



Interim Report  
Group 57

# Table of Content

Group details .....	3
Details of Project Supervisor, Co-supervisor, Advisors and Clients .....	4
Project Details:.....	5
Project Title.....	5
The Goal and Objectives.....	5
Tentative Problem Definition .....	6
A brief introduction to the project .....	6
The scope of the project .....	7
Tentative Technologies .....	11
Feasibility Study .....	12
Social Feasibility.....	12
Technical Feasibility .....	15
Operational Feasibility .....	16
Economic feasibility .....	17
Legal and Ethical Feasibility .....	18
Schedule Feasibility .....	18
Detailed Time Breakdown: .....	19
Main deliverables of the system .....	35
The Project Plan.....	42
References .....	44
Appendix.....	44
Technology Justification:.....	44
Project Scope Justification for 3 Credits per Member: .....	45
Declaration.....	48

## Project Proposal

Proposed Project Title: **StructuraX**

Project Group Details:

Group number: 57

Group members:

Name	Reg. No	Index No	Email address	Mobile Phone
(i) B. S. Madhushika	2022/CS/117	22001174	2022cs117@stu.ucsc.cmb.ac.lk	0715657995
(ii) S. R. Gamage	2022/CS/060	22000607	2022cs060@stu.ucsc.cmb.lk	07659971197
(iii) M. A. R. Peshala	2022/CS/155	22001557	2022cs155stu.ucsc.cmb.ac.lk	0762163506
(iv) Dilshani Nadeesha Kodithuwakku	2022/CS/092	22000925	2022cs092@stu.ucsc.cmb.lk	0757741833
(v) Arulsivam Thagshan	2022/IS/098	22020985	2022is098stu.ucsc.cmb.ac.lk	0779984032
(vi) Rathnayaka R.A.T.M.S	2022/IS/082	22020829	2022is082@stu.cmb.ac.lk	0768534750

## Details of Project Supervisor, Co-supervisor, Advisors and Clients

### **Proposed Project Supervisor:**

Name of the supervisor: Dr. H. N. D. Thilini

Signature of the supervisor:

Date:

### **Proposed Project Co-Supervisor:**

Name of the co-supervisor:

Signature of the co-supervisor:

Date:

### **The client of the Project**

Name of the client: None

Address of the client: None

Contact person at client: None

Contact number of the contact person: None

e-mail address of the contact person: None

## Project Details:

### **Project Title**

StructuraX - Smart Construction Project Management System

### **The Goal and Objectives**

#### **Main Goal:**

To develop a centralized, web-based construction site management system that facilitates effective coordination, monitoring, and decision-making among all stakeholders involved in a construction project—from design to completion.

#### **Objectives:**

- Create a centralized platform for managing multiple construction projects simultaneously
- Implement role-based access control for 12 different user types with specialized functionalities
- Develop automated workflow systems for project approval processes and documentation
- Establish real-time communication channels between all project stakeholders
- Integrate financial tracking and budget management capabilities
- Provide comprehensive reporting and analytics for project monitoring
- Ensure compliance tracking and legal document management
- Create mobile-responsive interfaces for on-site personnel
- Implement document version control and secure file sharing
- Develop notification and alert systems for critical project updates

## **Tentative Problem Definition**

Construction projects typically involve multiple stakeholders such as architects, engineers, site supervisors, managers, and property owners. Currently, many construction sites rely heavily on manual communication (phone calls, emails, paper-based approvals) and isolated tools (Excel sheets, CAD files, messaging apps) that do not integrate well. This causes inefficiencies like delays, miscommunication, lack of transparency, and cost overruns.

Moreover, real-time updates from the site are often not visible to stakeholders such as owners or financial managers, creating information gaps. Legal compliance and procurement processes are also delayed due to a lack of streamlined communication and documentation tools.

The proposed system aims to solve these problems by creating a centralized platform where all construction-related processes—from design approvals to daily progress tracking—can be managed in one integrated environment with role-specific access and automation.

## **A Brief introduction to the project**

The **StucturaX** is designed to revolutionize how construction companies manage their projects by providing a unified digital platform that connects all stakeholders in the construction ecosystem. This web-based system addresses the critical need for streamlined communication, efficient project tracking, and comprehensive resource management in the construction industry.

Our motivation stems from observing the recurring challenges faced by construction companies in coordinating multiple projects simultaneously while ensuring quality, staying within budget, and meeting deadlines. The system will serve as a central hub where architects can upload designs, engineers can approve technical specifications, project managers can coordinate timelines, financial managers can track budgets, and site supervisors can report real-time progress.

The platform is particularly valuable for construction companies handling multiple concurrent projects, as it provides scalable solutions that can adapt to projects of varying sizes and complexities. Unlike traditional project management tools that focus on single aspects of construction management, our system provides an integrated approach that covers the entire construction lifecycle from initial client contact to project completion.

This project represents an innovative approach to construction technology, combining modern web development practices with industry-specific requirements to create a solution that truly understands and addresses the unique challenges of construction project management.

## **The scope of the project**

### **What will be covered:**

- Role-based dashboards for 10+ user types (architect, QS, project manager, etc.)
- Blueprint upload, versioning, and approval workflows.
- Financial planning, approval, and cost tracking.
- Daily progress reporting and labor tracking.
- Material request handling and procurement interface for suppliers.
- Chat and messaging system for communication.
- Legal contract and document storage.
- Mobile access for site supervisors to submit updates and media files.
- Alerts, notifications, and scheduling tools for meetings or milestones.
- Admin module for managing system users and permissions.

### **What will not be covered:**

- Real-time GPS tracking or drone integration.
- Integration with external financial software (optional in future).
- AI-based decision making or predictive analytics (out of scope for now).
- Full-fledged HR payroll systems (only relevant cost-tracking features are included).

### **Users (possible actors) of the system:**

1. Architect
2. QS
3. Senior QS
4. Project Manager
5. Financial Department
6. Director
7. Marketing Manager
8. Legal Officer
9. Property Owner
10. Site Supervisor
11. Supplier
12. System Admin

## **Main functionalities of the system:**

- Work breakdown structure management.
- Budget planning and approval.
- Procurement management and supplier quotations.
- Daily progress tracking with site reports.
- Document and contract management.
- Worker and subcontractor coordination.
- Task management and internal communication.
- Reporting dashboards for management.
- Legal compliance documentation.
- Calendar and milestone planning.

## **Role based Functions in the system**

### **1. Quantity Survey Officer**

Review the design/plan/drawing and create the BOQ (Bill of quantity).

Estimate the project value

Create the work breakdown list and mark the notification needed works, mark the milestones.

View the project progress, financial status (payments), remaining time and related everything.

Create site visit logs and report.

Create the material list and manage it.(With Selected Brands)

Update the costings according to the changes.

View the daily updates of the site supervisor.

View the materials remain amounts and renew requests, approve the requests.

View the purchasing details.

Chat with project owners, finance department.

Request quotations for material purchases.

Add a project design/plan/drawing.

Use a to do list.

### **2. Senior Quantity Survey Officer**

All the functions of the QS officer.

Assign a QS officer to a project.

Create a customer login.

View Confirm or Edit the estimated value (BOQ) for the project.



### **3. Project Manager**

View the project progress, financial status (payments), remaining time and related everything.

View the materials remain amounts and renew requests.

Create site visit logs and report.

View the daily updates of the site supervisor.

Chat with project owners.

Use a to do list.

### **4. Site Supervisor**

View the project plan/drawing.

See the work breakdown structure, update it.

Update the daily work progress.

Update the daily workers list

Request materials and tools from the QS officers.

Manage site inventory.

Use a to do list.

Upload site photos, videos of the project to the system

### **5. Product owner**

Login to the system and change the password

View the project plan/drawing.

View project contract.

View the construction progress of the project.

View the materials that use in the project (brands)

Track the payments related to the project.

Request site visits.

Submit payment confirmation.

Chat with site supervisor, project manager and QS officer.

View remaining work, payment and time for finish the project.

Receive payment reminder notification.

View if there any cost changes.

### **6. Legal officer**

Submit the contract documents to the system.

Add the legal document processes related to a project.

Change the status of the processes (completed, pending, unsuccessful)

## **7. Supplier**

Maintain a product catalogue.  
View and respond to quotations.  
Receive e-mails with supply material orders.  
Confirm the delivery status.  
Track the payment history and current pending.  
View supply history.  
Submit payment requesting invoices.

## **8. Director**

Initiate the project.  
View all pending and finished projects and their details.  
Assign project managers, site supervisors to the projects.  
View, Update site visit logs.  
View the daily update reports, purchasing details.  
View all system generated reports, download them.  
View the resource inventories.  
Change the project status (Ongoing, terminate, hold)

## **9. Architecture**

Upload the project plan/drawing.  
Chat with Senior QS officer.

## **10. Finance Department**

Create the payment plan, according to the estimated value.  
Add the cash paid payment reports to the related projects  
Approve the confirmed payments.  
View the payment history of the projects.  
View the daily updated labor worker count and types.  
Enter the values for the labor charges of each site.

## **11. System Administrator**

Create, view, update and delete the system users.  
Remove inactive users (unaccepted, terminated project's owners)

## **Tentative Technologies**

Our project adopts a modern and scalable technology stack that includes React for the frontend, Spring Boot for the backend, MySQL for the database, and Flutter for the mobile application. Each technology is chosen with careful consideration for performance, maintainability, and user experience.

### **Frontend: React.js**

React is a widely used JavaScript library developed by Meta for building interactive user interfaces. It allows developers to build reusable components, manage state efficiently, and render pages dynamically using a virtual DOM. Compared to traditional approaches like plain HTML or jQuery, React promotes better organization, faster development, and enhanced scalability. Its strong community support and compatibility with RESTful APIs make it an ideal choice for our web frontend.

### **Backend: Spring Boot**

Spring Boot is a powerful, production-ready backend framework based on Java. It simplifies the creation of RESTful APIs and offers built-in support for security, dependency injection, and database integration. It is well-suited for enterprise applications requiring robust architecture and long-term maintainability. Compared to other backends like Express.js or Laravel, Spring Boot offers strong typing, greater modularity, and extensive documentation — making it easier to manage and secure in the long run.

