CHAPTER 1

COMPANY PROFILE

CHAPTER 2

ABOUT THE DEPARTMENT

CHAPTER 3

TASKS PERFORMED

3.1 FRONTEND DEVELOPMENT

- **HTML:** Hyper Text Markup Language forms the backbone of any web application, including those built with Spring Boot. It serves as the structural framework, defining the content and layout of a webpage. HTML consists of various elements such as headings, paragraphs, lists, images, tables, and forms, which help organize information in a meaningful way. It uses tags to mark different components of a webpage, ensuring they are properly displayed in browsers. In a Spring Boot application, HTML can be used with templating engines like Thymeleaf, which allows dynamic content rendering by embedding Java logic within the markup. This enables developers to create reusable and interactive web pages while integrating data from backend services efficiently.

- **CSS:** Cascading Style Sheets is responsible for styling and enhancing the visual appeal of a web application. It controls aspects such as colors, fonts, spacing, positioning, and responsiveness, ensuring an intuitive user experience across different devices. CSS can be written inline, within a tag in an HTML document, or externally in a separate stylesheet. Spring Boot applications often use CSS to style static HTML files or dynamically rendered pages. Additionally, modern CSS frameworks like Bootstrap or Tailwind CSS can be integrated to provide pre-designed components and utility classes, reducing development time while maintaining a professional and consistent design.

- **JavaScript:** Is a powerful scripting language that adds interactivity and dynamic behaviour to web applications. It enables functionalities such as user input validation, animations, event handling, and asynchronous communication with backend APIs. In a Spring Boot application, JavaScript can be used to interact with RESTful services, fetching data from the server and updating the webpage without requiring a full reload. This is commonly achieved using AJAX or modern frontend frameworks like React, Angular, or Vue.js, which enhance the application’s interactivity.

3.2 BACKEND DEVELOPMENT

A backend Spring Boot application using Maven for web development provides a robust and scalable framework for building enterprise-grade applications. Spring Boot simplifies the development process by offering built-in configurations, auto-configuration, and dependency management, making it easier to create production-ready applications with minimal setup. Maven acts as the project management tool, handling dependencies, build lifecycle, and packaging. In a typical Spring Boot web application, Maven manages libraries such as Spring Web for handling HTTP requests, Spring Data for database interactions, and Thymeleaf or RESTful services for communication between frontend and backend. The Spring Boot framework includes an embedded Tomcat server, allowing developers to run the application without requiring additional deployment configurations. The application follows a layered architecture, where controllers process requests, services contain business logic, and repositories manage data persistence with JPA or Hibernate. With Maven’s dependency management, developers can easily integrate authentication, logging, and third-party APIs, streamlining the development process. Additionally, Spring Boot provides extensive support for RESTful APIs, enabling seamless interaction between the frontend and backend through JSON-based communication. By combining Spring Boot with Maven, developers can build secure, scalable, and efficient web applications that are well-suited for modern software development.

3.3 DATABASE MANAGEMENT

In a Spring Boot application, MySQL database management using XAMPP provides a flexible and efficient approach for handling data. XAMPP is a local server environment that includes MySQL, Apache, and PHPMyAdmin, making database administration and development more convenient. MySQL serves as the database system, storing structured data and enabling efficient retrieval and manipulation using SQL queries. Spring Boot integrates with MySQL seamlessly through JPA (Java Persistence API) and Hibernate, allowing developers to interact with the database using object-relational mapping (ORM). By configuring the application. properties or application.yml file, developers specify the database connection details, including the MySQL driver, username, password, and schema. The application can then define entity classes representing database tables and use Spring Data JPA repositories to perform CRUD (Create, Read, Update, Delete) operations effortlessly. With XAMPP, MySQL can be managed using PHPMyAdmin, providing a user-friendly graphical interface for handling tables, queries, and database users. Additionally, Spring Boot applications often use RESTful APIs to retrieve and store data dynamically, ensuring seamless interaction between the backend and frontend. By leveraging MySQL with XAMPP in Spring Boot, developers gain a robust and efficient way to manage data for web applications, enhancing performance, scalability, and usability.

CHAPTER 4

REFLECTION NOTES

4.1 EXPERIENCE

Working on the Site Logistics Management System project was a valuable and enriching experience that significantly improved my technical, problem-solving, and collaboration skills. The objective of this project was to develop a full-stack application that streamlines the logistics operations on construction or industrial sites including managing material delivery, tracking inventory, assigning transport vehicles, and monitoring worker activities.Developing this system using Spring Boot allowed me to gain hands-on experience with industry-standard backend technologies. worked with Spring MVC, Spring Data JPA, and RESTful APIs, and learned how to structure a layered architecture (Controller, Service, Repository). Integrating a MySQL database enhanced knowledge of relational database design and query optimization.also used Spring Security for authentication and role-based access control, which was a great learning curve in understanding how to protect enterprise applications. On the frontend, used Thymeleaf for templates and Bootstrap for responsive UI design.real-time logistics updates and ensuring data consistency between multiple modules. overcame this by designing a clean API structure and implementing clear entity relationships in the database. used Git and GitHub for version control and followed Agile practices with daily standups and weekly sprints. These practices helped us stay organized and accountable throughout the development cycle.

4.2 TECHNICAL OUTCOMES

4.2.1 SYSTEM REQUIREMENT SPECIFICATION

4.2.2 HARDWARE REQUIREMENTS

- Processor: Intel i5 or above (Quad-core recommended)
- Hard Disk: 100 GB (minimum)
- Memory (RAM): 8 GB (minimum), 16 GB recommended

4.2.3 SOFTWARE REQUIREMENTS

- Web Server: Embedded Apache Tomcat (included with Spring Boot)
- Operating System: Windows
- Platform: Java 17+, Spring Boot Framework
- Build Tool: Maven
- Frontend Technologies: HTML5, CSS3, JavaScript
- IDE (Editor): VS Code
- Database: MySQL / PostgreSQL / H2 (for development)
- DK: Java Development Kit 17 or later

4.3 SYSTEM ANALYSIS AND DESIGN

4.3.1 EXISTING SYSTEM

The existing system for managing site logistics in construction or industrial projects is predominantly manual and inefficient, relying on traditional methods such as paper records, spreadsheets, phone calls, and informal communication tools. These outdated practices result in poor coordination between site staff, warehouse teams, and suppliers, leading to frequent miscommunication, delayed deliveries, and inaccurate tracking of materials, equipment, and labour. Due to the absence of real-time data and centralized control, resource allocation becomes guesswork, often causing overstocking or material shortages. Furthermore, the manual nature of record-keeping increases the risk of human error, data loss, and lack of accountability. Reporting is also a major challenge, as generating performance insights or audit trails from scattered and inconsistent data is time-consuming and unreliable. Overall, the existing system fails to support the dynamic and fast-paced requirements of modern project sites, resulting in inefficiencies, cost overruns, and project delays. The current site logistics management system in many construction and logistics companies is largely manual and fragmented. Operations such as tracking materials, assigning tasks, managing customer information, and handling payments are often performed using paper records, spreadsheets, and informal communication channels like phone calls or messaging apps. These methods are time-consuming, error-prone, and lack integration, leading to inefficiencies and delays in decision-making. In many cases, there is no centralized platform to monitor task progress, material usage, or employee assignments, making it difficult to coordinate activities effectively. Additionally, data security and access control are minimal or non-existent, increasing the risk of data loss and unauthorized access. The absence of automated reporting and real-time updates further hampers the ability of management to respond to on-site issues promptly. Overall, the existing system lacks the reliability, speed, and organization required to manage modern logistics operations efficiently.

4.3.2 DISADVANTAGES OF THE EXISTING SYSTEM

The existing site logistics management system, which heavily relies on manual processes and disconnected tools, presents several critical disadvantages that hinder the efficient execution of construction and industrial projects:

1. **Lack of Real-Time Data Access:**
   One of the most significant drawbacks is the absence of real-time visibility into the movement of materials, equipment, and manpower. This delay in information flow can lead to misinformed decisions and project delays.

2. **Inefficient Resource Allocation:**
   Without a centralized system to monitor inventory levels, deliveries, and equipment availability, site managers often face issues like overstocking, underutilization, or material shortages, all of which contribute to increased operational costs and wasted resources.

3. **High Risk of Human Errors:**
   Manual data entry in spreadsheets or paper logs is prone to errors, such as duplicate entries, incorrect quantities, or missing data. These errors can escalate into costly mistakes on-site.

4. **Poor Communication and Coordination:**
   Communication between site staff, suppliers, and logistics personnel is typically unstructured and informal, relying on phone calls or text messages. This lack of integration leads to confusion, miscommunication, and inefficient collaboration.

5. **Time-Consuming Reporting:**
   Generating reports and tracking performance metrics requires considerable manual effort.Data from different departments must be collected, verified, and compiled, resulting in slow and often inaccurate reporting.

6. **Limited Scalability:**

   As project size or complexity increases, the traditional system struggles to scale. It cannot efficiently handle large volumes of logistics data or coordinate multiple teams and locations simultaneously.

