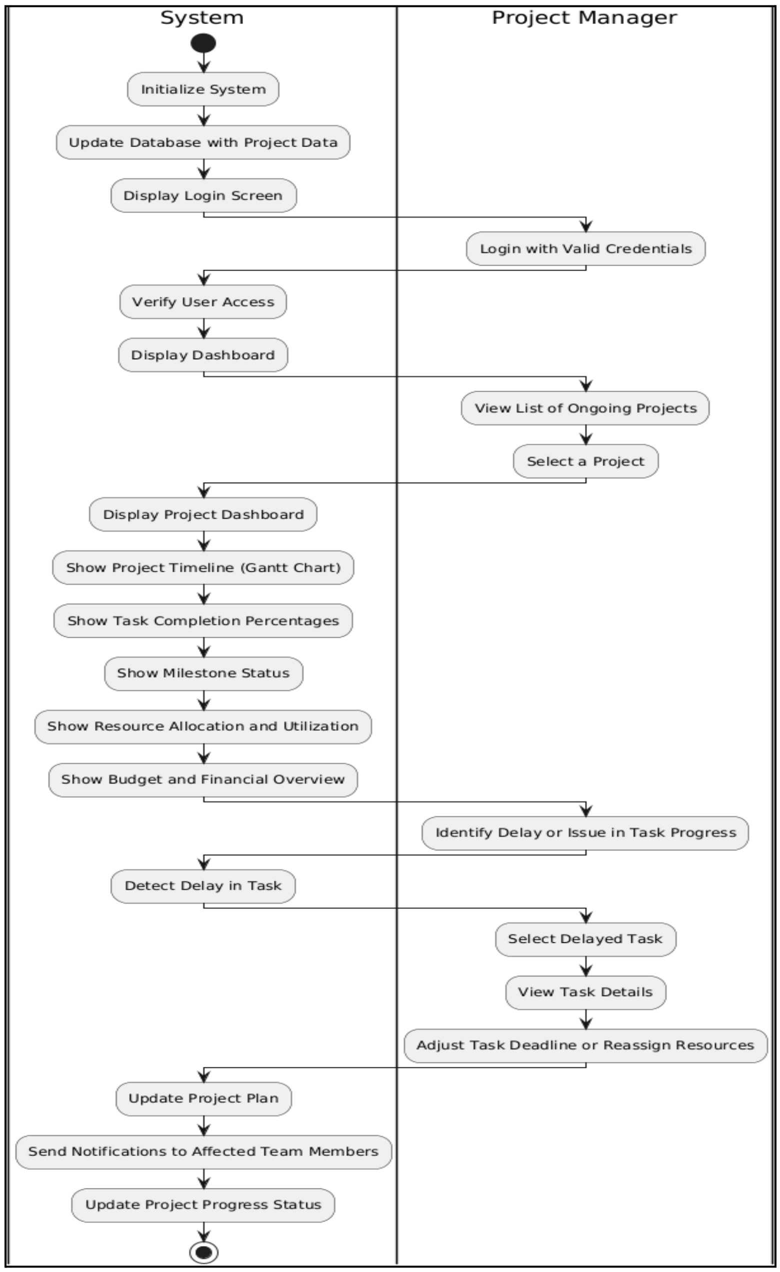


Figure 4.7.5.7.1 Activity Diagram for <Progress Monitoring>

Sequence Diagram

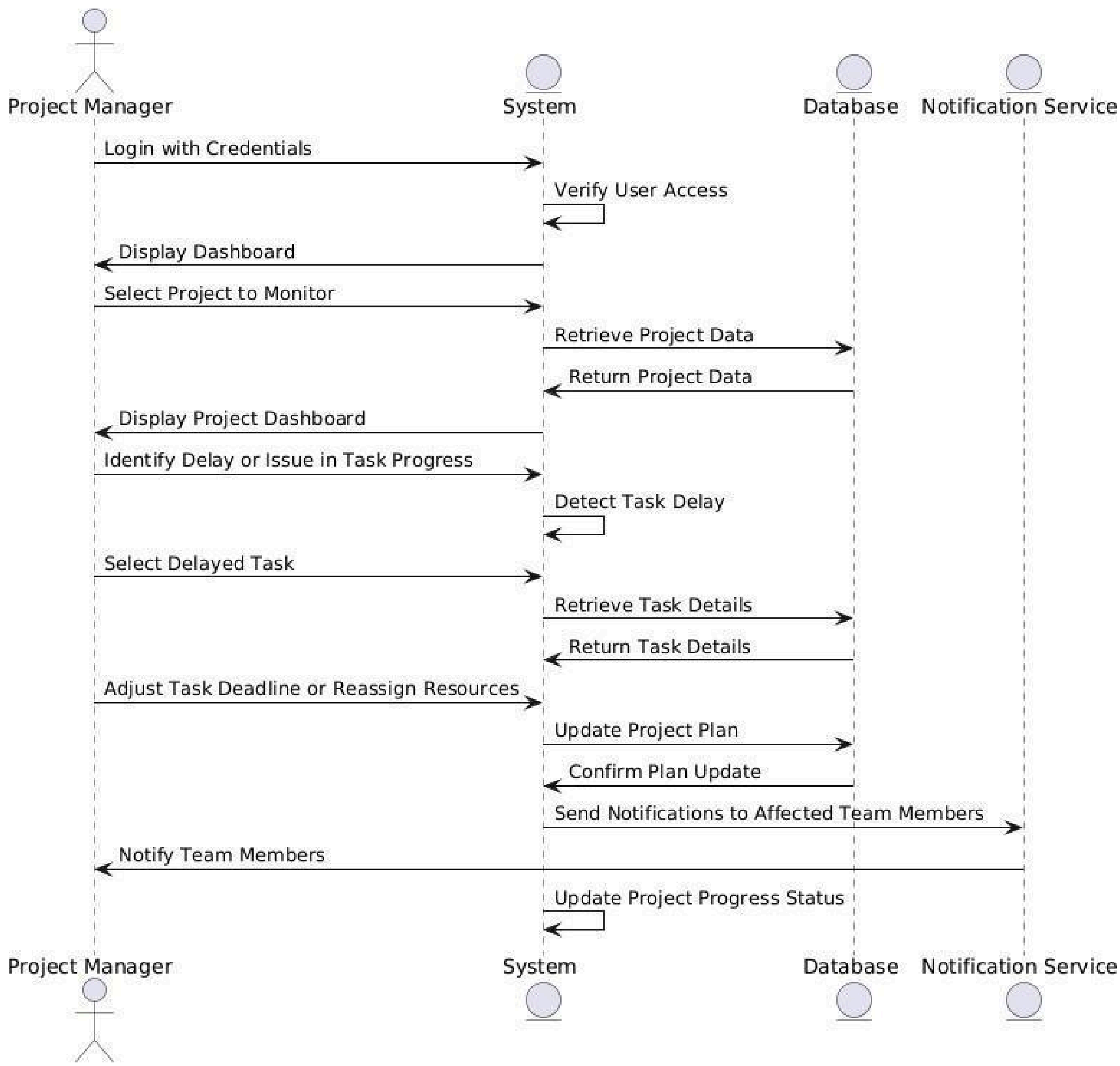


Figure 4.7.5.7.2 Sequence Diagram for <Progress Monitoring>

4.7.6 Database Design *(group)*

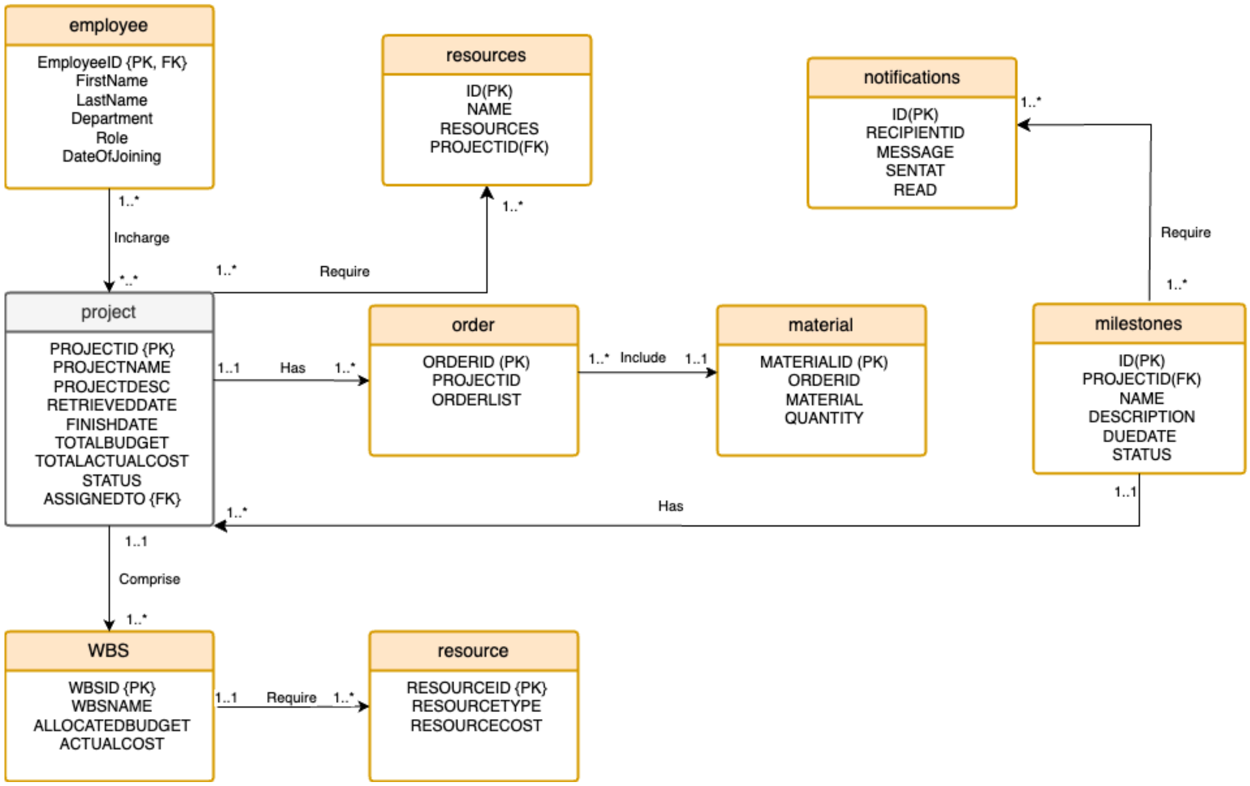
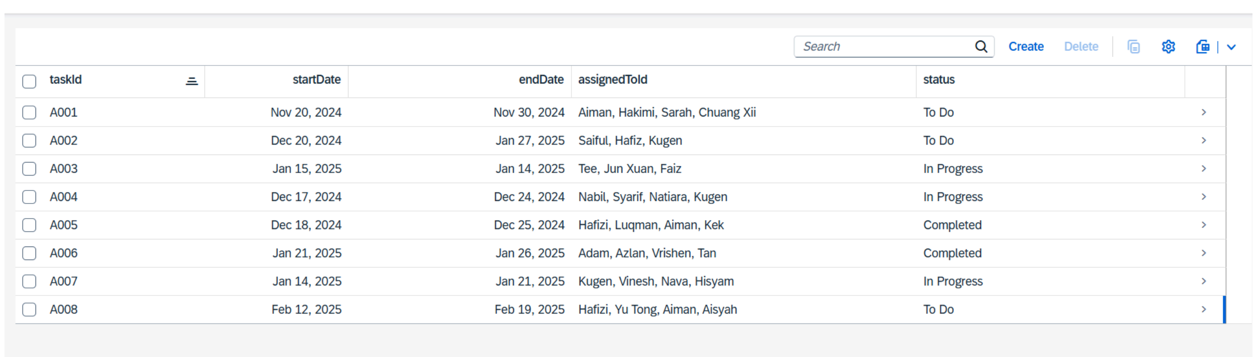


Figure 4.7.6.1 Database Design for Project Management Module

4.7.7 Interface Design

4.7.7.1 Submodule 1:Project Structure Management



taskId	startDate	endDate	assignedToId	status
A001	Nov 20, 2024	Nov 30, 2024	Aiman, Hakimi, Sarah, Chuang Xii	To Do
A002	Dec 20, 2024	Jan 27, 2025	Saiful, Hafiz, Kugen	To Do
A003	Jan 15, 2025	Jan 14, 2025	Tee, Jun Xuan, Faiz	In Progress
A004	Dec 17, 2024	Dec 24, 2024	Nabil, Syarif, Natiara, Kugen	In Progress
A005	Dec 18, 2024	Dec 25, 2024	Hafizi, Luqman, Aiman, Kek	Completed
A006	Jan 21, 2025	Jan 26, 2025	Adam, Azlan, Vrishen, Tan	Completed
A007	Jan 14, 2025	Jan 21, 2025	Kugen, Vinesh, Nava, Hisyam	In Progress
A008	Feb 12, 2025	Feb 19, 2025	Hafizi, Yu Tong, Aiman, Aisyah	To Do

Figure 4.7.7.1.1: Project Structure Management Page

Figure 4.7.7.1.1 shown Project Structure Management page which displays a task management table that organizes and tracks tasks with detailed information. Each task is identified by a unique **taskId** (e.g., A001, A002) and includes a **startDate** and **endDate** to indicate when the task is scheduled to begin and when it is expected to be completed. The **assignedToId** column lists the team members or individuals responsible for each task, while the **status** column reflects the current progress of the task, such as "To Do," "In Progress," or "Completed." At the top of the page, there is a search bar that allows users to filter or locate specific tasks, and buttons for "Create" and "Delete" provide functionality for managing tasks. This layout suggests a structured and user-friendly interface for task tracking and team coordination.

