

**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 Kelem** | **RU1543** | **ephratake@gmail.com** | **+251940468949** |
| **3** | **Wakuma Teshome** | **RU0036** | **wakegizaw@gmail.com** | **+251941344994** |

**Advisor: Mr. sDiriba. R**

**January, 2025**

**Fiche, Oromia, Ethiopia**

# DECLARATION

As required by the final year project I documentation, we hereby declare that the project documentation, "Construction Management System for 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 \_\_\_\_\_\_\_\_\_\_\_\_\_

1. Ephrata Kelemu Ru 1543/14 \_\_\_\_\_\_\_\_\_\_\_\_\_

1. 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

[DECLARATION I](#_Toc189034043)

[APPROVAL II](#_Toc189034044)

[ACKNOWLEDGMENT III](#_Toc189034045)

[TABLE OF CONTENTS iv](#_Toc189034046)

[LIST OF FIGURES vii](#_Toc189034047)

[LIST OF TABLES viii](#_Toc189034048)

[LIST OF ACRONYMS ix](#_Toc189034049)

[INTRODUCTION 1](#_Toc189034050)

[1.1. Background of the Study 1](#_Toc189034051)

[1.2. Description of The Existing System 1](#_Toc189034052)

[1.3 STATEMENT OF THE PROBLEM 2](#_Toc189034053)

[1.4 OBJECTIVE 3](#_Toc189034054)

[1.4.1 General objective 3](#_Toc189034055)

[1.4.2 Specific objective 3](#_Toc189034056)

[1.5 METHODOLOGY 3](#_Toc189034057)

[1.5.1 Data collection technique 3](#_Toc189034058)

[1.5.1.1 Primary Data Collection Methods: 3](#_Toc189034059)

[1.5.2 System Analysis and Design Methodology 4](#_Toc189034060)

[1.5.3 System Development Model 5](#_Toc189034061)

[1.6 FEASIBILITY OF THE PROJECT 6](#_Toc189034062)

[1.6.1 Technical Feasibility 6](#_Toc189034063)

[1.6.2 Economic Feasibility 6](#_Toc189034064)

[1.6.3 Operational Feasibility 6](#_Toc189034065)

[1.6.4 Legal Feasibility 6](#_Toc189034066)

[1.6.5 Schedule feasibility 6](#_Toc189034067)

[1.7 SCOPE OF THE PROJECT AND LIMITATION 7](#_Toc189034068)

[1.7.1 Scope of the project 7](#_Toc189034069)

[1.7.2 Limitation of our Project 8](#_Toc189034070)

[1.8 SIGNIFICANCE OF THE PROJECT 8](#_Toc189034071)

[1.9 ORGANAZATION OF THE PROJECT 9](#_Toc189034072)

[CHAPTER TWO 10](#_Toc189034073)

[2.1.1. Existing System Description 10](#_Toc189034074)

[2.2. New System 12](#_Toc189034075)

[2.2.1. New System Business Rules 12](#_Toc189034076)

[2.2.3. Non-Functional Requirements 15](#_Toc189034077)

[2.2.4 Actors Identification 16](#_Toc189034078)

[2.2.5 Use case diagram 17](#_Toc189034079)

[2.2.6 Use Case Description 18](#_Toc189034080)

[2.2.6 Sequence Diagram 32](#_Toc189034081)

[2.2.7 State chart Diagram 37](#_Toc189034082)

[2.2.8 Activity Diagram 39](#_Toc189034083)

[2.3 Key Abstraction with CRC Analysis 41](#_Toc189034084)

[2.4 Conceptual modeling class diagram 42](#_Toc189034085)

[2.5 Identifying change Cases 43](#_Toc189034086)

[2.6 User Interface Prototyping 43](#_Toc189034087)

[CHAPTER 3 OBJECT ORIENTED SYSTEM DESIGN 45](#_Toc189034088)

[3.1 Purpose and Goal of The Design 45](#_Toc189034089)

[3.1.1 Purpose of the System 45](#_Toc189034090)

[3.1.2 Goals of the System Using Various Criteria 45](#_Toc189034091)

[3.2 Current Software Architecture 46](#_Toc189034092)

[3.3 Proposed Software Architecture 47](#_Toc189034093)

[3.3.1 Subsystem decomposition 47](#_Toc189034094)

[3.3.2 Component diagram 48](#_Toc189034095)

[3.3.3. Deployment diagram 49](#_Toc189034096)

[3.4 Database design (E-R diagram) for relational database 51](#_Toc189034097)

[3.5 Persistence modelling for object-oriented database 52](#_Toc189034098)

[3.6 Access Control 52](#_Toc189034099)

[REFERENCES 55](#_Toc189034100)

[APPENDIX 56](#_Toc189034101)

LIST OF FIGURES

[Figure 1.1 Agile Model . 5](#_Toc187964914)

[Figure 1.2 Gantt chart 7](#_Toc187964915)

[Figure 2. 1 Existing system 11](#_Toc188952473)