7. **Low Transparency and Accountability:**

   The manual nature of record-keeping makes it difficult to track the history of material movement, identify responsible personnel, or audit past activities. This can lead to disputes and a lack of accountability during project evaluations.

8. **Difficulty in Forecasting and Planning:**

   Due to fragmented data and the absence of predictive tools, planning future material needs or scheduling deliveries becomes guesswork. This hampers project planning and risk management.

4.3.3 PROPOSED SYSTEM

The proposed Site Logistics Management System is a comprehensive, centralized, and digital solution designed to overcome the limitations of the existing manual system. It will streamline and automate the planning, tracking, and coordination of all logistics-related activities on a construction or industrial site. This system will be developed using modern full-stack technologies, offering a user-friendly interface and real-time access to critical information for site managers, warehouse personnel, procurement teams, and other stakeholders.

The proposed system will include modules for inventory management, material tracking, delivery scheduling, equipment allocation, and labour monitoring. Through integration with a central database, users can view live updates on stock levels, equipment usage, and worker deployment. The system will allow authorized users to update records, generate automated reports, and receive alerts or notifications for key logistics events such as low stock levels, delayed deliveries, or equipment breakdowns.Additionally, the system will support role-based access control, ensuring secure data management and accountability. Built-in analytics and reporting tools will provide insights into resource utilization, project efficiency, and logistics performance, assisting in better forecasting and decision-making. The web-based or cloud-enabled platform will also enable remote access and collaboration, facilitating better coordination across different departments and project sites.By implementing this digital logistics management system, the organization can significantly reduce errors, improve productivity, enhance transparency, and maintain tighter control over project timelines and costs.

4.3.4 ADVANTAGES OF THE PROPOSED SYSTEM

1. **Real-Time Monitoring and Updates:**
   The system enables real-time tracking of materials, equipment, and labor on-site, allowing site managers to make timely and informed decisions.

1. **Centralized Data Management:**
   All logistics-related data is stored in a centralized database, reducing data duplication and inconsistencies while improving accessibility and accuracy.

2. **Improved Communication and Coordination:**
   By providing a shared platform for all stakeholders, the system ensures smooth communication between site personnel, suppliers, and logistics teams, minimizing misunderstandings and delays.

3. **Efficient Resource Allocation:**
   The system provides insights into material consumption, equipment usage, and workforce distribution, allowing for optimal allocation of resources and minimizing waste.

4. **Automated Reporting and Analytics:**
   Built-in reporting tools generate automatic, accurate reports on inventory levels, delivery schedules, and work progress, reducing manual effort and supporting data-driven decision-making.

5. **Enhanced Transparency and Accountability:**
   With role-based access and activity logs, the system ensures that every transaction or update is recorded and traceable, promoting accountability among users.

6. **Scalability and Flexibility:**
   The system is designed to scale with the size of the project and can be adapted for use in multiple sites or projects simultaneously, providing long-term value.

7. **Improved Forecasting and Planning:**

   Historical data and analytics help in predicting future material requirements, labor needs, and equipment availability, which enhances planning and reduces the risk of shortages or delays.

8. **Reduced Human Errors:**

   Automation of key logistics functions, such as inventory updates and delivery notifications, reduces the chances of manual mistakes and enhances operational accuracy.

9. **Remote Access and Mobility:**

   Being a web-based or cloud-enabled solution, the system allows users to access and manage site logistics from any location using internet-connected devices.

4.4 SYSTEM ARCHITECTURE

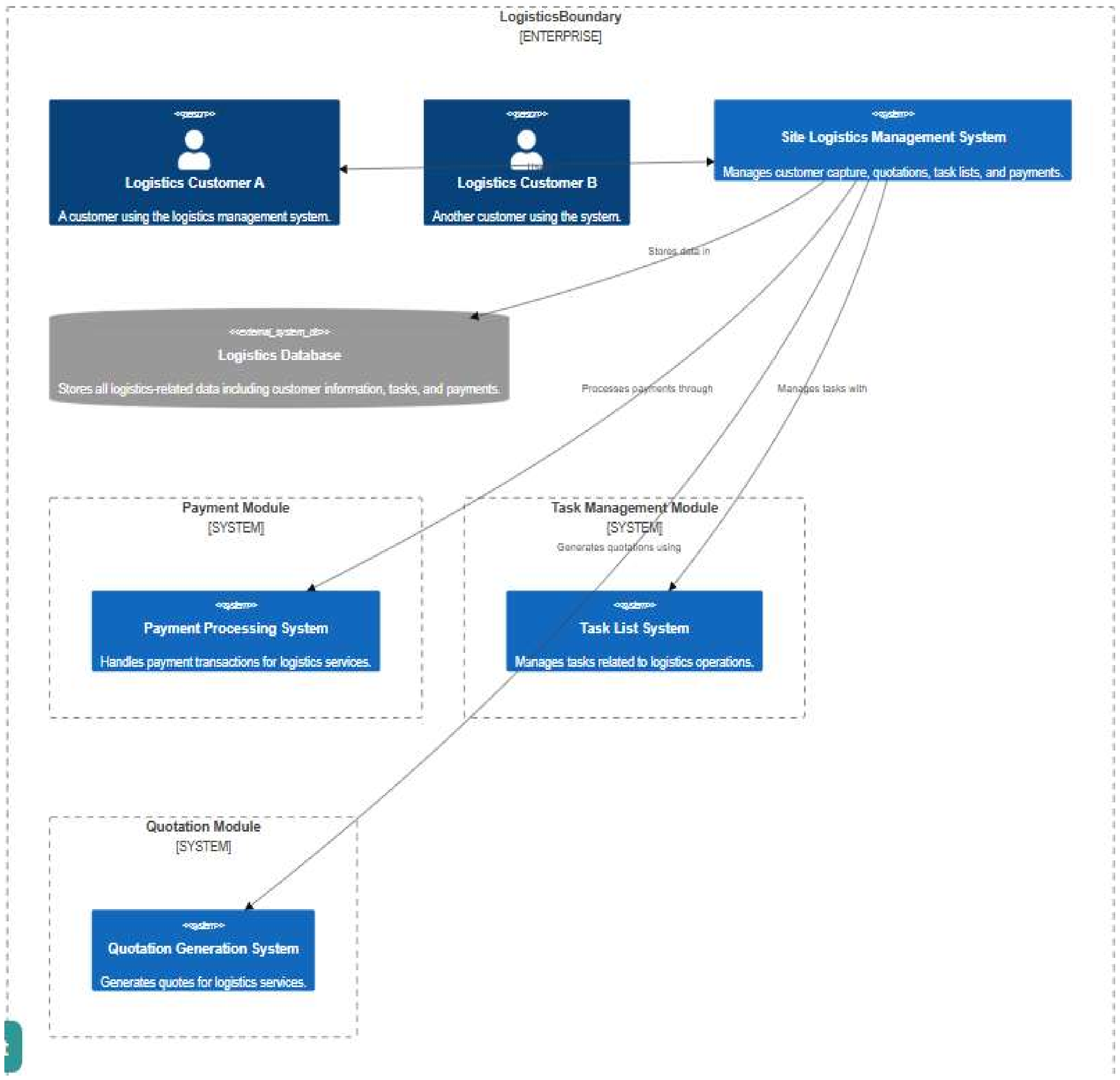

LogisticsBoundary
[ENTERPRISE]
<<person>>
Logistics Customer A
A customer using the logistics management system.
<<person>>
Logistics Customer B
Another customer using the system.
<<system>>
Site Logistics Management System
Manages customer capture, quotations, task lists, and payments.
Stores data in
<<external_system_db>>
Logistics Database
Stores all logistics-related data including customer information, tasks, and payments.
Processes payments through
Manages tasks with
Payment Module
[SYSTEM]
<<system>>
Payment Processing System
Handles payment transactions for logistics services.
Task Management Module
[SYSTEM]
Generates quotations using
<<system>>
Task List System
Manages tasks related to logistics operations.
Quotation Module
[SYSTEM]
<<system>>
Quotation Generation System
Generates quotes for logistics services.

