SALALE UNIVERSITY

COLLEGE OF NATURAL SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

PROJECT ON

WEB BASED CONSTRUCTION MANAGEMENT SYSTEM IN THE CASE OF SALALE UNIVERSITY

PREPARED BY:

No.	Name	Id	Email	Phone
1	Ephraim Yared	RU1544	ephraimyared@gmail.com	+251973941296
2	Ephrata Kelemu	RU1543	ephratake@gmail.com	+251940468949
3	Wakuma Teshome	RU0036	wakegizaw@gmail.com	+251941344994

Advisor: Mr. Diriba. R

January, 2025

Fiche, Oromia, Ethiopia

DECLARATION

As required by the final year project I documentation, we hereby declare that the project documentation, "Web Based Construction Management System In Case The Of Salale University," submitted to Salale University's department of computer science and college of natural science, is an academic record of our work.

Group Members	Id Number	Signature
1. Ephraim Yared	Ru 154414	______________
2. Ephrata Kelemu	Ru 1543/14	______________
3. Wakuma Teshome	Ru 1543/14	______________

APPROVAL

This certifies that Ephraim Yared, Ephrata Kelemu, and Wakuma Teshome Under Mr. Diriba's supervision, regular Salale University students completed this project paperwork, which is prepared for submission to Salale University's computer science department

Advisor Name	Date	Signature
Mr Diriba R.	___________________	_____________________

ACKNOWLEDGMENT

First and foremost, we give our deepest thanks to the Almighty God for His guidance, strength, and blessings, which have been our foundation throughout the journey of completing the first phase of our group project, titled "Web-Based Construction Management In case of Salale University “, undertaken at Salale University.

we extend our sincere thanks to Mr. Diriba R., our esteemed supervisor, for his invaluable guidance, encouragement, and constructive feedback throughout the project. His expertise and support have been instrumental in shaping our ideas and bringing this project to fruition.

We are also grateful to the faculty and staff of the Department of Computer Science at Salale University for providing us with the necessary resources and a conducive learning environment. Their support has been pivotal in enabling us to achieve our academic and project goals.

Additionally, we extend our sincere thanks to our colleagues, friends, and families for their continuous support, patience, and understanding during this demanding yet rewarding journey.

Finally, we deeply appreciate the cooperation and collaboration of our team members, whose dedication, hard work, and team spirit have been the driving force behind the success of this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

CMS: Construction Management System

CRC: Class Responsibility Collaborator

CPU: Central Processing Unit

DBMS: Database Management System

HTML: Hypertext Markup Language

MP: Mega-pixel

OOSAD: Object-Oriented System Analysis and Design

PDF: Portable Document Format

PHP: Hypertext Preprocessor

RAM: Random Access Memory

ROI: Return on Investment

UC- Use case

UML: Unified Modeling Language

USB: Universal Serial Bus

XAMPP: Cross platform Apache MySQL PHP Perl

CRC: Class Responsibility Collaboration

INTRODUCTION

1.1. Background of the Study

Salale University is located 114 kilometers from Addis Ababa, the Ethiopian capital, near the town of Fiche in northern Ethiopia. One of Ethiopia's more recent universities, Salale University was established in 2015. The institution is still undergoing rapid development because it was only recently established, and many construction projects are currently underway to meet its expanding administrative and academic needs. Since these construction projects will determine the university's foundation and destiny, they must be managed carefully. Construction projects entail intricate procedures that, in order to be completed successfully, call for efficient resource management, communication, and scheduling [1].

Traditionally, construction management at Salale University relies on manual processes and fragmented digital tools, leading to inefficiencies such as delays, miscommunication, and the inability to track project progress in real-time. The current system uses physical documentation, spreadsheets, and basic communication methods like emails and phone calls, all of which lack integration, further complicating the management of construction projects as the university grows [1].

1.2. Description of The Existing System

Traditionally, construction management at Salale University has relied heavily on manual processes and a range of fragmented digital tools, which introduces a host of challenges and inefficiencies. As a result, critical project information is scattered across various formats and locations, making it difficult to access and share when needed.

This lack of an integrated system creates inefficiencies at multiple levels. Delays are common due to the time required to manually update information, compile reports, or locate specific documents.

Key Features of the Existing System:

- ✓ Manual Record-Keeping
- ✓ Lack of Organized web-based system
- ✓ Limited Project Tracking

1.3 STATEMENT OF THE PROBLEM

Salale University, established in 2015, is undergoing rapid development and expansion, with multiple construction projects underway to support its growing academic infrastructure. However, the current construction management system (CMS) relies heavily on manual processes which leads to significant inefficiencies. These inefficiencies include:

- ✓ **Time-Consuming Processes:** Tasks such as updating schedules, generating reports, and managing approvals are slow and inefficient.
- ✓ **Poor Document Management:** Manual processes for managing construction documents lead to errors and complications, resulting in misplaced or outdated information.
- ✓ **Paper-Based Communication:** Disorganized and inefficient communication methods among project stakeholders cause misunderstandings, delays, and misalignment of project goals.
- ✓ **Challenging Record Retrieval:** Searching for specific information in large volumes of physical files is time-intensive.
- ✓ **Ineffective Project Tracking:** The absence of a centralized system for tracking project progress makes it difficult to monitor timelines, resource allocation, and budget expenditures, often resulting in delays and cost overruns.

1.4 OBJECTIVE

1.4.1 General objective

The general objective of this project is to develop web-based construction management system in case of Salale University.

1.4.2 Specific objective

To effectively address the challenges in construction project management at Salale University, the following specific objectives were established. Each objective was designed to ensure a comprehensive approach to improving project efficiency and collaboration.

- To Study problem of the existing system.
- To identify the appropriate model of the system.
- To design, database architecture.
- To develop user friendly user interface.
- To test the proposed system.
- To implement the proposed system.

1.5 METHODOLOGY

1.5.1 Data collection technique

Data collection for the **Web-Based Construction Management System** utilized both primary and secondary research methods to ensure comprehensive understanding and accuracy. Below are some methods for gathering data through both approaches:

1.5.1.1 Primary Data Collection Methods:

A. Interview: Team members interviewed Mr. Firew Balcha, who works in the Construction Department under the Vice President for Administration and Development, in order to obtain background and foundational knowledge regarding the existing construction management processes. This gave information about the services given, typical problems, and particular needs of stakeholders.

B. Observation: Observations was conducted within the construction management processes at Salale University to gain a detailed understanding of current workflows and identify areas needing improvement.

1.5.1.2 Secondary Data Collection Method:

Document Analysis: Relevant project documents, including previous reports, plans, and budgets, was reviewed to extract valuable information. Special attention was given to documents that could inform the system's design and contribute beneficial features to the project.

1.5.2 System Analysis and Design Methodology

We adopted the **Object-Oriented System Analysis and Design (OOSAD)** methodology throughout the entire project life-cycle, encompassing analysis, design, and implementation. This approach offered numerous advantages over traditional structured methods, including re-usability, enhanced quality, improved maintainability, and effective complexity management [2].

By applying the principles of object-oriented development, we will focus on managing and organizing objects that represent various components within our system. This methodology organizing objects that represent various components within our system. This methodology emphasizes the relationships and interactions between these objects, which were crucial for creating a cohesive and efficient construction management system [2].

The implementation of this methodology will be categorized into two distinct phases:

- ✓ **Analysis Phase**: In this phase, we identified the key objects, their attributes, and behaviors relevant to the construction management processes at Salale University. Stakeholder requirements were gathered, and use cases were developed to illustrate the interactions between users and the system.
- ✓ **Design Phase**: This phase focused on the architectural design of the system, including the development of class diagrams, sequence diagrams, and other relevant models. The goal

was to establish a detailed blueprint that outlines how the objects will interact and function within the system, ensuring a robust and scalable solution.

1.5.3 System Development Model

For the development of the **Web-Based Construction Management System** at Salale University, we will adopt the **Agile Development Model**. This approach is well-suited for our project due to its flexibility, iterative nature, and emphasis on collaboration. The Agile model allows us to adapt to changing requirements and deliver functional components in short cycles, ensuring continuous feedback and improvement.

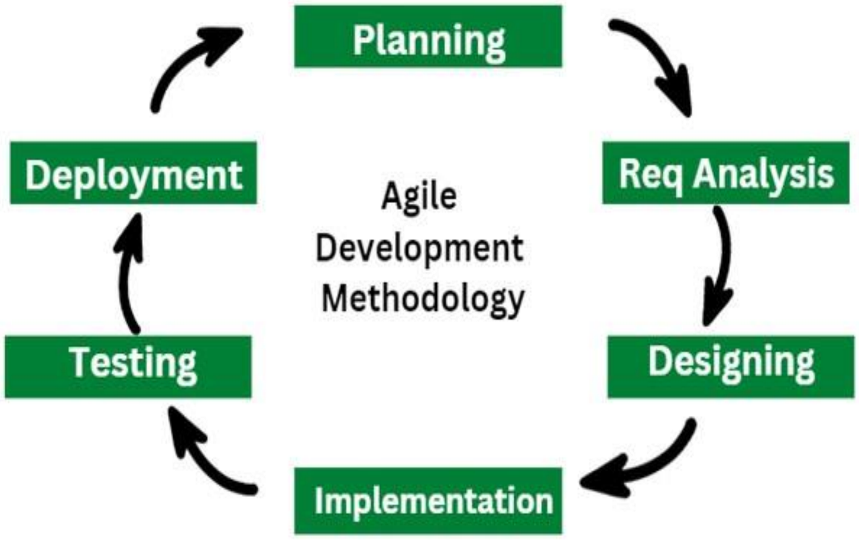


Figure 1.1 Agile Model [3].

1.6 FEASIBILITY OF THE PROJECT

The feasibility of the **Web-Based Construction Management System** for Salale University was evaluated across four key dimensions:

1.6.1 Technical Feasibility

The **Web-Based Construction Management System** is technically viable as it aimed to resolve the technical challenges faced by the current system. The necessary hardware and software components will be easily accessible and cost-effective, promoting a seamless implementation process. This will ensure that the system can be efficiently developed and maintained to meet the requirements of Salale University.

1.6.2 Economic Feasibility

A cost-benefit analysis was conducted, comparing system development costs (server, hosting, development) with expected savings (reduced paperwork, fewer delays, faster approvals). The project provides a positive return on investment.

1.6.3 Operational Feasibility

The system was developed with input from project managers and staff to ensure user acceptance. Training and ongoing support was provided to facilitate smooth adoption and effective use.

1.6.4 Legal Feasibility

The project complies with relevant regulations, including data protection laws, and ensure that any third-party software used meets licensing agreements.

1.6.5 Schedule feasibility

The project is implemented and configured within the scheduled timeline, demonstrating schedule feasibility and effective time management. The development team adhere to the project timeline and allocate resources efficiently to ensured timely completion.

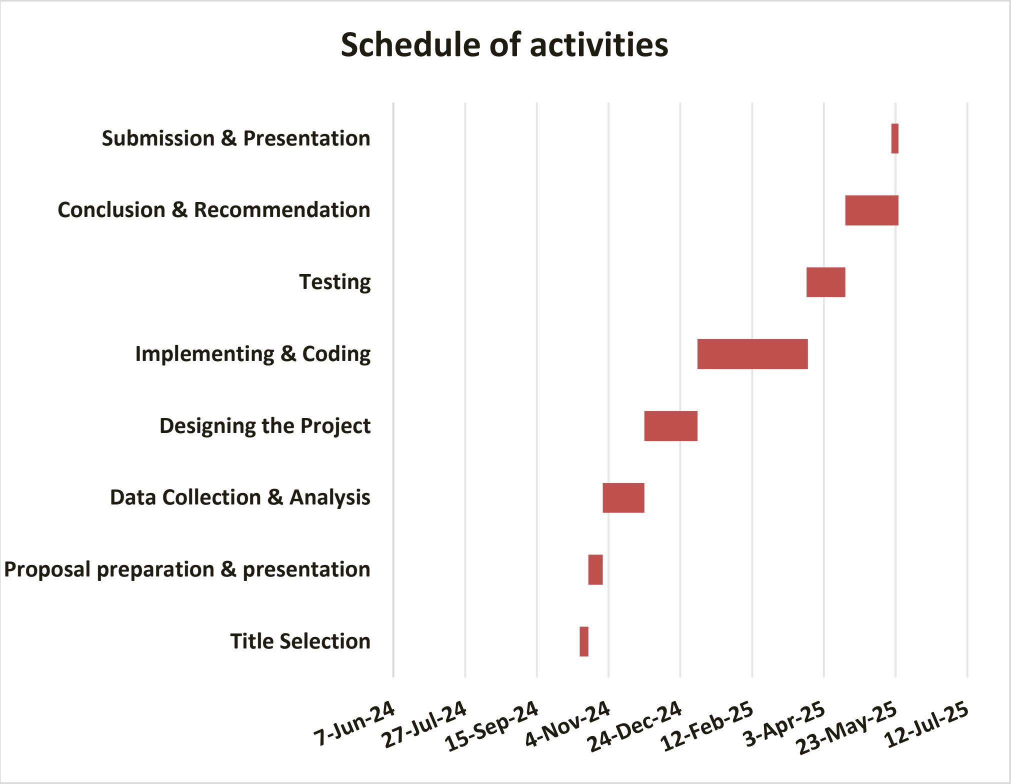


Figure 1.2 Gantt chart

Generally, the **Web-Based Construction Management System** is feasible across all evaluated dimensions, positioning it for successful implementation and enhanced project management at Salale University.

1.7 SCOPE AND LIMITATION OF THE PROJECT

1.7.1.Project scope

The scope of this project focuses on developing a Web-Based Construction Management System specifically designed for Salale University, . The primary goal is to address the university’s unique needs in managing construction projects, ensuring efficient collaboration among stakeholders, streamlining project tracking, and enhancing data accessibility and accuracy. This system will facilitate real-time updates, document management, and progress tracking to improve decision-making and resource allocation for university construction activities.

Our project will cover the following activities

- ✓ Provide a unified platform to store, manage, and access all project-related data.
- ✓ Assign tasks to specific team members, track their progress, and monitor task completion timelines.
- ✓ Generate detailed reports on projects.
- ✓ Enable users to view project tasks, schedules, and resource allocations based on their roles.

This scope outlines a tailored solution for Salale University that enhances construction management through modernized, digital processes, effectively replacing outdated manual methods with a reliable, accessible system.

1.7.2 Limitation of our Project

A web-based construction management system designed in case of Salale University can bring many benefits, but like any system, it also comes with limitations. Here are some common challenges and limitations specific to this context:

- Our project supports only English language
- Our project only applies to the constructions in salale University not landscapes or others.
- Works only for projects that has already won the bid

1.8 SIGNIFICANCE OF THE PROJECT

The Web-Based Construction Management System for Salale University holds significant importance for various stakeholders, contributing to the improvement of construction project management within the institution and beyond. The significance can be outlined as follows:

- ➢ **Enhanced Efficiency:** Automating scheduling, reporting, and approval workflows speeds up these tasks and ensuring timely completion of projects.

- **Improved Document Management:** Implementing streamlined, automated processes for managing construction documents minimizes errors, ensuring that information is accurate and up-to-date, reducing miscommunication and lost information.

- **Efficient Record Retrieval:** A centralized, searchable digital system enables instant access to required information, saving time and ensuring faster decision-making.

- **Optimized Project Tracking:** A centralized digital tracking system provides real-time updates on timelines, resources, and enabling proactive management.

Generally, the proposed system is used to:

- Save the time that is lost while recording project information manually
- Handle construction information in better way
- Facilitate better and fast process scheduling

1.9 ORGANIZATION OF THE PROJECT

Our Project is Grouped into six chapters, which is arranged accordingly in terms of each chapter the first is the introductory part the second covers the requirement analysis section and chapter three discusses the project design phase which includes the front-end user interface and database design on the back-end which explains the object-oriented graphical presentation. Chapter four explains about the implementation, testing and maintenance. Chapter five talks about result, and discussion lastly chapter six conclusion and recommendation.

CHAPTER TWO

REQUIREMENT SPECIFICATION AND OBJECT-ORIENTED ANALYSIS

This chapter provides an overview of the current system, its purpose, and the key features of the proposed system. It also outlines the necessary prerequisites for the new system, identifies the essential behaviors to retain from the existing system, and presents a proposed solution that addresses the challenges and inefficiencies of the current system.

2.1. Existing system

2.1.1. Existing System Description

The current construction management practices at Salale University predominantly rely on manual processes, resulting in a number of operational inefficiencies. Critical project data is not digitized, which makes it challenging to access and share information efficiently when needed. This lack of digitization creates several issues, such as delays and errors, particularly in tasks like updating information and compiling reports.The absence of a unified system contributes to inefficiencies in project management, making it difficult for involved parties to track progress, manage timelines, and ensure accurate reporting. The manual handling of records, combined with the lack of real-time updates, leads to delays and complications, affecting the timely completion of construction projects.