### **Database: MySQL**

MySQL is a reliable and mature relational database system that supports structured data storage, SQL queries, and relationships between entities. It is highly suitable for applications where data integrity and consistency are crucial, such as managing user records, projects, tasks, and timelines. Compared to NoSQL alternatives like MongoDB, MySQL is preferable for systems with well-defined schemas and complex queries.

## **Feasibility Study**

### **Social Feasibility**

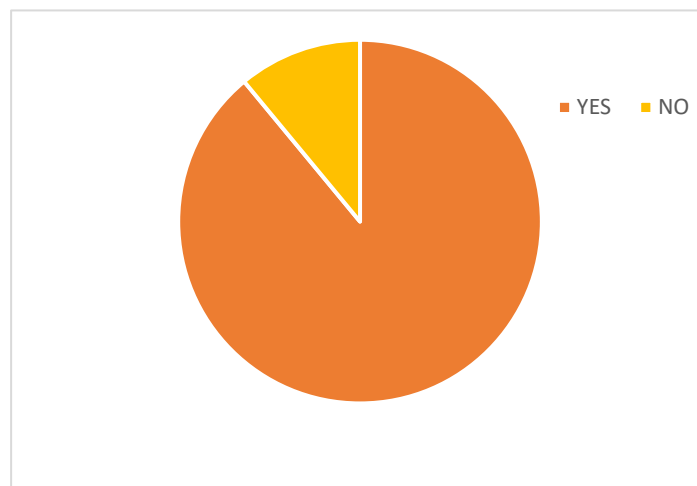
The construction industry in Sri Lanka faces significant challenges in project management and coordination due to fragmented communication systems and lack of integrated platforms. Traditional construction project management relies heavily on manual processes, paper-based documentation, and separate communication channels, leading to inefficiencies, delays, and cost overruns.

Our proposed Construction Management System addresses these critical issues by providing a centralized platform that connects all construction stakeholders. The system will benefit various user groups, including construction companies managing multiple projects, property owners seeking transparency in their investments, and professionals requiring streamlined workflows.

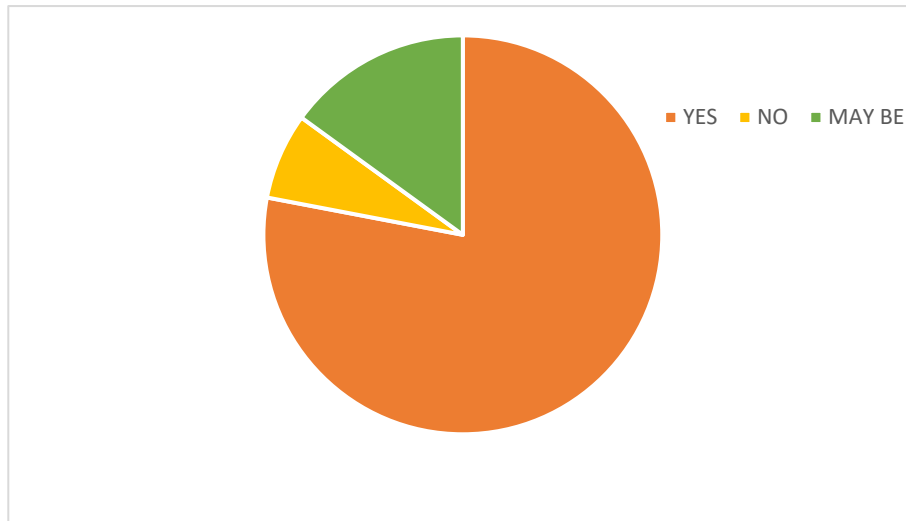
### **Survey Results on Construction Management System Adoption:**

*Given below are the results for a hypothetical survey conducted on the subject:*

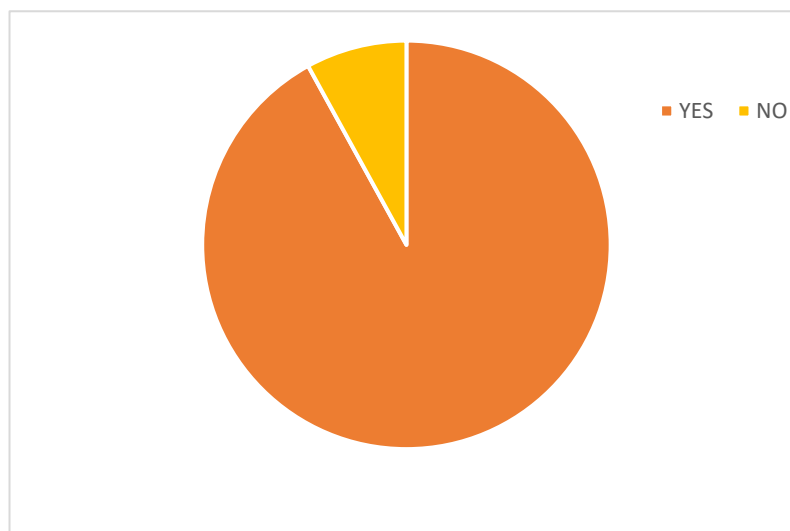
Q1. Do you face challenges in coordinating multiple construction projects simultaneously?



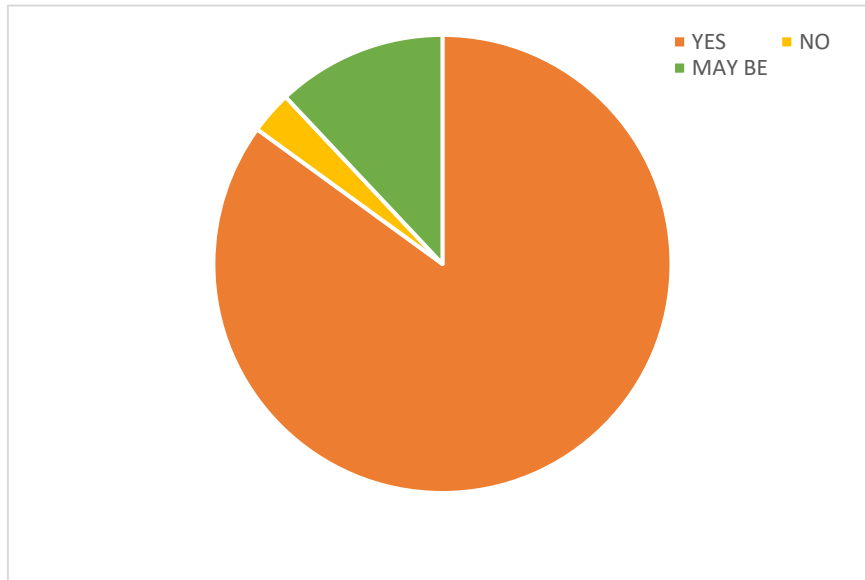
Q2. Would you be interested in using a digital platform to manage construction project workflows?



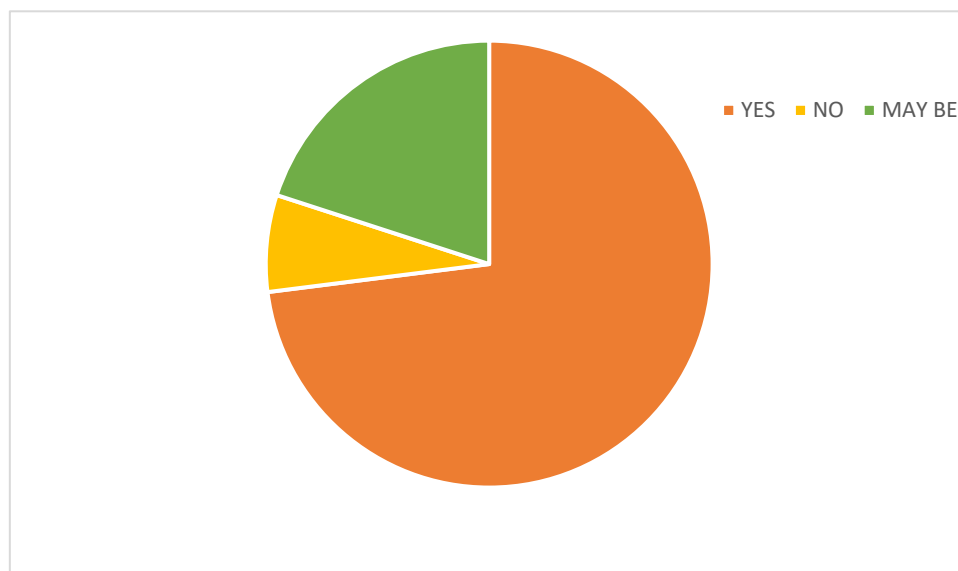
Q3. Do you believe real-time project tracking would improve construction project outcomes?



Q4. Would you find value in a system that integrates all construction stakeholders (architects, engineers, supervisors, suppliers) in one platform?



Q5. Are you willing to adopt new technology solutions to improve construction project efficiency?



Based on these survey insights, there is a strong market demand for such a system. The construction industry's growing trend of digitalization and the need for improved project coordination make this system highly socially feasible with long-term viability.

## Technical Feasibility

The main deliverable of this project is a comprehensive construction management platform that includes both web and mobile applications. Our project adopts a modern and scalable technology stack carefully selected based on specific technical requirements and comparative advantages over alternative solutions.

### Frontend: React.js

React was chosen over alternatives like Angular or Vue.js due to its component-based architecture that promotes code reusability and maintainability - crucial for a construction management system with complex UI requirements. Unlike traditional HTML/CSS/JavaScript approaches, React's virtual DOM ensures optimal performance when handling frequent data updates from construction sites. Compared to Angular's steep learning curve and Vue.js's smaller ecosystem, React offers extensive third-party libraries, robust community support, and seamless integration with RESTful APIs. Its declarative programming model reduces development complexity and allows for efficient state management across multiple construction project interfaces.

### Backend: Spring

Spring Boot was selected over Node.js (Express.js), PHP (Laravel), or Python (Django) frameworks due to its enterprise-grade capabilities essential for construction project management. While Express.js offers rapid development, it lacks the type safety and structured architecture required for handling complex construction workflows, financial calculations, and regulatory compliance. Unlike Laravel or Django, Spring Boot provides superior performance for concurrent user access, built-in security features for sensitive construction data, and robust transaction management for financial operations. Its dependency injection and auto-configuration capabilities reduce boilerplate code while maintaining enterprise-level security standards required in construction industry applications.