Fig : 4.4 System Architecture

4.4.1 DATA FLOW DIAGRAM

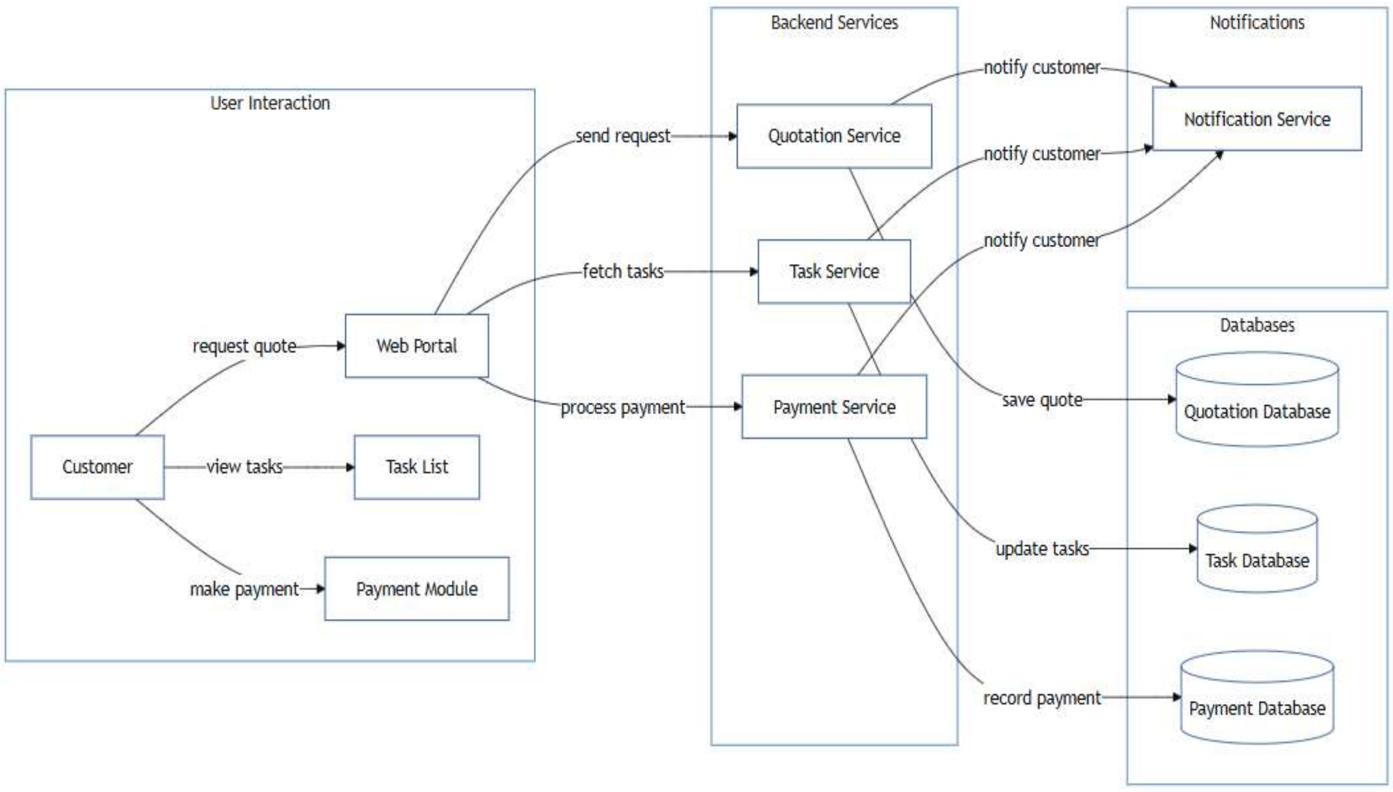


Fig 4.4.1 Data Flow Diagram

4.4.2 UML DIAGRAM

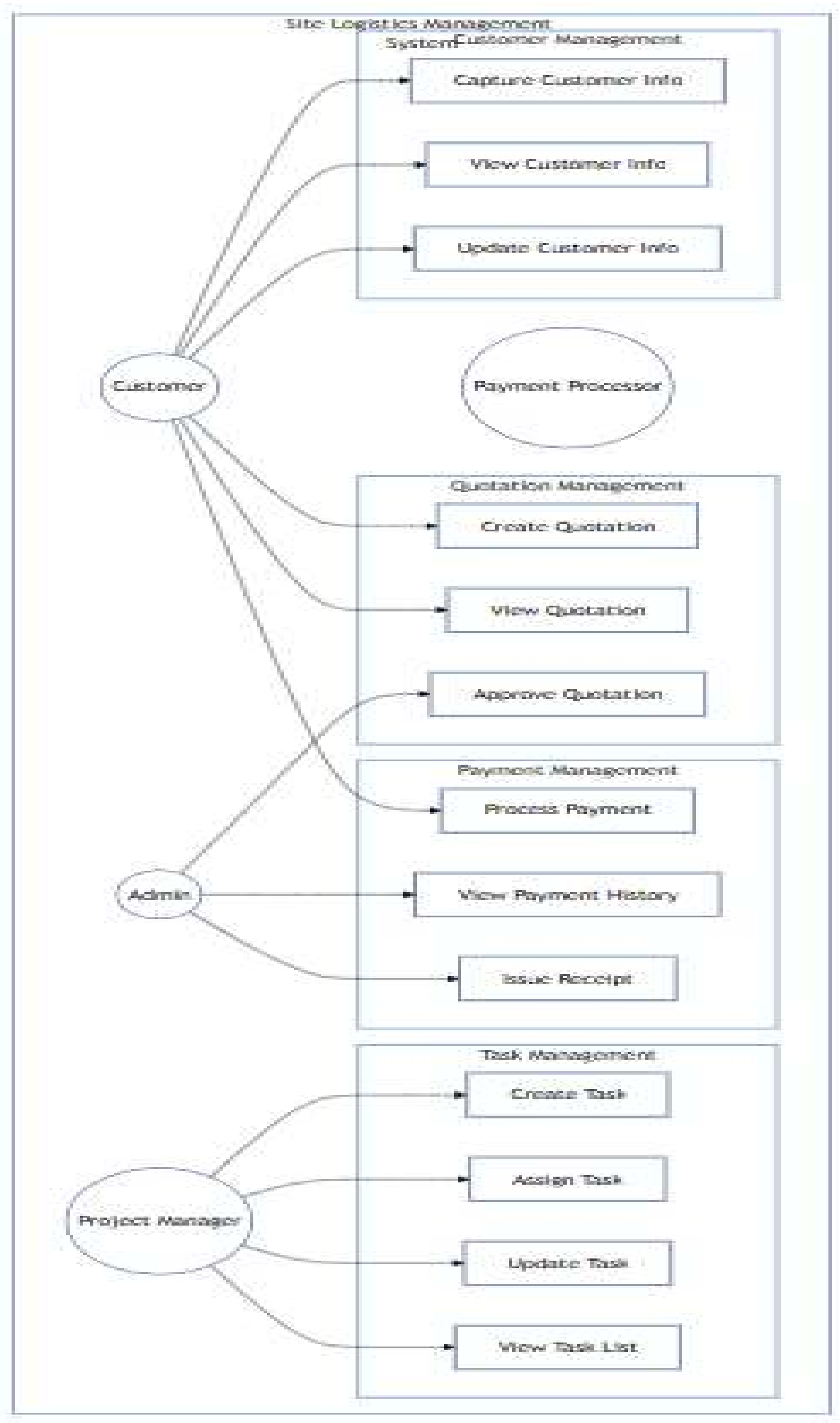


Fig 4.4.2UML Diagram

4.4.3 USE CASE DIAGRAM

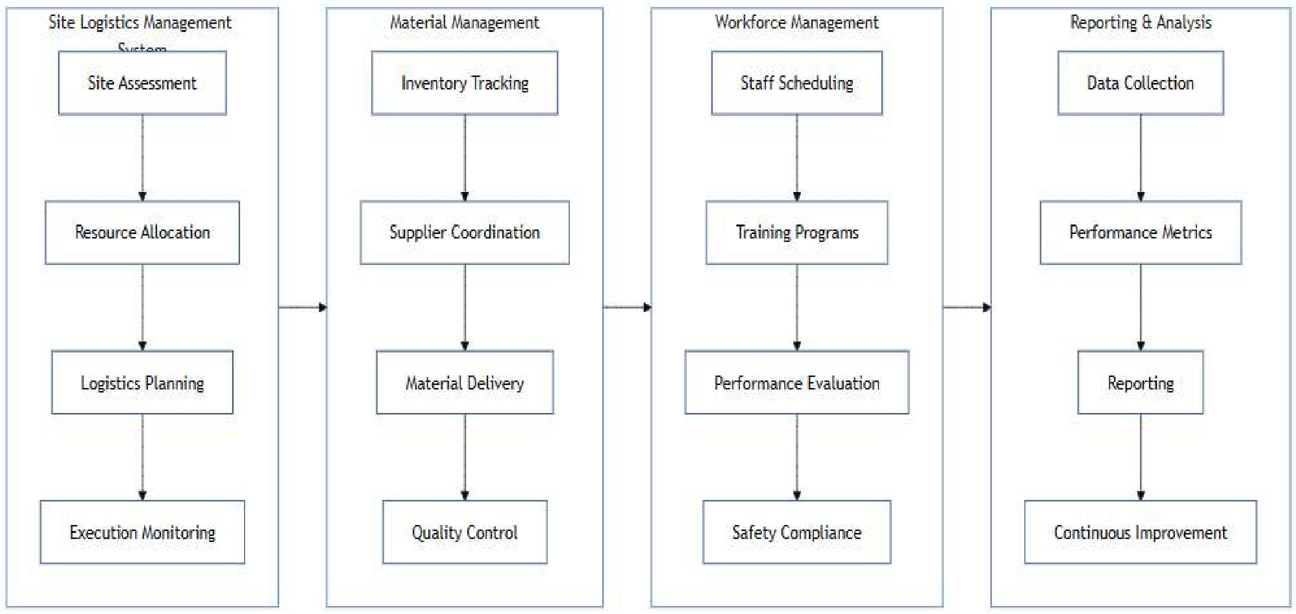


Fig 4.4.3 Use Case Diagram

4.4.4 CLASS DIAGRAM

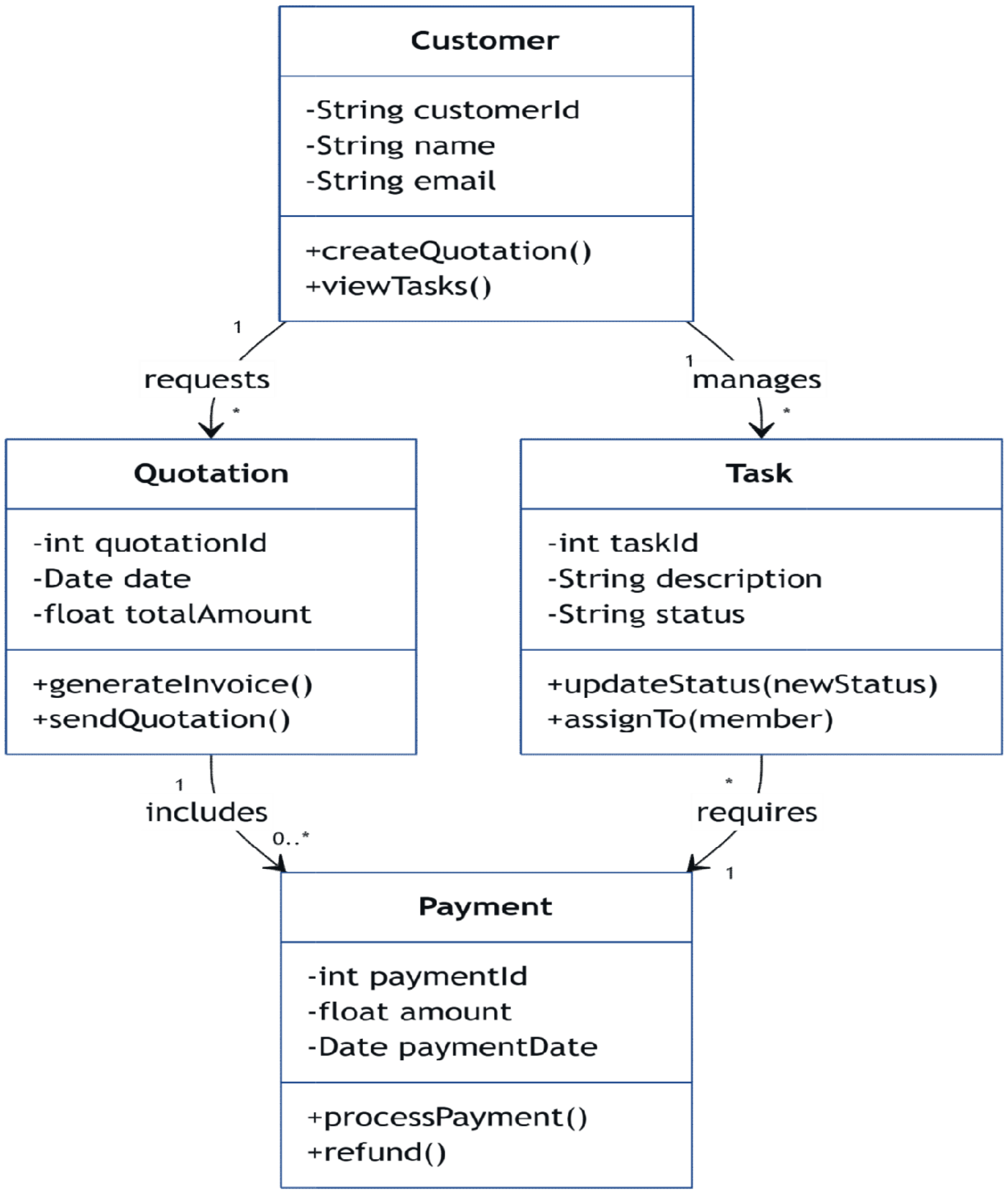


Fig : 4.4.4 Class Diagram

4.4.5 SEQUENCE DIAGRAM

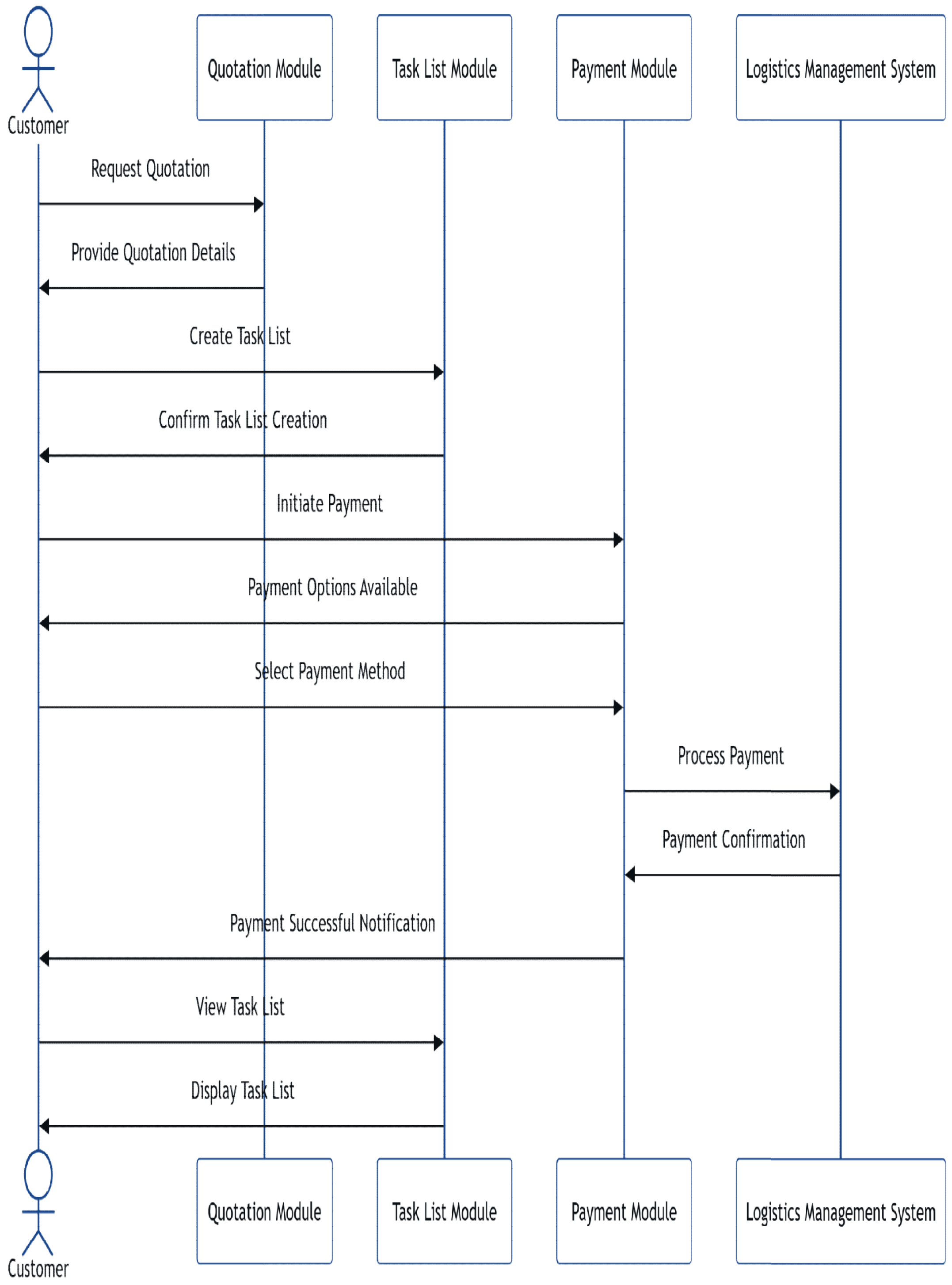


Fig 4.4.5 Sequence Diagram

4.4.6 ACTIVITY DIAGRAM

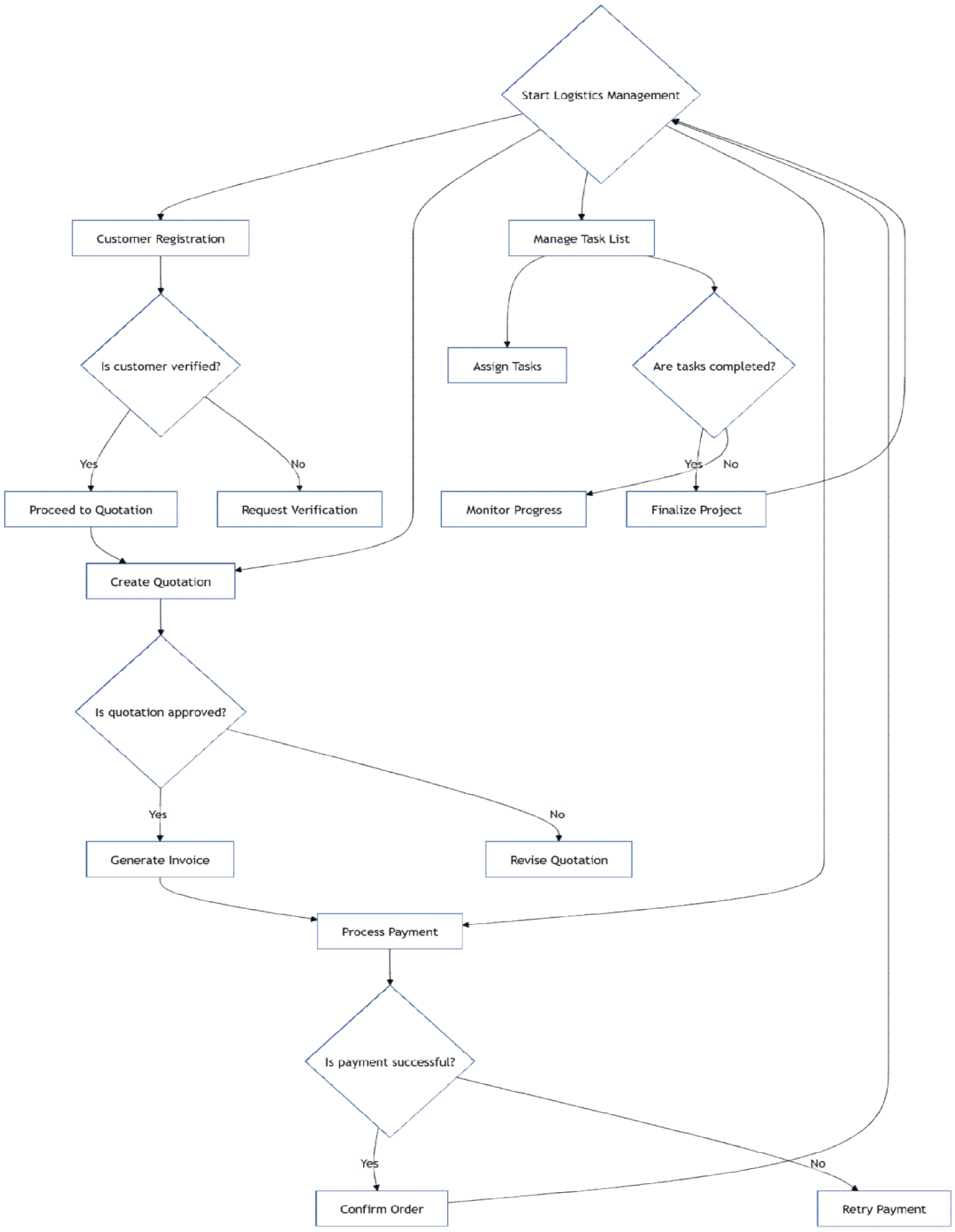


Fig 4.4.6 Activity Diagram

4.4 IMPLEMENTATION

4.5.1 MODULES

1. Customer Management Module

- Purpose: To manage all customer-related data and interactions.
- Key Features are as follows.
- Add, view, update, and delete customer information.
- Store contact details, company information, and site addresses.
- Track customer history (quotations, projects, payments).

2. Quotation Management Module

- Purpose: To handle cost estimations and service proposals for customers.
- Key Features are as follows.