The following figures illustrate the current construction management system at Salale University, which is manual and paper-based. [4]

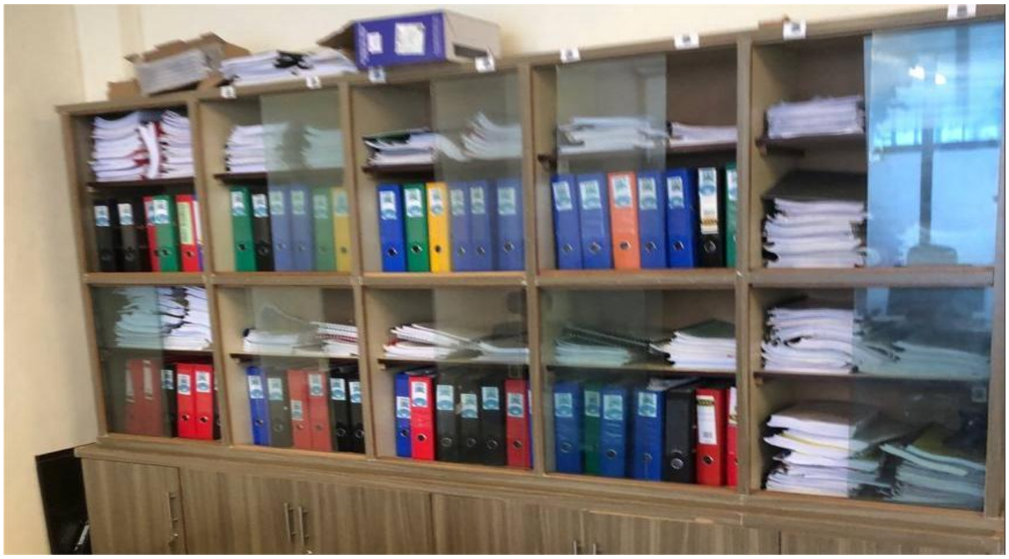
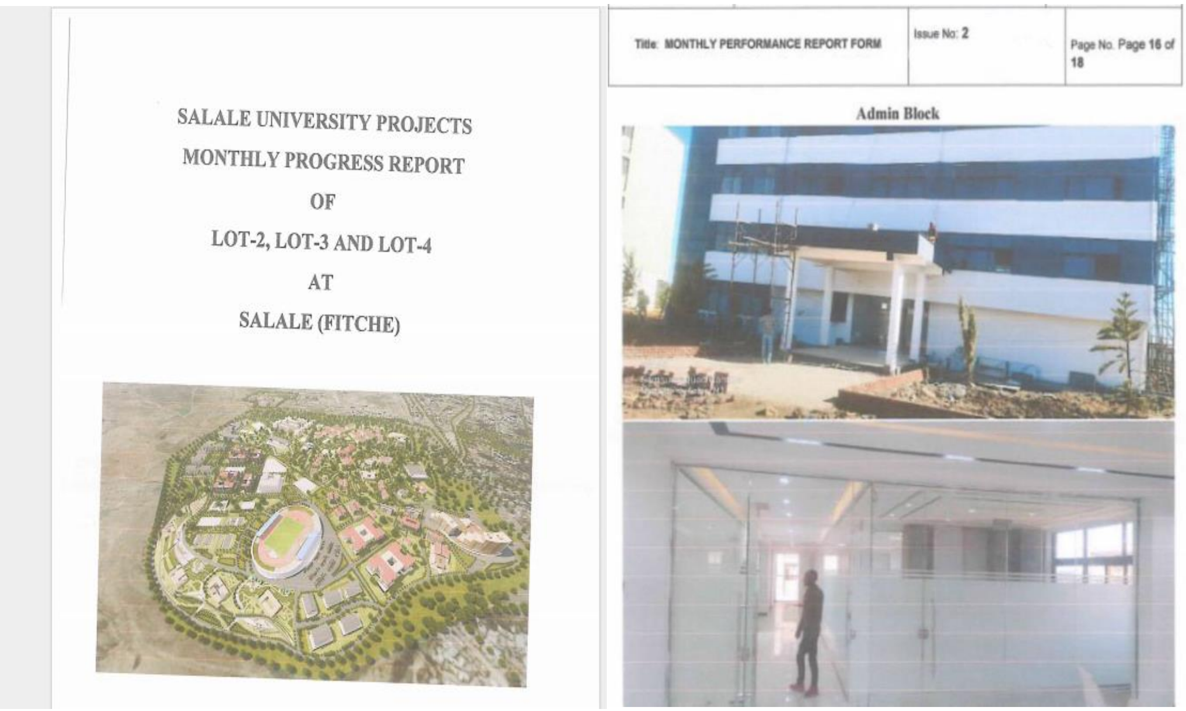


Figure 2. 1 Existing system [4]



SALALE UNIVERSITY PROJECTS
MONTHLY PROGRESS REPORT
OF
LOT-2, LOT-3 AND LOT-4
AT
SALALE (FITCHE)

Title: MONTHLY PERFORMANCE REPORT FORM	Issue No: 2	Page No. Page 16 of 18

Admin Block

Figure 2. 2 Existing system report [4]

2.2. New System

The transition to a web-based construction management system will address issues such as the inefficiency of accessing and sharing project data, delays in project updates, lack of centralized tracking, and difficulties in managing project timelines and resources. The proposed system includes various functionalities and will use multiple UML diagrams to provide a clear understanding of the system's internal workings for project managers, stakeholders, contractors, and other relevant parties.

2.2.1. New System Business Rules

BR1: All actors of the system must log in with valid credentials to access the system.

BR2: Project managers, Admins, contractors, and other authorized users must be able to interact with the system to access the information and features that are permitted for them.

BR3: An admin must log in with a verified username and password (with optional two-factor authentication) to create, modify, or delete user accounts and assign role-based permissions.

BR4: A project manager must manage projects and assign tasks to contractors before they can proceed with the project work.

BR5: A system Admin must review and approve any updates or changes made to project details by authorized users.

BR6: Project and contractor data must be kept confidential and should only be used for managing construction projects at Salale University.

BR7: All project-related inquiries and requests should be submitted through web-based form to ensure proper documentation and tracking.

BR8: The project manager and other actors are required to log in to view project schedules, timelines and to do other things

BR9: Only authorized users, such as project managers or Admins, can submit issues related to project delays or other concerns.

BR10: Contractors can assign tasks to an Employee

BR11: Employee can view assigned tasks to them

BR12: Contractors must submit the finished tasks

BR13: Contractors must prepare report on the status of the task

BR14: Project stakeholders (such as the Consultant, project manager) can view construction projects.

2.2.2. Functional Requirement

The **functional requirements** define the necessary functions that the Web-Based Construction Management System for Salale University is expected to perform. These requirements describe what the system should do to achieve the goal of efficient and effective project management. Based on the project scope, the following functional requirements are defined:

Account Management

- **Create account**: Authorized users able to create accounts by entering their necessary details, including roles and permissions.
- **Update account**: Users should be able to modify their account information (e.g., personal details, change password) as necessary.
- **Delete account**: Users with appropriate privileges should be able to deactivate or delete accounts.

Project Data Management

- **Register project**: Project managers must be able to register new won construction projects.
- **View project information**: Users should be able to view the status and detailed information of ongoing or completed projects.
- **Update project details**: Project managers can update project details.

Task Management

- **Assign tasks**: Project managers and other eligible actors should assign specific tasks to the appropriate users like contractors, site engineers and consultants.

Report Management

- **Generate reports**: Users should be able to generate various reports related to projects.

User Role and Access Control

- Restrict access based on roles likes admins, project managers, contractors, site engineers, consultants and employees.

2.2.3. Non-Functional Requirements

Non-functional requirements define the qualities or characteristics that the system must have in order to operate efficiently and effectively. These are the "how" aspects that describe the system's behavior and performance:

Performance

- It must handle a large volume of data without significant slowdowns, particularly as the number of projects and users increases.

Security

- Authentication and authorization: The system must have secure login procedures and ensure that only authorized users can access specific functionalities based on their roles (e.g., project manager, contractor, admin).

Reliability

- The system should be reliable and able to operate continuously without unexpected outages.
- Error handling: The system should display appropriate error messages and guide users to resolve issues without crashing or losing data.

Usability

- The user interface must be intuitive and easy to navigate for users with varying levels of technical expertise.
- The system should offer clear navigation paths and user-friendly forms for entering and managing project data, tasks, and reports.

Maintainability

- The system must be easy to maintain, with modular components and clear, well-documented code to allow for quick updates or fixes.

Scalability

- The system must be scalable, capable of handling increasing numbers of users, projects, and data without performance degradation.
- It should allow for future expansions, such as the addition of new features or handling a larger volume of concurrent users.

2.2.4 Actors Identification

Based on the roles defined for the Web-Based Construction Management System for Salale University, the following actor specifications outline the key roles and their associated actions:

Admin:

- Create new user accounts
- Update user account information
- Deactivate or delete user accounts
- Generate System Reports
- Define and manage roles and their associated permissions
- Approve project tasks submitted by Project Managers

Project Manager:

- Create projects that won the bid
- Manage project tasks
- Assign tasks to contractors, site engineers and consultants.
- Monitor project reports and prepare reports for admin
- Generate progress reports

Contractor:

- Register Employees
- View tasks which are assigned by project manager
- Submit report to project manager
- Manage Employee information
- Assign Tasks to employee

Site Engineer:

- Manage Daily Labor
- Prepare Report

Consultant:

- Track Project Status
- Prepare Report

Employee:

- View assigned tasks

2.2.5 Use case diagram

A **Use Case Diagram** is a type of diagram in the Unified Modeling Language (UML) that visually represents the functional requirements of a system. It illustrates how different users (actors) interact with the system to accomplish specific goals (use cases). This type of diagram is useful for understanding the system's functionality and identifying the roles and responsibilities of various actors [5].

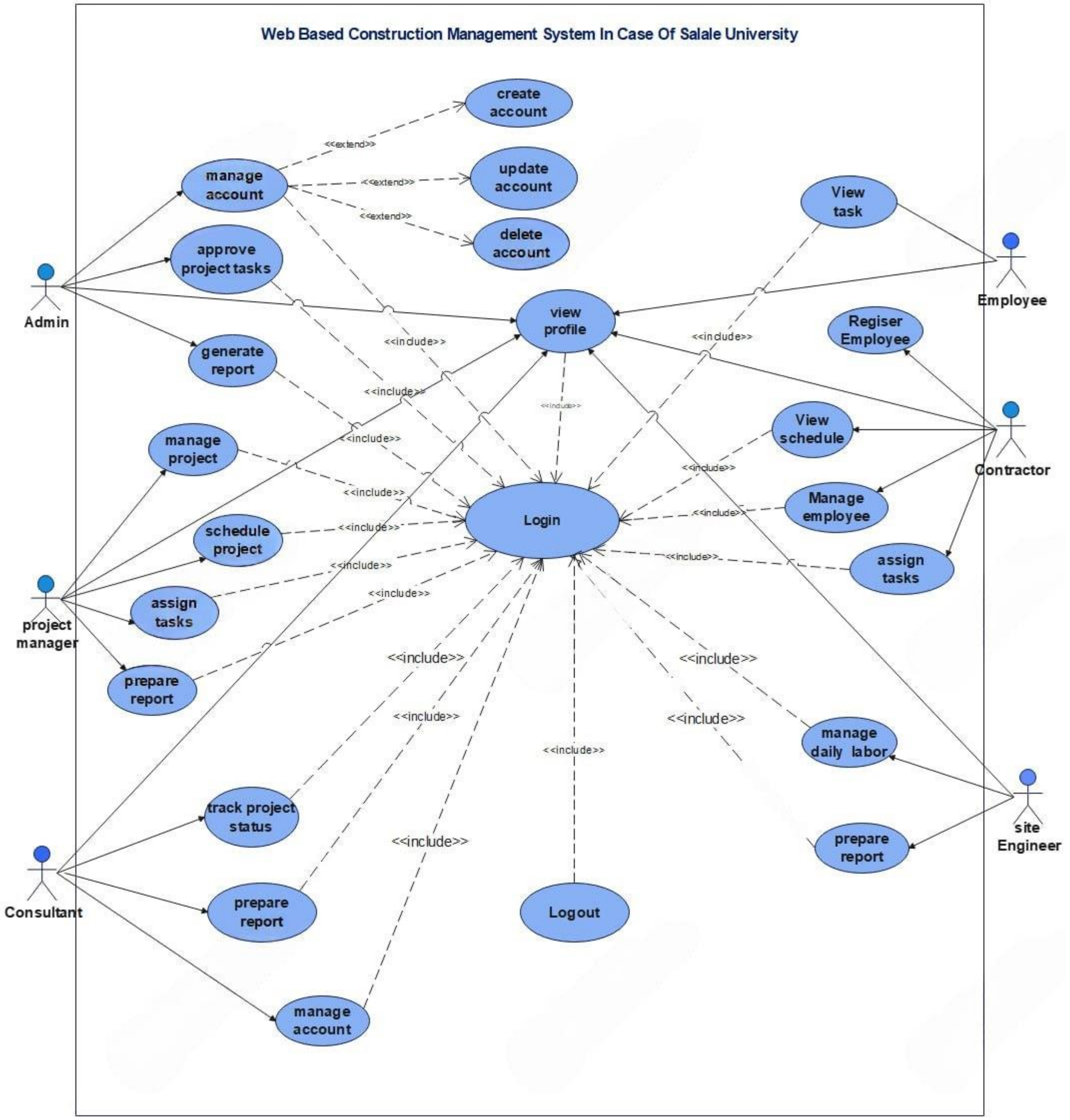


Figure 2. 3 Use case Diagram for the Proposed system

2.2.6 Use Case Description

A use case Description is a document or narrative that describes the interaction between users and system to achieve a specific goal. It is commonly used in software development and requirements engineering to capture the functional requirements of a system. The following table are describing about the use case.

Table 2. 1 Use case description for login

Use case Name	**Login**	
Use Case Id	UC#01	
Descriptions	As long as any users are given access to the system’s functions, users should be able to log in whenever they want to use it.	
Actors	Admin, project manager, contractor, site engineer, consultant and Employee	
Pre conditions	They must have a user account, username and password	
Basic course of Action:	Users Action	System Response
	1. The user Start the system. 3. The user inputs their username & password. 5. The user selects the login option.	2. The system displays Home page of the system. 4. The system verifies if the user’s name is valid or not. 6. The system takes the user to the next interface.
Alternative course of Action	The user is asked to re-enter is valid username and password if he made a mistake	
Post condition	The person who is authenticated gets the appropriate page.	

Table 2. 2 Use case description for Manage account

Use case name	**Manage Account**	
Use Case ID	UC#02	
Actor	Admin, Consultant	
Description	This use case describes the process by which users can manage their account details, such as updating profile information, changing passwords	
Basic course of action	User action	System action
	Step 1: Navigates to the "Manage Account" section via the dashboard.	Step 2: The system displays account management options (e.g., update profile, change password, set preferences).
	Step 3: The user updates personal information (e.g., email, phone number, password).	Step 4: the system validates inputs and saves changes to the database. Displays a confirmation message like "Your profile has been updated successfully".
		Step 5: Use case ends
Alternative course of action	The system verify information is not correctly, the system displays error message as ➢ Invalid ➢ "An error occurred while saving your changes. Please try again later."	
Post condition	User details are updated in the system database.	

Table 2. 3 Use case description for Approve projects

Use case name	**Approve Project**	
Use Case ID	UC#03	
Actor	Admin	
Description	This use case allows authorized users to approve a construction project that has been submitted for review. Approval indicates that the project is ready for execution or further processing.	
Basic course of action	User action	System action
	Step 1: Navigates to the "Manage Account" section via the dashboard.	Step 2: The system displays account management options (e.g., update profile, change password, set preferences).
	Step 3: The user updates personal information (e.g., email, phone number, password).	Step 4: the system validates inputs and saves changes to the database. Displays a confirmation message like "Your profile has been updated successfully”.
		Step 5: Use case ends
Alternative course of action	The system verify information is not correctly, the system displays error message as ➢ Invalid ➢ "An error occurred while saving your changes. Please try again later."	
Post condition	User details are updated in the system database.	

Table 2. 4 Use case description for view submitted projects

Use case Name	**View submitted projects**	
Use Case Id	UC#04	
Descriptions	This use case describes the process of viewing submitted projects by the managers	
Actors	Admin	
Pre conditions	The users must be authenticated and logged into the system, and has the necessary permissions to Approve submitted projects.	
Basic course of Action:	Users Action	System Response
	Step 1: The manager goes too project submission Step 3: The user clicks on a specific project entry to submit	Step 2: Displays a list of submitted projects awaiting approval. Step 4: Loads and displays the submitted projects if there are any
Alternative course of Action	If there are no submitted projects available the system wouldn’t submit any available projects	
Post condition	The admin successfully views submitted project details	

Table 2. 5 Use case description for prepare report

Use case Name	**Prepare Report**	
Use Case Id	UC#05	
Descriptions	This functionality allows authorized users to generate various reports.	
Actors	Project Manager, contractor, Consultant, Site Engineer	
Pre conditions	The users must be authenticated and logged into the system, and has the necessary permissions to generate reports.	
Basic course of Action:	Users Action	System Response
	Step 1: The user navigates to the "Reports" section of the navigation. Step 3: The user selects the desired report type (e.g., Project Progress Report project name, Step 5: The user clicks on the "Generate Report" button.	Step 2: Displays a list of available report types with a selection interface. Step 4: Dynamically loads input fields for the filters and validates inputs as they are entered. Step 6: The report is successfully generated and available for review, export, or sharing.
Alternative course of Action	Selects a report type and applies a filter with no matching data.	
Post condition	The system logs the report generation event for auditing purposes.	

Table 2. 6 Use case description for Schedule project

Use case name	**Schedule project**	
Use Case ID	UC#06	
Actor	Project manager	
Description	Project manager plans and schedules the project activities	
Basic course of action	User action	System action
	Step 1: The Project Manager logs into the system.	Step 3: The system displays the form to enter project details, milestones and deadlines.
	Step 2: The project manager clicks on the "Schedule Project" link.	Step 5: The system saves the scheduled project information.
	Step 4: The project manager enters the required information and selects the desired dates	Step 6: Use case end.
Alternative course of action	The system verify information is not correctly, the system displays error message as ➢ Invalid ➢ Go to step4. ➢ Use case ends.	
Post condition	The project will be scheduled.	

Table 2. 7 Use case description for Assign task

Use case name	**Assign task**	
Use Case ID	UC#07	
Actor	Project manager and contractor	
Description	Project Manager, contractor assigns tasks to contractors.	
Basic course of action	User action	System action
	Step1: The Project Manager and contractor logs into the system.	Step 3: System displays tasks associated with the selected project manager and contractor.
	Step 2: The project manager and contractor clicks on contractor’s and Employee button respectively and selects a Contractor and Employee from the list of available contractors and Employees respectively.	Step 5: The System displays task details with the selected contractor and Employee assigned.
	Step 4: Project Manager and Contractor selects a task and confirms the task assignment.	Step 6: Use case end.
Alternative course of action	The system verify information is not correctly, the system displays error message as: ➢ Not available ➢ The system redirects to step 4 ➢ Use case ends	
Post condition	The task assignment will be generated.	

Table 2. 8 Use case description for generate report

Use case name	**Generate Report**	
Use Case ID	UC#08	
Actor	Project manager	
Description	Project Manager Generates reports	
Basic course of action	User action	System action
	Step 1: The Project Manager logs into the system.	Step 3: System displays a list of available report templates or options.
	Step 2: Project Manager selects the "Generate Report" option.	Step 4: System generates the report based on the selected template from contractor, consultants and site engineers.
	Step 5: the project manager views the generated reports	
Alternative course of action	The system shows if there are no submitted reports	
Post condition	The report will be generated.	

Table 2. 9 Use case description for Submit task

Use case name	**Submit task**	
Use Case ID	UC#09	
Actor	Contractor	
Description	Contractor submits the project activities	
Basic course of action	User action	System action
	Step 1: Contractor logs into their page	Step 3: The system displays to contractor the submit task file form
	Step 2: The contractor select submit task file link	Step 5:The system saves submitted task
	Step 4: contractor can submit the task and click the submit button.	Step 6: Use case end.
Alternative course of action	The system verify information is not correctly, the system displays error message as ➢ Incorrect ➢ Go to step4. ➢ Use case ends.	
Post condition	The task will be submitted.	

Table 2. 10 Use case description view task

Use case name	**View task**
Use Case ID	UC#10
Actor	Employee
Description	The Employee views details of a specific task assigned to them.
Pre-condition	The Employee has been assigned a task within a project.

Basic course of action	User action	System action
	Step 1: Employee logs into the system.	Step 3: The system displays the contractor to select assigned tasks
	Step 2: Employee click on the view task link	Step 5: System displays task details (task description, Assigned to, due date)
	Step 4 Employee select the tasks assigned	Step 6: Use case end
Alternative course of action	The system verify information is not correctly, the system displays error message as ➢ Not available ➢ Go to step 2. ➢ Use case ends.	
Post condition	The Contractor has viewed the details of the selected task.	