[Figure 2. 2 Existing system report 11](#_Toc188952474)

[Figure 2. 3 Use case Diagram for the Proposed system 18](#_Toc188952475)

[Figure 2. 4 Sequence diagram for Login 33](#_Toc188952476)

[Figure 2. 5 Sequence diagram for Generate report 34](#_Toc188952477)

[Figure 2. 6 Sequence diagram for Registration 35](#_Toc188952478)

[Figure 2. 7 Sequence diagram for view task 36](#_Toc188952479)

[Figure 2. 8 sequence diagram for logout 36](#_Toc188952480)

[Figure 2. 9 State chart diagram for Login 37](#_Toc188952481)

[Figure 2. 10 state chart diagram for Assign contractor 38](#_Toc188952482)

[Figure 2. 11 state chart diagram for Generate report 38](#_Toc188952483)

[Figure 2. 12 Activity diagram for Login 39](#_Toc188952484)

[Figure 2. 13 Activity diagram for Assign contractor 40](#_Toc188952485)

[Figure 2. 14 Activity diagram for View task 40](#_Toc188952486)

[Figure 2. 15 Key Abstraction with CRC analysis 41](#_Toc188952487)

[Figure 2. 16 Conceptual Model of class diagram 42](#_Toc188952488)

[Figure 2. 17 User Interface Prototype 44](#_Toc188952489)

[Figure 3. 1 Subsystem Decomposition 48](#_Toc188642920)

[Figure 3. 2 Component Diagram 49](#_Toc188642921)

[Figure 3. 3 Deployment Diagram 50](#_Toc188642922)

[Figure 3. 4 E-R Diagram 51](#_Toc188642923)

# LIST OF TABLES

[Table 2. 1 Use case description for login………………………………………………………...19](#_Toc188644901)

[Table 2. 2 Use case description for Manage account 20](#_Toc188644902)

[Table 2. 3 Use case description for Approve projects 21](#_Toc188644903)

[Table 2. 4 Use case description for Approve budget 22](#_Toc188644904)

[Table 2. 5 Use case description for Generate report 23](#_Toc188644905)

[Table 2. 6 Use case description for Schedule project 24](#_Toc188644906)

[Table 2. 7 Use case description for Assign task 25](#_Toc188644907)

[Table 2. 8 Use case description for Prepare report 26](#_Toc188644908)

[Table 2. 9 Use case description for Submit task 27](#_Toc188644909)

[Table 2. 10 Use case description view task 27](#_Toc188644910)

[Table 2. 11 Use case description for Manage project 28](#_Toc188644911)

[Table 2. 12 Use case description for view schedule 29](#_Toc188644912)

[Table 2. 13 Use case description for manage Employee 30](#_Toc188644913)

[Table 2. 14 Use case description for Register Employee 31](#_Toc188644914)

[Table 3. 1 Activity Control……….…………………………………………………………….. 52](#_Toc188644804)

# LIST OF ACRONYMS

**CMS**: Construction Management System

**CRC**: Class Responsibility Collaborator

**CPU**: Central Processing Unit

**DBMS**: Database Management System

**ETB:** Ethiopian Birr

**HTML**: Hypertext Markup Language

**MP:** Megapixel

**OOSAD**: Object-Oriented System Analysis and Design

**PDF**: Portable Document Format

**PHP**: Hypertext Preprocessor

**RAM:** Random Access Memory

**ROI:** Return on Investment

**UML**: Unified Modeling Language

**USB:** Universal Serial Bus

**XAMPP:** Cross platform Apache MySQL PHP Perl

# INTRODUCTION

* 1. Background of the Study

Salale University is located 114 kilometers from Addis Ababa, the African 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, budgeting, and scheduling [1].

Traditionally, construction management at Salale University relies on manual processes and fragmented digital tools, leading to inefficiencies such as delays, miscommunication, budget overruns, 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].

## 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 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 our 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 functional and non-functional requirements from specific stakeholders.
* 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**: To gather foundational and background information about the current construction management practices, team members conducted interviewed Mr Friew who works in Construction Department, university administrative staff and Head of Engineering college. This provided insights into the services offered, common challenges, and specific requirements 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.

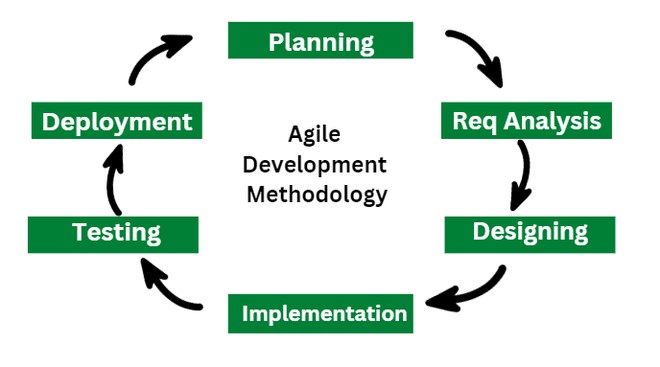


Figure 1.1 Agile Model [3].