- Generate new quotations for specific site logistics services.
- Store quotation details (description, cost breakdown, validity).
- Track quotation status (pending, approved, rejected).
- Generate downloadable PDFs or printable versions.

3. Task List / Work Order Management Module

- Purpose: To manage and track site-specific tasks and work schedules.
- Key Features are as follows.
- Create, assign, and schedule tasks for specific projects.
- Set deadlines and priorities.
- Track task progress (Not Started, In Progress, Completed).
- Assign resources and employees to tasks.
- Send task notifications/reminders to staff.

4. Payment Management Module

- Purpose: To manage financial transactions and billing processes.
- Key Features are as follows.
- Generate and track invoices based on approved quotations.
- Record customer payments (mode, amount, date).
- Track outstanding balances and due payments.
- Generate payment reports and receipts.

5. Inventory & Material Management Module (Optional but useful)

- Purpose: To track and manage materials, equipment, and logistics supplies on site.
- Key Features are as follows.
- Record stock levels, issue and receive materials.
- Manage vendors and procurement details.
- Track material usage against specific tasks or projects.

6. User & Role Management Module

- Purpose:
- To manage system access and user roles.
- Key Features are as follows.
- Admin can create users with different roles (admin, site manager, accountant, etc.).
- Role-based access to features/modules.
- Secure login and session management.

7. Reporting & Dashboard Module

- Purpose: To provide visual insights and reports for decision-makers.
- Key Features are as follows.

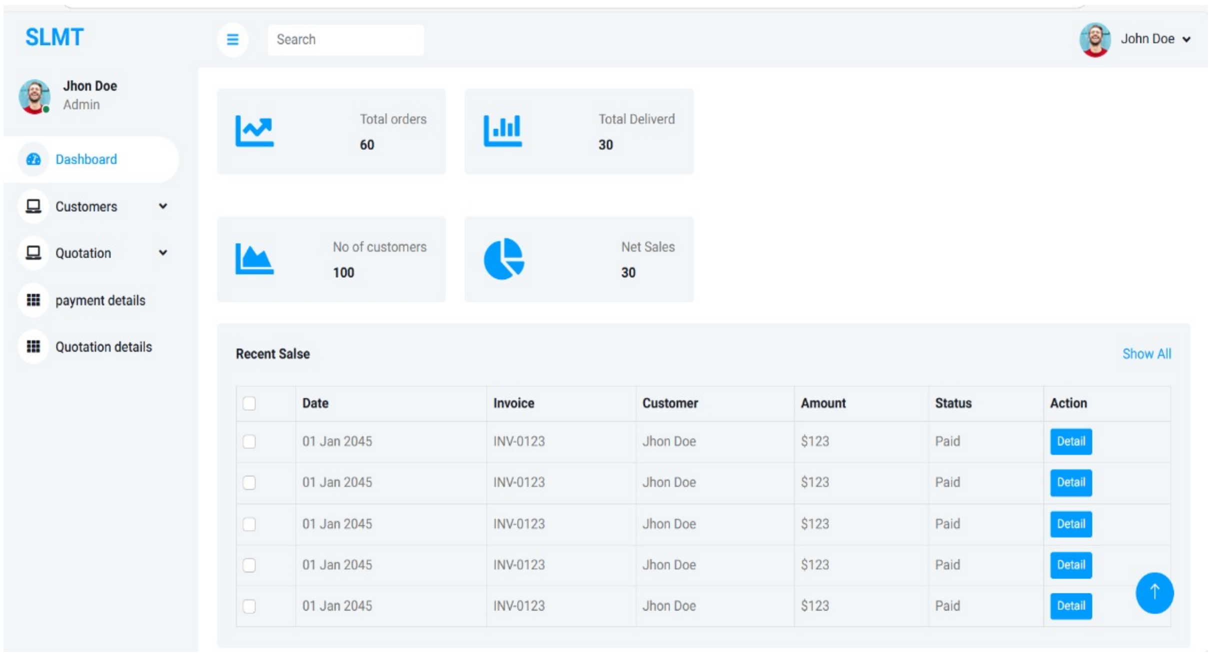
- Dashboard showing key KPIs (tasks completed, revenue, pending payments).
- Reports on customer activity, material usage, task performance, etc.
- Export options (PDF, Excel).

8. Notification & Alerts Module

- Purpose: To notify users of important updates or deadlines.
- Key Features are as follows.
- Email or in-app notifications for task deadlines, payment dues, or approvals.
- Alert system for low stock levels or delayed tasks.