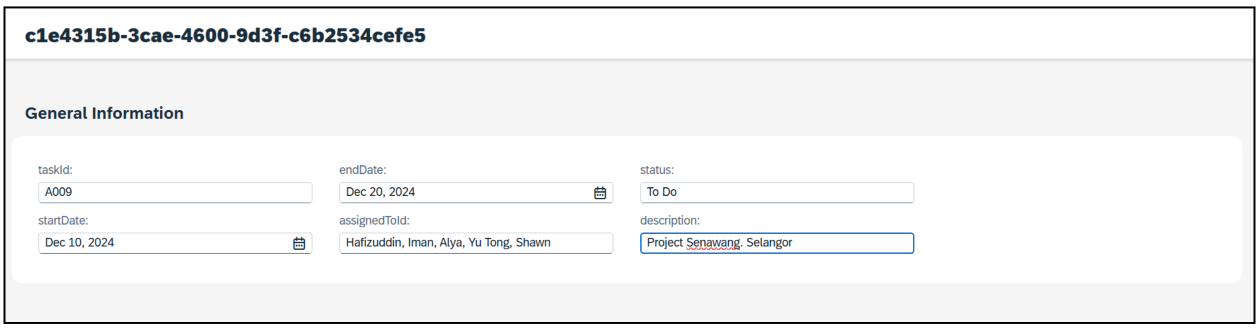


Figure 4.7.7.1.2: Create New Project Page

Figure 4.7.7.1.2 shown a Create New Project page with a detailed task view for a task identified as **A009**, with additional information under "General Information." The task has a **startDate** of December 10, 2024, and an **endDate** of December 20, 2024. Its current **status** is listed as "To Do." The task is assigned to multiple team members: Hafizuddin, Iman, Alya, Yu Tong, and Shawn. There is also a **description** field specifying the task's context as "Project Senawang, Selangor." The interface appears structured, with editable fields for easy updates to the task details.

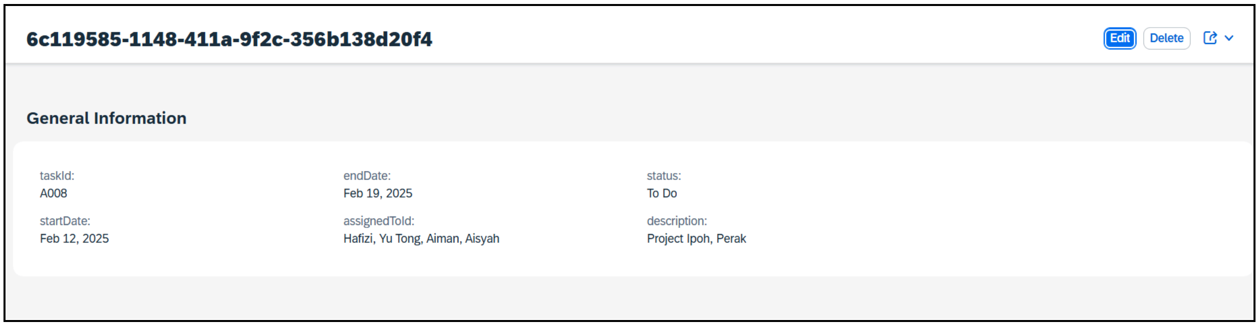
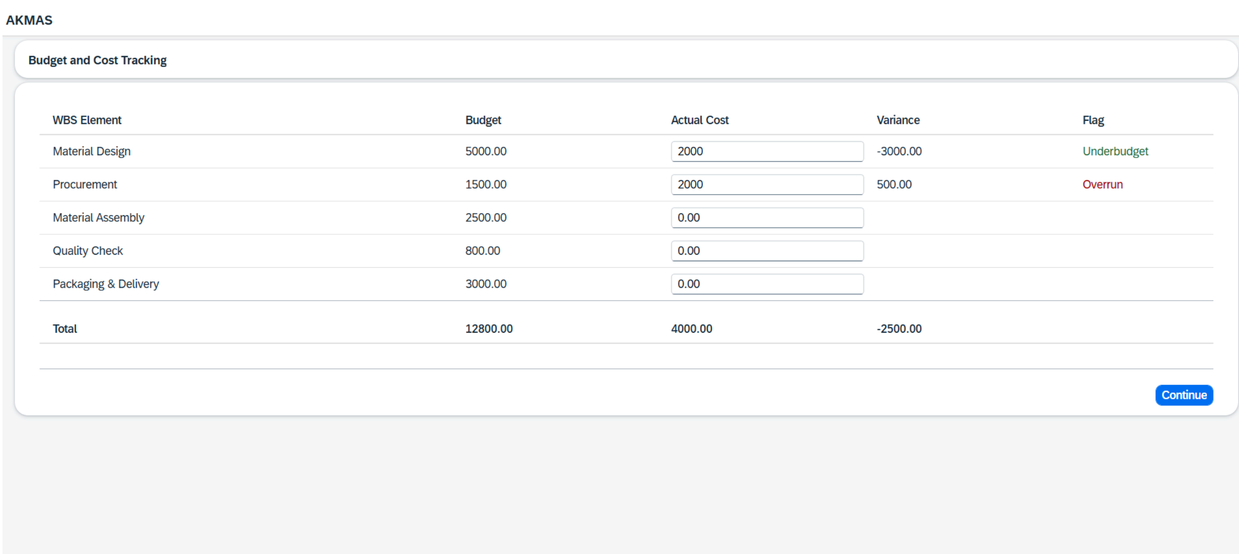


Figure 4.7.7.1.3: Edit and Delete Project Page

Figure 4.7.7.1.3 shown a task detail interface with several key components. At the top, there is a unique identifier displayed prominently. Below, a section labeled "General Information" is present, containing labeled fields such as **taskId**, **startDate**, **endDate**, **status**, **assignedToId**, and **description**, each displaying corresponding information. At the top-right corner, there are buttons for **Edit** and **Delete**, allowing the user to modify or remove the task. The overall layout is simple and organized, designed for easy viewing and management of task details.

4.7.7.2 Submodule 2: Budget and Cost Tracking



WBS Element	Budget	Actual Cost	Variance	Flag
Material Design	5000.00	2000	-3000.00	Underbudget
Procurement	1500.00	2000	500.00	Overrun
Material Assembly	2500.00	0.00		
Quality Check	800.00	0.00		
Packaging & Delivery	3000.00	0.00		
Total	12800.00	4000.00	-2500.00	

Figure 4.7.7.2.1: Budget and Cost Tracking Page

Figure 4.7.7.2.1 displays the budget, actual cost, variance, and flags for various work breakdown structure (WBS) elements. The budget is calculated and shown based on the resource allocated. After the user enters the actual cost, the variances (budget - actual cost) and flag (Underbudget or Overrun) will be shown.

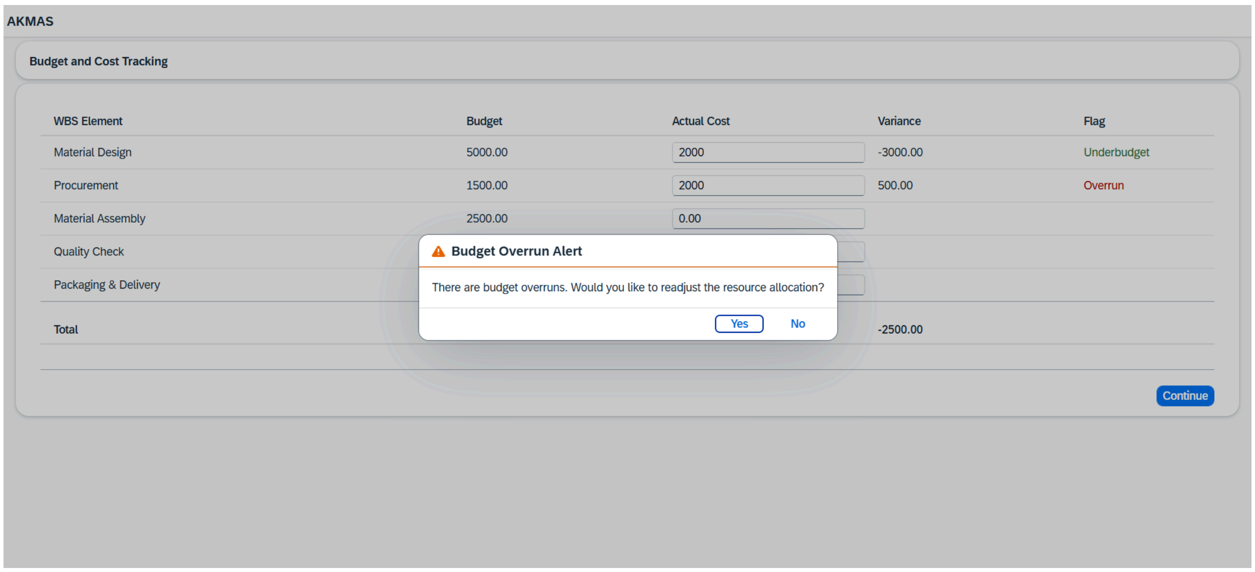


Figure 4.7.7.2.2: Budget Overrun Alert

Figure 4.7.7.2.2 shows an alert when there is any overrun (actual cost exceeds the budget), prompting the user to adjust resource allocation. Selecting "Yes" will go to the Resource Allocation page, selecting "No" will go to the Financial Performance page.

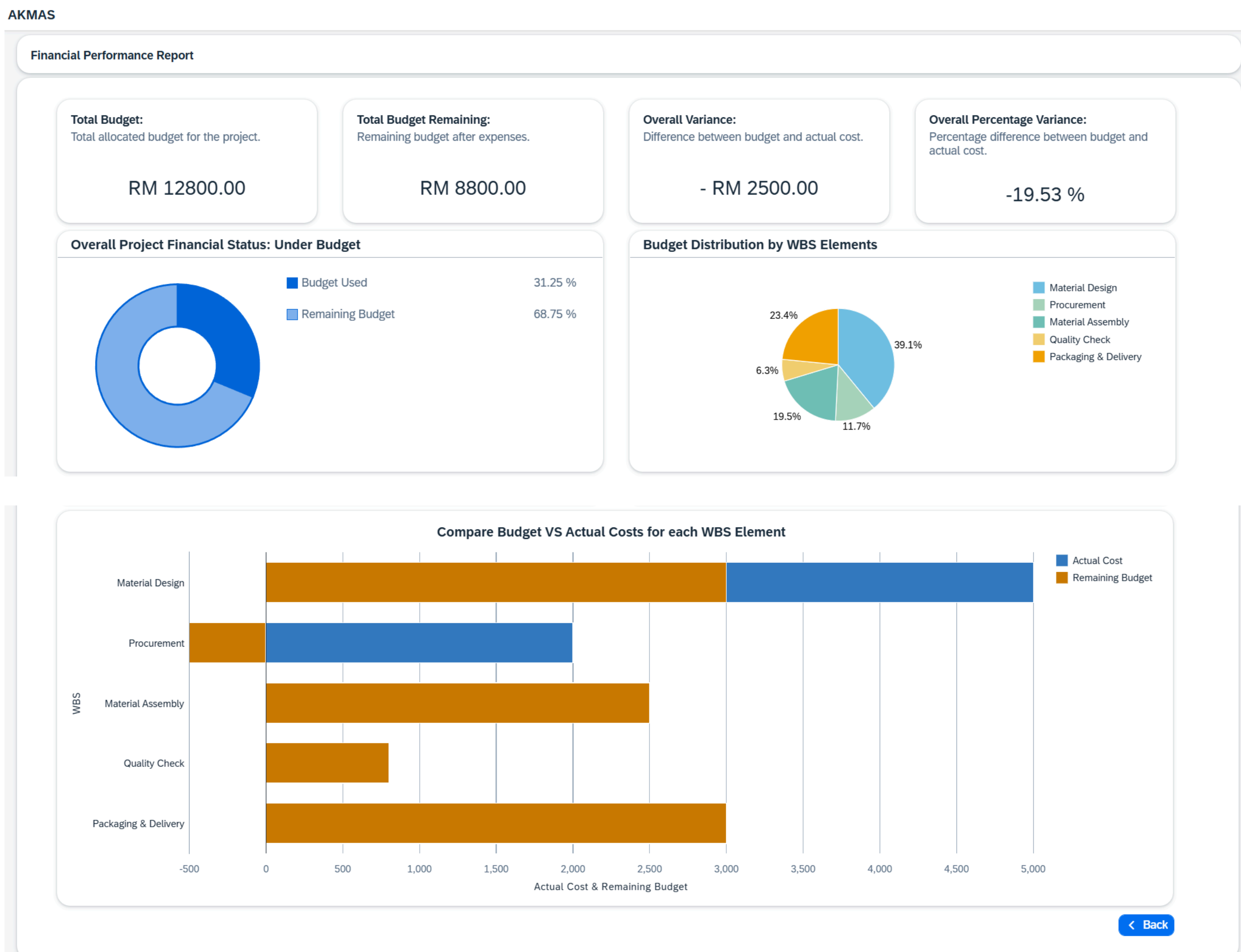


Figure 4.7.7.2.3: Financial Performance Report Page

Figure 4.7.7.2.3 displays the Financial Performance Report, showing Total Budget, Budget Remaining, Overall Variances, Percentage Variances, Overall Project Financial Status, Budget Distribution by WBS Elements and Comparison of Budget and Actual Costs.