Table 2. 11 Use case description for Manage project

Use case name	**Manage project**	
Use Case ID	UC#11	
Actor	Project Manager	
Description	Use case to add /delete/update/ and to view project information consequently.	
Pre condition	The Project manager must be login to perform those activities.	
Basic course of action	User action	System action

	Step 1: Project Manager login to the system	Step 3: The system displays project registration form/enter project Id to delete/enter project Id to edit/enter project Id to view consequently.
	Step 2: Project Manager clicks the “add new project /delete project/edit project /search project” link consequently	Step 5: system displays the data which are added /deleted /updated /search stored in systems database consequently
	Step 4: Project Manager fills project data/enter project Id and save it.	Step 6: Use case end
Alternative course of action	When there is not filled the project data correctly ➢ Please fill project data correctly message will be displayed. ➢ Go to step 2. ➢ Use case ends.	
Post condition	System has successfully add/delete/update/ and display project information to the project manager.	

Table 2. 12 Use case description for view schedule

Use case name	**View schedule**	
Use Case ID	UC#12	
Actor	Contractor	
Description	The Contractor views schedule	
Pre condition	The Contractor must be login to perform those activities	
Basic course of action	User action	System action

	Step 1: Contractor logs into the system.	Step 3: The system displays the schedule information.
	Step 2: Contractor click on the view schedule button	Step 4: Use case end
Post condition	System shows schedule information to contractor.	

Table 2. 13 Use case description for manage Employee

Use case name	**Manage Employee**	
Use Case ID	UC#13	
Actor	Contractor	
Description	use case to add/delete/update/ and to view employee information.	
Pre condition	The Contractor must be login to perform those activities	
Basic course of action	User action	System action
	Step 1: Contractor logs into the system.	Step 3: The system displays the Employee information.
	Step 2: Contractor click on the Manage Employee button	Step 5: The system displays employee registration form/enter employee Id to delete/enter employee Id to edit/enter employee Id to view consequently.
	Step 4: . Contractor clicks the “add new employees/delete employee/edit employee/search employee” button consequently	Step 7: system displays the data which are added /deleted /updated /search stored in systems database consequently
	Step 6: Contractor fills employee data/enter employee Id and save it	Step 8: Use case end

Alternative course of action	The system validates the entered data is not correct, the system displays incorrect entered data message ➢ “The system redirects to go step 4 i.e.to fill the data again. ➢ Go to step 4
Post condition	The Employee will be registered

Table 2. 14 Use case description for Register Employee

Use case name	**Register Employee**	
Use Case ID	UC#14	
Actor	Contractor	
Description	The Contractor can register Employee information into the data base	
Pre condition	The Contractor must be login to perform those activities	
Basic course of action	User action	System action
	Step 1: Contractor logs into his/her page.	Step 3: The system displays the schedule information registration form
	Step 2: Contractor selects the register Employee button	Step 5: The system validates the input data
	Step 4: Contractor fills the required fields.	Step 6: The system displays the successful message.
		Step 7: Use case ends
Alternative course of action	The system validates the entered data is not correct, the system displays	

	➢ Incorrect entered data message. ➢to go step 4 to fill the data again. ➢ Use case ends
Post condition	The Employee will be registered

Use case name	**View project status**	
Use Case ID	UC#15	
Actor	Consultant	
Description	The consultant views project status	
Pre condition	The consultant must be login to perform those activities	
Basic course of action	User action	System action
	Step 1: consultant logs into the system.	Step 3: The system displays the list of available projects
	Step 2: consultant click on the view project status button	Step 4: Use case end
Post condition	System shows schedule information to contractor.	

2.2.6 Sequence Diagram

A Sequence Diagram is a key component of Unified Modelling Language (UML) used to visualize the interaction between objects in a sequential order. It focuses on how objects communicate with each other over time, making it an essential tool for modeling dynamic behavior in a system. Sequence diagrams illustrate object interactions, message flows, and the sequence of operations, making them valuable for understanding use cases, designing system architecture, and documenting complex processes [6].

The following diagram shows our sequence diagram.

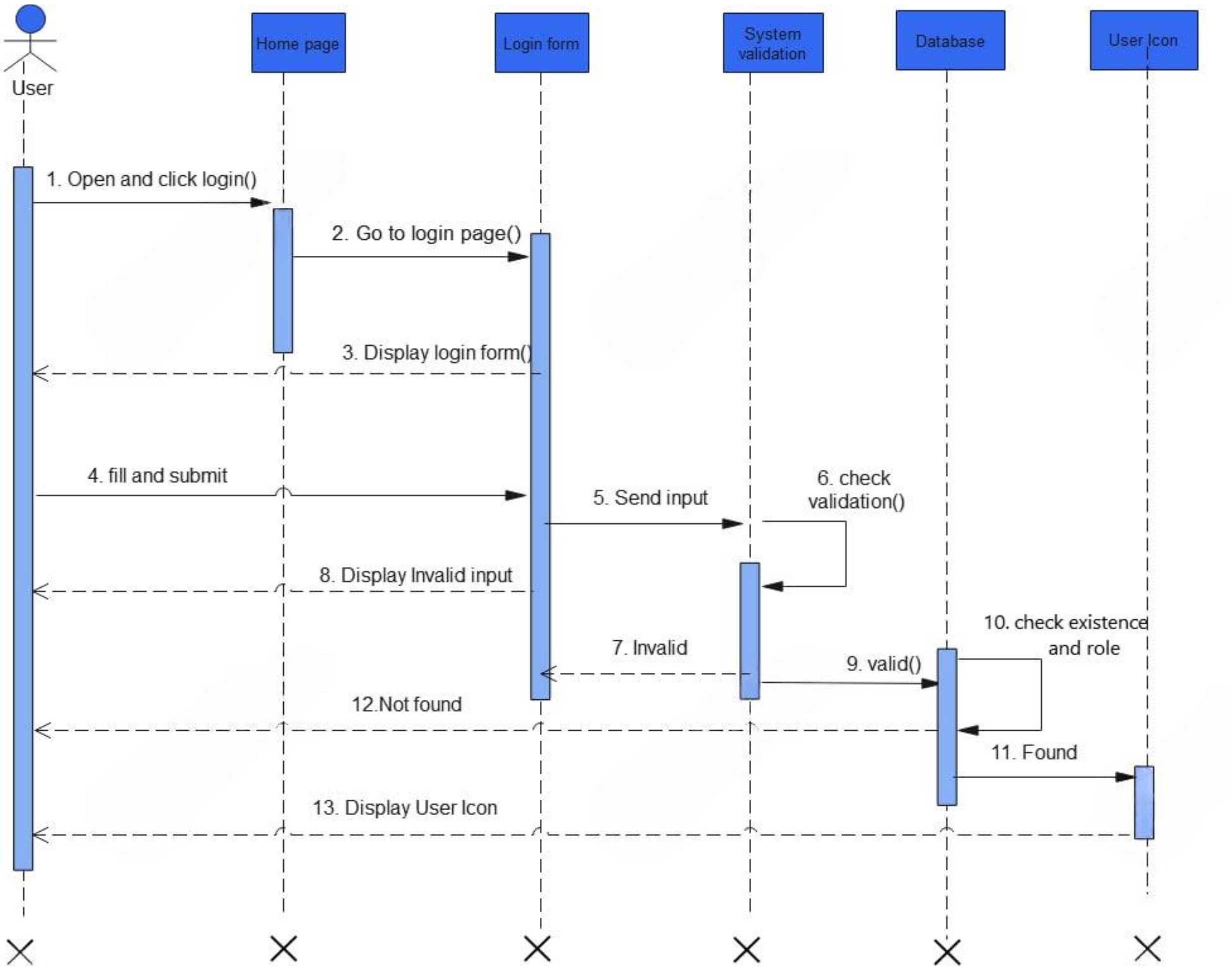


Figure 2. 4 Sequence diagram for Login

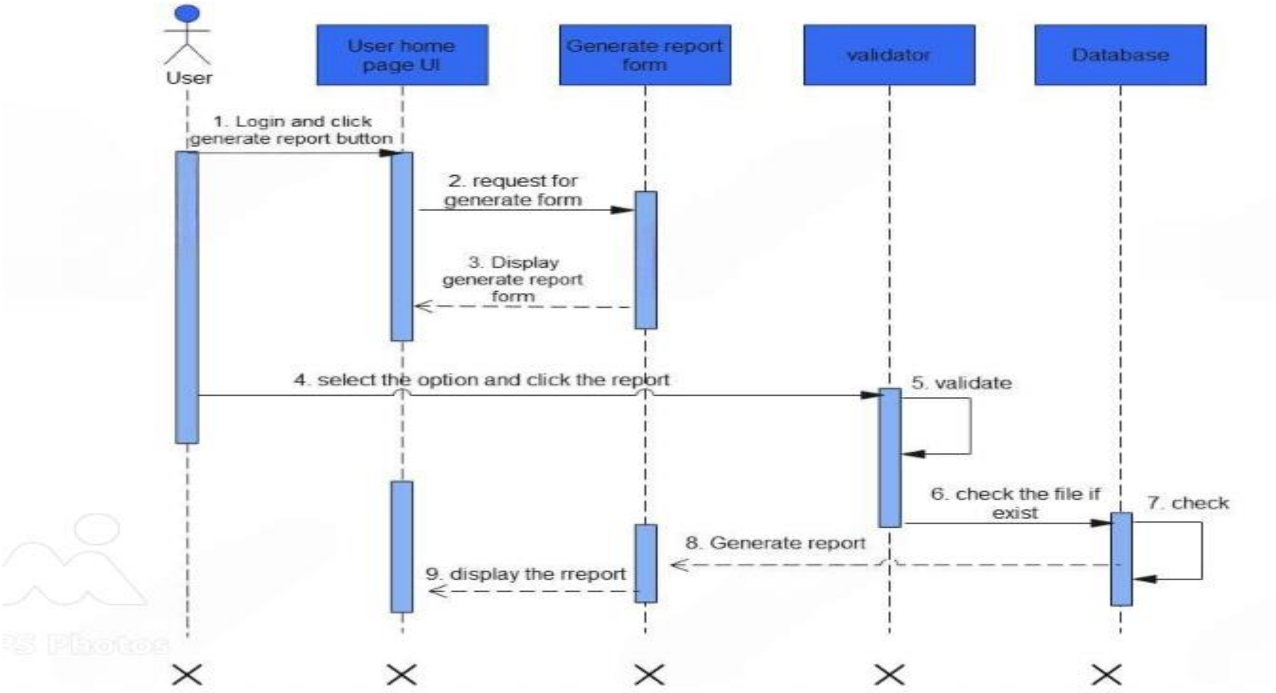

User
User home page UI
Generate report form
validator
Database
1. Login and click generate report button
2. request for generate form
3. Display generate report form
4. select the option and click the report
5. validate
6. check the file if exist
7. check
8. Generate report
9. display the rreport

Figure 2. 5 Sequence diagram for Generate report

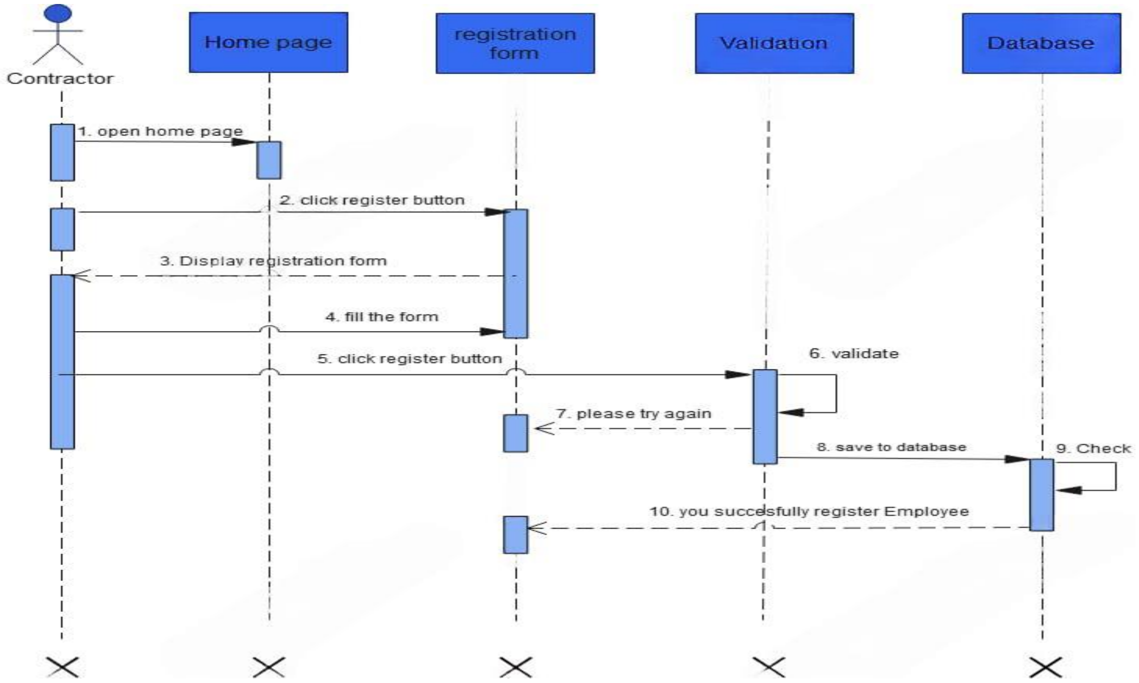


Figure 2. 6 Sequence diagram for Registration

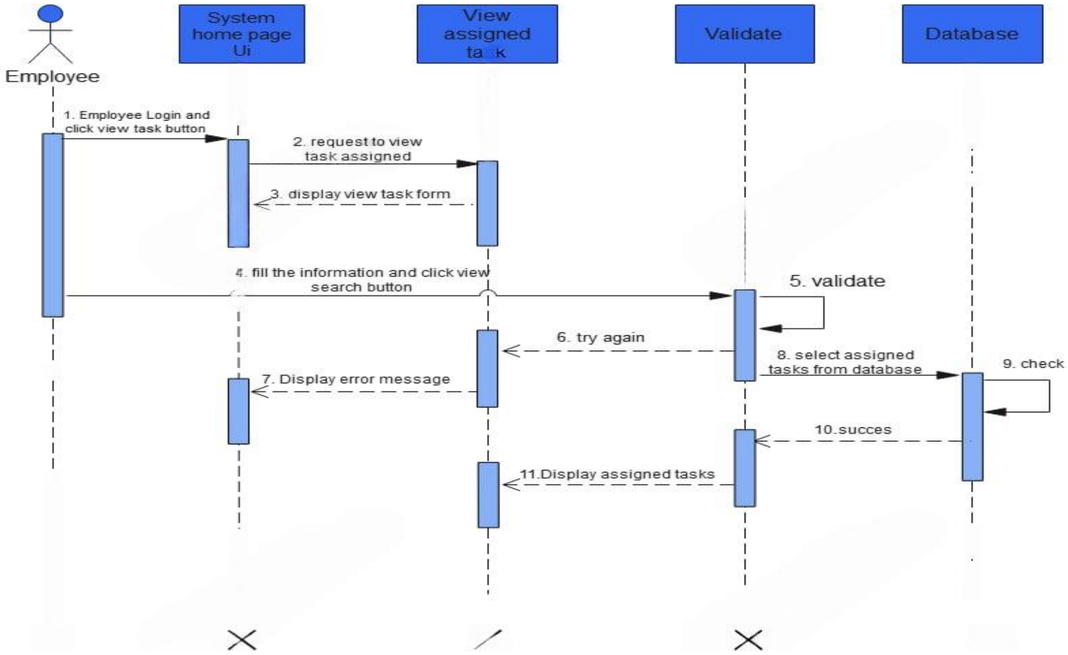

Employee
System home page Ui
View assigned task
Validate
Database
1. Employee Login and click view task button
2. request to view task assigned
3. display view task form
4. fill the information and click view search button
5. validate
6. try again
7. Display error message
8. select assigned tasks from database
9. check
10.succes
11.Display assigned tasks

Figure 2. 7 Sequence diagram for view task

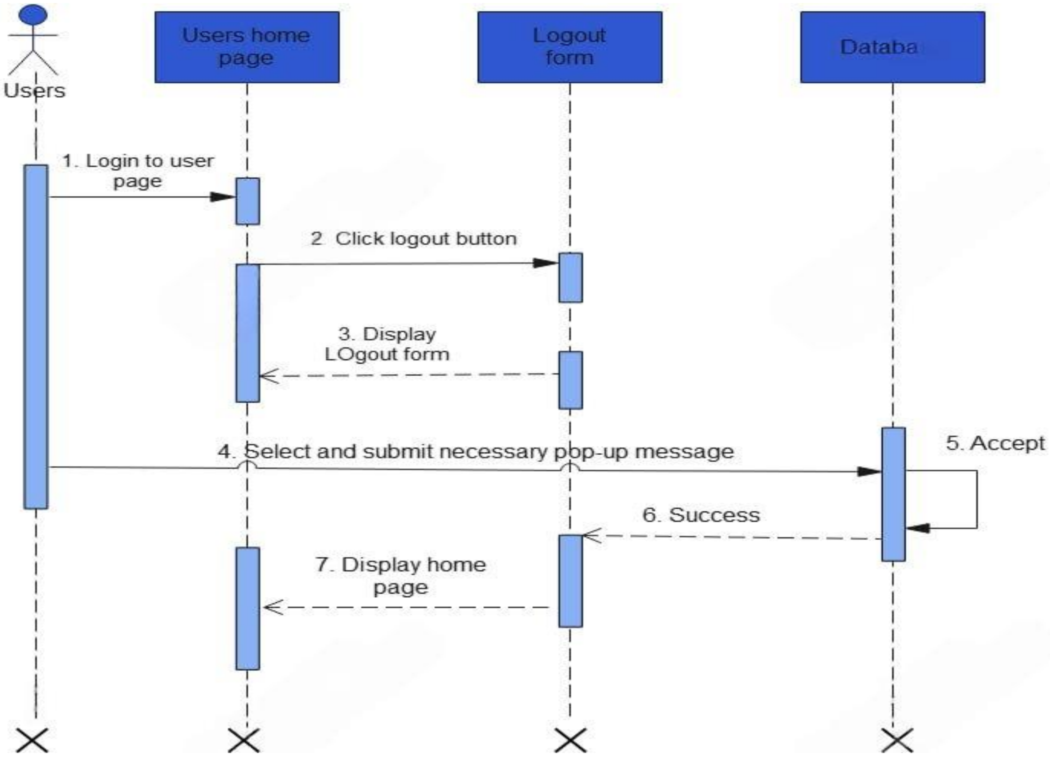


Figure 2. 8 sequence diagram for logout

2.2.7 State chart Diagram

State chart diagram is one of the UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. State chart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered

Our state chart diagrams are the followings

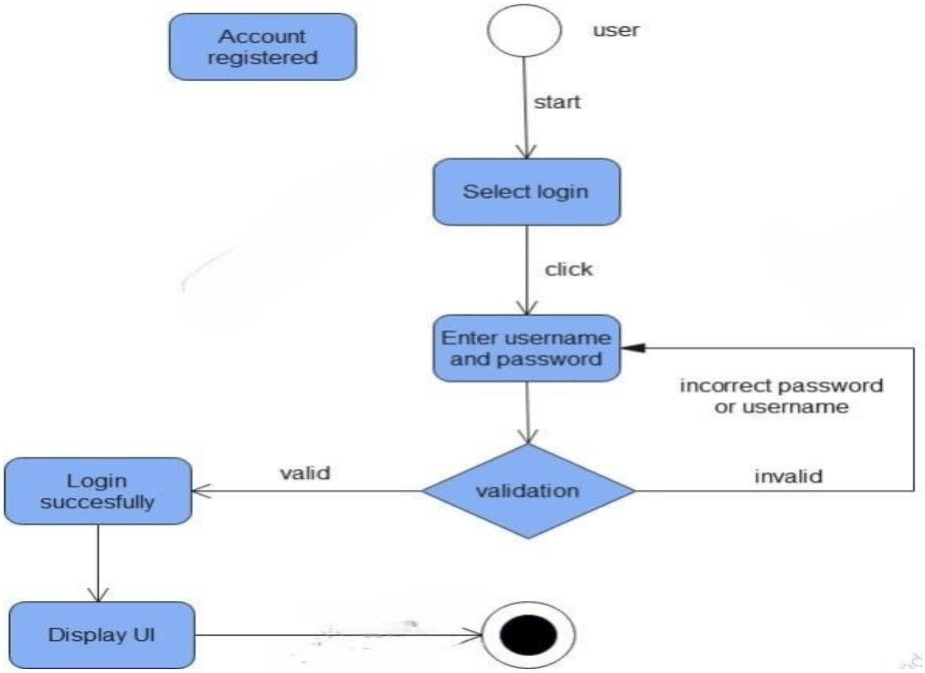


Figure 2. 9 State chart diagram for Login

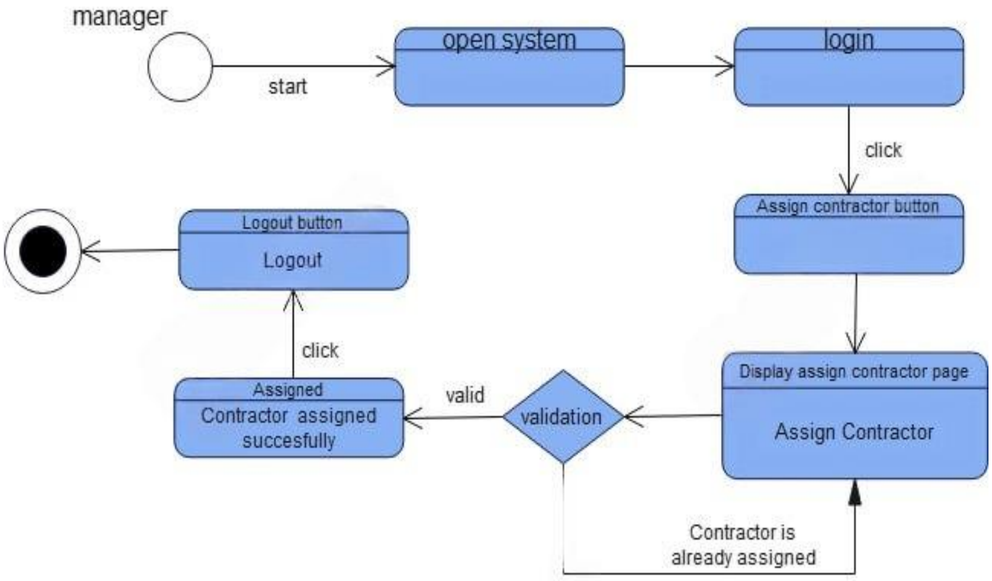


Figure 2. 10 state chart diagram for Assign contractor

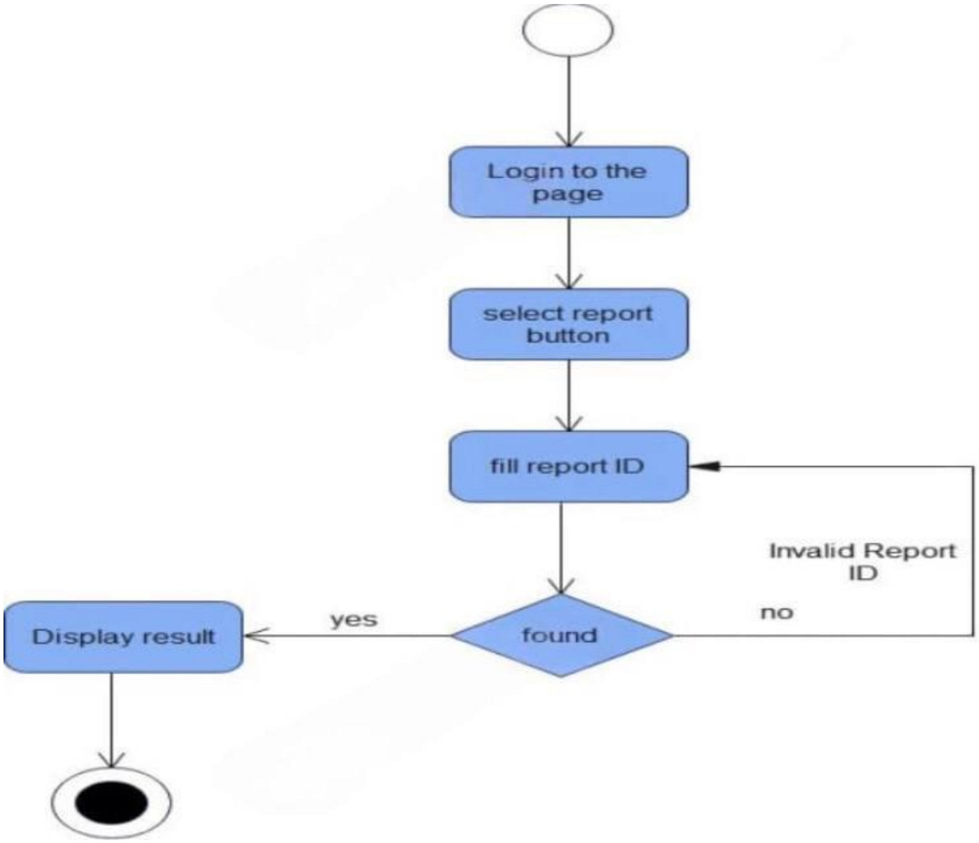


Figure 2. 11 state chart diagram for Generate report

2.2.8 Activity Diagram

We use Activity Diagrams to illustrate the flow of control in our system and refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram.

Our activity diagrams are the followings

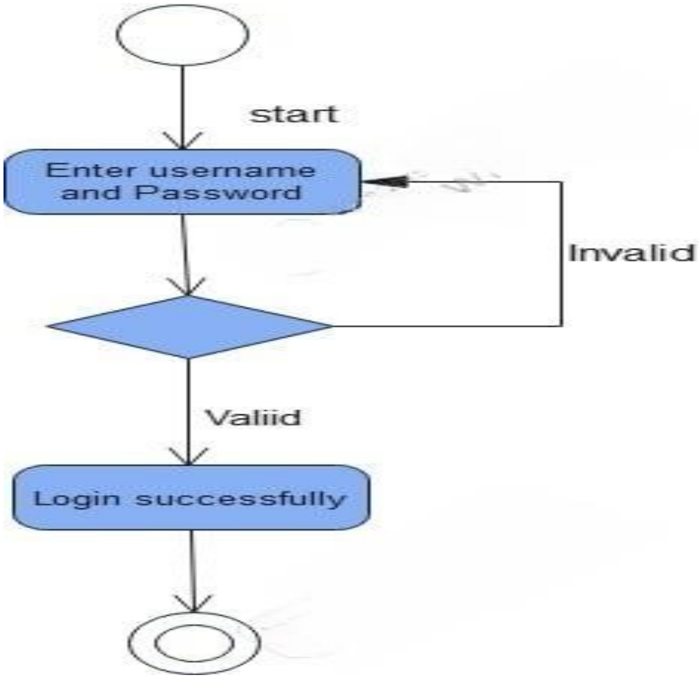


Figure 2. 12 Activity diagram for Login

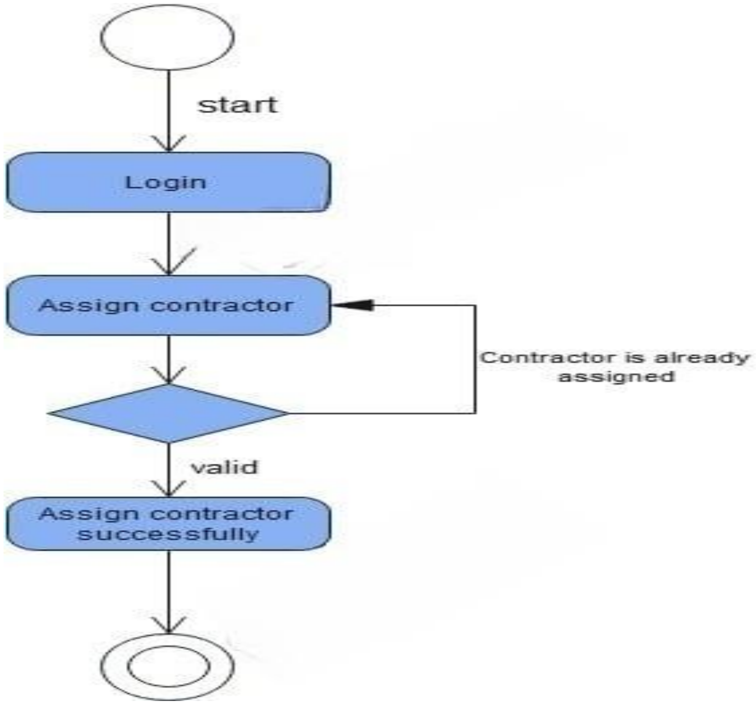

start
Login
Assign contractor
Contractor is already assigned
valid
Assign contractor successfully

Figure 2. 13 Activity diagram for Assign contractor

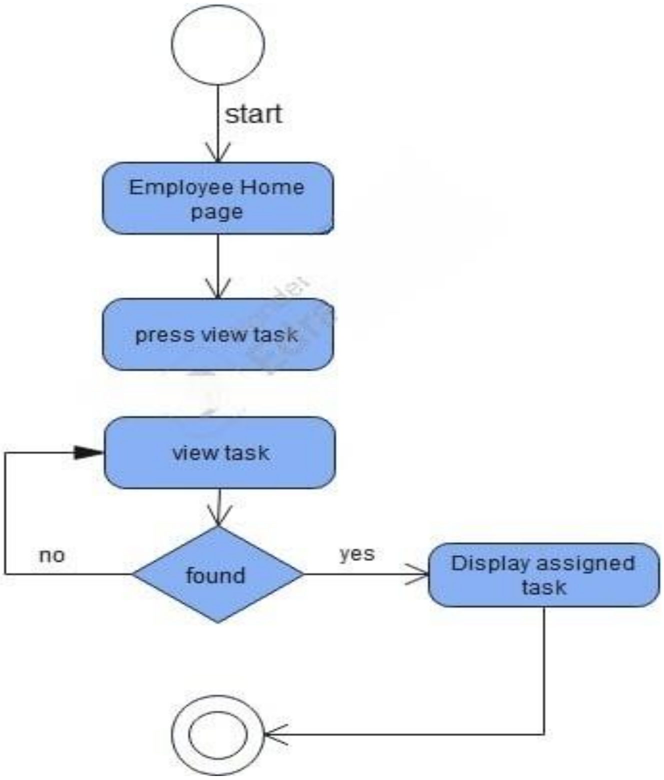
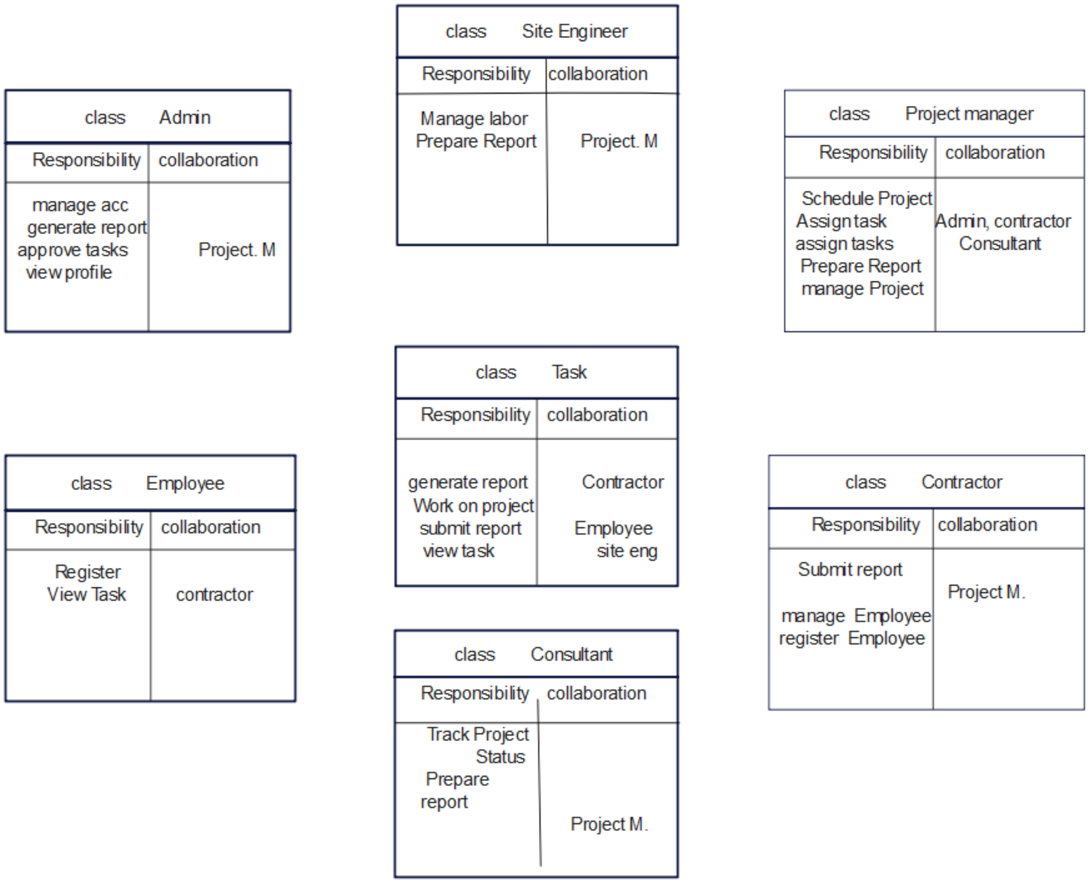


Figure 2. 14 Activity diagram for View task

2.3 Key Abstraction with CRC Analysis

Key abstraction with **CRC analysis** is concepts used in object-oriented design to identify and model the essential components and their interactions in a system. Together, they help break down complex systems into manageable, meaningful components. This evolves into a class diagram of your system when you create a class diagram for the new system [7].



class	Site Engineer
Responsibility	collaboration
Manage labor Prepare Report	Project. M

class	Admin
Responsibility	collaboration
manage acc generate report approve tasks view profile	Project. M

class	Project manager
Responsibility	collaboration
Schedule Project Assign task assign tasks Prepare Report manage Project	Admin, contractor Consultant

class	Task
Responsibility	collaboration
generate report Work on project submit report view task	Contractor Employee site eng

class	Employee
Responsibility	collaboration
Register View Task	contractor

class	Contractor
Responsibility	collaboration
Submit report manage Employee register Employee	Project M.

class	Consultant
Responsibility	collaboration
Track Project Status Prepare report	Project M.

Figure 2. 15 Key Abstraction with CRC analysis

2.4 Conceptual modeling class diagram

Conceptual modeling in class diagrams is a foundational step in understanding and analyzing a problem domain. It focuses on capturing the essential concepts and relationships within a system without delving into specific implementation details. We can use it to model the objects that make up the system to display the relationship between the objects do and services that they develop.

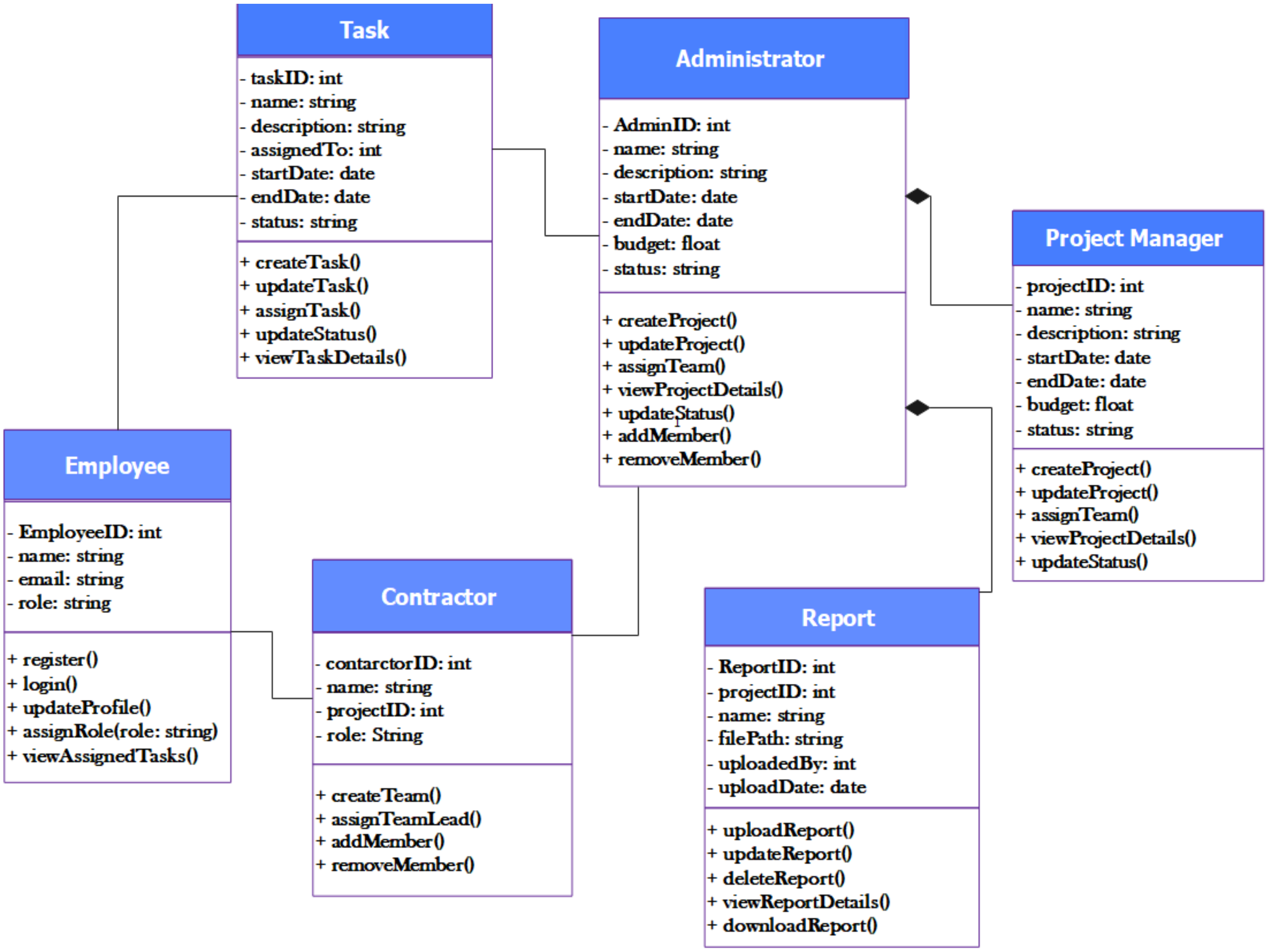


Figure 2. 16 Conceptual Model of class diagram

2.5 Identifying change Cases

Change cases are used to describe new potential requirements for a system or modifications to existing requirements in our system. The existing construction management practices at Salale University likely involve manual processes, leading to inefficiencies, delays, and potential errors.