4.6 SCREENSHOTS

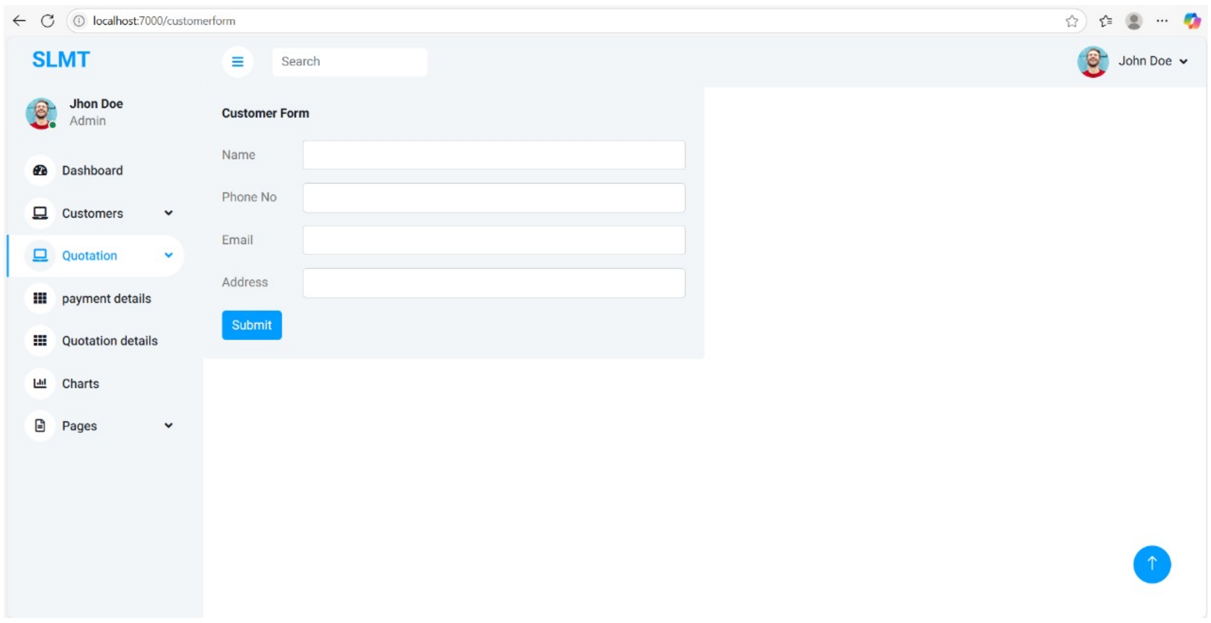
● Dashboard page



Recent Salse

	Date	Invoice	Customer	Amount	Status	Action
☐	01 Jan 2045	INV-0123	Jhon Doe	$123	Paid	Detail
☐	01 Jan 2045	INV-0123	Jhon Doe	$123	Paid	Detail
☐	01 Jan 2045	INV-0123	Jhon Doe	$123	Paid	Detail
☐	01 Jan 2045	INV-0123	Jhon Doe	$123	Paid	Detail
☐	01 Jan 2045	INV-0123	Jhon Doe	$123	Paid	Detail

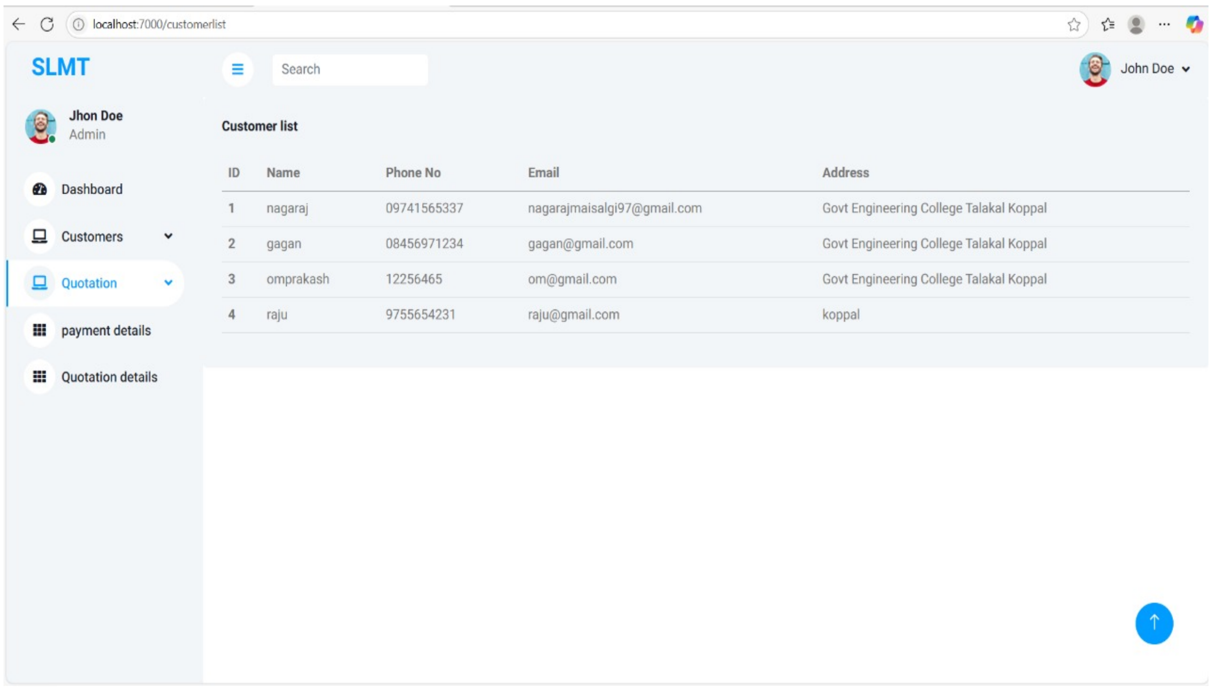
● Customer Form



Customer Form

Name	
Phone No	
Email	
Address	

● Customer list

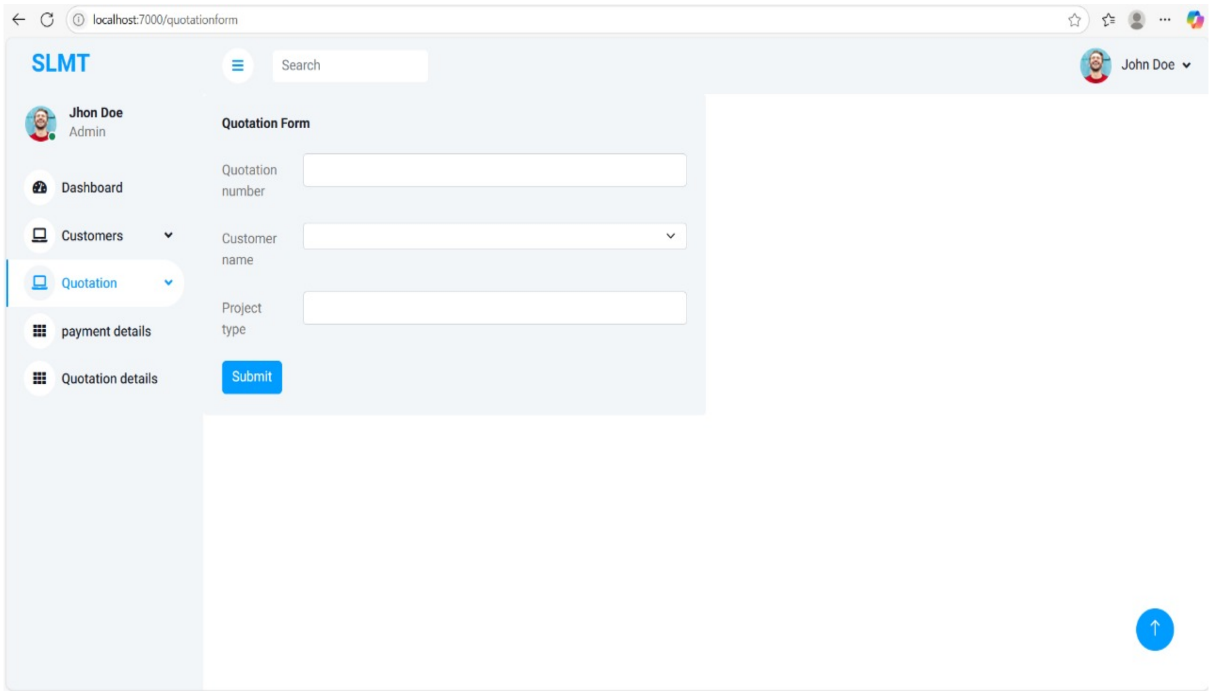


localhost:7000/customerlist

Customer list

ID	Name	Phone No	Email	Address
1	nagaraj	09741565337	nagarajmaisalgi97@gmail.com	Govt Engineering College Talakal Koppal
2	gagan	08456971234	gagan@gmail.com	Govt Engineering College Talakal Koppal
3	omprakash	12256465	om@gmail.com	Govt Engineering College Talakal Koppal
4	raju	9755654231	raju@gmail.com	koppal