### Database: MySQL

MySQL was chosen over PostgreSQL, MongoDB, or SQLite based on specific construction management requirements. Unlike NoSQL databases like MongoDB, construction projects require strict data consistency for financial records, project timelines, and regulatory documentation where ACID properties are non-negotiable. Compared to PostgreSQL, MySQL offers simpler administration and better performance for read-heavy operations common in construction reporting. SQLite lacks the concurrent access capabilities needed for multiple users accessing project data simultaneously. MySQL's mature replication features ensure data availability crucial for construction sites with varying connectivity, while its structured schema approach perfectly suits the hierarchical nature of construction project data (projects → phases → tasks → resources).

### **Alternative Technologies Considered and Rejected:**

- MEAN/MERN Stack: Rejected due to JavaScript's type safety limitations for financial calculations and lack of enterprise-grade security features
- Ruby on Rails: Rejected due to slower performance and smaller talent pool for long-term maintenance
- Oracle Database: Rejected due to licensing costs and over-engineering for project requirements
- Native Mobile Development: Rejected due to increased development time and maintenance complexity

### **Additional Tools:**

- GitHub for version control and team collaboration
- Jira for project management and issue tracking
- Postman for API testing and documentation

The development team possesses foundational knowledge in these technologies, and the 4-5 month timeline accommodates skill enhancement through extensive documentation and community resources. The chosen technology stack's maturity, active community support, and proven track record in enterprise applications ensures technical implementation feasibility within project constraints while meeting construction industry's specific requirements for reliability, security, and performance

### **Operational Feasibility**

This section evaluates how effectively our proposed solution addresses current construction project management challenges. The system must meet operational requirements to ensure user adoption and success.

### **Currently, construction companies struggle with:**

- Fragmented communication between stakeholders
- Manual document management and version control issues
- Lack of real-time project visibility
- Inefficient resource and budget tracking
- Delayed decision-making processes



**Our Construction Management System addresses these issues by providing:**

- Centralized communication platform for all stakeholders
- Automated workflow management and approval processes
- Real-time project dashboards and reporting
- Integrated financial tracking and budget management
- Mobile-responsive interfaces for field personnel

**User Requirements:**

- Internet connectivity and modern web browser
- Basic computer literacy for system navigation
- Mobile device capability for site supervisors
- Understanding of construction project workflows

The system is designed to be intuitive and user-friendly, requiring minimal training for adoption. As construction professionals increasingly embrace digital solutions, operational feasibility is high.

### **Economic Feasibility**

Here the cost for the project will be taken into consideration.

Since the team uses open-source technologies for the development process, there will be no cost for purchasing them, and the team will not be making profits. Therefore, the development will be cost-free.

Although for hosting the web application as well as the database, free web hosting can be used completely freely or shared hosting packages can be used at a low cost, the growth of the user base and platform will require a more expensive hosting package.

The website requires a domain name for which it will cost around Rs.4000 annually for a domain of “.lk”.

Hence the cost for building and maintaining the web application is affordable and economically feasible.

## Legal and Ethical Feasibility

### Licensing and Attribution

- All technologies used are open-source with permissive licenses (MIT, Apache 2.0)
- No proprietary software or pirated versions will be used
- Proper attribution will be provided for third-party libraries and resource

### Handling

- User registration data will be collected with explicit consent
- Personal and project information will be protected with encryption
- Compliance with data protection regulations (GDPR principles)
- No data sharing with unauthorized third parties
- Secure backup and disaster recovery procedures

### Intellectual Property

- Construction documents and blueprints remain property of respective owners
- System provides secure access control and sharing permissions
- Version control maintains document integrity and audit trails

### Transparency

- All project activities and financial transactions are logged and auditable
- Users have access to their data and can request data export/deletion
- Clear terms of service and privacy policy

## Schedule Feasibility

**Project Duration:** 4–5 months (16–20 weeks)

**Resource Allocation:**

- **Team Size:** 6 members

**Working Hours:**

- Weekdays:  $1.5 \text{ hours/day} \times 5 \text{ days} = 7.5 \text{ hours/week}$  per member
- Weekends:  $4 \text{ hours/day} \times 2 \text{ days} = 8 \text{ hours/week}$  per member
- Total Weekly Hours per Member:  $7.5 + 8 = 15.5 \text{ hours}$
- Total Weekly Hours (All Members):  $15.5 \times 6 = 93 \text{ hours/week}$

**Project Duration:** 16 weeks

**Total Estimated Man-hours:**

93 hours/week  $\times$  16 weeks = **1,488 hours**

**Detailed Time Breakdown:**

<i>Task</i>	<b>Estimated Hours</b>	<b>% of Total</b>
<i>Requirements Analysis &amp; Planning</i>	75 hrs	5%
<i>System Design &amp; Architecture</i>	150 hrs	10%
<i>Database Design &amp; Implementation</i>	112 hrs	7.5%
<i>Backend Development</i>	450 hrs	30%
<i>Frontend Development</i>	375 hrs	25%
<i>Integration &amp; Testing</i>	225 hrs	15%
<i>Documentation</i>	75 hrs	5%
<i>Deployment &amp; Final Testing</i>	38 hrs	2.5%

**Development Methodology:** Agile methodology with 2-week sprints, allowing for iterative development and regular progress assessment.

**Risk Mitigation:** The timeline includes 10% buffer time for unexpected challenges and requirement changes.

**Credit Justification:** With 4,000 total man-hours distributed among 6 team members over 20 weeks, each member will contribute approximately 800-1,000 hours, which adequately justifies the 3-credit requirement per member for this comprehensive system development project.

## **System Architecture**

Our project platform adopts a modular, Scalable architecture designed to support a multi-role , multi-project construction management system . The System consists of several deployable components that communicate through secure RESTful APIs and real-time protocols. Below is an overview of the high- level system architecture

### **Frontend (Web Application – React.js)**

- Interfaces tailored for each user role (e.g., QS, Architect, Project Manager, Site Supervisor, etc.)
- Communicates with the backend via RESTful APIs.
- Displays real-time data through dashboards and notifications.

- Mobile-responsive design, enabling seamless access for site supervisors and other stakeholders via phones and tablets.

### **Backend Services (Spring Boot):**

- Handles business logic, user role permissions, workflow automation, document management, notifications, and real-time updates.
- Manages project data, financial operations, documentation workflows, and communication logic.
- Integrates with external services such as SMS gateways or email services for alerts.
- Provides secure API endpoints for all client interactions.

### **Database (MySQL)**

- Stores structured data such as user profiles, project data, material records, reports, financial logs, and document metadata.
- Ensures data integrity and consistency with relational schemas and ACID-compliant operations

### **Authentication & Identity Management**

- Spring Security handles login, session management, and role-based access.
- Each user role has tailored access levels and permissions to ensure secure workflows.

### **File Storage System**

- Stores blueprints, contracts, reports, and site documentation (photos, PDFs).
- Secure upload/download functionality with version control.

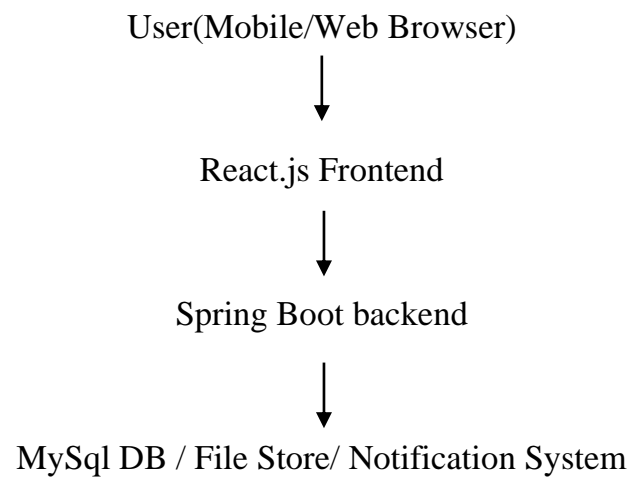
### **Notification & Messaging System**

- Enables in-system notifications, status updates, and reminders for tasks, deadlines, and payments.
- Chat feature allows real-time messaging between stakeholders

### **Hosting & Deployment**

- Hosted on a cloud-based or shared hosting environment.
- Version control via GitHub; deployment pipeline supports efficient updates.

## Overview of System Communication



## **Requirements Specification**

### Functional Requirements

- The system shall allow QS officers to create and manage BOQs.
- The system shall support uploading and versioning of construction blueprints.
- The system shall provide real-time chat between users.
- The system shall allow project managers to view progress dashboards.
- The system shall restrict each user role to its specific permissions
- The system shall allow site supervisors to submit daily work reports.