4.7.7.3 Submodule 3: Scheduling

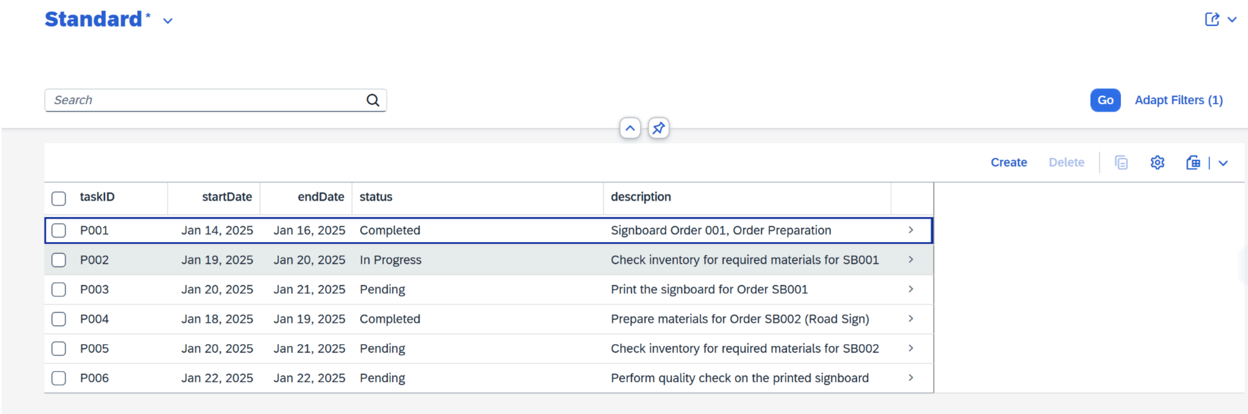


Figure 4.7.7.3.1 displays a list of created tasks

This figure presents a cohesive and professional interface for managing tasks, displayed in a table format that includes key details such as Task ID, Start and End Dates, Status, and Description. The interface features a Create button to add new tasks, a Delete button to remove selected tasks, and an Export to File option for exporting the task list as a CSV file. Additionally, users can filter the view based on specific criteria, including task assignee and department, enabling efficient organization and streamlined task management.

4.7.7.4 Submodule 4: Resource Allocation

Figure 4.7.7.4.1 Homepage

The homepage of the system as shown in Figure 4.7.7.4.1 displays the logo of AK Maju Resources Sdn Bhd and serves as the entry point for navigation. From this page, users can navigate to the Project Management page, where they can allocate resources and manage project details. This central hub may also provide access to various modules, ensuring efficient navigation throughout the system.

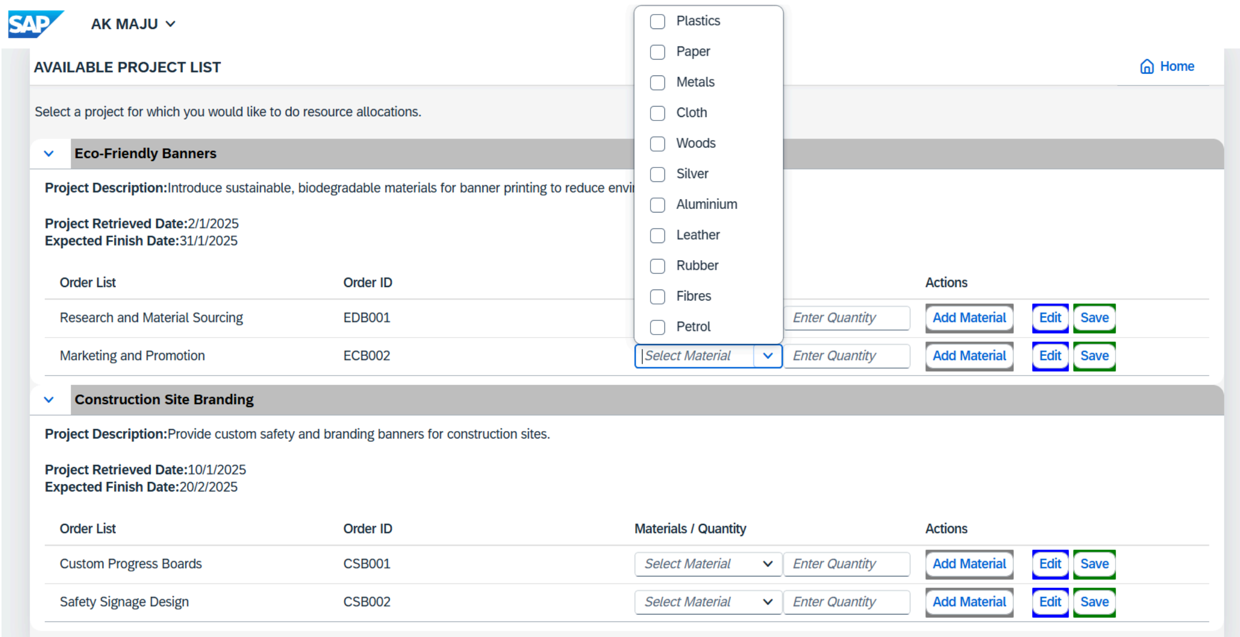


Figure 4.7.7.4.2 Project Management

Figure 4.7.7.4.2 shows a Project Management page where users can view comprehensive details about ongoing projects, including project names, timelines, and associated orders. The page provides essential information such as unique order IDs, available materials list, and the current status of each order. These details allow users to manage resource allocation effectively by reviewing and monitoring all project and order-related information in one place.

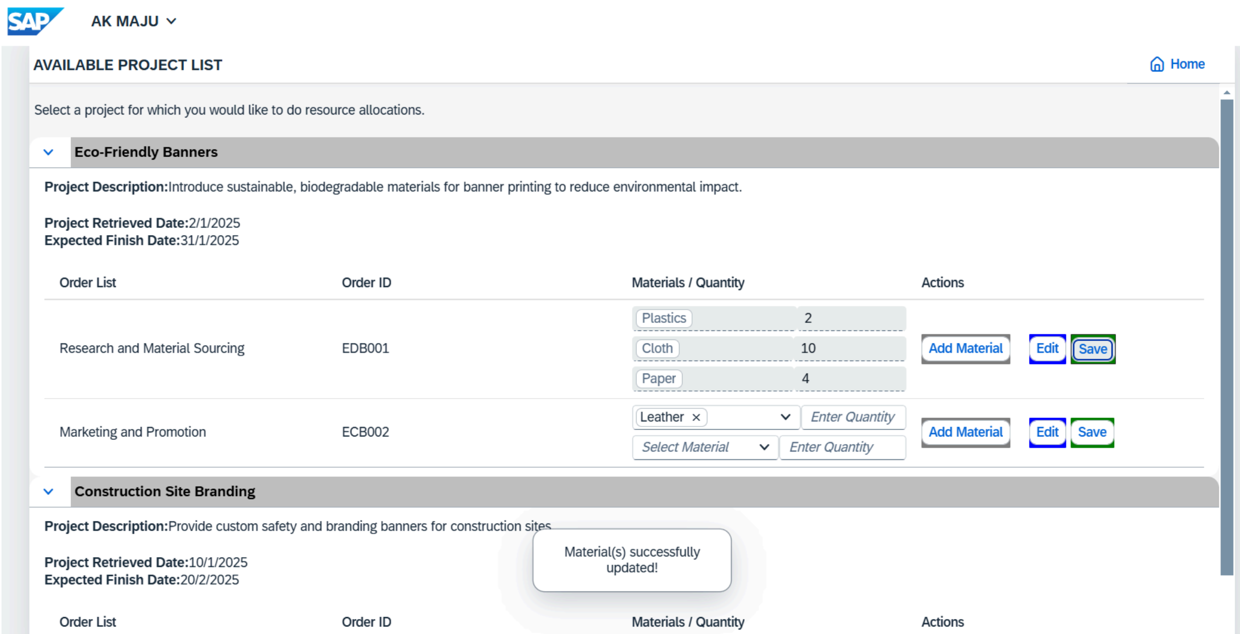


Figure 4.7.7.4.3 Materials Allocation

Figure 4.7.7.4.3 outlines several actions for managing materials associated with each order. When the user clicks the "Save" button, a confirmation message, "Material(s) successfully updated," appears, indicating that all changes have been securely saved. Clicking the "Edit" button enables editing of materials, and a message, "Material(s) are now editable," is displayed. Once the "Save" button is clicked after editing, the materials become locked for further modification unless the "Edit" button is clicked again. Additionally, the "Add Material" button allows users to add a new row for material selection and quantity input, facilitating the assignment of additional materials to specific orders. Users can specify the material type and its quantity as needed. These functionalities enhance flexibility and ensure efficient resource allocation and order management within the system.

4.7.7.5 Submodule 5 : Progress Monitoring

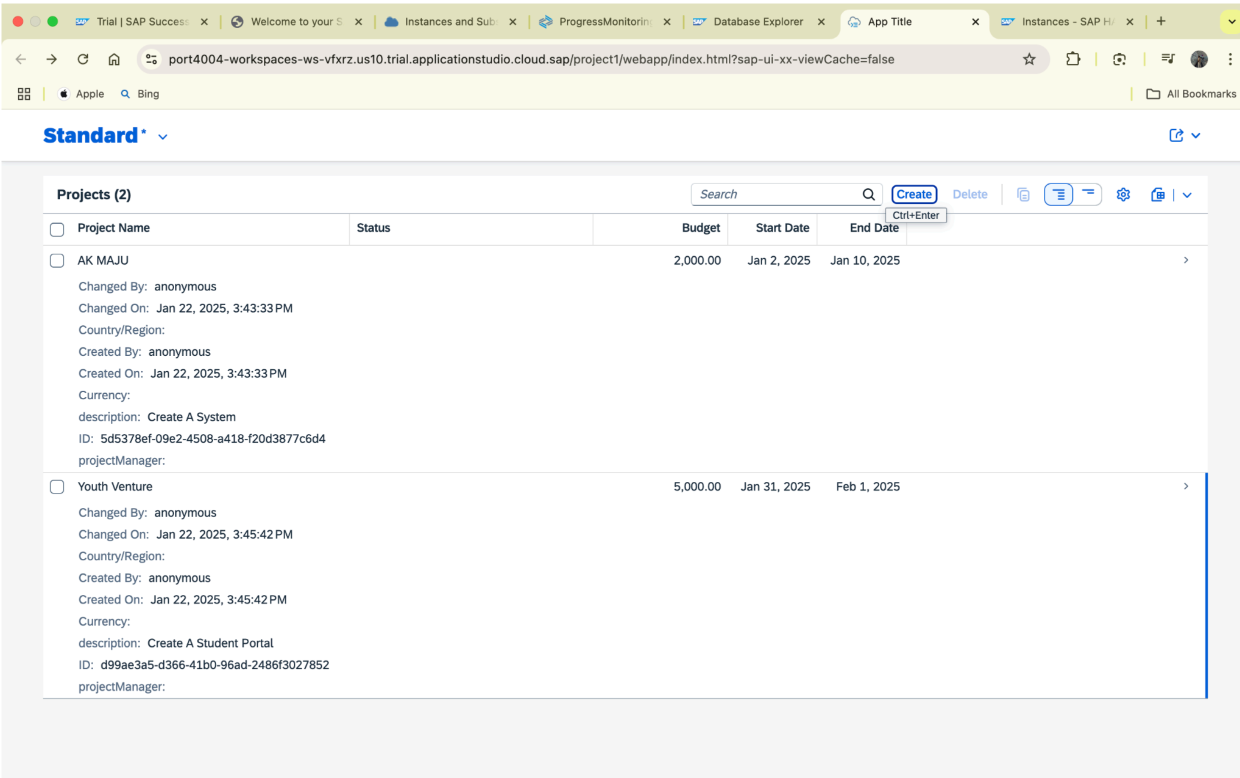


Figure 4.7.7.5.1: Overview Page

Figure 4.7.7.5.1 shows where users can view and manage projects. Each project is listed with details like its name, budget, timeline, and description. Users can also see when a project was created or last modified, along with a unique ID for each project. The toolbar at the top allows users to search for projects, create new ones, or delete existing entries. This interface serves as a central place to track and manage organizational projects efficiently.