● Quotation Form



localhost:7000/quotationform

Quotation Form

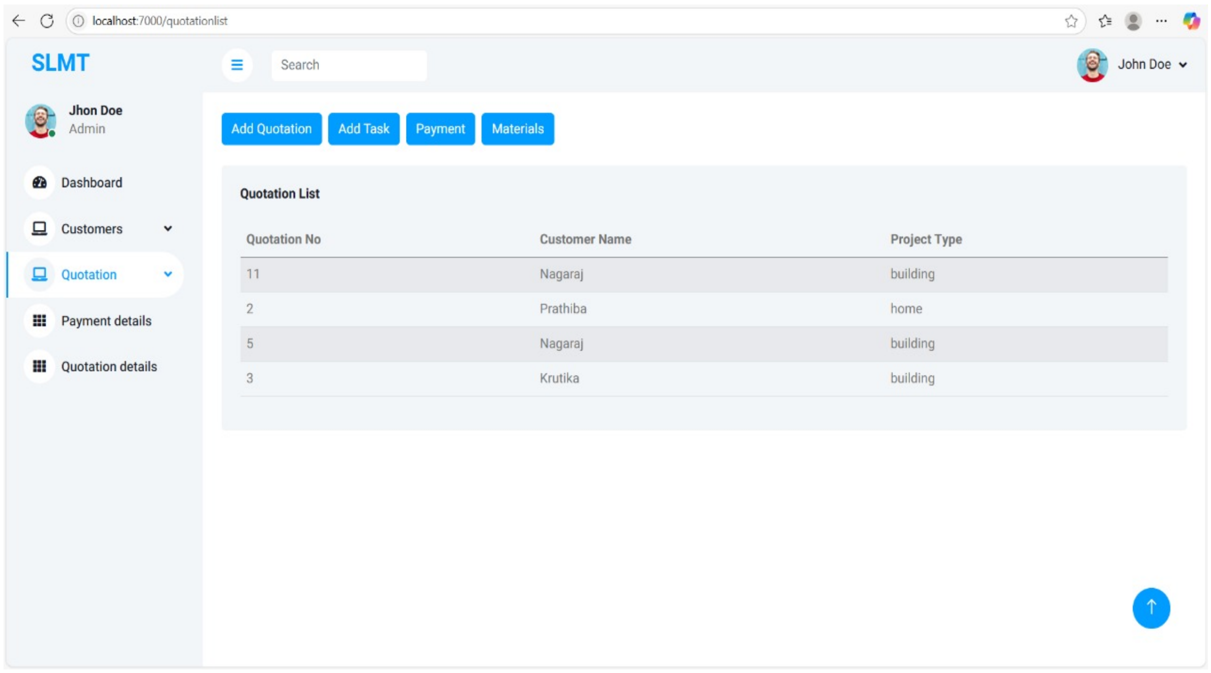
Quotation number

Customer name

Project type

Submit

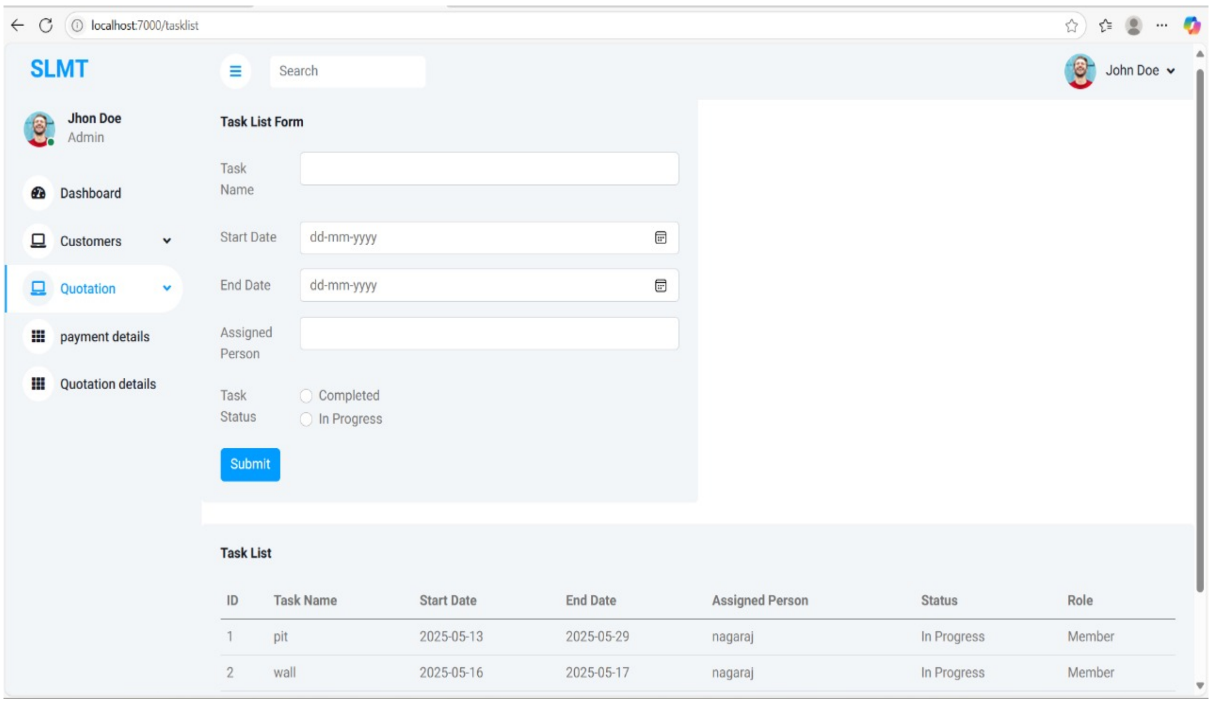
● Quotation List



Quotation List

Quotation No	Customer Name	Project Type
11	Nagaraj	building
2	Prathiba	home
5	Nagaraj	building
3	Krutika	building

● Task List Form



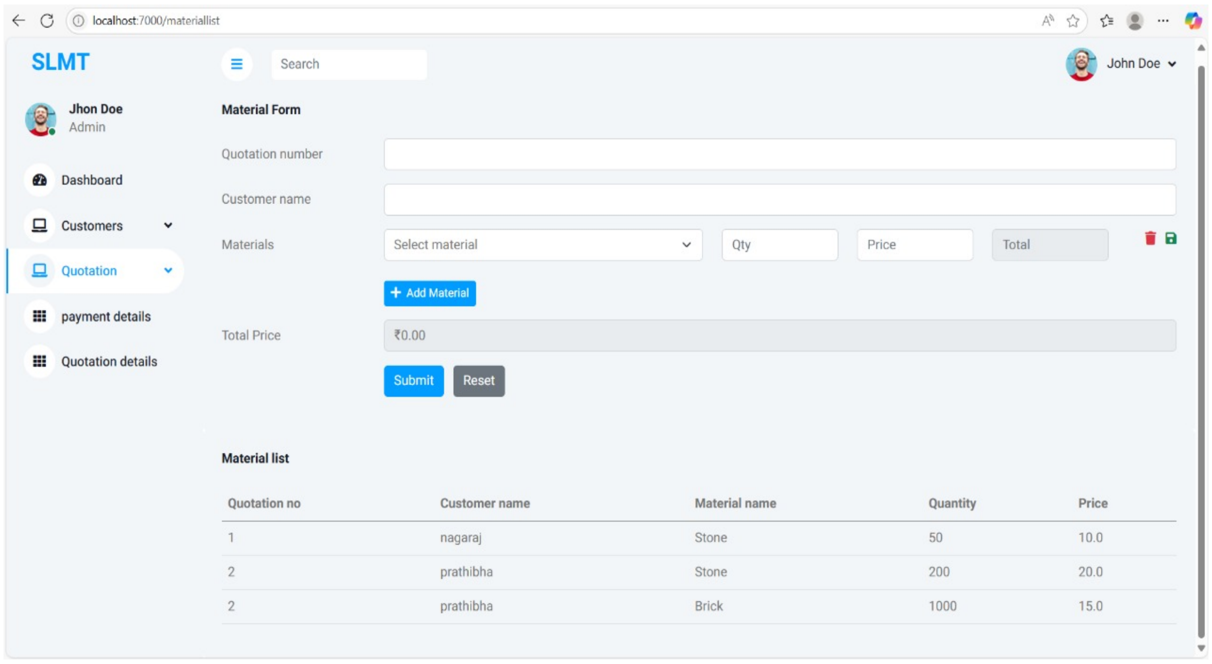
Task List Form

Task Name	
Start Date	dd-mm-yyyy
End Date	dd-mm-yyyy
Assigned Person	
Task Status	Completed / In Progress

Task List

ID	Task Name	Start Date	End Date	Assigned Person	Status	Role
1	pit	2025-05-13	2025-05-29	nagaraj	In Progress	Member
2	wall	2025-05-16	2025-05-17	nagaraj	In Progress	Member