### Non- functional Requirements

- The system shall be mobile responsive for field usage.
- The system shall encrypt all sensitive data (e.g., login credentials)
- The system shall support concurrent access by at least 50 users.
- The system shall load any dashboard in under 3 seconds on average.
- The system shall provide backup and recovery options

## Use-Case Diagrams

Supplier :-



## Property Owner

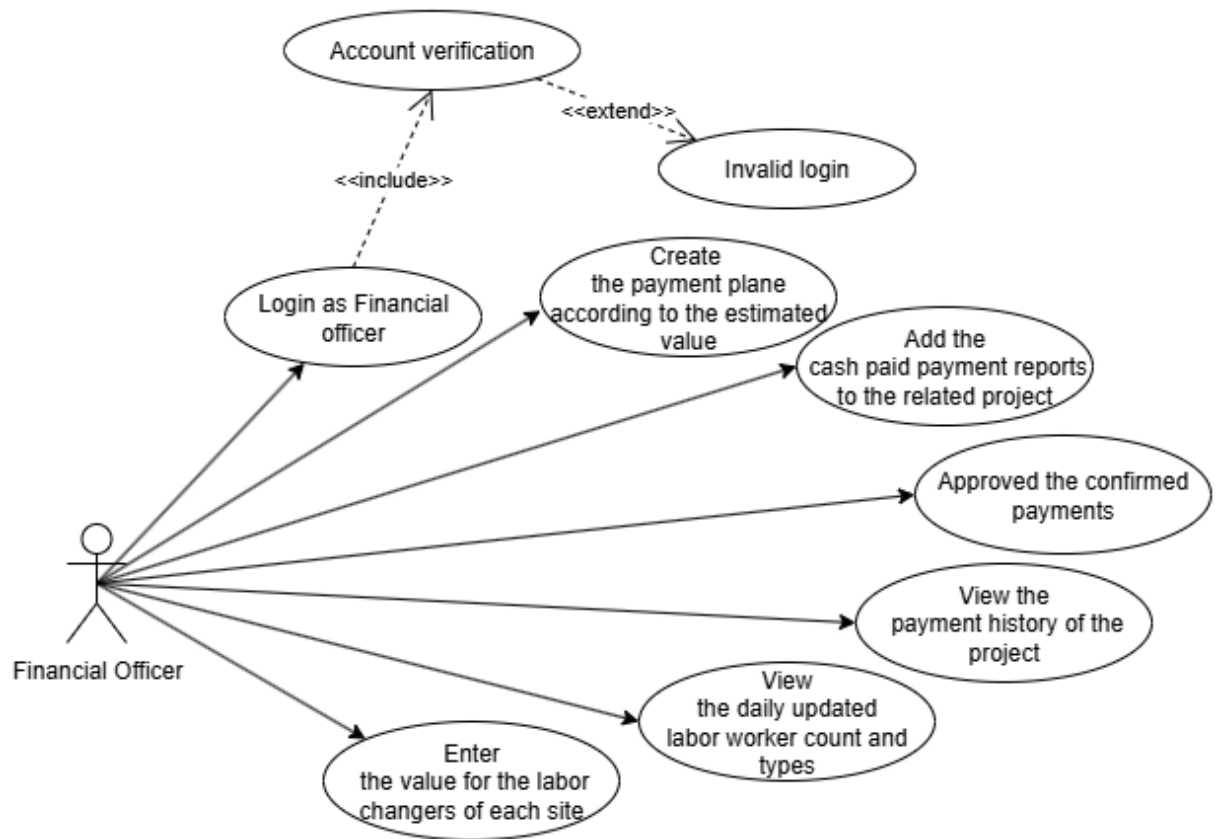




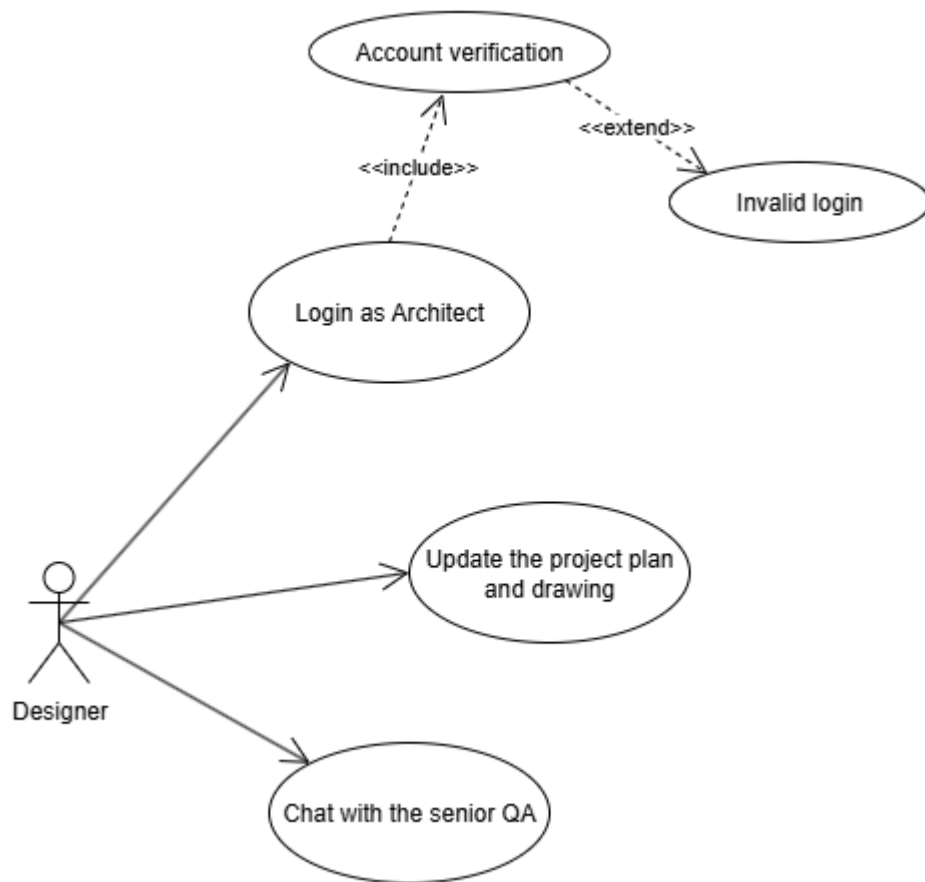
## QS offices



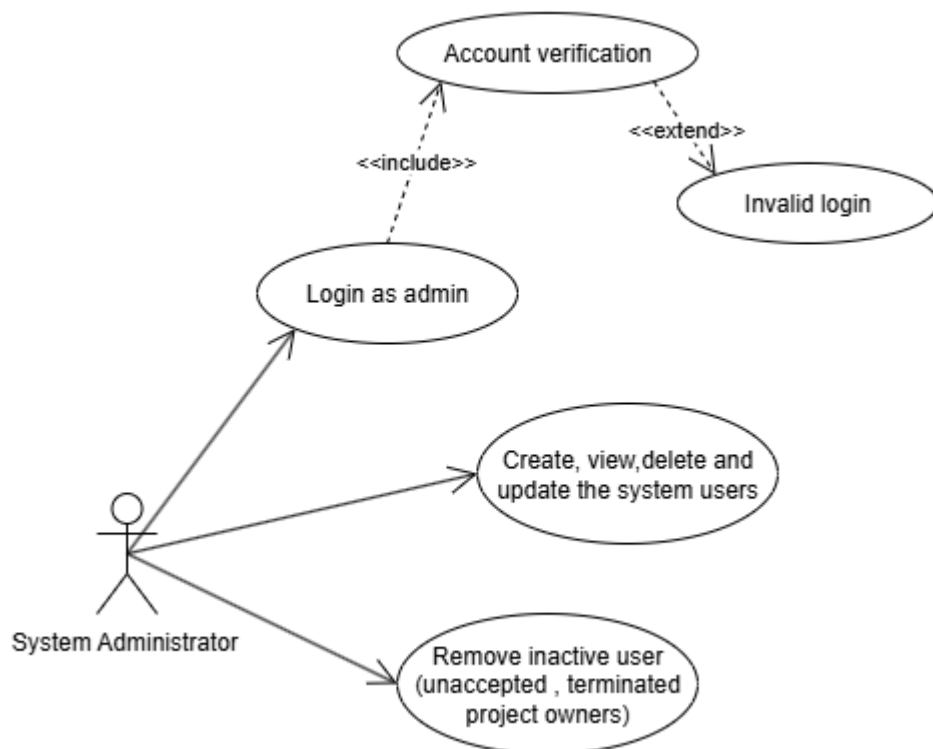
## Financial Officer



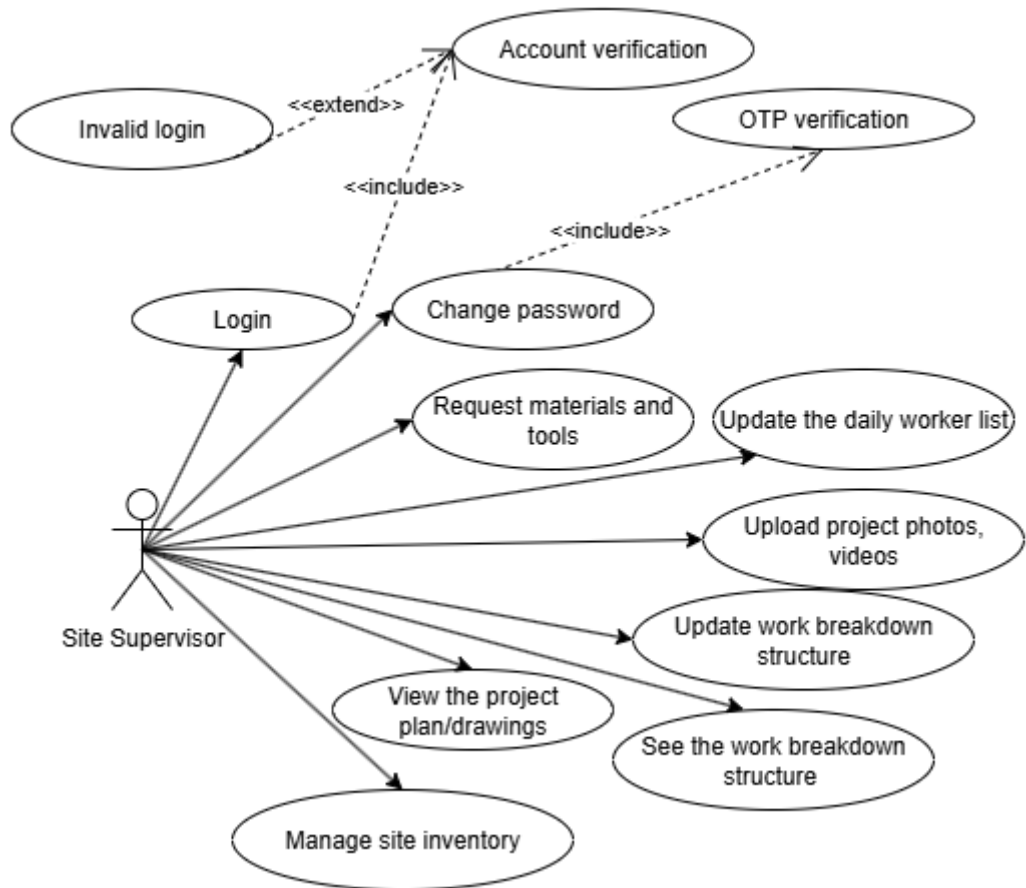
Designer



Admin



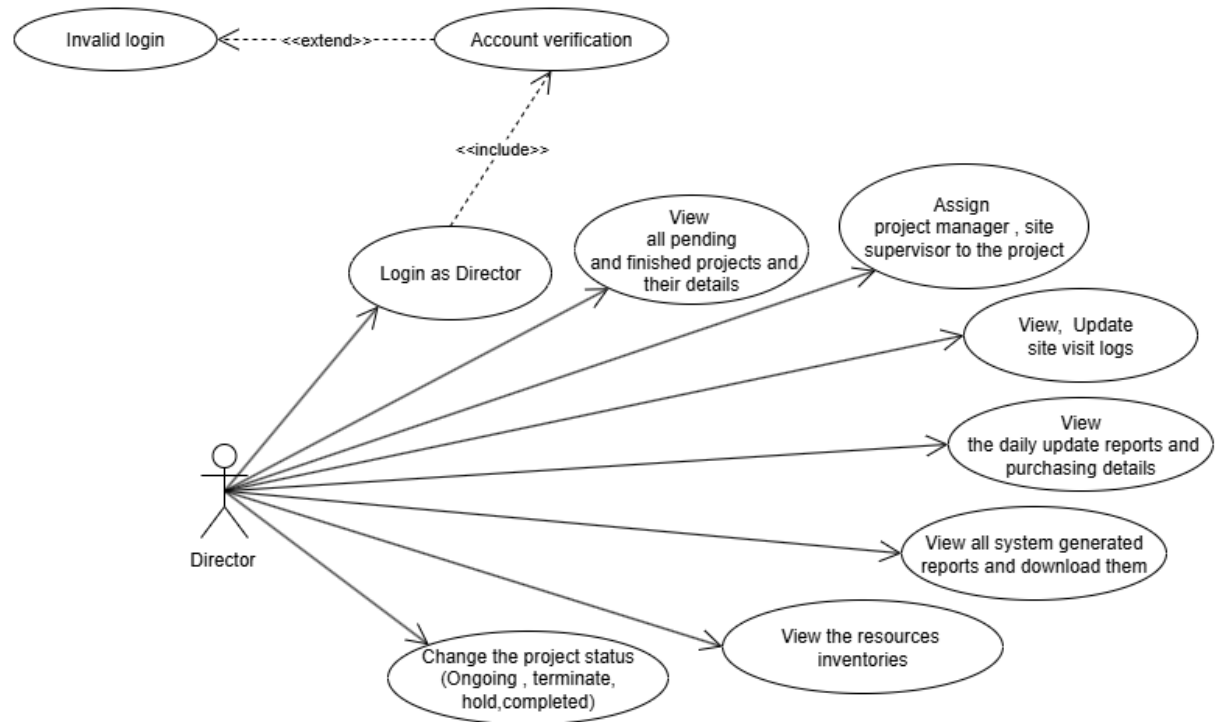
## Site Supervisor



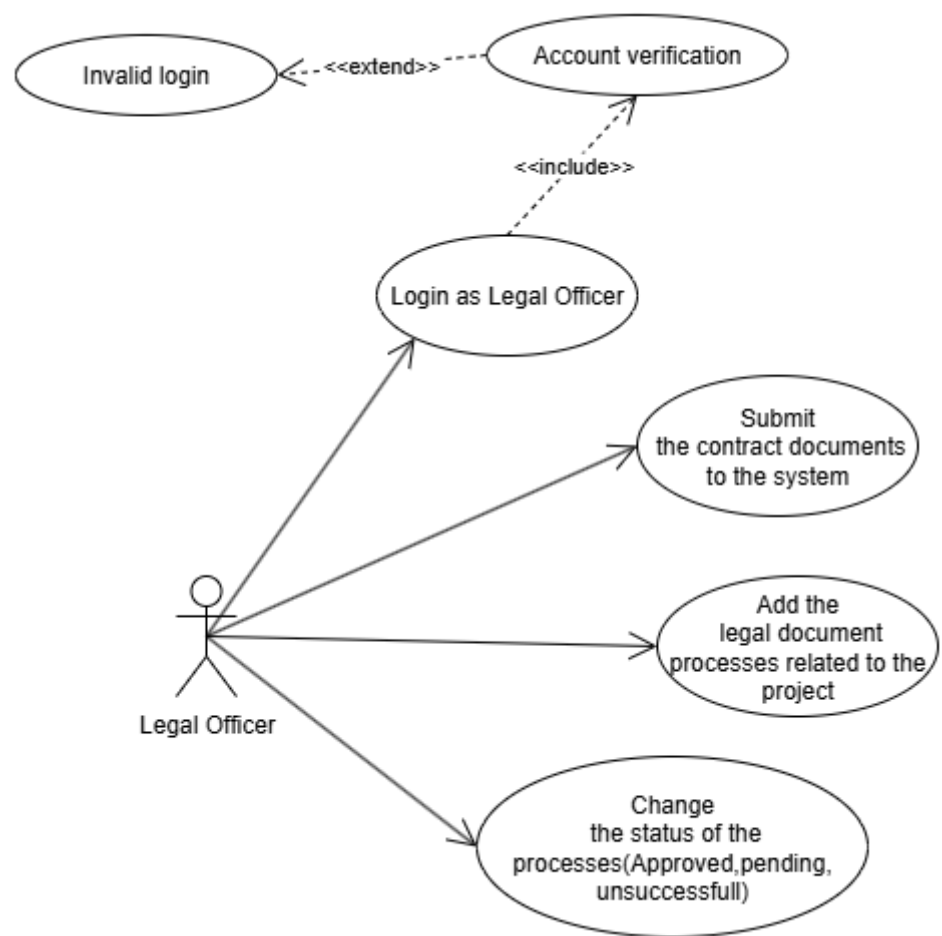
## Project Manager



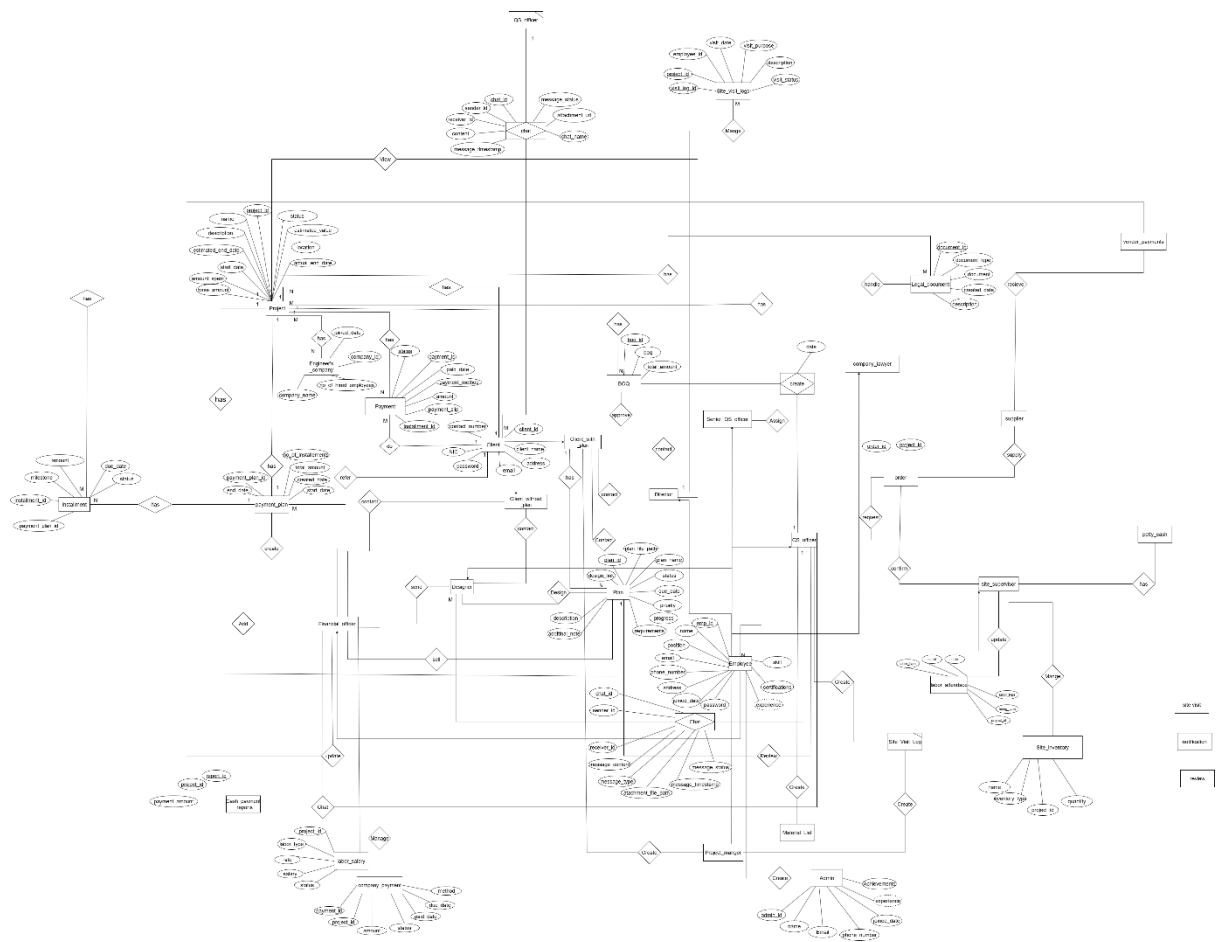
## Director



## Legal Officer

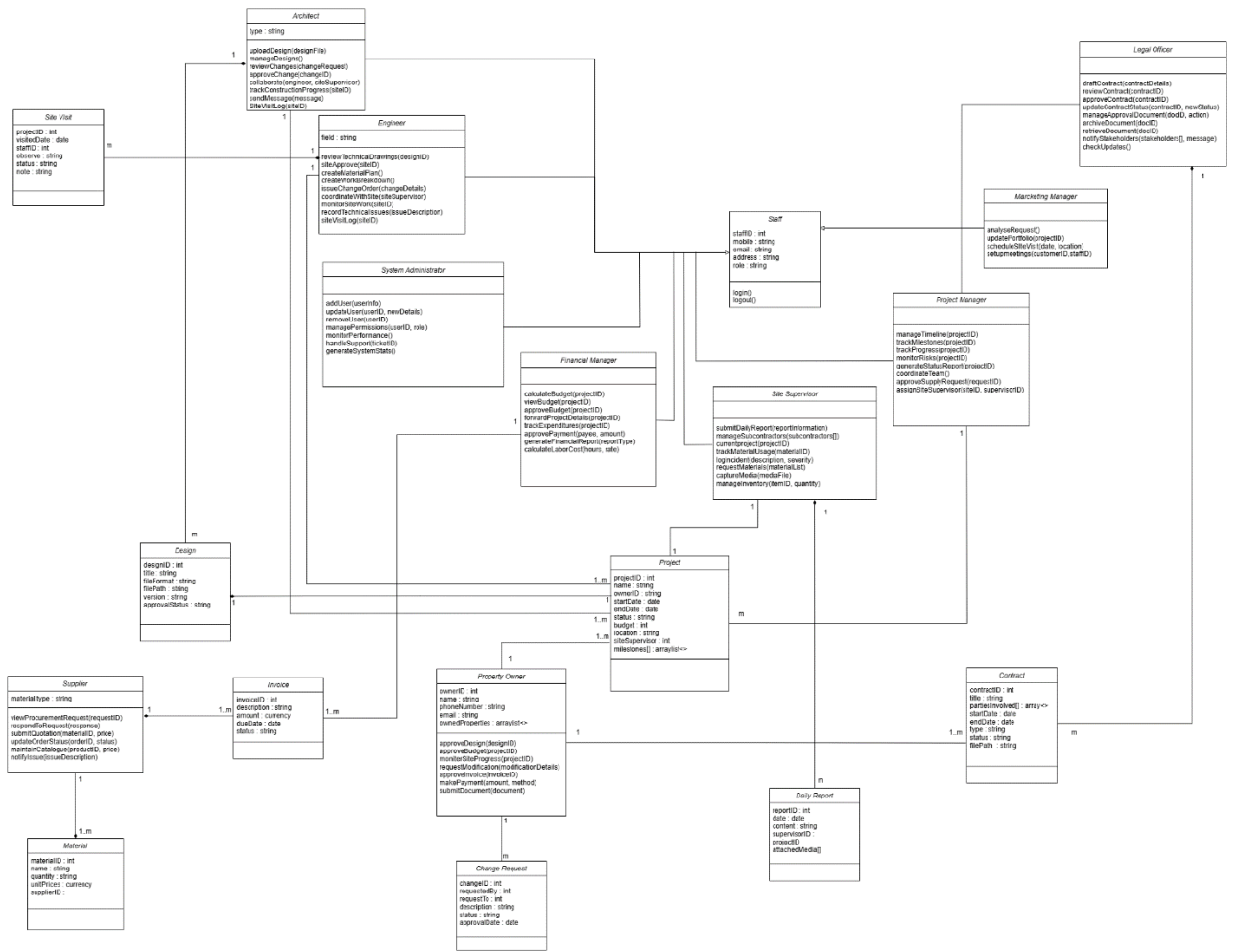


### ER Diagram

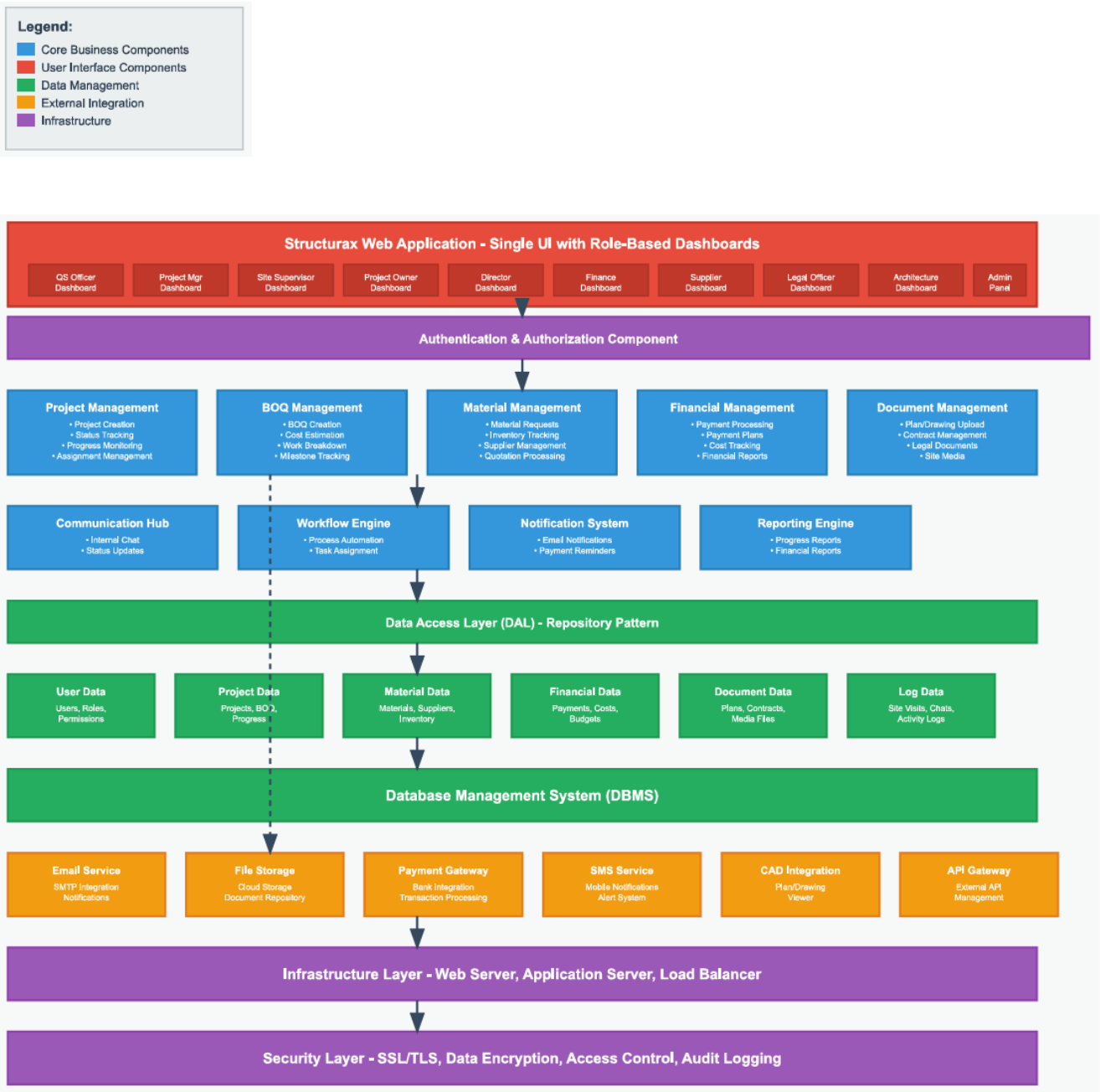




# Class Diagram



# Component Diagram



**Key Features:**

- Single web application with 11 role-based dashboards
- Complete project lifecycle management from initiation to completion
- Integrated communication system with chat and notifications