Any future updates will follow an Agile development cycle, allowing for incremental improvements. System updates will be reviewed by stakeholders to incorporate evolving requirements. Key benefits include:

- ✓ Enhancing the Efficiency: Streamlining workflows, automating tasks, and improving data flow will significantly increase project efficiency and reduce delays.
- ✓ Improving Data Management: for the future centralized database will ensure accurate and timely data access, enabling better tracking of project progress, costs, and resource utilization.
- ✓ Enhancing Collaboration: The system will facilitate seamless communication and collaboration among all stakeholders, including contractors, consultants, and university officials.
- ✓ Improving Decision Making: Real-time data and insights will empower stakeholders to make informed decisions, proactively address challenges, and optimize resource.
- ✓ Making our system Multilingual to support different languages.

2.6 User Interface Prototyping

User Interface prototypes are invaluable tools for stakeholder engagement. By providing an interactive preview of the system, they enable stakeholders to actively participate in the design process, provide valuable feedback, and ensure the final product meets their expectations and user needs.

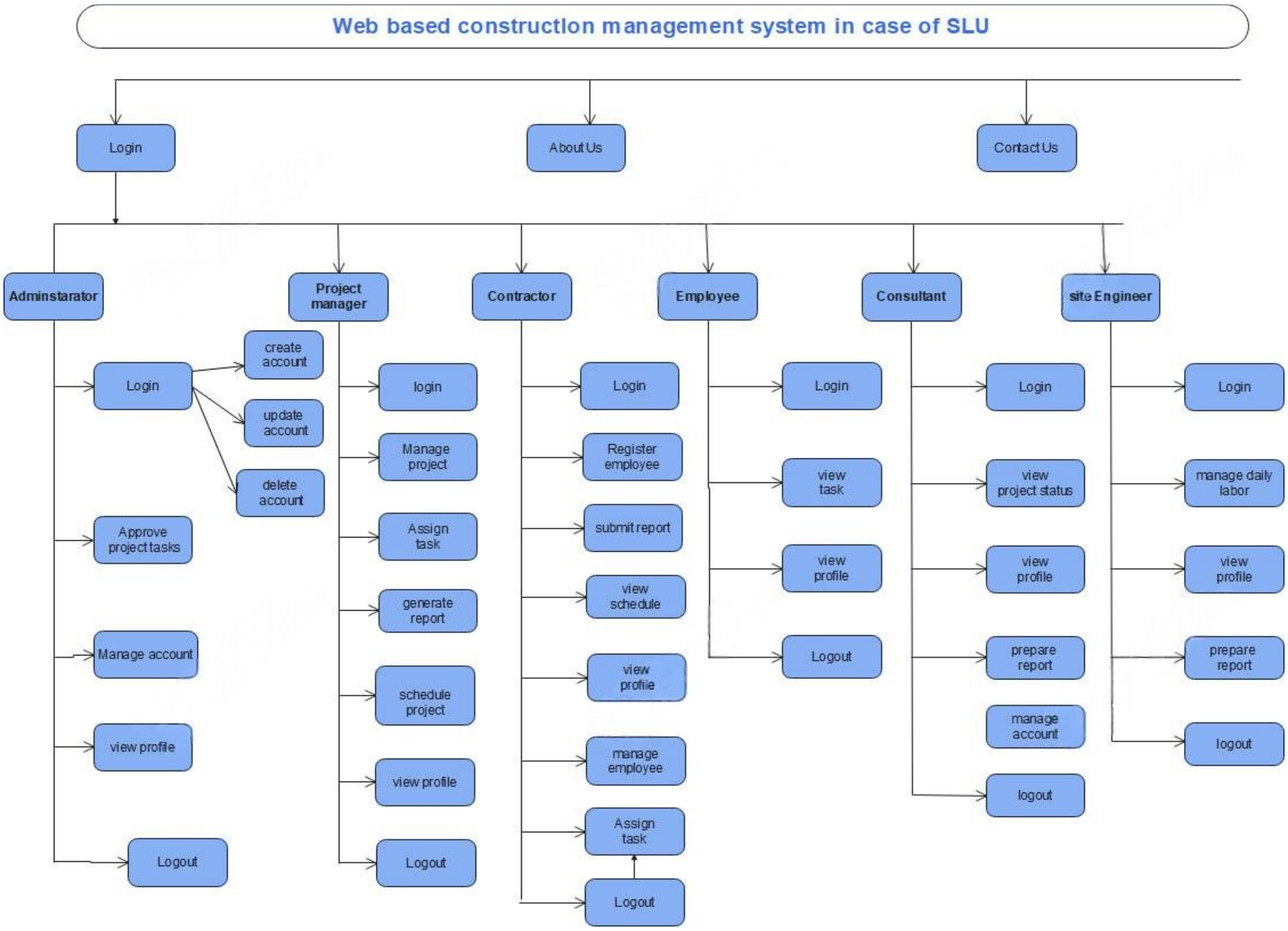


Figure 2. 17 User Interface Prototype

CHAPTER 3

OBJECT ORIENTED SYSTEM DESIGN

3.1 Purpose and Goal of The Design

3.1.1 Purpose of the System

The primary purpose of the web-based construction management system for Salale University is to modernize the management of construction projects, replacing the manual processes with an automated solution, efficient, reliable, and user-friendly platform and aims to address inefficiencies, ensure effective communication, and enhance decision-making by providing real-time data access and project tracking.

3.1.2 Goals of the System Using Various Criteria

1. Performance

- ➢ Fast Response Times: Ensure that the system responds quickly to user requests, minimizing delays when accessing project data, generating reports, or updating records.
- ➢ Efficient Data Processing: Enable the system to perform complex computations with minimal delay.

2. Dependability

- ➢ Data Integrity: Prevent data corruption by using validated inputs
- ➢ Maintainability: The system should be easy to maintain and update, with minimal downtime.

3. End User Considerations

- ➢ Accessibility: Make the system accessible to all stakeholders, including Admins, project managers, contractors, and others.

4. Availability

- Ensure the system is available and accessible to users at all times, with minimal downtime.

5. Problem-solving

- Real-time Tracking: Address inefficiencies in project progress monitoring by providing real-time status updates.

6. Security

- Authentication: Use strong password policy
- Authorization: Ensure that only authorized users can access the system and its resources.

7. Scalability

- Handling Increased User Load: The system should be able to handle an increasing number of users (e.g., Admins, project managers, contractors, staff) as the university's construction projects expand.

3.2 Current Software Architecture

The system relies on fragmented tools like spreadsheets, emails, and physical documents, leading to:

- Delays due to lack of real-time updates.
- Limited automation for task assignments, risk management, and reporting.

3.3 Proposed Software Architecture

The most suitable system architecture for our System is a Three-Tier Architecture. This structure includes a Presentation Layer (front-end) for user interaction via web browsers, the Application Layer (business logic) processes user requests, enforces business rules, and handles core functionalities like task assignments and generate report using technologies like PHP. The Data Layer (database) securely stores and retrieves project data, user accounts, and logs, utilizing MySQL for efficient data management.

Benefits of Three-Tier Architecture:

- Performance: Handles increasing data and users efficiently by distributing workloads across layers.
- Security: Protects data with encryption and secure communication between layers.
- Maintainability: Simplifies updates and debugging with modular design.

- Scalability: Easily scales by adding more resources to specific layers.
- Flexibility: Allows changes in one layer (e.g., front end updates) without impacting others.

3.3.1 Subsystem decomposition

Sub system decomposition refers to the process by which a complex problem or system is broken down into parts that are easier to conceive, understand, program and maintain. To simplify the development and management of the construction management system, it can be decomposed into subsystems. Our web-based construction management system consisted of the following subsystems [8].

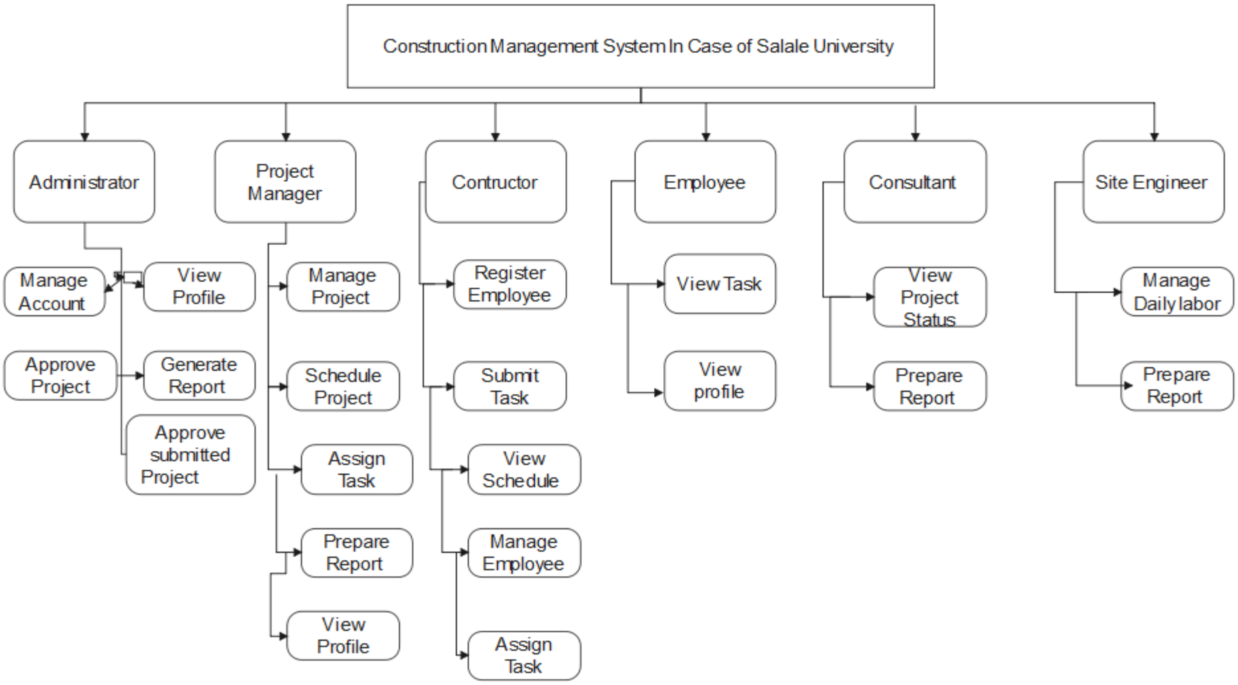


Figure 3. 1 Subsystem Decomposition

3.3.2 Component diagram

In Unified Modeling Language (UML), a component diagram depicts how components are wired together to form larger components or software systems.

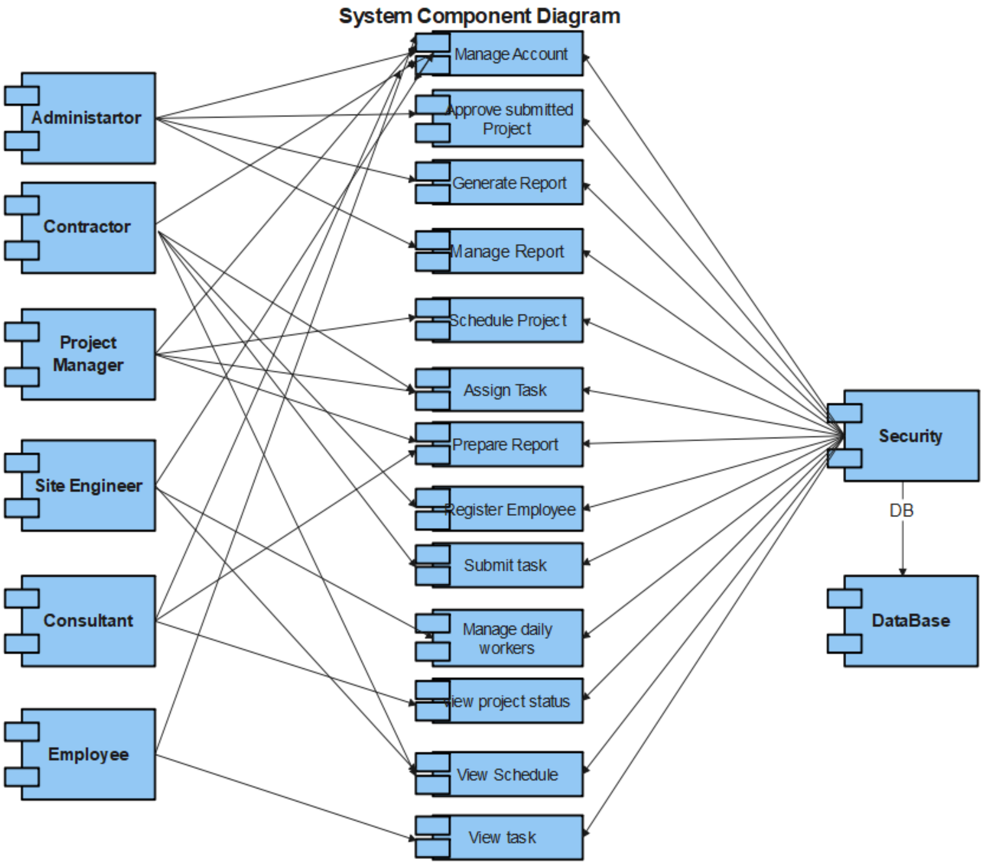


Figure 3. 2 Component Diagram

3.3.3. Deployment diagram

Deployment diagram (modeling) is used to show the hardware of the system, the software that is installed in the hardware and also the middleware that is used to connect the dissimilar machines to one and other. It also shows how the software and the hardware components work together in order perform the task [9].

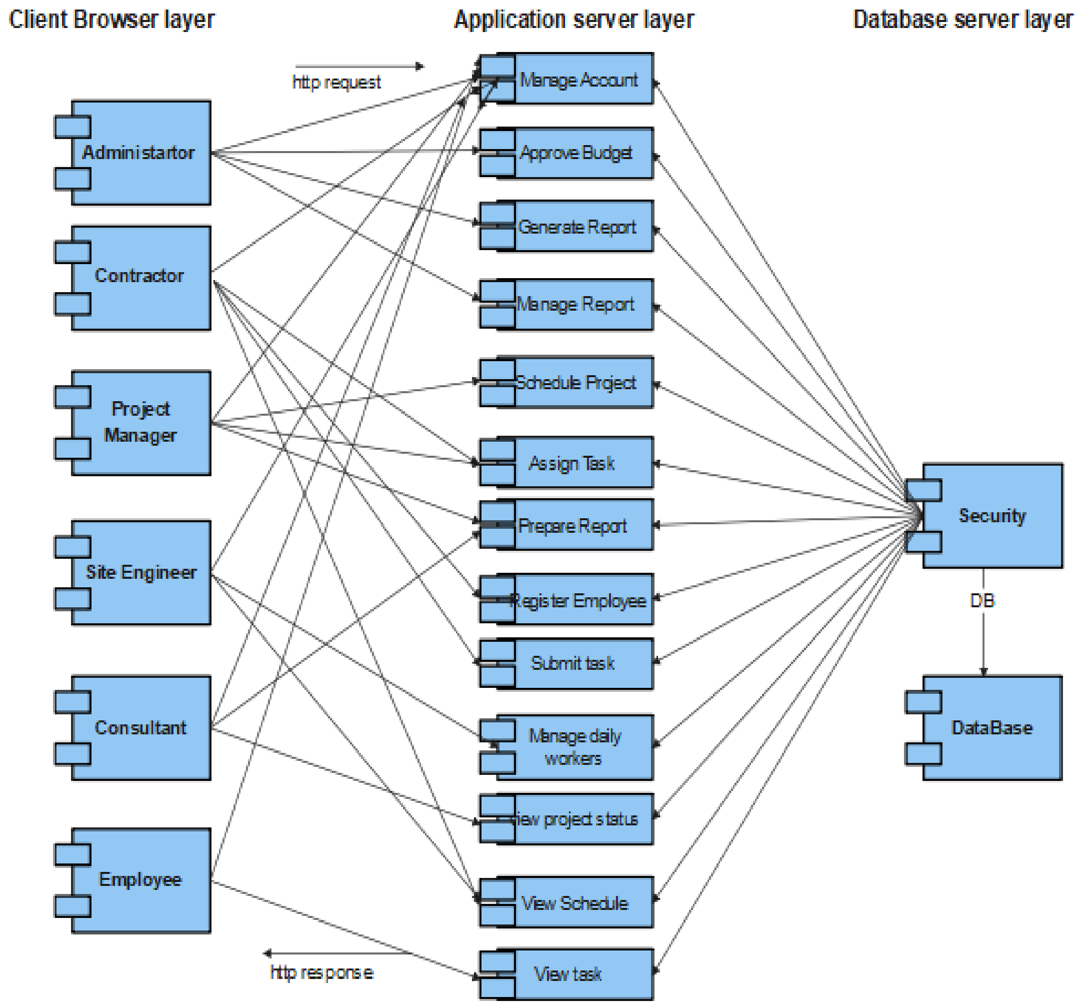


Figure 3. 3 Deployment Diagram

3.4 Database design (E-R diagram) for relational database

This relationship model defines data elements and their relationships within a system. It creates a conceptual database design and provides a simple, user-friendly data view.