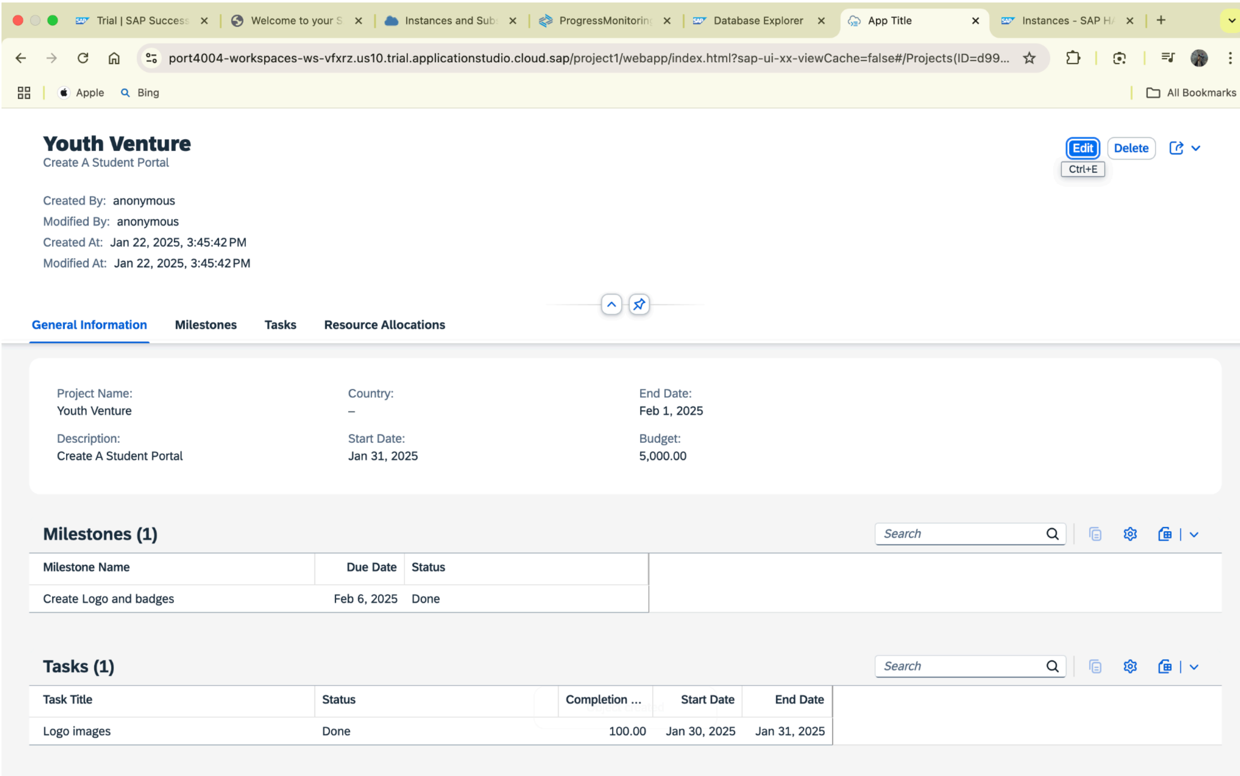


Figure 4.7.7.5.2: Progress Monitoring Page in detail

Figure 4.7.7.5.2 shows a project named **"Youth Venture"**, aimed at creating a student portal. It was created and last modified by "anonymous" on January 22, 2025. The project has a budget of $5,000 and runs from January 31 to February 1, 2025. A milestone, "Create Logo and badges," was completed by February 6, 2025. A task, "Logo images," was fully completed between January 30 and January 31, 2025. The interface includes options to edit or delete the project and provides tabs for managing milestones, tasks, and resources. It helps track the project's progress effectively.

4.8 Summary

This topic focuses on the analysis and design of a new enterprise system for AK Maju to replace its existing system, which is built on a basic architecture using PHP, HTML, CSS, JavaScript, Bootstrap, and a relational database. The current system has significant limitations, including data silos, a lack of real-time tracking, resource allocation, progress monitoring, and visual tools such as Gantt charts and KPI dashboards.

The proposed enterprise system leverages SAP S/4HANA for enterprise-grade project management and SAP Business Technology Platform (BTP) for seamless integration with existing PHP interfaces. It features a unified architecture with integrated sub-modules for Project Structure Management, Budgeting, Scheduling, Resource Allocation, and Progress Monitoring. The system includes advanced tools like Work Breakdown Structure (WBS), real-time project tracking, cost control, Gantt charts for scheduling, and KPI dashboards for performance tracking.

With centralized data management and fully integrated modules, the new system enhances project management depth, resource utilization, and operational visibility. This topic also provides an overview of the enterprise and system architecture, highlighting the improvements in scalability and efficiency for AK Maju.

Chapter 5

System Implementation

5.1 Introduction

This report section focuses on the implementation phase of the AK Maju Project Management Subsystem. It shows the tools, technologies, and processes utilized to develop and integrate the subsystem. Utilizing SAP Business Technology Platform (BTP) and SAP HANA, the system development emphasized scalability, efficiency, and adaptability, providing a strong foundation for modern project management practices.

The implementation process showcases how the database has been designed, how the core functionalities have been coded, and the methodologies followed in order to make the system effective. Detailed database structures were developed to support seamless data storage and retrieval, while coding practices ensured that core functionalities such as budget tracking, scheduling, and progress monitoring were in a user-friendly manner. Additionally, all the processes were carefully done to ensure that the system met all functional and technical requirements.

By integrating modules such as project structure management, resource allocation, and financial performance reporting, the implementation phase was key to addressing critical operational challenges and delivering a scalable, reliable project management solution tailored to AK Maju's needs.

5.2 System Development

1. Tools and Technologies

The development of this system relied on two key tools which are SAP Business Technology Platform (SAP BTP) and SAP HANA.

- **SAP BTP:** This cloud platform provided the essential environment for developing, deploying, and managing the AK Maju system. SAP BTP enabled seamless integration of services, ensured scalability, and supported efficient cloud-based operations. It also provided APIs and tools that simplified the development of custom functionalities and enhanced the system's adaptability to evolving business needs.
- **SAP HANA:** Serving as the backend database, SAP HANA's in-memory processing capabilities allowed real-time data storage and retrieval. Its columnar data organization ensured high-speed querying and analytics, making it ideal for handling resource allocation and project management tasks.

These tools, combined with modern coding practices, ensured that the system was efficient, reliable, and future-proof.

2. Development Process

The development of the system was a **collaborative** effort, where tasks were divided among team members to ensure efficient progress and effective use of individual strengths. Each team member was assigned specific responsibilities based on the system's core functionalities:

- **Project Structure Management**: One team member, *Daniel Hakim* focused on creating the project framework, including defining the project structure, managing project details, and ensuring proper organization of all related components in the system.

- **Budget and Cost Tracking**: Another member, *Yan Qing* was responsible for integrating features to monitor project budgets, track expenses, and provide cost analysis through the database.
- **Scheduling**: The task of implementing project scheduling functionalities, such as setting start and finish dates and visualizing timelines, was handled by another team member, *Yu Tong*.
- **Resource Allocation**: The development of features for assigning materials and other resources to specific projects or orders was managed by another team member, *Li Hui*.
- **Progress Monitoring**: A team member, *Vinesh* focused on designing functionalities to track the status and progress of ongoing projects, including updates on material usage and order completion.

The team collectively contributed to developing the user interface, ensuring a cohesive and user-friendly design across all modules. The system was built entirely by the team using **SAP BTP** as the development environment and **SAP HANA** as the database backend. This approach allowed seamless collaboration and ensured that the system met the functional requirements of project management, cost tracking, and resource allocation.

5.3. Create Database (individual)

5.3.1 Submodule 1: Project Structure Management

Table Name: APP_SCHEDULING_TASKS

Schema: DBADMIN

Open Data

Columns | Indexes | Properties | Runtime Information

	Name	SQL Data Type	Key	Not Null	Default	Comment
1	ID	NVARCHAR(36)	1	X	NULL	
2	TASKID	NVARCHAR(20)		X	NULL	
3	STARTDATE	DATE		X	NULL	
4	ENDDATE	DATE		X	NULL	
5	ASSIGNEDTO_ID	NVARCHAR(50)			NULL	
6	STATUS	NVARCHAR(20)			NULL	
7	DESCRIPTION	NVARCHAR(500)			NULL	

Figure 5.3.1.1: APP_SCHEDULING_TASKS Table

Figure 5.3.1.1 shown the **APP_SCHEDULING_TASKS** table, located in the **DBADMIN** schema, manages scheduled tasks with the following columns: **ID** (primary key, NVARCHAR(36)), uniquely identifies each record; **TASKID** (NVARCHAR(20)), specifies the task identifier; **STARTDATE** and **ENDDATE** (both DATE), define the task's start and end dates; **ASSIGNEDTO_ID** (NVARCHAR(50)), indicates the individual or entity assigned to the task; **STATUS** (NVARCHAR(20)), tracks the task's current status; and **DESCRIPTION** (NVARCHAR(500)), provides details or notes about the task. All fields are mandatory (not null), with a default value of NULL, ensuring efficient scheduling and management of tasks.

5.3.2 Submodule 2: Budget and Cost Tracking

View Name	Schema	Type	
PROJECTBUDGETINGSERVICE_PROJECTS	BUDGETCOSTTRACKING_HDI_BUDGETCOSTTF	ROW	Open Data

Columns CREATE Statement

	View Column	SQL Data Type	Not Null	Default	Comment
1	ID	NVARCHAR(36)	X	NULL	
2	CREATEDAT	TIMESTAMP		NULL	
3	CREATEDBY	NVARCHAR(255)		NULL	
4	MODIFIEDAT	TIMESTAMP		NULL	
5	MODIFIEDBY	NVARCHAR(255)		NULL	
6	PROJECTNAME	NVARCHAR(100)		NULL	
7	TOTALBUDGET	DECIMAL(10,2)		0	
8	TOTALACTUALCOST	DECIMAL(10,2)		0	

Figure 5.3.2.1: Project Table

Figure 5.3.2.1 shows the Project Table which includes financial information for each project. Each row of data represents a unique project, identified by the **ID** column. The **CREATEDAT** and **CREATEDBY** columns track when and by whom the project was created, while the **MODIFIEDAT** and **MODIFIEDBY** columns record the timestamp and user responsible for the latest updates. These attributes (**CREATEDAT**, **CREATEDBY**, **MODIFIEDAT**, and **MODIFIEDBY**) are system-generated fields automatically added by the SAP framework or database configuration to track metadata. They are not explicitly created by the developer. The **PROJECTNAME** column holds the name of the project, while the **TOTALBUDGET** and **TOTALACTUALCOST** columns store the allocated budget and the total actual costs incurred for the project, respectively. The details, such as **ID**, **PROJECTNAME**, are generated when a project is created. The **TOTALBUDGET** data is stored after summing the allocated budgets for all WBS elements. Similarly, the **TOTALACTUALCOST** data is stored by summing the actual costs entered for each WBS element. The **TOTALBUDGET** and **TOTALACTUALCOST** are later used in generating key metrics on the Financial Performance Report page, such as Total Budget, Total Budget Remaining, Overall Variances, Percentage Variances, and Overall Project Financial Status. This helps track and report project finances clearly.

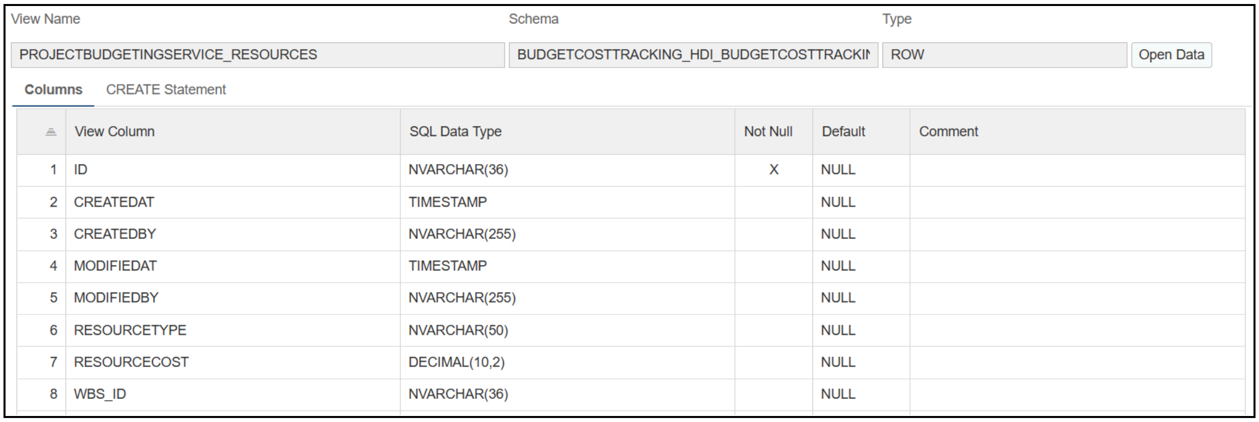
View Name	Schema	Type	
PROJECTBUDGETINGSERVICE_WBS	BUDGETCOSTTRACKING_HDI_BUDGETCOSTTR/	ROW	Open Data

Columns CREATE Statement

	View Column	SQL Data Type	Not Null	Default	Comment
1	ID	NVARCHAR(36)	X	NULL	
2	CREATEDAT	TIMESTAMP		NULL	
3	CREATEDBY	NVARCHAR(255)		NULL	
4	MODIFIEDAT	TIMESTAMP		NULL	
5	MODIFIEDBY	NVARCHAR(255)		NULL	
6	WBSNAME	NVARCHAR(100)		NULL	
7	ALLOCATEDBUDGET	DECIMAL(10,2)		NULL	
8	ACTUALCOST	DECIMAL(10,2)		0	
9	PROJECT_ID	NVARCHAR(36)		NULL	

Figure 5.3.2.2: WBS Table

Figure 5.3.2.2 shows the WBS Table which provides information of individual Work Breakdown Structure (WBS) elements for each project. Each row represents a unique WBS element, identified by the **ID** column. The **CREATEDAT**, **CREATEDBY**, **MODIFIEDAT**, and **MODIFIEDBY** columns capture creation and modification details. These attributes (**CREATEDAT**, **CREATEDBY**, **MODIFIEDAT**, and **MODIFIEDBY**) are system-generated fields automatically added by the SAP framework or database configuration to track metadata. The **WBSNAME** column specifies the name of the WBS element, while the **ALLOCATEDBUDGET** and **ACTUALCOST** columns store the budget and actual cost for that specific WBS element. The **PROJECT_ID** column links the WBS element to a corresponding project in the Project Table. The **ALLOCATEDBUDGET** data is calculated by summing the resource allocation costs (calculated by multiplying the allocated resources with the resource cost) for the WBS element. The **ACTUALCOST** data is recorded after users input the actual cost for each WBS element. The **PROJECT_ID** column acts as a foreign key, linking the WBS element to its corresponding project in the Project table. The **WBSNAME** columns is filled when the WBS element is created. Key fields like **WBSNAME**, **ALLOCATEDBUDGET**, and **ACTUALCOST** are used to visualize Budget Distribution across WBS Elements and Compare the Budget against the Actual Cost for each WBS Element in the Financial Performance page.