**Architecture Pattern:**

- Layered Architecture (N-Tier)
- Repository Pattern for data access
- Service-oriented component design

## **User Interfaces**

### **Introduction to User Interface Design**

**Purpose of the UI :** The primary goal of our user interface is to provide a clear, intuitive, and role-specific experience to a diverse set of users, including Architects, QSs, Project Managers, Suppliers, and more. The system ensures ease of use, responsiveness across devices, and accessibility so that users at all levels can navigate and perform their tasks with minimal friction.

**Design Philosophy :** We adopted a user-centered and minimalist design approach, ensuring clean layouts and clear content hierarchy. The UI is mobile-first, supporting desktop, tablet, and mobile resolutions seamlessly.

**Tools & Frameworks Used :**

- Frontend Framework: React.js
- Styling Frameworks: Tailwind CSS
- Design Prototyping: Figma

### **Design Decisions**

**Color Scheme :** We selected a warm and energetic palette to reflect professionalism and vibrancy:

- #E05F00 (Burnt Orange): Used for primary actions and highlights.
- #FAAD00 (Golden Yellow): Used for alerts, notifications, and hover states
- #FFC746 / #FFE169 (Soft Yellows): Used for background shades and contrast in UI sections.

**Layout:** The design features a responsive grid layout, with adaptive panels and collapsible menus on mobile. Critical sections like dashboards are built to prioritize quick scanning and action.

**Navigation:**

- Sidebar for desktop, collapsible on mobile
- Top bar with role-based shortcuts, notifications, and user profile access.\

**Icons and Buttons:** All buttons maintain consistency in size, spacing, color, and hover feedback. Icons are used to support actions visually, such as upload/download, edit, delete, and alert.

**Accessibility Features:**

- Sufficient contrast between text and background.
- Keyboard-navigable components.

**Target Users and Personalization**

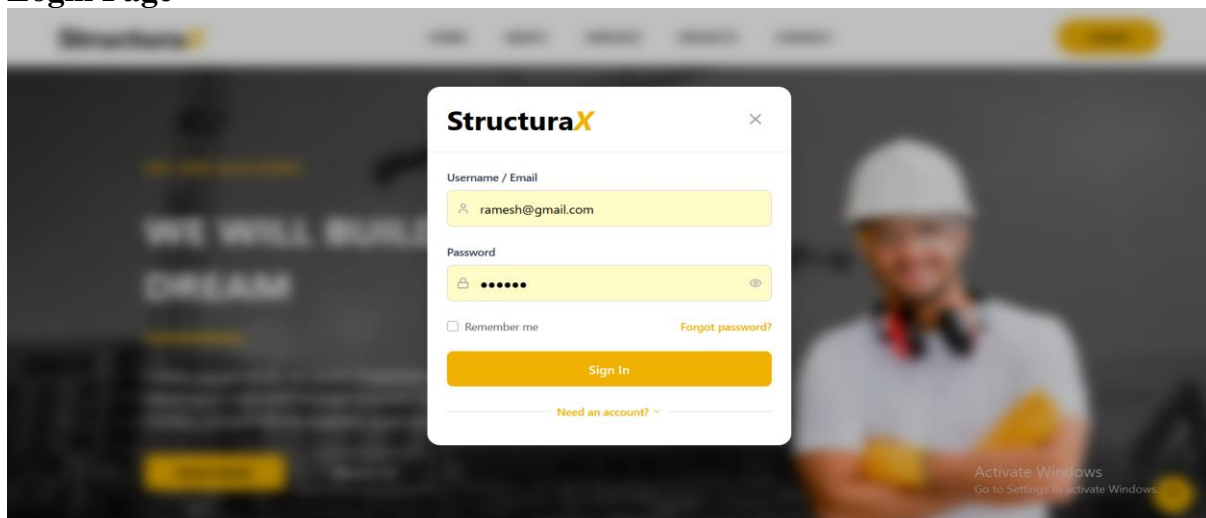
The interface is personalized based on the user's role:

- Architects & Qs: Access to project plans, BOQs, revisions.
- Project Managers: Dashboard , updates, reports
- Suppliers: Quotation submission, material delivery schedules.
- Legal Officers: Contract views, approval workflows.
- System Admins: Role management, audit logs.