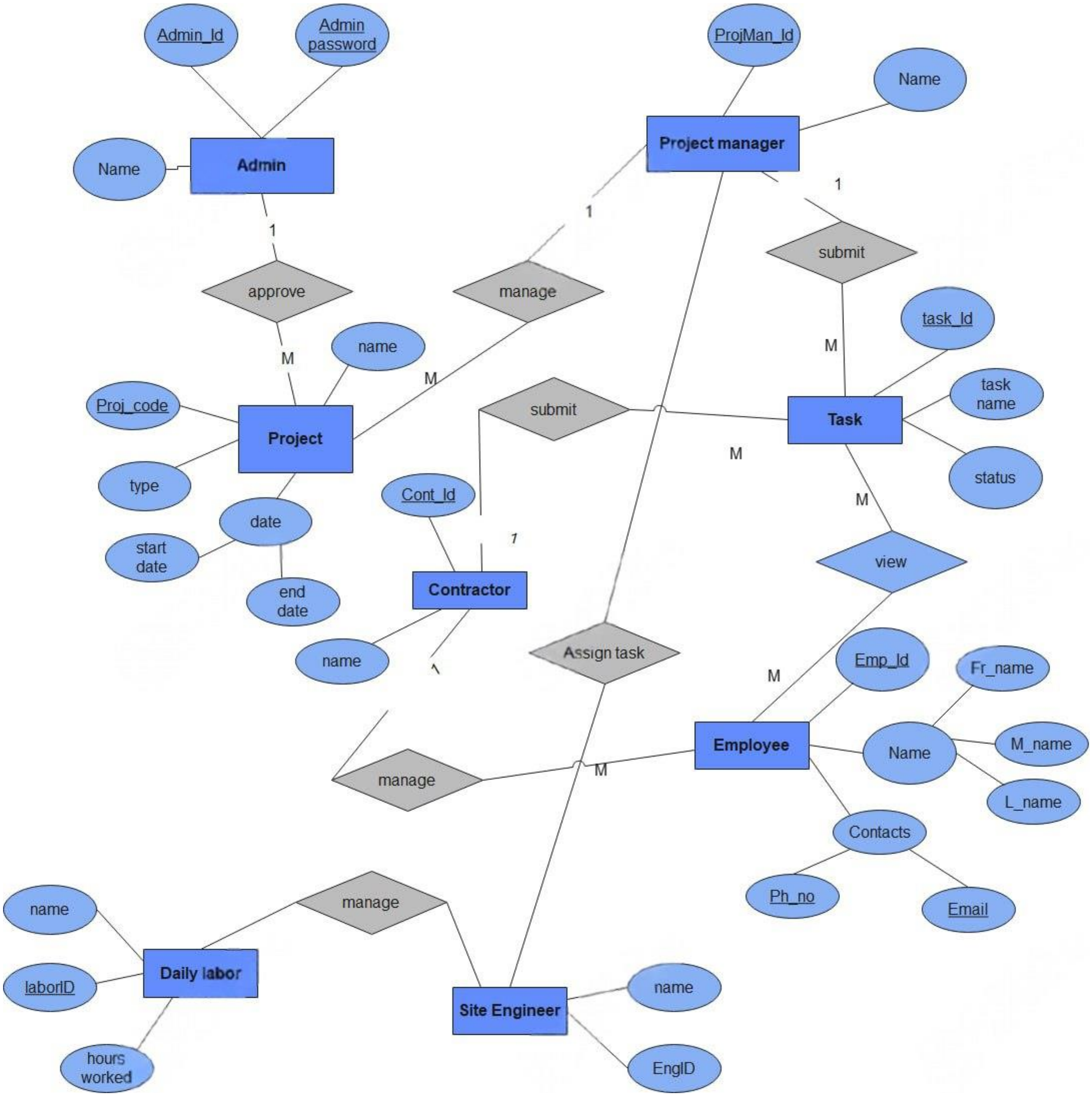


Figure 3. 4 E-R Diagram

3.5 Persistence modelling for object-oriented database

Persistence modeling in object-oriented databases manages persistent objects, which retain their state and values across program executions.

3.6 Access Control and Security

Access Control, also known as Authorization, is the process of mediating access to resources based on role. It’s an essential element of security that determines who is allowed to access certain data, apps, and resources and under what circumstances [10].

The system incorporates the following access control measures:

Role-Based Access Control

Admin:

- account management (create, update, delete accounts, deactivate).
- Approve project tasks
- Generate reports.

Project Manager:

- Access to project and task management (manage, schedule projects, assign tasks).
- Assign tasks to contractors, site engineers and consultants.
- Prepare reports.

Consultant:

- Prepare reports.
- Track project status.

Contractor:

- Register employees.
- View tasks.

- Manage employees.
- Assign tasks to daily labor.

Site Engineer:

- Manage daily labor.
- Prepare reports.
- View assigned tasks.

Employee:

- View assigned tasks.

Table 3. 1 Activity Control

	Actors					
Function	Admin	Project manager	contractor	Employee	Consultant	Site Engineer
Approve project	✓					
Login	✓	✓	✓	✓	✓	✓
Generate report		✓				
Approve submitted projects	✓					
Update user account	✓					
Manage project		✓				
Schedule project		✓				
Assign task		✓	✓			
View task			✓	✓	✓	✓
View profile	✓	✓	✓	✓	✓	✓

Register employee			✓			
View schedule			✓			
logout	✓	✓	✓	✓	✓	✓
Manage account	✓				✓	
View project status					✓	
Manage daily labor						✓

Chapter Four

Implementation, Testing and Maintenance

4.1 Introduction to Languages, IDE's, Tools and Technologies used

This section provides an overview of the core technologies utilized throughout the implementation, testing, and maintenance phases of the system. It outlines the rationale behind choosing specific programming languages and frameworks, the development environments used to write and manage code, as well as various tools and libraries that supported tasks such as debugging, version control, UI design. Understanding these foundational components is essential, as they significantly influence the overall performance, maintainability, and scalability of the system.

4.1.1 Programming Languages

Frontend Programming Languages

1. HTML5
   - Provides the structural framework for all user interfaces
   - Implements semantic markup for accessibility and SEO
   - Creates forms for data input and user interactions
   - Structures content in a logical and navigable hierarchy
2. CSS3, boot strap
   - Styles the user interface with modern design principles
   - Implements responsive layouts that adapt to different screen sizes
   - Creates custom styling with gradients and color schemes
   - Enhances typography using web fonts like 'Poppins'
   - Provides visual feedback for user interactions
3. JavaScript

   We used JavaScript for client-side interactivity and dynamic content loading.

   - Enhances the user experience with client-side functionality

- Implements form validation before submission
- Provides clipboard operations for copying generated credentials
- Enables print functionality for reports and credentials
- Creates interactive UI elements and dynamic content updates
- Manages modal dialogs and popup notifications

Backend Programming Languages

1. PHP

   We used PHP as our primary server-side programming language due to its widespread support, ease of integration with databases, and robust web development capabilities

   - Serves as the primary server-side programming language
   - Handles user authentication and session management
   - Processes database operations through prepared statements
   - Implements business logic and data validation
   - Generates dynamic content based on user roles and permissions
   - Manages file uploads and document processing

2. MySQL: We chose MySQL for our database needs due to its reliability, performance, and compatibility with PHP.

   - Used for database operations and data management
   - Implements complex queries with JOINs for related data retrieval
   - Handles data manipulation (INSERT, UPDATE, DELETE operations)
   - Maintains data integrity through constraints and transactions
   - Supports schema modifications as the application evolves

4.1.2 Integrated Development Environments

1. **Visual Studio Code (VS Code)**

- The primary integrated development environment used for our project.
- VS Code was chosen for its lightweight nature, extensive extension marketplace, and strong support for web development technologies. Its cross-platform compatibility allowed developers to maintain a consistent development environment across different operating systems.
- Features utilized include:
  - PHP language support with syntax highlighting and IntelliSense
  - Live Server for real-time preview of frontend changes
  - Extensions for PHP debugging and code quality
  - Integrated terminal for command-line operations
  - Multi-file editing and project-wide search capabilities
  - Code snippets for faster development
  - Customizable workspace settings for team consistency

2. **XAMPP**

XAMPP; stands for Cross-Platform (X), Apache (A), MySQL (M), PHP (P), and Perl (P). XAMPP was chosen for its ease of setup, cross-platform compatibility, and comprehensive inclusion of all necessary components for PHP web development. It provided a consistent development environment that closely mimicked the production server configuration.

It was utilized as the local development environment, providing an integrated package of:

- Apache HTTP Server: Serves the PHP application and handles HTTP requests
- MySQL: Provides the relational database system for storing application data
- PHP: Processes server-side scripts and business logic. It enables dynamic content generation and interaction with databases.

- phpMyAdmin: Offers a web interface for database management and administration.

Database Management

The system utilizes a relational database management system with:

- Normalized table structure to minimize data redundancy
- Foreign key constraints to maintain data integrity
- Indexes for optimized query performance
- Prepared statements to prevent SQL injection attacks

Security Implementations

Security is a priority in the CMS implementation, with measures including:

- Password hashing for secure credential storage
- Session-based authentication with timeout mechanisms
- Role-based access control for different user types
- Input validation and sanitization to prevent XSS attacks
- CSRF protection for form submissions

Testing Tools

The development process incorporated various testing methodologies:

- Unit testing for individual components
- Integration testing for feature interactions
- Security testing to identify and address vulnerabilities

4.2 Coding Standards of Language Used

In our PHP implementation, we adhered to the following coding standards:

Naming Conventions

- Files: Used descriptive names with CamelCase (e.g., ProjectAssignment.php, SiteEngineerDashboard.php)
- Variables: Used lowercase with underscores for readability (e.g., $project_id, $user_role)

- Database Tables: Named using plural nouns (e.g., projects, users)

Code Organization

- Separation of Concerns: We separated database connections, business logic, and presentation layers.
- Includes: Used PHP includes to modularize code and prevent duplication.
- Session Management: Implemented consistent session handling for user authentication and authorization.

Security Practices

- Prepared Statements: Used parameterized queries to prevent SQL injection attacks.
- Input Validation: Validated all user inputs before processing.
- Output Escaping: Used htmlspecialchars() to prevent XSS attacks when displaying user data.
- Role-Based Access Control: Implemented session checks to ensure users can only access authorized pages.

4.3 Tools and Technologies used for Implementation

We used the following tools and technologies in the development of the construction Management System:

Programming Languages

Backend

1. **PHP** - The primary server-side language used throughout the application for processing logic, database operations, and generating dynamic content.
2. **SQL** - Used for database queries, data manipulation, and retrieval.

Frontend

1. **HTML5** - Used for structuring web pages and forms.
2. **CSS3** - Used for styling and responsive design.
3. **JavaScript** - Used for client-side functionality like form validation, copying to clipboard, and printing credentials.

Development Environment

1. **XAMPP** - Local development environment that includes:
   - Apache HTTP Server
   - MySQL database
   - PHP interpreter
   - phpMyAdmin for database management
2. **Visual Studio Code (VS Code)** - Primary code editor/IDE used for development.

Frameworks and Libraries

1. **Bootstrap 5.3** - CSS framework for responsive design and UI components, as evidenced by the inclusion of Bootstrap CSS (bootstrap@5.3.2/dist/css/bootstrap.min.css).

The frontend is built on Bootstrap 5.3, providing:

- A responsive grid system for layout management
- Pre-styled UI components (buttons, forms, alerts, modals)
- Consistent design patterns across the application
- Mobile-first approach to ensure compatibility across devices

2. **Font Awesome 6** - Icon library used throughout the interface for visual elements, as seen in code snippets with classes like fas fa-print, far fa-copy, and fas fa-exclamation-circle.

This icon library enhances the user interface with:

- Intuitive visual cues for actions and information
- Consistent iconography throughout the application
- Scalable vector icons that maintain quality at any size

Database

1. **MySQL** - Relational database management system used for storing application data, as evidenced by the SQL queries and the inclusion of db_connection.php.

Security Implementations

1. **Prepared Statements** - Used throughout the codebase to prevent SQL injection attacks.
2. **Password Hashing** - Implemented for secure storage of user credentials.
3. **Session Management** - Used for user authentication and maintaining user state.
4. **Input Validation** - Implemented to sanitize and validate user inputs.
5. **Role-Based Access Control** - Different user roles (Project Manager, Contractor, Consultant, Employee) with specific permissions.

Additional Technologies

1. **AJAX** - Likely used for asynchronous operations (though not explicitly shown in the provided snippets).
2. **File Upload Handling** - Functionality for document attachments and file management.

These tools and technologies work together to create a comprehensive web-based system for managing contractors, projects, tasks, and reports with different user roles and permissions.

4.4 Testing Techniques and Test Plans

We implemented a comprehensive testing strategy to ensure our CMS functions correctly and securely:

Testing Techniques

1. Unit Testing

Unit testing would focus on testing individual components and functions in isolation:

- Database Operations: Tested individual database operations to ensure correct data retrieval and manipulation. Testing CRUD operations for each entity (users, projects, tasks, reports)
- Authentication Functions: Testing login, password validation, and session management
- Form Processing: Testing validation and processing of form submissions
- Business Logic: Testing specific business rules and calculations
- File Upload Functionality: Tested file upload features with various file types and sizes to ensure proper handling.

2. Integration Testing

Integration testing would verify that different components work together correctly:

- User Role Interactions: Testing how different user roles (Project Manager, Contractor, Consultant) interact with the system
- Workflow Testing: Testing complete workflows like project approval processes
- Database Integration: Testing the application's interaction with the database across multiple operations

3. System Testing

System testing would evaluate the complete application:

- End-to-End Testing: Testing complete user journeys through the application
- Performance Testing: Evaluating response times and resource usage
- Security Testing: Checking for vulnerabilities like SQL injection, XSS, CSRF
- Browser Compatibility: Testing across different browsers and devices
- Responsive Design: Verified that the interface adapts correctly to different screen sizes.

Test Plan

1. Test Environment Setup

- Local Development Environment: Using XAMPP with Apache, PHP, and MySQL
- Testing Database: Separate database instance with test data
- Browser Testing Tools: Multiple browsers and responsive design testing tools

2. Test Cases for Key Functionality

User Authentication and Management

- Verify user login with valid credentials
- Test login failure with invalid credentials
- Test password reset functionality
- Verify role-based access restrictions

- Test user account creation and management

Project Management

- Test project creation and submission for approval
- Verify project approval/rejection workflows
- Test project assignment to consultants
- Verify project details viewing based on user roles

Task Management

- Test task assignment to employees
- Verify task status updates
- Test task attachment uploads and viewing
- Verify task scheduling and deadline notifications

Reporting

- Test report generation and submission
- Verify report viewing permissions
- Test report filtering and searching
- Verify report printing functionality

3. Security Testing

- Test for SQL injection vulnerabilities in all input fields
- Verify XSS protection in output display
- Test session security and timeout functionality
- Verify CSRF protection in forms
- Test file upload security

4. Performance Testing

- Measure page load times under different conditions
- Test database query performance with large datasets

Chapter Five

Results and Discussions

5.1 User Interface Representation

Our Construction Management System (CMS) features a modern, responsive user interface designed to provide intuitive access to various system functions based on user roles. The interface utilizes Bootstrap 5.3 for consistent styling and responsive design, ensuring usability across different devices and screen sizes.

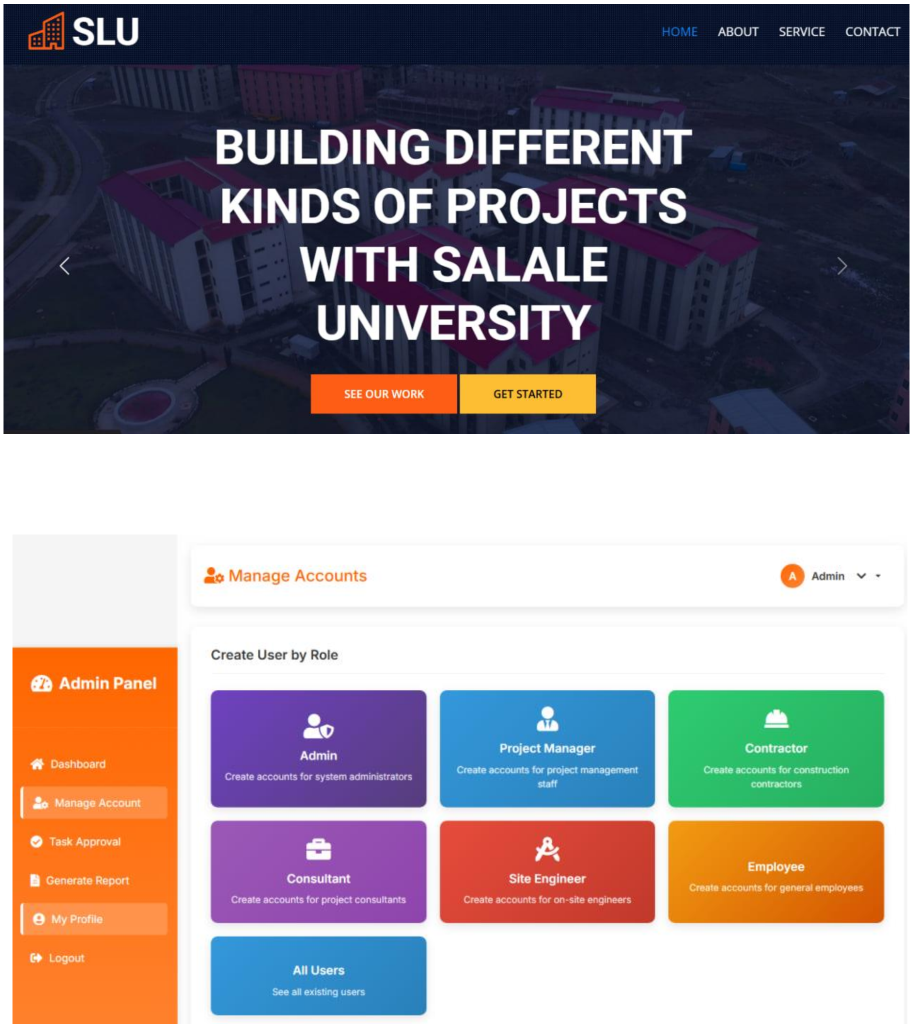

SLU
HOME
ABOUT
SERVICE
CONTACT
BUILDING DIFFERENT KINDS OF PROJECTS WITH SALALE UNIVERSITY
SEE OUR WORK
GET STARTED
Manage Accounts
Admin
Admin Panel
Dashboard
Manage Account
Task Approval
Generate Report
My Profile
Logout
Create User by Role
Admin
Create accounts for system administrators
Project Manager
Create accounts for project management staff
Contractor
Create accounts for construction contractors
Consultant
Create accounts for project consultants
Site Engineer
Create accounts for on-site engineers
Employee
Create accounts for general employees
All Users
See all existing users

figure 5. 1 user interface

5.1.1 Brief Description of Various Modules of the system

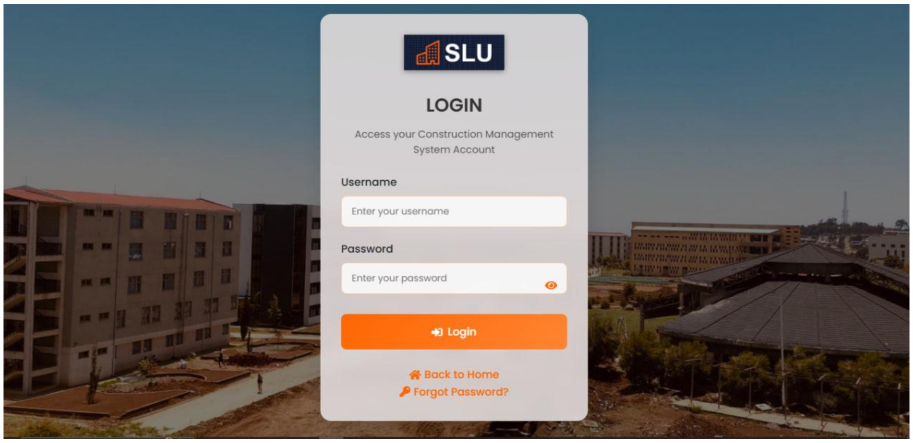
1. **Authentication Module**
   - Handles user login, account creation, and password management
   - Implements role-based access control for different user types
   - Manages session security and user verification
2. **Admin Module**
   - Manages system users and their roles
   - Handles construction project approval and rejection
   - Provides system-wide configuration options
   - Monitors system performance and usage
3. **Project Management Module**
   - Allows Project Managers to create and manage construction projects
   - Facilitates project approval workflows
   - Provides project status tracking and reporting
   - Enables assignment of projects to consultants
4. **Task Management Module**
   - Supports assignment of construction tasks to employees
   - Tracks task status and completion
   - Manages task documentation and attachments
   - Provides scheduling and deadline management for construction activities
5. **Contractor Management Module**
   - Enables contractors to manage their employee accounts
   - Facilitates assignment of construction workers to specific tasks
   - Provides contractor-specific reporting and tracking
   - Manages contractor credentials and profiles

6. **Consultant Module**
    - Allows consultants to view assigned construction projects
    - Facilitates project implementation and tracking
    - Provides reporting capabilities for consultants
    - Manages consultant schedules and assignments
7. **Reporting Module**
    - Generates various construction reports for different user roles
    - Provides filtering and search capabilities
    - Supports printing and exporting of reports
    - Displays project and task statistics for construction activities

5.2 Snapshots of system with brief detail of each

Login Screen

The login interface would provide secure access to the system with username and password fields, along with appropriate validation and error messaging for failed login attempts.