View Name: PROJECTBUDGETINGSERVICE_RESOURCES
Schema: BUDGETCOSTTRACKING_HDI_BUDGETCOSTTRACKIN
Type: ROW
Open Data

Columns | CREATE Statement

	View Column	SQL Data Type	Not Null	Default	Comment
1	ID	NVARCHAR(36)	X	NULL	
2	CREATEDAT	TIMESTAMP		NULL	
3	CREATEDBY	NVARCHAR(255)		NULL	
4	MODIFIEDAT	TIMESTAMP		NULL	
5	MODIFIEDBY	NVARCHAR(255)		NULL	
6	RESOURCETYPE	NVARCHAR(50)		NULL	
7	RESOURCECOST	DECIMAL(10,2)		NULL	
8	WBS_ID	NVARCHAR(36)		NULL	

Figure 5.3.2.3: Resource Table

Figure 5.3.2.3 shows the Resource Table, which provides information on resource allocations. Each row represents a unique resource entry, identified by the **ID** column. The **CREATEDAT, CREATEDBY, MODIFIEDAT,** and **MODIFIEDBY** columns capture details regarding the creation and modification of each record. These fields are system-generated by the SAP framework or database configuration. The **RESOURCETYPE** column specifies the type of resource allocated to a WBS element, such as labor, equipment and so on. The **RESOURCECOST** column records the associated cost of each resource, providing essential financial data for budgeting and cost tracking. The **WBS_ID** column acts as a foreign key, linking the resource entry to a corresponding WBS element in the Work Breakdown Structure. The **RESOURCECOST** data is calculated based on the cost estimation for each resource type, which contributes to the overall budgeting of the project. All **RESOURCECOST** data associated with the same **WBS_ID** will be added and later recorded as the **ALLOCATEDBUDGET** in the WBS table.

5.3.3 Submodule 3: Scheduling

	Name	SQL Data Type	Key	Not Null
1	ID	NVARCHAR(36)	1	X
2	TASKID	NVARCHAR(10)		
3	STARTDATE	DATE		
4	ENDDATE	DATE		
5	ASSIGNEDTO_ID	NVARCHAR(36)		
6	STATUS	NVARCHAR(50)		
7	DESCRIPTION	NVARCHAR(36)		

Figure 5.2.3.1

Figure 5.2.3.1 is the database schema that defines attributes essential for task management within a Work Breakdown Structure (WBS). The **ID** serves as the primary key, ensuring each task is uniquely identifiable within the WBS framework, while **TASKID** offers a concise, human-readable reference for quick identification. **STARTDATE** and **ENDDATE** capture the task's timeline, enabling scheduling and progress tracking against the WBS hierarchy. The **ASSIGNEDTO_ID** acts as a foreign key, linking tasks to specific employees or team members from the employee database, facilitating resource allocation and accountability. The **STATUS** field reflects the task's progress (e.g., "Planned," "In Progress," "Completed"), ensuring clear visibility into project status. **DESCRIPTION** provides additional details or context for each task, offering clarity within the broader WBS structure. This design integrates seamlessly with project management processes, supporting effective planning and execution.

Rows (5)

	ID	EMPLOYEEID	FIRSTNAME	LASTNAME	DEPARTMENT	ROLE	DATEOFJOINING
1	12345678-90AB-CDEF-1234-567890ABCDEF	ST01	Abby	Dowry	Logistic	Logistics Manager	2004-04-25
2	22345678-90AB-CDEF-1234-567890ABCDEF	ST02	John	Doe	Inventory	Inventory Specialist	2005-06-15
3	32345678-90AB-CDEF-1234-567890ABCDEF	ST03	Smith	Brown	Production	Production Operator	2010-02-10
4	42345678-90AB-CDEF-1234-567890ABCDEF	ST04	Lee	Chen	Quality Control	QC Supervisor	2007-09-20
5	52345678-90AB-CDEF-1234-567890ABCDEF	ST05	Lucy	Walker	Administration	System Administrator	2003-01-12

Figure 5.2.3.2

Figure 5.2.3.2 is the **Employee** table. It is designed to manage staff information critical for resource allocation and task assignments in the system. The **ID** serves as a unique identifier in UUID format, specifically required for integration with the SAP system, ensuring database consistency. The **EMPLOYEEID** represents the staff ID, providing a human-readable reference for quick identification. **FIRSTNAME** and **LASTNAME** store the employee's full name, supporting personalization and clear communication. The **DEPARTMENT** field categorizes employees into functional areas, facilitating organizational structure and task allocation. The **ROLE** field defines the employee's job position or responsibility within the company, aiding in assigning tasks to suitable personnel. Lastly, the **DATEOFJOINING** field captures the employee's start date, enabling tenure tracking and compliance with HR policies. This table plays a crucial role in linking employees to tasks and projects, enhancing collaboration and efficient resource management.

5.3.4 Submodule 4: Resource Allocation

	View Column	SQL Data Type	Not Null	Default
1	PROJECTID	NVARCHAR(36)	X	NULL
2	PROJECTNAME	NVARCHAR(100)		NULL
3	PROJECTDESC	NVARCHAR(255)		NULL
4	RETRIEVEDDATE	TIMESTAMP		NULL
5	FINISHDATE	TIMESTAMP		NULL

Figure 5.2.4.1 Project Table

Figure 5.2.4.1 shows a Project table which stores the fundamental information about all projects managed within the system. Each project is uniquely identified using the PROJECTID, ensuring that no two projects have conflicting identifiers. The PROJECTNAME provides a descriptive title for the project, and the optional PROJECTDESC column allows for further elaboration, such as objectives or requirements. The RETRIEVEDDATE and FINISHDATE fields track the project's timeline, helping to manage deadlines and monitor progress. This table plays a crucial role in the overall system as it integrates with other tables (such as Order) to ensure that all orders and materials are associated with the correct project.

	View Column	SQL Data Type	Not Null	Default
1	ORDERID	NVARCHAR(36)	X	NULL
2	PROJECTID	NVARCHAR(36)		NULL
3	ORDERLIST	NVARCHAR(255)		NULL

Figure 5.2.4.2 Order Table

The Order table as shown in Figure 5.2.4.1 links directly to the Project table through the PROJECTID column, establishing a relationship between a project and its associated orders. Each order is uniquely identified by ORDERID, ensuring traceability and precise management of tasks or orders within a project. The ORDERLIST column captures the details of each order, providing essential information about what the order entails. This table facilitates the tracking of multiple orders for a single project, ensuring that the system can manage complex projects with multiple deliverables or stages.

	View Column	SQL Data Type	Not Null	Default
1	MATERIALID	NVARCHAR(36)	X	NULL
2	ORDERID	NVARCHAR(36)		NULL
3	MATERIAL	NVARCHAR(255)		NULL
4	QUANTITY	NVARCHAR(36)		NULL

Figure 5.2.4.3 Material Table

Figure 5.2.4.3 shows a Material table which manages the resources required to fulfill the orders defined in the Order table. Each material is uniquely identified by the MATERIALID, and its association with a specific order is defined by the ORDERID column. The MATERIAL column stores the name of materials, while the QUANTITY column ensures that the exact amount of materials needed is recorded. This table plays a critical role in resource allocation, as it provides detailed insights into the materials required for each task or order, helping project managers track and optimize resource usage.

5.3.5 Submodule 5 : Progress Monitoring

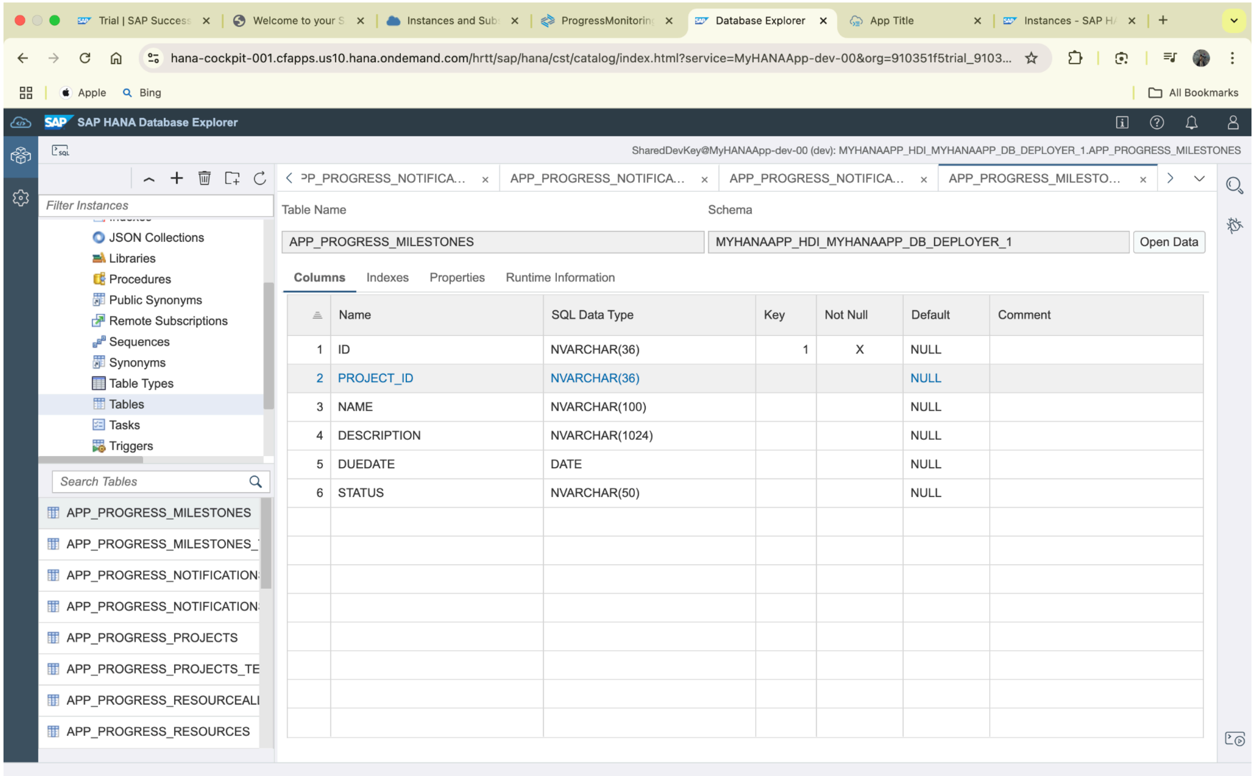


Figure 5.2.5.1 Milestones Table

Figure 5.2.5.1 shows the APP_PROGRESS_MILESTONES table, which stores key information about project milestones managed within the system. Each milestone is uniquely identified by the ID column, ensuring distinct records. The PROJECT_ID column links each milestone to a specific project, establishing a relationship with the project management system. The NAME column provides a descriptive title for the milestone, while the DESCRIPTION column allows for detailed information about the milestone's objectives or requirements. The DUEDATE field tracks the expected completion date, helping manage deadlines and monitor progress. The STATUScolumn indicates the current state of the milestone, such as planned, in-progress, or completed. This table plays a crucial role in integrating project milestones with the overall project management system through PROJECT_ID, ensuring accurate tracking and efficient progress monitoring.