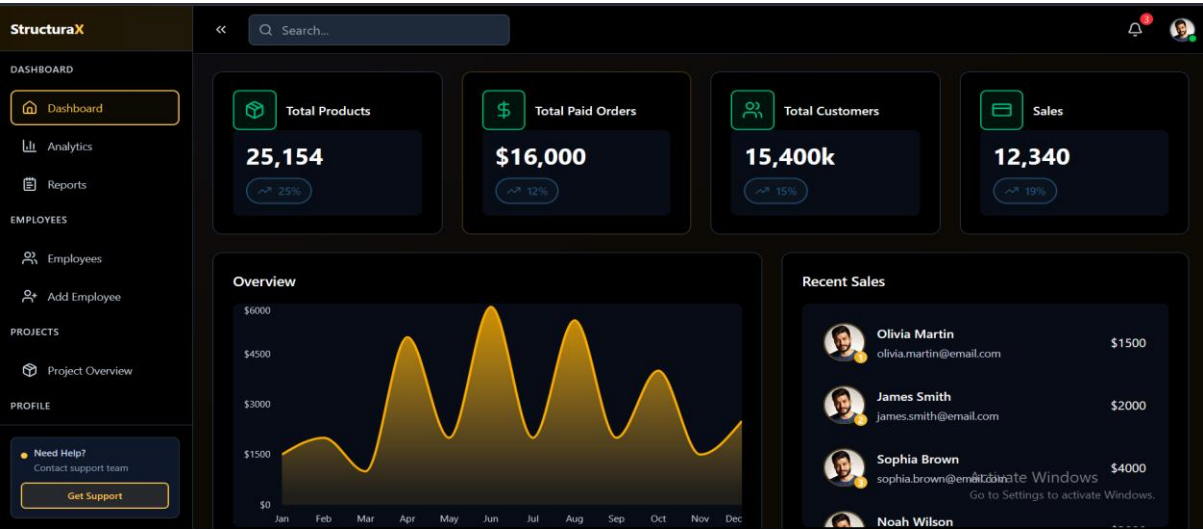
Each user sees only relevant modules and can personalize dashboards

**Sample Screens with Descriptions**

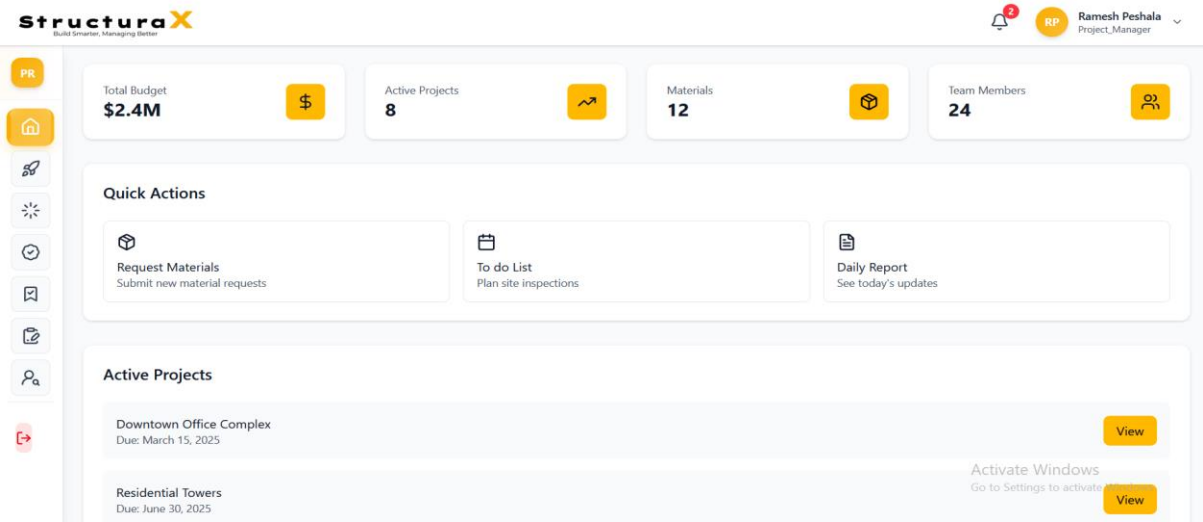
### **Login Page**



# Admin Dashboard



# Project Manager View



Legal Officer

structuraX

Build Smarter. Managing Better

2

RP

Ramesh Peshala

Legal Officer

Projects Assigned for Legal Processing

Q Search projects...

Downtown Commercial Complex

active

Mixed-use development project requiring comprehensive legal documentation

Due: 6/30/2024

Legal Processes2 total

1 completed

1 pending

Residential Tower Phase II

pending

Second phase of residential development with zoning considerations

Due: 8/15/2024

Legal Processes1 total

0 completed

0 pending

1 unsuccessful

Industrial Park Expansion

active

Expansion of existing industrial facilities with environmental clearances

Due: 7/20/2024

Legal Processes0 total

0 completed

0 pending

Activate Windows

Go to Settings to activate Windows.

Designer

structuraX

Build Smarter. Managing Better

2

JD

John Doe

Designer

DE

15

Ongoing Projects

Active projects in development

80% average progress

48

Completed Projects

Successfully delivered projects

100% completion rate

Q Search projects by name, client, or project ID...

Filter Projects

Project Portfolio

6 projects in your workspace

Activate Windows

Go to Settings to activate Windows.

QS

StructuraX

2

M

Malith

QSOffer

QS

Welcome, Quantity Survey Officer!

Manage BOQs, track project costs, and handle material requests here.

Project Count

8

Pending Tasks

12

Upcoming Milestones

4

Total Budget

Rs 45M

Monday

7

July 2025

Quick Links

Projects

BOQs

Material Requests

Site Visits

Monthly Cost Analysis

Actual vs Budget Comparison

Hide Budget

2024

TOTAL COST

Rs. 1.4M

AVG MONTHLY

Rs. 116K

HIGHEST

Rs. 150K

BUDGET VAR

-1.6%

Payment Status Overview

4

3

2

Activate Windows

Go to Settings to activate Windows.

Senior QS

StructuraX

2

M

Malith

SeniorQSOffer

SE

Show Customer Registration

Assign QS Officers to Projects

Projects Awaiting QS Assignment

Project Name	Type	Client	Priority	Action
Shopping Mall Complex	Commercial	ABC Corp	High	Assign QS
Highway Extension	Infrastructure	Government	Medium	Assign QS
Residential Tower	Residential	XYZ Developers	High	Assign QS