● Material Form



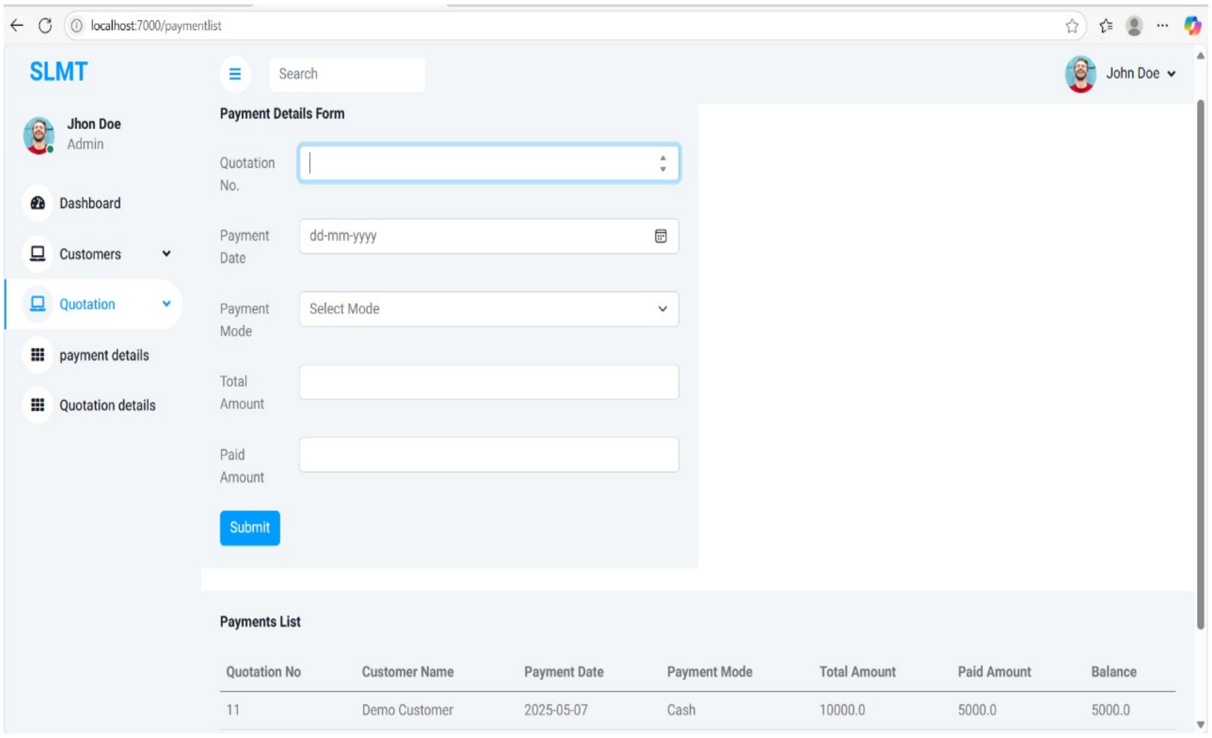
Material Form

Quotation number	
Customer name	
Materials	Select material · Qty · Price · Total
Total Price	₹0.00

Material list

Quotation no	Customer name	Material name	Quantity	Price
1	nagaraj	Stone	50	10.0
2	prathibha	Stone	200	20.0
2	prathibha	Brick	1000	15.0

● Payment Details Form



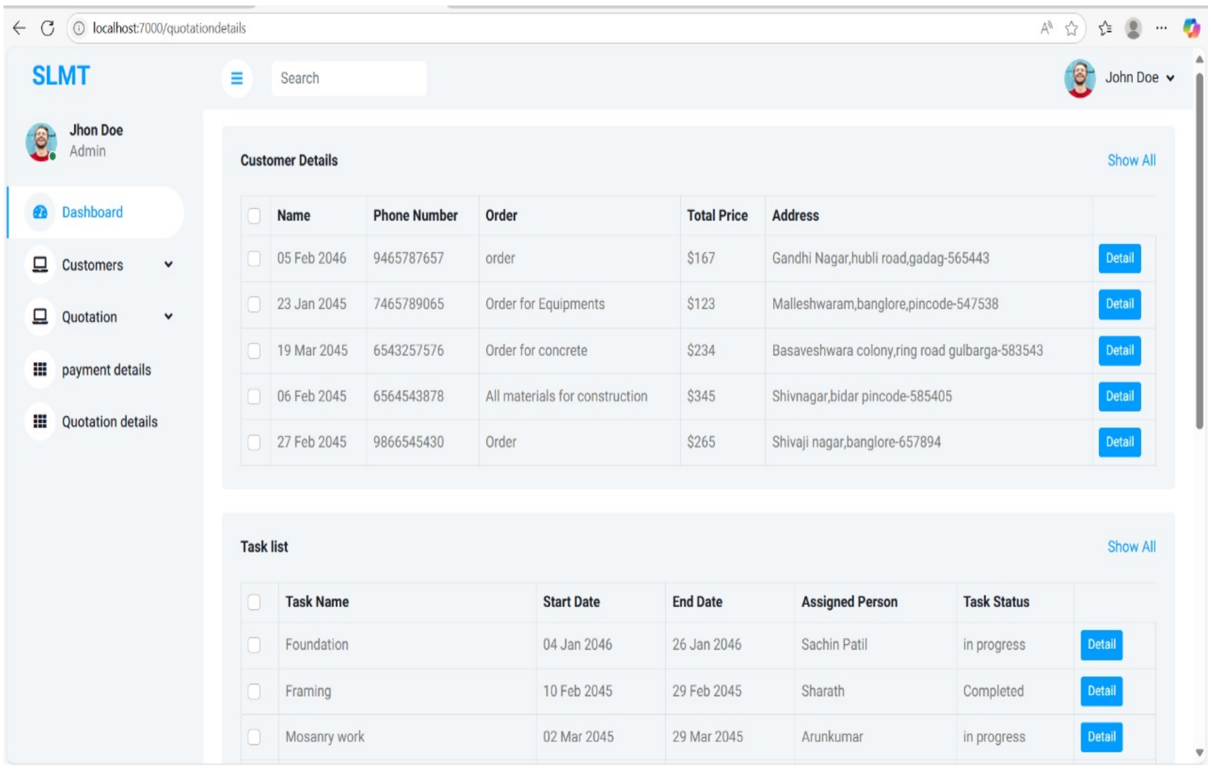
Payment Details Form

Quotation No.	
Payment Date	dd-mm-yyyy
Payment Mode	Select Mode
Total Amount	
Paid Amount	

Payments List

Quotation No	Customer Name	Payment Date	Payment Mode	Total Amount	Paid Amount	Balance
11	Demo Customer	2025-05-07	Cash	10000.0	5000.0	5000.0

● Quotation Details



Customer Details

Name	Phone Number	Order	Total Price	Address	
05 Feb 2046	9465787657	order	$167	Gandhi Nagar,hubli road,gadag-565443	Detail
23 Jan 2045	7465789065	Order for Equipments	$123	Malleshwaram,banglore,pincode-547538	Detail
19 Mar 2045	6543257576	Order for concrete	$234	Basaveshwara colony,ring road gulbarga-583543	Detail
06 Feb 2045	6564543878	All materials for construction	$345	Shivnagar,bidar pincode-585405	Detail
27 Feb 2045	9866545430	Order	$265	Shivaji nagar,banglore-657894	Detail

Task list

Task Name	Start Date	End Date	Assigned Person	Task Status	
Foundation	04 Jan 2046	26 Jan 2046	Sachin Patil	in progress	Detail
Framing	10 Feb 2045	29 Feb 2045	Sharath	Completed	Detail
Mosanry work	02 Mar 2045	29 Mar 2045	Arunkumar	in progress	Detail

CHAPTER 5

CONCLUSION

The Site Logistics Management System developed using Spring Boot provides a comprehensive and efficient solution for managing logistics operations across various project sites. By modularizing the application into distinct components—Customer, Quotation, Task List, and Payment—the system ensures organized and scalable development, enabling users to easily manage clients, generate and track quotations, monitor project-related tasks, and handle financial transactions effectively. Leveraging the power of Spring Boot, along with Spring Data JPA for database interaction and RESTful APIs for communication, the system ensures robustness, maintainability, and high performance. This project not only digitizes and streamlines traditional logistics workflows but also lays a solid foundation for future enhancements like authentication, real-time tracking, and reporting. Ultimately, it serves as a valuable tool for improving operational efficiency and decision-making in site logistics management.

BIBLIOGRAPHY

The development of the Site Logistics Management System using Spring Boot was supported by various technical resources and documentation. Key references include Spring in Action by Craig Walls and Pro Spring Boot 3 by Johnson and Sharma, which provided in-depth guidance on building scalable Java applications. The official documentation from Spring Boot (spring.io) and Oracle's Java SE documentation were essential in understanding core framework and language features. Online platforms like Baeldung and Geeks for Geeks offered practical tutorials and code examples for implementing REST APIs, security, and database integration. For front-end development, resources from W3Schools and the Bootstrap 5 documentation were utilized to design responsive interfaces. Additional support was drawn from Thymeleaf's official site for integrating dynamic content in HTML templates and from the MySQL documentation for configuring and managing the relational database. These combined sources played a crucial role in shaping the architecture, coding standards, and functionality of the project.

APPENDIX

The appendix of the Site Logistics Management System involved several tools, technologies, and design components that support the core functionality of the application. The project was built using the Spring Boot framework for the backend, leveraging features such as Spring Data JPA for database operations and Spring Security for user authentication and role management. The frontend was developed using HTML, CSS, and Bootstrap for responsive design, while Thymeleaf served as the template engine for dynamic web content rendering. MySQL was used as the relational database to store and manage data related to customers, quotations, tasks, and payments. The development environment included IntelliJ IDEA and VS Code, with Git used for version control. Postman was utilized for testing REST APIs, and the project structure followed a modular MVC architecture. The appendix also includes sample code snippets, database schema diagrams, and screenshots of major application pages such as login, dashboard, task management, and quotation generation, which are available upon request or attached in the supporting documentation.