Access your Construction Management System Account

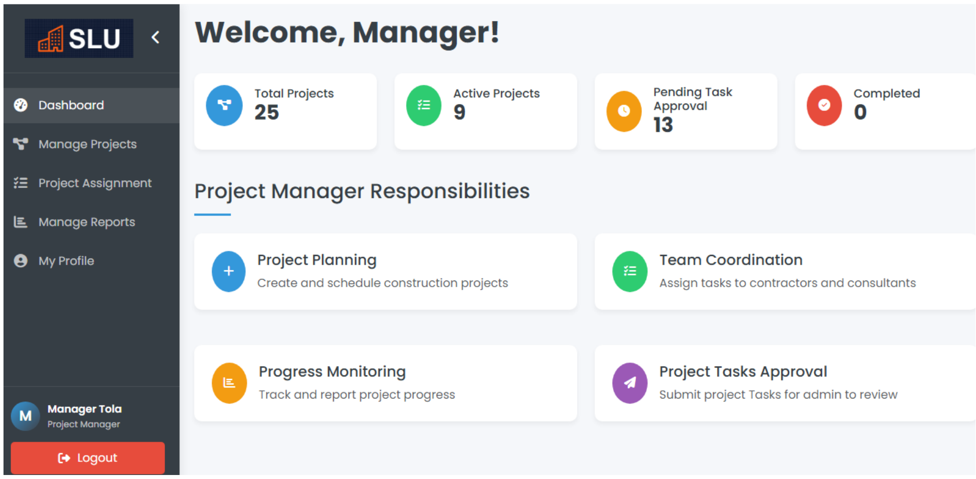
figure 5. 2 login interface

Project Manager Dashboard

The dashboard for Project Managers would display:

- Overview of current construction projects and their statuses

- Quick access to create new construction projects
- Reports requiring attention
- Recent activities and notifications



Welcome, Manager!

Total Projects 25

Active Projects 9

Pending Task Approval 13

Completed 0

Project Manager Responsibilities

Project Planning
Create and schedule construction projects

Team Coordination
Assign tasks to contractors and consultants

Progress Monitoring
Track and report project progress

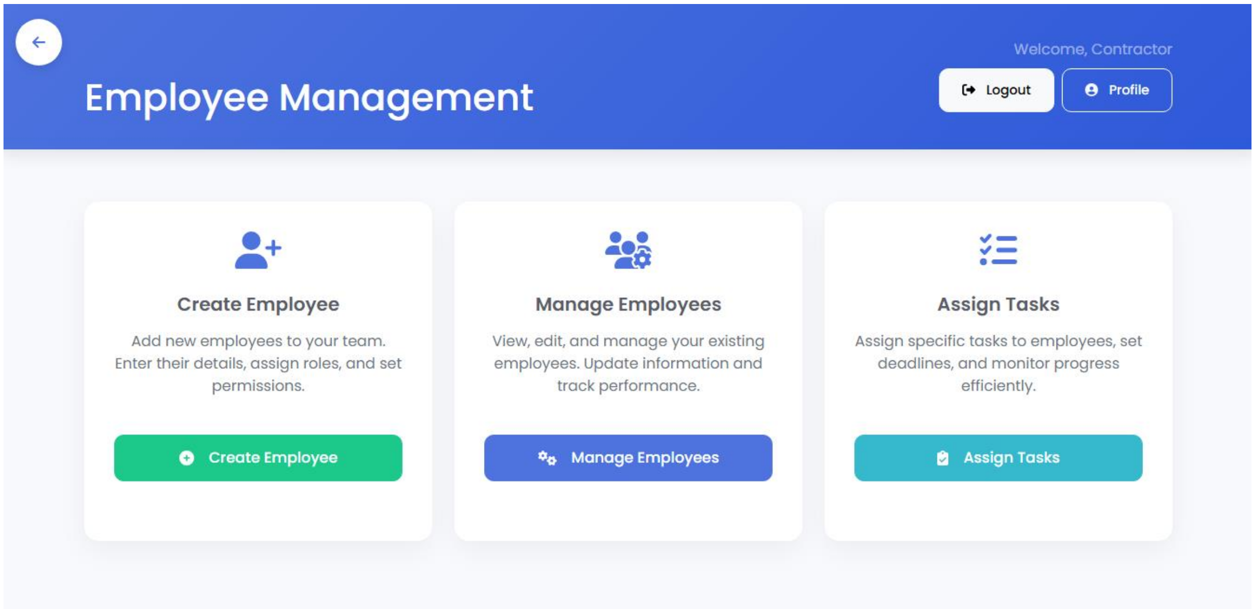
Project Tasks Approval
Submit project Tasks for admin to review

figure 5. 3 project manager's interface

Contractor Account Management

This interface allows contractors to:

- View and manage construction employee accounts
- Create new employee accounts with auto-generated credentials
- Deactivate or delete employee accounts as needed
- Track employee assignments and performance on construction sites



Employee Management

Create Employee

Add new employees to your team. Enter their details, assign roles, and set permissions.

Manage Employees

View, edit, and manage your existing employees. Update information and track performance.

Assign Tasks

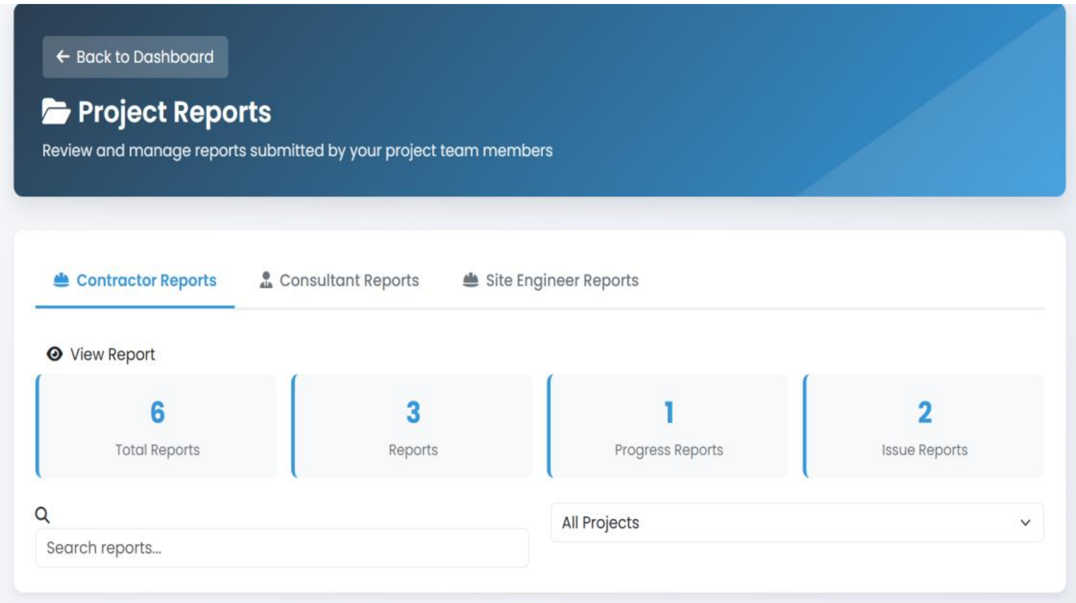
Assign specific tasks to employees, set deadlines, and monitor progress efficiently.

figure 5. 4 Employee management interface

Project Details View

This screen provides comprehensive construction project information including:

- Project specifications and requirements
- Assigned team members and their roles
- Task breakdown and status
- Construction timeline and milestones
- Attached documents and resources



← Back to Dashboard

Project Reports

Review and manage reports submitted by your project team members

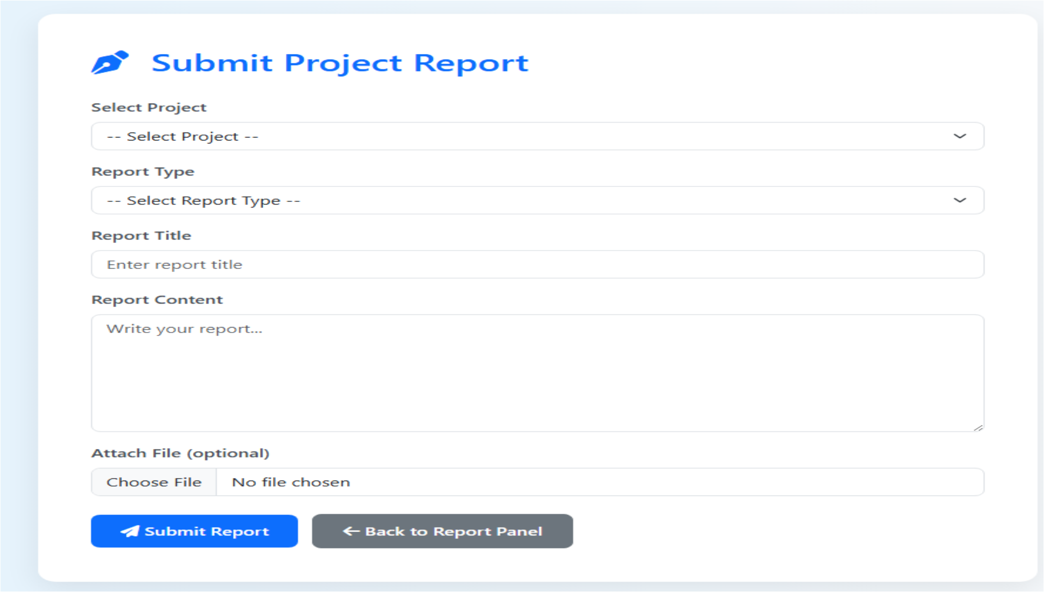
6	3	1	2
Total Reports	Reports	Progress Reports	Issue Reports

figure 5. 5 report generation interface

Report Generation

The reporting interface allows users to:

- Create different types of construction reports
- Include relevant project data and metrics
- Attach supporting documentation
- Submit reports for review



Submit Project Report

Select Project

-- Select Project --

Report Type

-- Select Report Type --

Report Title

Enter report title

Report Content

Write your report...

Attach File (optional)

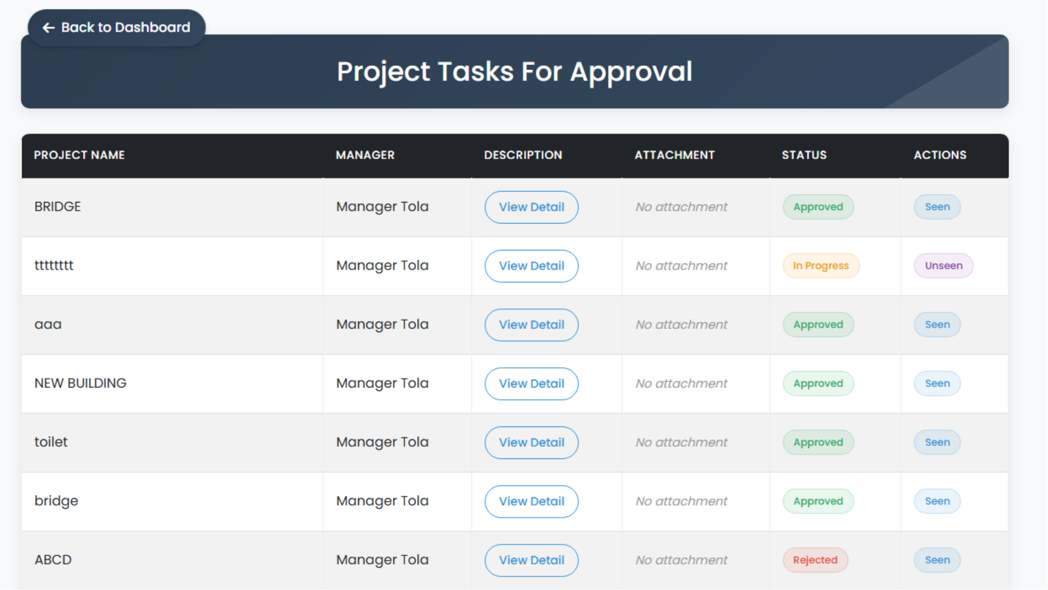
Choose File No file chosen

figure 5. 6 Project manager report

Project Approval Workflow

This interface shows:

- Construction projects pending approval
- Project details and justifications
- Options for approving or rejecting projects
- Fields for providing feedback and comments



Project Tasks For Approval

PROJECT NAME	MANAGER	DESCRIPTION	ATTACHMENT	STATUS	ACTIONS
BRIDGE	Manager Tola	View Detail	No attachment	Approved	Seen
ttttttt	Manager Tola	View Detail	No attachment	In Progress	Unseen
aaa	Manager Tola	View Detail	No attachment	Approved	Seen
NEW BUILDING	Manager Tola	View Detail	No attachment	Approved	Seen
toilet	Manager Tola	View Detail	No attachment	Approved	Seen
bridge	Manager Tola	View Detail	No attachment	Approved	Seen
ABCD	Manager Tola	View Detail	No attachment	Rejected	Seen

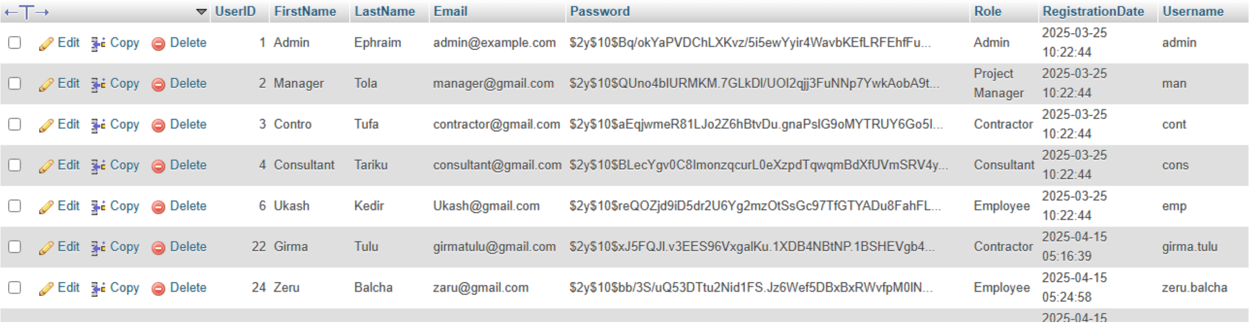
figure 5. 7 Task approval

5.3 Back Ends Representation (Database to be used)

This Construction Management System utilizes a relational database (MySQL) to store and manage all system data. The database schema is designed to support the various modules and their relationships while ensuring data integrity and performance.

5.3.1 Snapshots of Database Tables with brief description

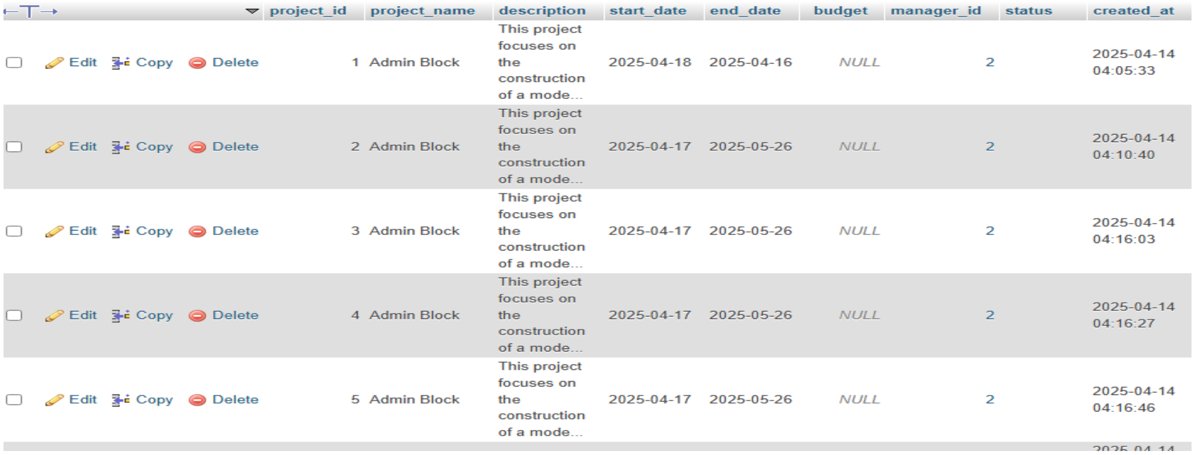
Users Table



←T→	UserID	FirstName	LastName	Email	Password	Role	RegistrationDate	Username
□ Edit Copy Delete	1	Admin	Ephraim	admin@example.com	$2y$10$Bq/okYaPVDChLXKvz/5i5ewYyir4WavbKEfLRFEhfFu...	Admin	2025-03-25 10:22:44	admin
□ Edit Copy Delete	2	Manager	Tola	manager@gmail.com	$2y$10$QUno4bIURMKM.7GLkDI/UOI2qjj3FuNNp7YwkAobA9t...	Project Manager	2025-03-25 10:22:44	man
□ Edit Copy Delete	3	Contro	Tufa	contractor@gmail.com	$2y$10$aEqjwmeR81LJo2Z6hBtvDu.gnaPslG9oMYTRUY6Go5l...	Contractor	2025-03-25 10:22:44	cont
□ Edit Copy Delete	4	Consultant	Tariku	consultant@gmail.com	$2y$10$BLecYgv0C8ImonzqcurL0eXzpdTqwqmBdXfUVmSRV4y...	Consultant	2025-03-25 10:22:44	cons
□ Edit Copy Delete	6	Ukash	Kedir	Ukash@gmail.com	$2y$10$reQOZjd9iD5dr2U6Yg2mzOtSsGc97TfGTYADu8FahFL...	Employee	2025-03-25 10:22:44	emp
□ Edit Copy Delete	22	Girma	Tulu	girmatulu@gmail.com	$2y$10$xJ5FQJl.v3EES96VxgalKu.1XDB4NBtNP.1BSHEVgb4...	Contractor	2025-04-15 05:16:39	girma.tulu
□ Edit Copy Delete	24	Zeru	Balcha	zaru@gmail.com	$2y$10$bb/3S/uQ53DTtu2Nid1FS.Jz6Wef5DBxBxRWvfpM0IN...	Employee	2025-04-15 05:24:58	zeru.balcha
							2025-04-15	