Available QS Officers

John Smith

Commercial Buildings

2 slots

Sarah Wilson

Infrastructure

1 slots

Mike Johnson

Residential

3 slots

Emily Davis

Industrial

1 slots

Project Count

8

Pending Tasks

12

Upcoming Milestones

4

Total Budget

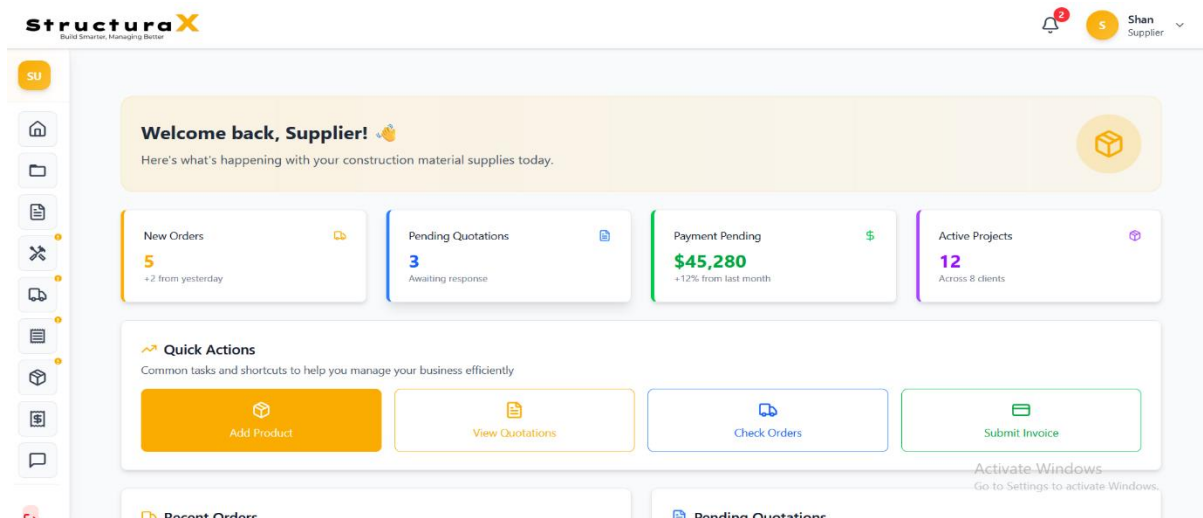
Rs 45M

Monday

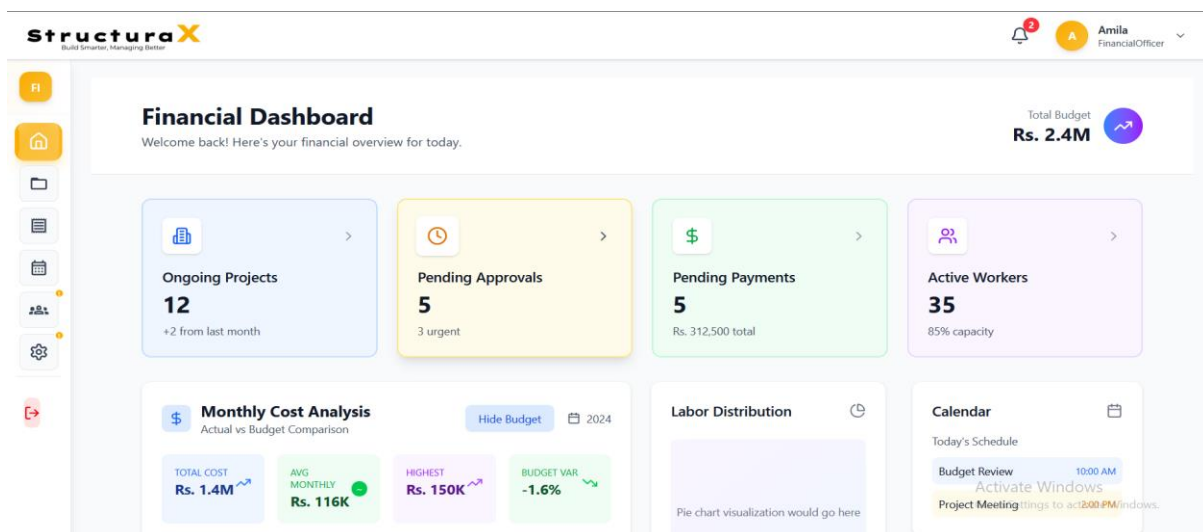
7

July 2025

## Supplier



## Financial officer



## Color and Typography Justification

- Warm tones (orange/yellow) support visibility, urgency, and friendliness.
- Buttons & alerts stand out with #E05F00 and #FAAD00.

## Navigation and Accessibility

- Fixed sidebar improves orientation for users working with large data modules.
- Mobile users benefit from a collapsible navigation bar



## Design Challenges

Challenges:

- Designing dashboards that remain uncluttered despite complex role-based features
- Ensuring contrast ratios are sufficient across all modules.

## Main deliverables of the system

- **Complete Working Software:** Fully functional web-based construction management system with all specified features
- **Source Code:** Complete, well-documented source code with proper version control history
- **Software Requirement Specification (SRS):** Comprehensive documentation of system requirements, use cases, and technical specifications
- **User Manual:** Detailed guides for each user role explaining system functionality and workflows
- **Administrator Manual:** Complete deployment instructions, system configuration guide, and maintenance procedures
- **API Documentation:** Complete documentation of all system APIs for future integrations
- **Database Schema Documentation:** Detailed database design and relationship documentation
- **Software License:** Open-source license (MIT License) for community contribution and transparency
- **Test Reports:** Comprehensive testing documentation including unit tests, integration tests, and user acceptance testing results
- **Project Presentation:** Final demonstration and presentation materials

## **The Project Plan**

### **Work Breakdown Structure:**

#### **Phase 1: Project Initiation & Analysis (Weeks 1-2)**

- Requirements Analysis & Planning (75 hours)
  - Stakeholder interviews with construction industry professionals
  - Functional and non-functional requirements documentation
  - User story creation and acceptance criteria definition
  - Technology stack finalization and development environment setup

#### **Phase 2: System Design & Architecture (Weeks 3-4)**

- System Design & Architecture (150 hours)
  - High-level system architecture design
  - Database schema design and entity relationship modeling
  - API design and endpoint specification
  - UI/UX wireframes and mockups for web and mobile interfaces
  - Security architecture and authentication flow design

#### **Phase 3: Foundation Development (Weeks 5-6)**

- Database Design & Implementation (112 hours)
  - MySQL database setup and configuration
  - Table creation with proper indexing and constraints
  - Stored procedures and triggers implementation
  - Database testing and optimization

#### **Phase 4: Backend Development (Weeks 7-12)**

- Backend Development (450 hours)
  - Spring Boot application setup and configuration
  - RESTful API development for all modules
  - Authentication and authorization implementation
  - Business logic implementation for construction management features
  - SMS gateway integration for notifications
  - File upload and document management system
  - Reporting and analytics module development

### **Phase 5: Frontend Development (Weeks 8-13)**

- Frontend Development (375 hours)
  - React.js application setup and component structure
  - User interface implementation for all modules
  - Dashboard and analytics visualization
  - Form handling and validation
  - API integration and state management
  - Responsive design implementation
  - Mobile-specific features (camera, GPS, offline functionality)

### **Phase 6: Integration & Quality Assurance (Weeks 14-16)**

- Integration & Testing (225 hours)
  - Frontend-backend integration testing
  - Mobile-web synchronization testing
  - User acceptance testing with construction industry stakeholders
  - Performance testing and optimization
  - Security testing and vulnerability assessment
  - Cross-browser and cross-device compatibility testing

### **Phase 7: Documentation & Deployment (Weeks 17-18)**

- Documentation (75 hours)
  - Technical documentation and API documentation
  - User manuals and training materials
  - Installation and deployment guides
  - System maintenance documentation
- Deployment & Final Testing (38 hours)
  - Production environment setup
  - Application deployment and configuration
  - Final system testing in production environment
  - Performance monitoring setup

### **Phase 8: Project Closure (Weeks 19-20)**

- Final project presentation preparation
- Stakeholder demonstration and feedback collection
- Project handover and knowledge transfer
- Post-deployment support planning

## **Reference**

### **Industry Consultation**

**MN Home Developers PVT LTD** - Primary industry consultation partner with 20 years of experience in the construction field. Provided insights into:

- Current construction project management challenges and pain points
- Industry-specific workflow requirements and business processes
- User requirements for both web and mobile platforms
- Real-world construction site operational needs
- Financial management and reporting requirements in construction projects

## **Appendix**

### **Technology Justification:**

#### **Frontend Framework - React.js**

React.js is chosen for its component-based architecture that enables the development of reusable UI components essential for a construction management system with 11 different user roles requiring distinct interfaces. The virtual DOM ensures optimal performance for real-time updates crucial in construction project monitoring. React's extensive ecosystem, strong community support, and compatibility with RESTful APIs make it ideal for building scalable, maintainable frontend applications. Compared to traditional HTML/CSS/JavaScript approaches, React significantly reduces development time while providing better code organization and state management capabilities.

#### **Backend Framework - Spring Boot**

Spring Boot is selected for its enterprise-grade capabilities and robust architecture suitable for construction management systems handling sensitive project data and financial information. Its built-in security features, dependency injection, and database integration capabilities provide a solid foundation for developing secure, scalable APIs. The framework's strong typing through Java reduces runtime errors and improves code maintainability. Compared to other backend frameworks like Express.js or Laravel, Spring Boot offers superior modularity, extensive documentation, and better long-term maintainability, which is crucial for enterprise-level construction management applications.

## Database Management System – MySQL

MySQL is chosen for its reliability, maturity, and excellent support for structured data storage with complex relationships. Construction management systems require robust data integrity and consistency for managing project hierarchies, user relationships, financial transactions, and document associations. MySQL's ACID compliance ensures data consistency crucial for financial tracking and legal document management. Its support for complex SQL queries and joins is essential for generating comprehensive reports and analytics. Compared to NoSQL alternatives like MongoDB, MySQL is preferable for systems with well-defined schemas and complex relational data requirements.

## Version Control – GitHub

GitHub provides distributed version control essential for team collaboration, code backup, and release management. Its integration capabilities with deployment pipelines and project management tools streamline the development workflow.

## Project Scope Justification for 3 Credits per Member:

Despite utilizing established frameworks and libraries, this construction management project requires substantial custom development work that adequately justifies 3 credits per team member:

**1. Complex Multi-Role System Architecture (25% of development effort):** Implementing 11 distinct user roles with specialized functionalities, permissions, and workflows requires extensive custom business logic development beyond framework capabilities. Each role requires unique interfaces, data access patterns, and workflow implementations.

**2. Advanced Construction-Specific Business Logic (25% of development effort):**

Developing industry-specific features such as project lifecycle management, approval workflows, material tracking, budget calculations, and progress monitoring requires significant custom algorithm development and complex database design.

**3. Real-time Collaboration Features (20% of development effort):** Implementing live project updates, real-time notifications, chat systems, and collaborative document editing requires custom WebSocket integration, state synchronization, and conflict resolution mechanisms.

**4. Financial Management Integration (10% of development effort):** Developing comprehensive budget tracking, cost analysis, payment approval workflows, and financial reporting systems requires custom financial algorithms and complex database queries.

**5. Document Management and Version Control (8% of development effort):** Implementing secure document storage, version control, approval workflows, and access control systems requires custom file handling and workflow management logic.

**6. Advanced Security and Access Control (5% of development effort):** Implementing role-based access control, data encryption, audit trails, and security compliance requires extensive custom security implementations beyond framework defaults.

**7. Reporting and Analytics System (4% of development effort):** Developing comprehensive reporting dashboards with custom charts, data visualization, and analytics requires significant custom development work.

**8. Integration and Testing (3% of development effort):** Comprehensive testing across multiple platforms, user roles, and integration scenarios requires substantial custom test development and validation work.

The frameworks and libraries chosen (React, Spring Boot, MySQL, Flutter) accelerate development of standard functionalities such as basic CRUD operations, authentication, and UI components, allowing the team to focus more time and effort on implementing

complex, construction-specific business logic and advanced features. This approach ensures the project maintains adequate complexity and learning opportunities while delivering a professionally viable solution that addresses real industry challenges.

The estimated 6,624 total man-hours, with each team member contributing approximately 1,104 hours over 16 weeks, demonstrates the substantial scope and complexity required to justify the 3-credit academic requirement per member.

## **Declaration**

*We as members of the project titled **StructuraX** - Smart Construction Project Management System certify that we will carry out this project according to guidelines provided by the coordinators and supervisors of the course as well as we will not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university. To the best of our knowledge and belief, the project work will not contain any material previously published or written by another person or ourselves except where due reference is made in the text of appropriate places.*

<i>Name</i>	<i>Signature</i>
(i) B. S. Madhushika	
(ii) S. R. Gamage	
(iii) M. A. R. Peshala	
(iv) Dilshani Nadeesha Kodithuwakku	
(v) Arulsivam Thagshan	
(vi) Rathnayaka R.A.T.M.S	