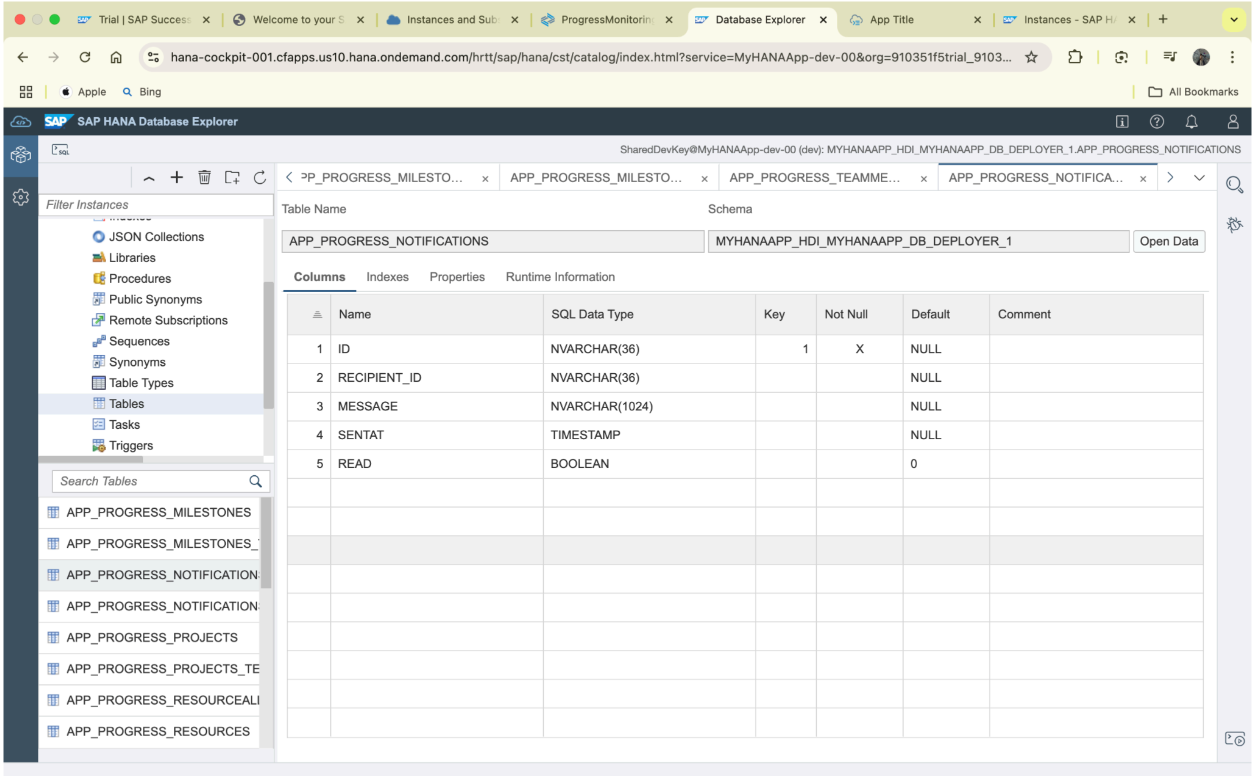


Figure 5.2.5.2 Notifications Table

Figure 5.2.5.2 shows the APP_PROGRESS_NOTIFICATIONS table, which stores essential information about notifications within the system. Each notification is uniquely identified by the ID column, ensuring distinct records. The RECIPIENT_ID column links each notification to a specific user, identifying the intended recipient. The MESSAGE column stores the content of the notification, providing relevant information or updates. The SENTAT field records the timestamp when the notification was sent, allowing for tracking communication history. The READ column, which uses a Boolean data type, indicates whether the notification has been read (true) or remains unread (false). This table plays a critical role in facilitating effective communication and status updates within the system.

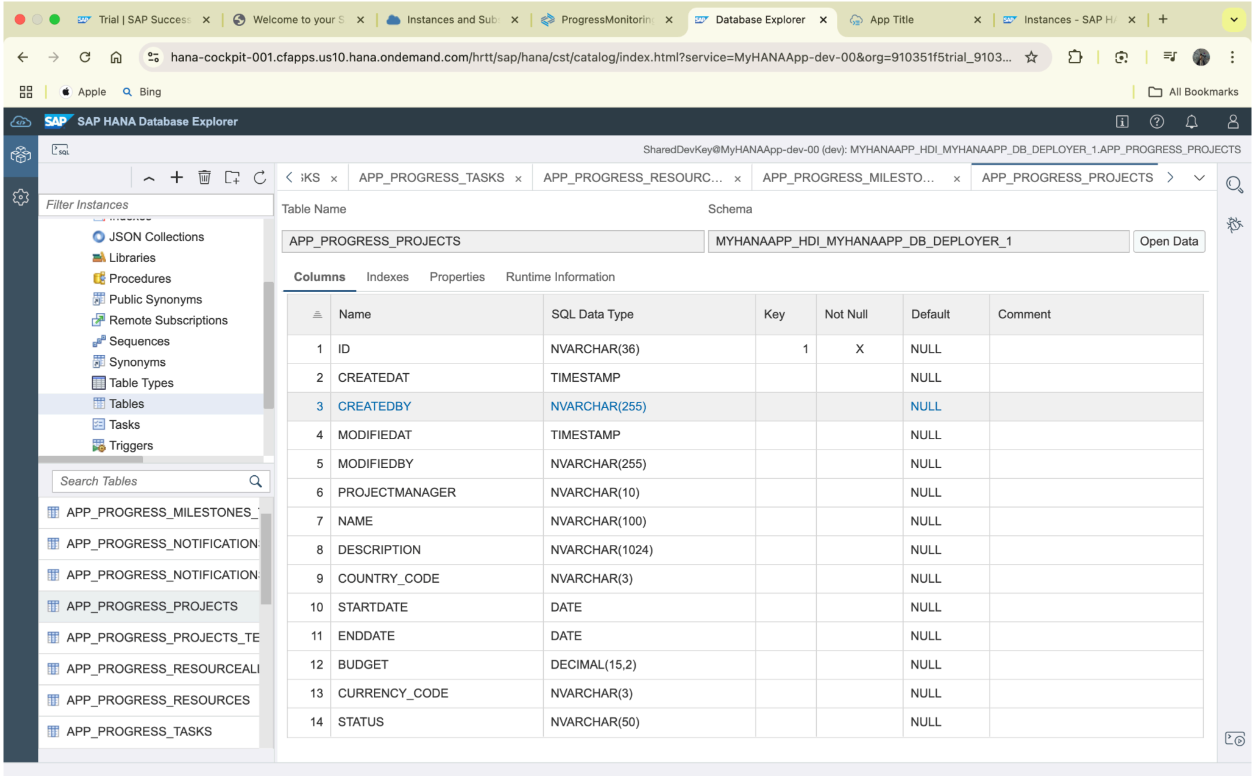


Figure 5.2.5.3 Projects Table

Figure 5.2.5.3 shows the APP_PROGRESS_PROJECTS table, which stores fundamental information about projects managed within the system. Each project is uniquely identified by the ID column. The CREATEDAT and CREATEDBY columns record the timestamp and user who created the project, respectively, while MODIFIEDAT and MODIFIEDBY capture updates made to the project over time. The PROJECTMANAGER column specifies the individual responsible for managing the project. The NAME and DESCRIPTION columns provide a descriptive title and additional information about the project. The COUNTRY_CODE column allows localization tracking. The STARTDATE and ENDDATE fields help manage the project timeline, while BUDGET and CURRENCY_CODE columns handle financial aspects. The STATUS column indicates the current state of the project, such as active or completed. This table plays a vital role in tracking and managing project lifecycle details.

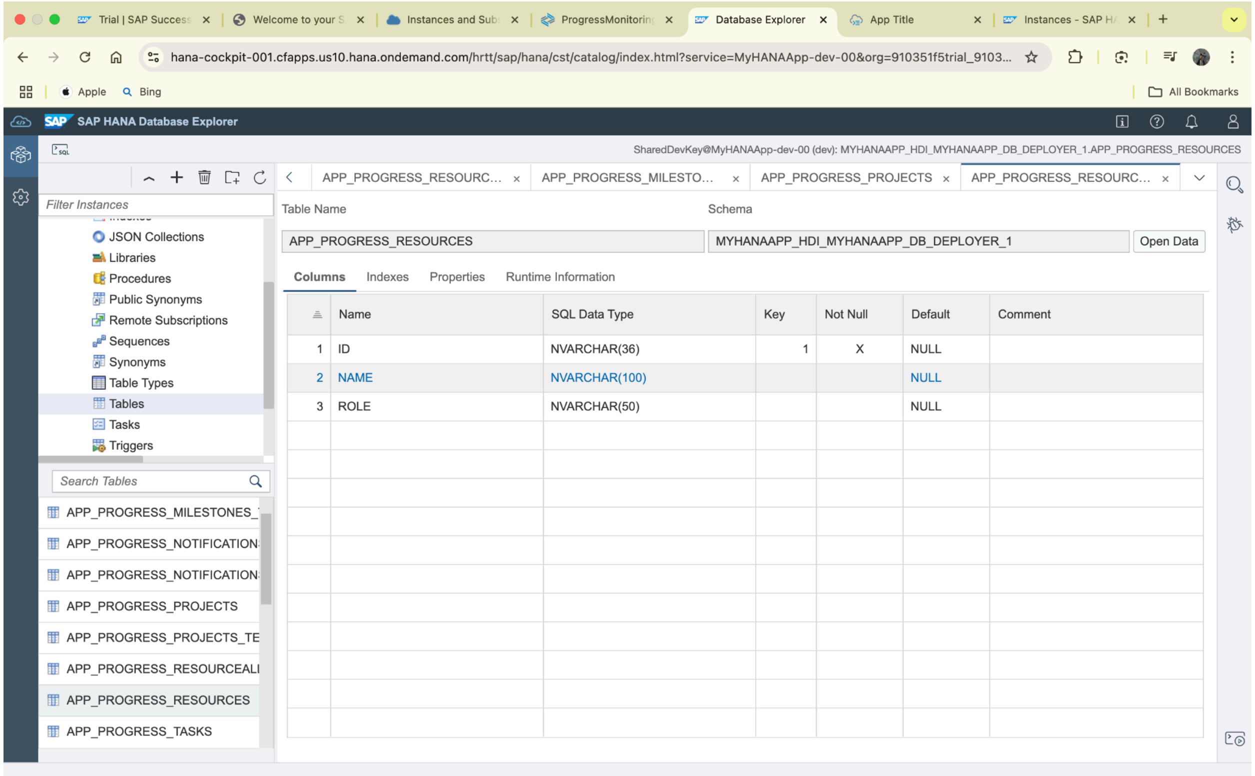


Figure 5.2.5.4 Resources Table

Figure 5.2.5.4 shows the APP_PROGRESS_RESOURCES table, which stores information about resources associated with projects within the system. Each resource is uniquely identified by the ID column, which serves as the primary key. The NAME column captures the name of the resource, while the ROLE column specifies the function or responsibility of the resource within the project.

5.4 Coding of the system's main functions/Process (individual)

5.4.1 Submodule 1: Project Structure Management

1. **Displaying Task Details**

```
using CatalogService as service from '../../srv/interaction_srv';
annotate service.SchedulingTasks with @(
    UI.FieldGroup #GeneratedGroup : {
        $Type : 'UI.FieldGroupType',
        Data : [
            {
                $Type : 'UI.DataField',
                Label : 'taskId',
                Value : taskId,
            },
            {
                $Type : 'UI.DataField',
                Label : 'startDate',
                Value : startDate,
            },
            {
                $Type : 'UI.DataField',
                Label : 'endDate',
                Value : endDate,
            },
            {
                $Type : 'UI.DataField',
                Label : 'assignedToId',
                Value : assignedToId,
            },
            {
                $Type : 'UI.DataField',
                Label : 'status',
                Value : status,
            },
            {
                $Type : 'UI.DataField',
                Label : 'description',
                Value : description,
            },
        ],
    },
```

Figure 5.4.1.1: Annotations.cds code for Project Structure Management

Figure 5.4.1.1 shown CDS annotation for the Project Structure Management entity in the CatalogService, defining a UI.FieldGroup named #GeneratedGroup. It organizes fields (taskId, startDate, endDate, assignedToId, status, and description) for UI purposes, specifying their labels and values. This annotation is used to structure data for SAP Fiori or CAP-based applications, enabling tools like SAP Fiori Elements to automatically generate user interfaces with grouped fields for better organization and display.

2. Database Table

```
namespace app.interactions;

using {
    cuid,
    managed
} from '@sap/cds/common';

entity SchedulingTasks : cuid, managed {
    taskId       : String(20);  // TASKID
    startDate    : Date;        // STARTDATE
    endDate      : Date;        // ENDDATE
    assignedToId : String(50);  // ASSIGNEDTO_ID
    status       : String(20);  // STATUS
    description  : String(500); // DESCRIPTION
}
```

Figure 5.4.1.2: Interactions.cds code for Project Structure Management

Figure 5.4.1.2 shown a CDS entity called SchedulingTasks within the app.interactions namespace, inheriting the cuid (unique identifier) and managed (standard metadata fields like createdBy and modifiedAt) aspects from @sap/cds/common. The entity contains fields such as taskId (20-character string for task identification), startDate (date when the task starts), endDate (date when the task ends), assignedToId (50-character string for the assigned person or team ID), status (20-character string for task status), and description (500-character string for task details). This entity is typically used to model scheduling data for database storage and UI display in SAP CAP applications.