figure 5. 8 user table

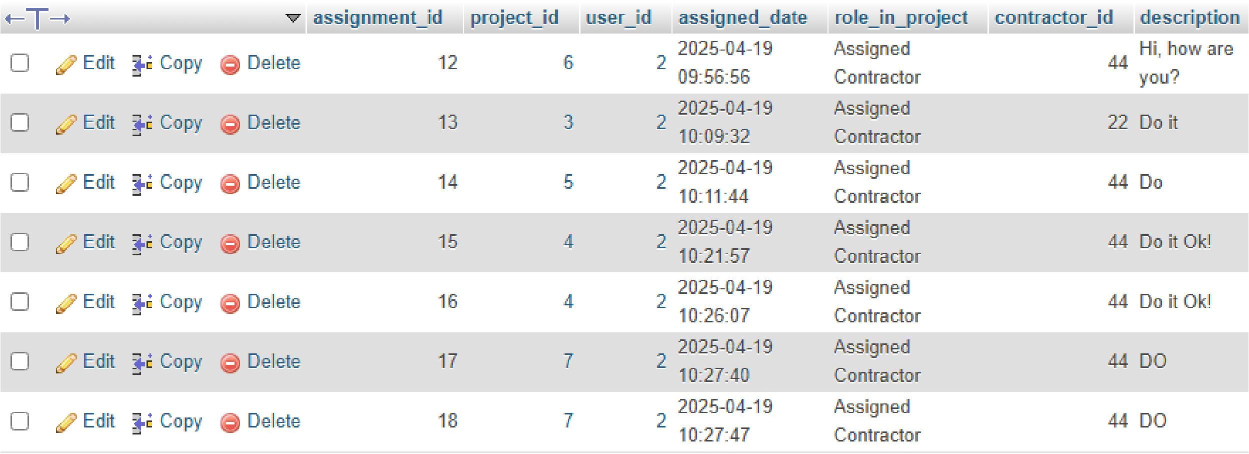
Projects Table



←T→	project_id	project_name	description	start_date	end_date	budget	manager_id	status	created_at
□ Edit Copy Delete	1	Admin Block	This project focuses on the construction of a mode...	2025-04-18	2025-04-16	NULL	2		2025-04-14 04:05:33
□ Edit Copy Delete	2	Admin Block	This project focuses on the construction of a mode...	2025-04-17	2025-05-26	NULL	2		2025-04-14 04:10:40
□ Edit Copy Delete	3	Admin Block	This project focuses on the construction of a mode...	2025-04-17	2025-05-26	NULL	2		2025-04-14 04:16:03
□ Edit Copy Delete	4	Admin Block	This project focuses on the construction of a mode...	2025-04-17	2025-05-26	NULL	2		2025-04-14 04:16:27
□ Edit Copy Delete	5	Admin Block	This project focuses on the construction of a mode...	2025-04-17	2025-05-26	NULL	2		2025-04-14 04:16:46

figure 5. 9 project table

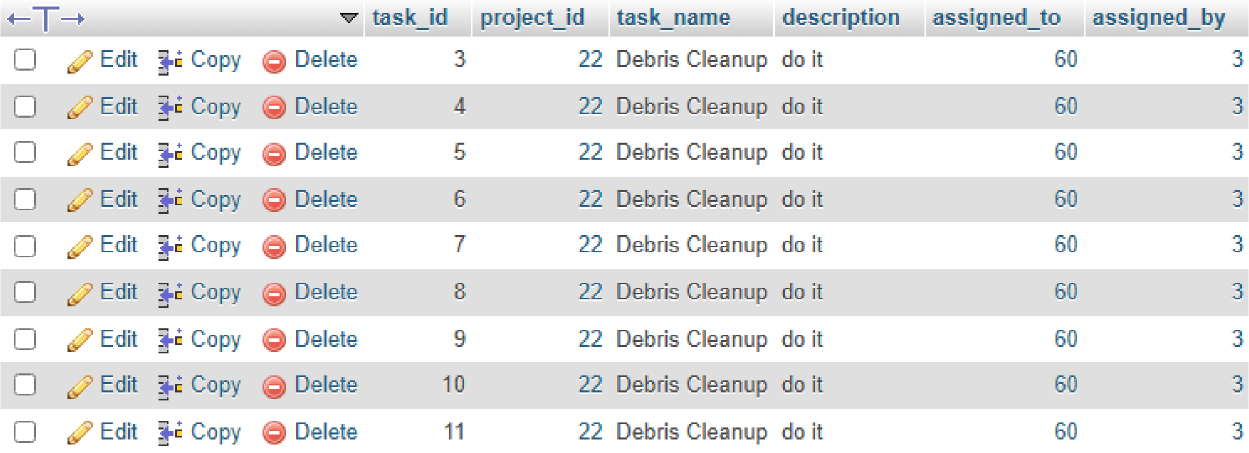
Project Assignments Table



←T→	assignment_id	project_id	user_id	assigned_date	role_in_project	contractor_id	description
□ Edit Copy Delete	12	6	2	2025-04-19 09:56:56	Assigned Contractor	44	Hi, how are you?
□ Edit Copy Delete	13	3	2	2025-04-19 10:09:32	Assigned Contractor	22	Do it
□ Edit Copy Delete	14	5	2	2025-04-19 10:11:44	Assigned Contractor	44	Do
□ Edit Copy Delete	15	4	2	2025-04-19 10:21:57	Assigned Contractor	44	Do it Ok!
□ Edit Copy Delete	16	4	2	2025-04-19 10:26:07	Assigned Contractor	44	Do it Ok!
□ Edit Copy Delete	17	7	2	2025-04-19 10:27:40	Assigned Contractor	44	DO
□ Edit Copy Delete	18	7	2	2025-04-19 10:27:47	Assigned Contractor	44	DO

figure 5. 10 project assignment table

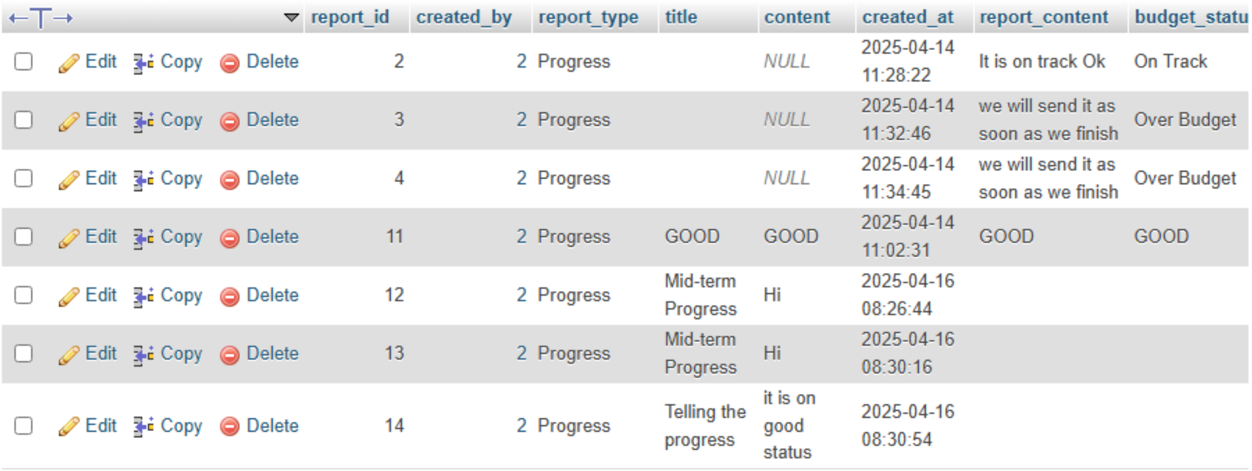
Tasks Table



←T→		▼	task_id	project_id	task_name	description	assigned_to	assigned_by
☐	Edit Copy	Delete	3	22	Debris Cleanup	do it	60	3
☐	Edit Copy	Delete	4	22	Debris Cleanup	do it	60	3
☐	Edit Copy	Delete	5	22	Debris Cleanup	do it	60	3
☐	Edit Copy	Delete	6	22	Debris Cleanup	do it	60	3
☐	Edit Copy	Delete	7	22	Debris Cleanup	do it	60	3
☐	Edit Copy	Delete	8	22	Debris Cleanup	do it	60	3
☐	Edit Copy	Delete	9	22	Debris Cleanup	do it	60	3
☐	Edit Copy	Delete	10	22	Debris Cleanup	do it	60	3
☐	Edit Copy	Delete	11	22	Debris Cleanup	do it	60	3

figure 5. 11 task table

Reports Table



←T→		▼	report_id	created_by	report_type	title	content	created_at	report_content	budget_statu
☐	Edit Copy	Delete	2	2	Progress		NULL	2025-04-14 11:28:22	It is on track Ok	On Track
☐	Edit Copy	Delete	3	2	Progress		NULL	2025-04-14 11:32:46	we will send it as soon as we finish	Over Budget
☐	Edit Copy	Delete	4	2	Progress		NULL	2025-04-14 11:34:45	we will send it as soon as we finish	Over Budget
☐	Edit Copy	Delete	11	2	Progress	GOOD	GOOD	2025-04-14 11:02:31	GOOD	GOOD
☐	Edit Copy	Delete	12	2	Progress	Mid-term Progress	Hi	2025-04-16 08:26:44		
☐	Edit Copy	Delete	13	2	Progress	Mid-term Progress	Hi	2025-04-16 08:30:16		
☐	Edit Copy	Delete	14	2	Progress	Telling the progress	it is on good status	2025-04-16 08:30:54		

figure 5. 12 report table

- ➢ The database schema supports the complex workflows and role-based access control required by the Construction Management System, allowing for efficient data storage, retrieval, and management across all construction project modules.

CHAPTER SIX

CONCLUSION AND RECOMMENDATION

6.1 Conclusion

In conclusion our Construction Management System (CMS) was developed with the primary objective of creating an integrated digital platform to streamline construction project management processes. We aimed to develop a comprehensive system that would facilitate efficient communication between project managers, site engineers, contractors, and consultants while providing tools for project planning, task assignment, progress tracking, and reporting. The goal was to replace fragmented manual processes with a cohesive digital solution that enhances productivity, transparency, and accountability in construction project management.

6.1.2 Summary of Key Findings and Results

The implementation of our Construction Management System has yielded several significant outcomes:

1. Streamlined Project Workflow: The system successfully established a structured workflow from project creation to completion, with clear role-based responsibilities and handoffs.

2. Enhanced Communication: The integrated dashboard interfaces for different user roles (managers, engineers, contractors, consultants) have significantly improved information sharing and reduced communication gaps.

3. Improved Documentation: The centralized storage of project documents, reports, and attachments has eliminated document loss and version control issues that were common in paper-based systems.

4. Real-time Progress Tracking: The reporting system allows for timely updates on project status, enabling proactive management of potential delays or issues.

5. Accountability: The system's tracking features have increased accountability among all stakeholders by clearly documenting task assignments and completion status.

6.1.3 Discussion of Implications

The successful implementation of the CMS has several important implications for construction project management:

1. Operational Efficiency: The digitization of project management processes has reduced administrative overhead and allowed team members to focus more on their core responsibilities.

2. Data-Driven Decision Making: The centralized collection of project data enables more informed decision-making based on accurate, up-to-date information.

3. Resource Optimization: Better visibility into project timelines and resource allocation has led to more efficient use of human and material resources.

4. Quality Improvement: The structured approach to task assignment and reporting has contributed to improved quality control and adherence to project specifications.

6.1.4 Acknowledgment of Limitations

Despite the system's success, we acknowledge several limitations in the current implementation:

1. Mobile Accessibility: While the system is responsive, a dedicated mobile application would provide better accessibility for on-site personnel.

2. Integration Capabilities: The system currently lacks integration with other enterprise systems such as accounting or inventory management software.

3. Advanced Analytics: The reporting functionality is primarily descriptive rather than predictive, limiting proactive risk management capabilities.

4. Offline Functionality: The system requires internet connectivity, which can be problematic in remote construction sites with limited connectivity.

5. Scalability Concerns: As project volume increases, additional optimization may be needed to maintain system performance.

6.1.5 Suggestions for Future Work

Based on our experience and the identified limitations, we propose the following areas for future development:

1. Mobile Application Development: Creating dedicated mobile apps for iOS and Android to enhance on-site usability.

2. API Development: Building a comprehensive API layer to enable integration with other business systems.

3. Advanced Analytics Implementation: Incorporating predictive analytics and business intelligence features to enhance decision-making capabilities.

4. Offline Mode: Developing offline functionality that synchronizes when connectivity is restored.

5. Expanded Document Management: Enhancing the document management system with version control and collaborative editing features.

6. Automated Notifications: Implementing a more sophisticated notification system with customizable alerts and reminders.

7. Client Portal: Adding a dedicated portal for clients to monitor project progress without requiring full system access.

6.1.6 Closing Statement

The Construction Management System we have developed represents a significant step forward in digitalizing construction project management processes. By providing a centralized platform for communication, task management, and reporting, the system addresses many of the challenges faced in traditional construction management approaches.

6.2 Recommendation:

1. Practical Suggestions

Based on the development and implementation of the Construction Management System, we recommend the following practical actions for Salale University:

1. **Establish a Dedicated System Administration Team**: Appoint and train a dedicated team responsible for system maintenance, user support, and ongoing development of the CMS.

2. **Implement a Comprehensive Training Program**: Develop role-specific training materials and conduct regular training sessions for all user types to ensure maximum adoption and effective use of the system.
3. **Create a Feedback Mechanism**: Establish a structured process for collecting user feedback to inform future system improvements and address any usability issues.
4. **Develop Standard Operating Procedures**: Document standard procedures for using the CMS in different construction project scenarios to ensure consistency and compliance.
5. **Implement Regular Data Backup and Recovery Protocols**: Establish robust data backup procedures to protect against data loss and ensure business continuity.

2. Recommendations Stemming Directly from Project Findings

1. **Expand System Integration**: Based on the identified limitation of isolated systems, we recommend that Salale University prioritize the integration of the CMS with existing financial and resource management systems.
2. **Enhance Mobile Accessibility**: Given the nature of construction work and the need for on-site access, we recommend developing enhanced mobile capabilities, potentially including a dedicated mobile application.
3. **Implement Advanced Analytics**: The current reporting capabilities, while functional, could be significantly enhanced with data analytics tools to provide deeper insights into project performance and trends.
4. **Establish a Phased Rollout Strategy**: To address the complexity of full system adoption, we recommend implementing a phased approach to system rollout across different departments and project types.
5. **Develop a Sustainability Plan**: To ensure long-term system viability, develop a plan for ongoing maintenance, updates, and funding for the CMS.

3. Guidance on Actions Based on Project Findings

1. **Technology Infrastructure Assessment**: Conduct a comprehensive assessment of the university's IT infrastructure to ensure it can support the CMS effectively, particularly in remote construction sites.
2. **Policy Development**: Develop and implement policies governing the use of the CMS, including data privacy, security protocols, and user responsibilities.
3. **Performance Metrics Establishment**: Define key performance indicators to measure the effectiveness of the CMS in improving construction project outcomes.

4. **Change Management Strategy**: Implement a structured change management approach to address resistance and ensure smooth transition from manual to automated processes.
5. **Continuous Improvement Framework**: Establish a framework for ongoing evaluation and improvement of the CMS based on user feedback and evolving university needs.

4. Recommendations for Salale University Stakeholders

1. **For University Administration**:
   - Allocate sufficient resources for system maintenance and enhancement
   - Use the CMS data for strategic planning of campus development
   - Consider the CMS as a model for digitizing other administrative processes
2. **For the Facilities Management Department**:
   - Leverage the CMS to standardize construction project management practices
   - Use the system's reporting capabilities to improve accountability and transparency
   - Integrate the CMS into the department's overall workflow
3. **For Project Managers and Contractors**:
   - Fully utilize the collaboration features to improve communication
   - Contribute to system improvement through regular feedback
   - Adopt the standardized processes facilitated by the CMS
4. **For the IT Department**:
   - Provide technical support for the CMS
   - Ensure system security and data protection
   - Collaborate with system users to identify and implement enhancements
5. **For Academic Departments**:
   - Consider incorporating the CMS as a case study in relevant courses
   - Explore research opportunities related to construction management technology
   - Use the system as a practical learning tool for students in construction-related fields

By implementing these recommendations, Salale University can maximize the benefits of the Construction Management System, address current limitations, and position itself as a leader in the application of technology for efficient infrastructure.

REFERENCES

[1] S. University, "Slu.edu.et," [Online]. Available: https://www.slu.edu.et. [Accessed 24 October 2024].

[2] [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/Javascript/Objects /object-oriented_programming. [Accessed 25 October 2024].

[3] [Online]. Available: https://search.brave.com/images?q=agile%20model. [Accessed 16 October 2024].

[4] M. C. A. a. E. PLC, "Salale University Monthly Progress Report," 2024 .

[5] B. Solomon, H. Mola and T. Deressa, "Online Research and project management system for Salale University," Fiche Salale, Ethiopia , 2024.

[6] [Online]. Available: https://sequencediagram.org/. [Accessed 24 December 2024].

[7] [Online]. Available: https://www.infoq.com/articles/iot-key-abstractions/. [Accessed 5 January 2025].

[8] [Online]. Available: https://www.cs.cmu.edu/afs/cs/project/classes-bob-1/owl-f1996/ group/architecture/SDD/SysDecomp.html. [Accessed 20 January 2025].

[9] "What is Deployement diagram," [Online]. Available: https://www.visual-paradigm.com/ guide/uml-unified-modeling-language/what-is-deployment-diagram/. [Accessed 22 January 2025].

[10] "Access Control," [Online]. Available: https://search.brave.com/search? q=Access+control+in+Er+diagram&source=web&summary=1&conversation=c1fbf6e4e430f8556e1cf3. [Accessed 25 January 2025].

APPENDIX

INTERVIEWEE

Name: -Mr. Firew Balcha.

Date:- 24 Decemeber 2024.

Interview Questions

1. How is Construction Management currently working?
2. How is the project data stored?
3. What are the limitations of the current system?
4. What is your current process for repository progress on tasks?
5. How are you assigning currently assigned tasks?
6. How do you report progress on tasks?
7. How many actors are there within the Salale university construction Management system?
8. What are the roles of each actor and ways of the roles of each actor and with the system?