5.4.2 Submodule 2: Budget and Cost Tracking

Budget and Cost Tracking Page

1. Data Fetching

```
onReadWBSData: function () {
    var oModel = this.getOwnerComponent().getModel();
    var aProjects = []; // Declare aProjects here
    var oBusyDialog = new sap.m.BusyDialog({
        title: "Loading Data",
        text: "Please wait ......"
    });
    oBusyDialog.open();

    // Fetch Project data
    oModel.bindList("/Projects").requestContexts().then(function (aContexts) {
        aProjects = aContexts.map(function (oContext) {
            return oContext.getObject();
        });

        // Fetch WBS data
        oModel.bindList("/WBS").requestContexts().then(function (aWBSContexts) {
            var aWBS = aWBSContexts.map(function (oContext) {
```

Figure 5.4.2.1: onReadWBSData Function

Figure 5.4.2.1 shows part of the code that loads and binds project and WBS (Work Breakdown Structure) data from the backend to the view.

2. Dynamic Calculations and Updates

```
updateTotals: function() {
    var oModel = this.getView().getModel("WBSDataModel");
    var aWBSData = oModel.getData().WBSData; // Get WBS data from the model

    var totalActualCost = 0.00;
    var totalVariance = 0.00;

    // Loop through each WBS data and accumulate the totals
    aWBSData.forEach(function(oData) {
        totalActualCost += parseFloat(oData.ActualCost) || 0; // Ensure it's a number
        totalVariance += (parseFloat(oData.Variance) || 0); // If Variance is empty or invalid,
    });

    // Update the totals directly in the model
    oModel.setProperty("/TotalActualCost", totalActualCost.toFixed(2));
    oModel.setProperty("/TotalVariance", totalVariance.toFixed(2));
```

Figure 5.4.2.2: updateTotals Function

Figure 5.4.2.2 shows part of the code that calculates the total actual cost and total variance for all projects, and then updates the displayed totals.

3. Input Validation

```
onLiveChange: function(oEvent) {
    var oInput = oEvent.getSource(); // Get the input field
    var sNewValue = oInput.getValue(); // Get the new value of ActualCost
    var oBindingContext = oInput.getBindingContext("WBSDataModel"); // Get the binding context
    var oData = oBindingContext.getObject(); // Get the object that contains Budget and ActualCost

    // Check if the input is empty (i.e., deleted everything)
    if (sNewValue.trim() === "") {
        // If the input is empty, reset the Variance and FlagState
        oData.Variance = ""; // Set Variance to an empty string or "nothing"
        oData.FlagState = "";  // Reset the FlagState
        oData.FlagText = ""; // Reset the FlagText

        // Reset the ActualCost to an empty value
        oData.ActualCost = ""; // Reset ActualCost field

        // Update the totals in the model
        this.updateTotals();

        // Refresh the model to reflect changes
        oBindingContext.getModel().refresh(true); // Force refresh to update the UI
        return; // Exit the function to prevent further processing
    }
```

Figure 5.4.2.3: onLiveChange Function

Figure 5.4.2.3 shows part of the code that gets the input in the actual cost field. It then checks if the input is valid, calculates the variance between the actual cost and the allocated budget, and updates the display with the new variance and flag (underbudget or overrun).

Financial Performance Report Page

1. Data Fetching and Processing

```
onReadFinancialData: function () {
    var oTempModel = this.getOwnerComponent().getModel("TempDataModel");
    var oFinancialDataModel = new JSONModel();  // Model for Total Data (Budget, Actual, etc.)
    var oWBSDataModel = new JSONModel();  // Model for WBS Data (WBS elements)

    var aWBSData = []; // Array to hold the WBS-related data
    var totalBudget = 0;
    var totalActualCost = 0;
    var totalVariance = 0;

    // Fetch all the relevant data from TempDataModel
    if (oTempModel) {
        var tempData = oTempModel.getData();        You, a few seconds ago • Uncommitted changes

        // Iterate over the WBS data
        for (var wbsId in tempData) {
            if (tempData.hasOwnProperty(wbsId)) {
                var data = tempData[wbsId]; // The data associated with each wbsId

                // Check if the data is an array (WBS data)
                if (Array.isArray(data)) {
                    for (var i = 0; i < data.length; i++) {
                        var wbs = data[i]; // Get the current WBS record
```

Figure 5.4.2.4: onReadFinancialData Function

Figure 5.4.2.4 shows part of the code that retrieves and processes the financial related data. It then calculates key financial metrics such as total budget, actual cost, variance, and percentages (like budget used and remaining budget) and stores them in the models. These models are later bound to the view to display the data in the UI.

5.4.3 Submodule 3: Scheduling

1. Displaying Task Details

```
annotate service.Tasks with @(
    UI.FieldGroup #GeneratedGroup : {
        $Type : 'UI.FieldGroupType',
        Data : [
            {
                $Type : 'UI.DataField',
                Label : 'taskID',
                Value : taskID,
            },
            {
                $Type : 'UI.DataField',
                Label : 'startDate',
                Value : startDate,
            },
            {
                $Type : 'UI.DataField',
                Label : 'description',
                Value : description,
            },
            {
                $Type : 'UI.DataField',
                Label : 'endDate',
                Value : endDate,
            },
            {
                $Type: 'UI.DataField',
                Label: 'Assigned To',
                Value: assignedTo_ID, // Correctly binds to the assignedTo_ID field
            },

            {
                $Type : 'UI.DataField',
                Label : 'status',
                Value : status,
            },

        ],
    },

UI.Facets : [
    {
        $Type : 'UI.ReferenceFacet',
        ID : 'GeneratedFacet1',
        Label : 'General Information',
        Target : '@UI.FieldGroup#GeneratedGroup',
    },
],
```

Figure 5.3.3.1

Figure 5.3.3.1 shows the code that defines how task details are presented in the user interface using the **@UI.FieldGroup** and **@UI.LineItem** annotations. This section groups task details into a "General Information" facet in the UI.

@UI.FieldGroup represents a grouping of fields. Each DataField specifies a task property (taskID, startDate,endDate, description, status) with its label and bound value.

@UI.Facets creates a tab or section in the UI that references the defined field group. Fields such as assignedTo_ID are linked to the relevant properties in the database.

2. Defining Task List (Tabular View)

```
UI.LineItem : [
    {
        $Type : 'UI.DataField',
        Label : 'taskID',
        Value : taskID,
    },
    {
        $Type : 'UI.DataField',
        Label : 'description',
        Value : description,
    },
    {
        $Type : 'UI.DataField',
        Label : 'status',
        Value : status,
    },
    {
        $Type : 'UI.DataField',
        Label : 'startDate',
        Value : startDate,
    },
    {
        $Type : 'UI.DataField',
        Label : 'endDate',
        Value : endDate,
    },
    {
        $Type: 'UI.DataField',
        Label: 'Assigned To',
        Value: assignedTo.firstName,
    },
    {
        $Type: 'UI.DataField',
        Label: 'Department',
        Value: assignedTo.department,
    },
],
```

Figure 5.3.3.2

This part specifies how the task details appear in a tabular or list format.Purpose: Defines the table columns in the list view. Fields: Includes task details such as taskID, description, status, and Assigned To. assignedTo.firstName and assignedTo.department: Uses navigation properties to fetch details (first name and department) from related entities.

3. Value Help for Assigned Employees

```
annotate service.Tasks with {
    assignedTo @Common.ValueList: {
        $Type: 'Common.ValueListType',
        CollectionPath: 'Employees',
        Parameters: [
            {
                $Type: 'Common.ValueListParameterInOut',
                LocalDataProperty: assignedTo_ID,
                ValueListProperty: 'ID',
            },
            {
                $Type: 'Common.ValueListParameterDisplayOnly',
                ValueListProperty: 'firstName',
            },
            {
                $Type: 'Common.ValueListParameterDisplayOnly',
                ValueListProperty: 'lastName',
            },
            {
                $Type: 'Common.ValueListParameterDisplayOnly',
                ValueListProperty: 'department',
            },
        ],
    };
};
```

Figure 5.3.3.3

This section configures the value help (dropdown) for assigning tasks to employees, enabling a dropdown for selecting an employee. The **CollectionPath** points to the *Employees* entity, storing employee data. **ValueListParameterInOut** maps the *assignedTo_ID* field in the task to the employee ID in the value list, while **ValueListParameterDisplayOnly** specifies additional fields, such as *firstName* and *department*, to be displayed in the dropdown for better user context. This setup improves the user experience when assigning tasks.

4. Supporting CRUD Operations and Excel Export

```
using app.scheduling from '../db/scheduling';
using {sap} from '@sap/cds-common-content';

service SchedulingService {

    @odata.draft.enabled: true
    entity Tasks as projection on scheduling.Tasks;

    entity Employees as projection on scheduling.Employees;

    @readonly
    entity Languages as projection on sap.common.Languages;

}
```

Figure 5.3.3.4

This code defines the SchedulingService, which serves as the entry point for OData services, enabling interaction with database entities through HTTP methods (e.g., POST, GET, PATCH, DELETE). It fully supports CRUD (Create, Read, Update, Delete) operations and is draft-enabled, providing flexibility in creating and updating tasks.

Purpose: The service acts as a bridge between your database and client applications, such as SAP Fiori apps or external systems. It ensures that entities like Tasks, Employees, and Languages are accessible via OData protocols, facilitating seamless integration and data manipulation.

Entities defined within this service are exposed to clients through OData, allowing them to perform operations such as querying data, creating new records, or updating existing ones. Features like Export to Excel are supported using UI annotations (e.g., @UI.DataField), which ensure that data fields are exportable. CAP (Cloud Application

Programming) also allows for custom export logic using tools such as SAP Fiori or Node.js modules like xlsx.

5.4.4 Submodule 4: Resource Allocation

1. Initializing the View (Setting up Models)

```
onInit: function () {
    var oData = {
        projects: [
            {
                project: "Eco-Friendly Banners",
                projectDescription: "Introduce sustainable, biodegradable materials for banner printing to reduce environmental impact.",
                retrievedDate: "2/1/2025",
                expectedFinishDate: "31/1/2025",
                orders: [
                    {
                        orderList: "Research and Material Sourcing",
                        orderId: "ED0001",
                        materials: [
                            { material: "", quantity: "" }
                        ],
                        editable: false
                    },
                    {
                        orderList: "Marketing and Promotion",
                        orderId: "EC0002",
                        materials: [
                            { material: "", quantity: "" }
                        ],
                        editable: false
                    }
                ]
            },
            {
                project: "Construction Site Branding",
                projectDescription: "Provide custom safety and branding banners for construction sites.",
                retrievedDate: "10/1/2025",
                expectedFinishDate: "20/2/2025",
                orders: [
                    {
                        orderList: "Custom Progress Boards",
                        orderId: "CS0001",
                        materials: [
                            { material: "", quantity: "" }
                        ],
                        editable: false
                    },
                    {
                        orderList: "Safety Signage Design",
                        orderId: "CS0002",
                        materials: [
                            { material: "", quantity: "" }
                        ],
                        editable: false
                    }
                ]
            }
        ]
    };

    var oModel = new JSONModel(oData);
    this.getView().setModel(oModel);
},
```

Explanation:

The onInit function initializes the view when the controller is loaded. It creates a static data model (oData) representing a list of projects, which will be displayed in the UI. This data is then set to the view using setModel(). The model binds to the UI elements, ensuring that the view is populated with this initial data. The model is a JSONModel, which stores the data in a simple format that is easy to bind to UI components.

2. Navigate to HomePage

```
onHomePress: function () {
    // Navigate to the Home page
    var oRouter = sap.ui.core.UIComponent.getRouterFor(this);
    oRouter.navTo("RouteHome");  // "home" is the route name defined in manifest
},
```

Explanation:

The onHomePress function handles the navigation when the user presses the Home button. It uses the sap.ui.core.UIComponent.getRouterFor(this) to get the router for the current controller and then navigates to the "RouteHome" route. This enables smooth navigation between views/screens in the application, allowing users to return to the home screen.

3. Edit Resources (Project Information)

```
// Enable editing and show message
onEdit: function (oEvent) {
    var oRow = oEvent.getSource().getParent().getParent();  // Get the row from the button press
    var oItem = oRow.getBindingContext().getObject();
    oItem.materials.forEach(function (material) {
        material.editable = true;  // Set the editable flag to true
    });

    // Update the model with the new editable state
    this.getView().getModel().refresh();

    // Show a message indicating the material(s) are now editable
    MessageToast.show("Material(s) are now editable!");
},
```

Explanation:

The onEdit function is triggered when the user presses the "Edit" button. It retrieves the context of the selected row and updates the project's materials to be editable. The function accesses the materials array of the project and sets the editable property of each material to true. After updating the data, this.getView().getModel().refresh() is called to refresh the view with the new editable state, allowing users to modify the project details.

4. Save Changes to Project

```
// Save changes and disable editing, then show message
onSave: function (oEvent) {
    var oRow = oEvent.getSource().getParent().getParent();  // Get the row from the button press
    var oItem = oRow.getBindingContext().getObject();
    oItem.materials.forEach(function (material) {
        material.editable = false;  // Set the editable flag to false
    });

    // Saving changes to HANA database
    this._saveToDatabase(oItem);

    // Update the model with the new editable state
    this.getView().getModel().refresh();

    // Show a message indicating that materials have been successfully updated
    MessageToast.show("Material(s) successfully updated!");
},
```

Explanation:

The onSave function is triggered when the user clicks the "Save" button. It first disables the editable state of each material in the project (to prevent further changes after saving). Then, it calls the _saveToDatabase function to persist the updated project data into the database. Finally, it refreshes the model, ensuring that the UI reflects the updated state of the data after saving.

5. Saving Data to SAP HANA Database

```
// Save changes to SAP HANA Cloud Database (OData service)
_saveToDatabase: function (oItem) {
    var oDataModel = new ODataModel("/path/to/your/odata/service"); // Define the correct OData service URL

    // Assuming your OData service has entity sets like "Projects", "Orders", "Materials"
    var oEntry = {
        "ProjectId": oItem.projectId, // Ensure the project has a proper ID
        "ProjectName": oItem.project,
        "ProjectDescription": oItem.projectDescription,
        "RetrievedDate": oItem.retrievedDate,
        "ExpectedFinishDate": oItem.expectedFinishDate,
        "Orders": oItem.orders // Include the orders and materials data
    };

    oDataModel.create("/Projects", oEntry, {
        success: function () {
            MessageToast.show("Changes saved to the database successfully!");
        },
        error: function (oError) {
            MessageToast.show("Error saving changes: " + oError.message);
        }
    });
},
```

Explanation:

The _saveToDatabase function handles the interaction with the SAP HANA Cloud database. It uses the ODataModel to connect to an OData service and create a new entry in the Projects entity. The project data (oEntry) is constructed and sent to the backend. If the save operation is successful, a success message is displayed using MessageToast.show(). If there is an error, the error message is displayed instead. This function ensures that changes made to the project data are persisted to the database.

6. Add Material to Project

```
onAddMaterial: function (oEvent) {
    // Get the button's context to identify the specific order
    var oButton = oEvent.getSource();
    var oContext = oButton.getBindingContext(); // Get the binding context for the button
    var sPath = oContext.getPath(); // Path to the specific order in the model

    // Retrieve the model
    var oModel = oContext.getModel();

    // Path to the materials array within the specific order
    var sMaterialsPath = sPath + "/materials";

    // Get the current materials array
    var aMaterials = oModel.getProperty(sMaterialsPath);

    // Add a new material entry to the materials array
    aMaterials.push({
        material: "", // Default blank material
        quantity: ""   // Default quantity nothing
    });

    // Update the model with the new materials array
    oModel.setProperty(sMaterialsPath, aMaterials);
}
```

Explanation:

The onAddMaterial function allows the user to add a new material to an existing project. When the "Add Material" button is pressed, it retrieves the current context and path for the materials array of the selected project. The function then adds a new empty material object to the array (aMaterials.push()). After updating the array, the model is refreshed using setProperty() to ensure that the UI is updated with the new material.

5.4.5 Submodule 5 : Progress Monitoring

1. Annotation of the Projects Entity for SAP Fiori UI

```
using CatalogService as service from '../../srv/progress_srv';

annotate service.Projects with @(
    UI.HeaderInfo: {
        Title: {
            $Type: 'UI.DataField',
            Value: name
        },
        TypeName: 'Project',
        TypeNamePlural: 'Projects',
        Description: { Value: description }
    },
    UI.HeaderFacets: [{
        $Type: 'UI.ReferenceFacet',
        Target: '@UI.FieldGroup#Admin'
    }],
    UI.FieldGroup #GeneratedGroup: {
        $Type: 'UI.FieldGroupType',
        Data: [
            {
                $Type: 'UI.DataField',
                Label: 'Project Name',
                Value: name
            },
            {
                $Type: 'UI.DataField',
                Label: 'Description',
                Value: description
            },
            {
                $Type: 'UI.DataField',
                Label: 'Country',
                Value: country.descr
            },
            {
                $Type: 'UI.DataField',
                Label: 'Start Date',
                Value: startDate
            },
            {
                $Type: 'UI.DataField',
                Label: 'End Date',
                Value: endDate
            },
            {
                $Type: 'UI.DataField',
                Label: 'Budget',
                Value: budget
            }
        ]
    },
    UI.FieldGroup #Admin: {
```

Explanation:
The image shows CDS annotations for the Projects entity in an SAP CAP model, defining how project data is displayed in SAP Fiori UI. The UI.HeaderInfo annotation sets the project name as the title and includes a description field for additional details. The UI.HeaderFacetsannotation references an Admin section to organize metadata such as created by and modified by. The UI.FieldGroup#GeneratedGroupannotation structures key project attributes, including Project Name, Description, Country, Start Date, End Date, and Budget, ensuring a clear and organized UI display. These annotations improve project management visibility by structuring data efficiently, making it easier for users to navigate and manage projects within the system.

2. UI Facets and Line Item Annotations for Projects

```
        {
            $Type: 'UI.ReferenceFacet',
            ID: 'GeneratedFacet1',
            Label: 'General Information',
            Target: '@UI.FieldGroup#GeneratedGroup'
        },
        {
            $Type: 'UI.ReferenceFacet',
            Label: 'Milestones',
            Target: 'milestones/@UI.LineItem'
        },
        {
            $Type: 'UI.ReferenceFacet',
            Label: 'Tasks',
            Target: 'tasks/@UI.LineItem'
        },
        {
            $Type: 'UI.ReferenceFacet',
            Label: 'Resource Allocations',
            Target: 'resources/@UI.LineItem'
        }
    ],
    UI.LineItem: [
        {
            $Type: 'UI.DataField',
            Label: 'Project Name',
            Value: name
        },
        {
            $Type: 'UI.DataField',
            Label: 'Status',
            Value: status
        },
        {
            $Type: 'UI.DataField',
            Label: 'Budget',
            Value: budget
        },
        {
            $Type: 'UI.DataField',
            Label: 'Start Date',
            Value: startDate
        },
        {
            $Type: 'UI.DataField',
            Label: 'End Date',
            Value: endDate
        }
    ]
```

Explanation:

The image shows CDS annotations for structuring the Projects entity in an SAP Fiori UI, defining facets and list views for better navigation and data presentation. The UI.Facets annotation organizes project-related data into sections such as General Information, Milestones, Tasks, and Resource Allocations, allowing users to view different aspects of a project efficiently.

The UI.LineItem annotation defines key attributes—Project Name, Status, Budget, Start Date, and End Date—displayed in a tabular list format, improving accessibility and usability for project management.

3. Admin Field Group Annotations for Projects

```
UI.FieldGroup #Admin: {
    Data: [
        {
            $Type: 'UI.DataField',
            Label: 'Created By',
            Value: createdBy
        },
        {
            $Type: 'UI.DataField',
            Label: 'Modified By',
            Value: modifiedBy
        },
        {
            $Type: 'UI.DataField',
            Label: 'Created At',
            Value: createdAt
        },
        {
            $Type: 'UI.DataField',
            Label: 'Modified At',
            Value: modifiedAt
        }
    ]
},
```

Explanation:

The image displays CDS annotations defining the Admin field group for the Projects entity in SAP Fiori UI, focusing on administrative metadata. The UI.FieldGroup#Admin annotation structures key audit fields, including Created By, Modified By, Created At, and Modified At, ensuring that project records maintain clear ownership and modification history. This setup enhances data transparency and accountability, allowing users to track changes and identify responsible stakeholders efficiently.

4. Catalog Service Definition for Project Management

```
using app.progress from '../db/progress';
using { sap } from '@sap/cds/common';

service CatalogService {
    @odata.draft.enabled: true
    entity Projects as projection on progress.Projects;

    entity Tasks as projection on progress.Tasks;
    entity Milestones as projection on progress.Milestones;
    entity ResourceAllocations as projection on progress.ResourceAllocations;

    entity Resources as projection on progress.Resources;
    entity TeamMembers as projection on progress.TeamMembers;

    @readonly
    entity Notifications as projection on progress.Notifications;

    @readonly
    entity Languages as projection on sap.common.Languages;
}
```

Explanation:

The code defines the CatalogService in an SAP CAP (Cloud Application Programming) model, enabling OData-based interactionswith project-related entities. The Projects, Tasks, Milestones, ResourceAllocations, Resources, and TeamMembers entities are projectionson their respective database tables, ensuring structured data exposure. The @odata.draft.enabled: true annotation allows draft handling, enabling users to save work-in-progress changes before final submission. Additionally, the Notifications and Languages entities are marked as @readonly, restricting modifications and ensuring they serve as reference data. The use of SAP common models (e.g., sap.common.Languages) ensures standardization and integration within SAP's ecosystem.

5.5 Summary

The implementation of the AK Maju Project Management Subsystem demonstrates the effective use of advanced technologies along with structured development practices. By utilizing SAP BTP for seamless integration and scalability, together with SAP HANA, which allows real-time data processing, the subsystem achieves its goal of delivering a reliable and efficient project management solution.

The section provided detailed insights into database designs, coding approaches, and the methodologies applied in bringing the system into reality. Each component, from project structure management to progress monitoring, was implemented with precision to ensure functionality and usability. Despite minor challenges encountered during coding and testing, the implementation phase successfully laid the groundwork for deploying a powerful project management solution.

Future enhancements could involve refining database optimization, expanding system functionalities, and adding advanced analytics to enhance decision-making capabilities. The implementation phase serves as a strong foundation for future evolution of the subsystem, which will assist AK Maju in achieving operational goals.

Chapter 6

Conclusion

6.1 Introduction

The development of the AK Maju Project Management Subsystem is a strategic initiative aimed at enhancing project efficiency and operational effectiveness. Built on SAP S/4HANA, SAP Business Technology Platform (BTP), and SAP Fiori, this system integrates essential project management functionalities to streamline processes and improve resource utilization. The implementation of this enterprise system addresses key challenges such as inefficient resource allocation, limited real-time synchronization, and restricted integration with third-party tools. By leveraging SAP's advanced capabilities, the system ensures seamless data flow, optimized project tracking, and improved decision-making. This project not only modernizes project management operations but also lays the foundation for future enhancements, ensuring scalability and adaptability to evolving business requirements.

6.2 System Contribution/Achievement

The AK Maju Project Management Subsystem has significantly improved the efficiency and effectiveness of project execution by integrating five core submodules: Project Structure Management, Budget and Cost Tracking, Scheduling, Resource Allocation, and Progress Monitoring. These modules provide a unified and scalable approach to managing projects while ensuring real-time synchronization and accurate data flow. The system's implementation has enhanced decision-making by offering better visibility into project status, financial tracking, and resource utilization. Additionally, the integration of SAP BTP has enabled improved connectivity with external applications, ensuring seamless data exchange and operational flexibility. Future advancements, including AI-driven automation, predictive analytics, and mobile accessibility via SAP Fiori, will further enhance system capabilities, promoting sustained efficiency, agility, and growth. This achievement marks a significant step toward a fully optimized and future-ready enterprise management system.

6.3 System Constraint

The AK Maju Project Management Subsystem faces several constraints that may impact its efficiency. The lack of real-time synchronization can cause delays in reflecting updated project statuses, potentially leading to miscommunication and operational setbacks. Additionally, the system's handling of concurrent resource allocation is inefficient, which may result in over-allocation or under-utilization of materials and personnel. Furthermore, restricted integration capabilities with third-party tools limit opportunities for advanced analytics and external resource management, reducing the overall flexibility and scalability of the system. The absence of intelligent automation features further limits operational efficiency in handling routine tasks such as schedule adjustments and resource allocation. Moreover, the current system lacks mobile accessibility, restricting stakeholders from monitoring and managing projects remotely. Limited user support and feedback channels may hinder effective system usage, while its architecture requires optimization to ensure seamless scalability and adaptation to evolving business needs.

6.4 Future Suggestion

To ensure the continuous improvement and scalability of the enterprise system, several recommendations are proposed. First and foremost, integrating advanced analytics tools can help leverage historical data for predictive insights, enabling proactive decision-making and better risk management. Incorporating AI-powered automation can further streamline operations by handling repetitive tasks such as scheduling adjustments, resource optimization, and anomaly detection in budgeting. Additionally, expanding the system's accessibility through mobile integration using SAP Fiori would allow stakeholders to manage and monitor projects on the go, enhancing flexibility and productivity. Regular user training sessions and the establishment of feedback mechanisms are essential to ensure users maximize the system's potential and to identify areas for refinement. Lastly, planning for future scalability and adopting continuous updates to the system with the latest SAP advancements will help maintain a competitive edge and accommodate AK Maju's growth. These enhancements will further strengthen the system's capabilities and ensure its long-term success.

6.5 Summary

The development of AK Maju's enterprise system using SAP S/4HANA, SAP BTP, and SAP Fiori introduces a transformative solution for managing projects efficiently. The integration of the five core submodules—Project Structure Management, Budget and Cost Tracking, Scheduling, Resource Allocation, and Progress Monitoring—ensures a unified and scalable system that addresses previous inefficiencies. Future enhancements, such as advanced analytics, AI-driven automation, and mobile integration, will further strengthen the system's capabilities, paving the way for sustained operational excellence and organizational growth.