# 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 assessed development, deployment, and maintenance costs against the anticipated benefits, such as improved efficiency in project management. The project provides a positive return on investment (ROI).

## 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 OF THE PROJECT AND LIMITATION

## 1.7.1 Scope of the project

The scope of this project focuses on developing a Web-Based Construction Management System specifically designed for Salale University, Tadesse Birru Campus. 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.
* Facilitate role-based communication and collaboration among stakeholders (e.g., Admin, Project Manager, Contractors, and Employees).
* Assign tasks to specific team members, track their progress, and monitor task completion timelines.
* Generate detailed reports on project progress, task completion, resource utilization, and financial status.
* 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
* Our project doesn’t apply to other campus of Salae University only Taddesse biru campus

# 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 Communication:** Organized, efficient communication channels among stakeholders prevent misunderstandings, accelerate decision-making
* **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 budgets, enabling proactive management and preventing delays and cost overruns.

Generally, the proposed system is used to:

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

# 1.9 ORGANAZATION OF THE PROJECT

Our Project is Grouped into six chapters, we have only arranged the documentation from chapter one up to chapter three into three-chapter conclusion. 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 backend which explains the object-oriented graphical presentation.

# 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. Existing system [4]

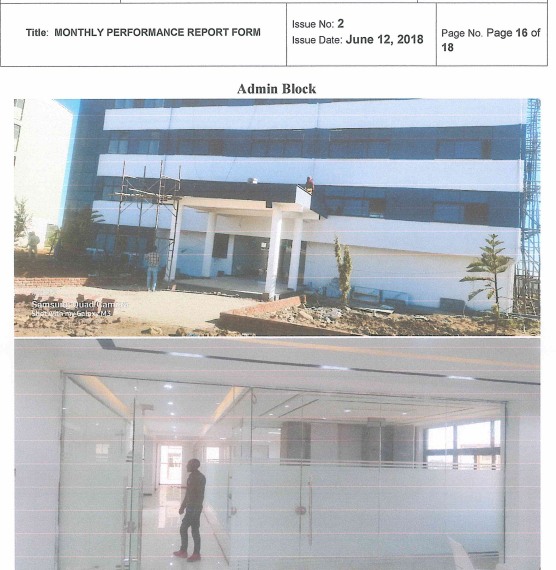
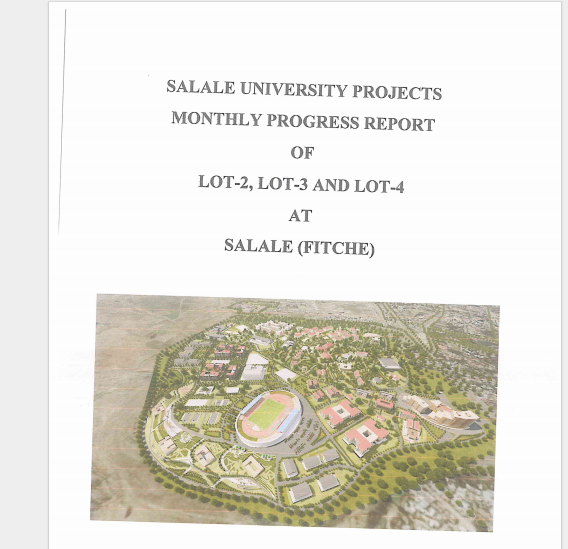


Figure 2. Existing system report [4]

# 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.

s**BR3:** An admin must have the correct credentials to log in and manage user access to different parts of the system.  
**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 (such as project managers, contractors, and Admins) 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, role) as necessary.
* **Delete account**: Users with appropriate privileges (Admins) should be able to deactivate or delete accounts.

**Project Data Management**

* **Register project**: Project managers must be able to register new construction projects, including project name, budget, timeline, location, and team members.
* **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.
* **Delete project**: Admins must have the ability to remove projects that are no longer needed or completed.

**Task Management**

* **Assign tasks**: Project managers and other eligible actors should assign specific tasks to the appropriate team members. (e.g., contractors) and set deadlines for completion.
* **Monitor task progress**: The system must allow the tracking of task completion, providing real-time status updates on assigned tasks.

**Communication and Collaboration**

* **Facilitate communication**: The system should allow seamless communication between stakeholders.
* **Task management**: The system should allow authorized users to assign tasks to team members, track progress, and collaborate on task completion through task comments or updates.

**Budget and Resource Management**

* **Track budget**: The system must monitor budget allocations for each project to prevent budget overruns and ensure accurate financial tracking.
* **Generate budget reports**: The system should allow generating detailed reports on the budget and expenses for each project.

**Document Management and Reporting**

* **Store project documents**: The system must provide a unified platform to store and organize all project-related documents (e.g., contracts, inspection reports).
* **Generate reports**: Users should be able to generate various reports related to project progress, budget status, and resource allocation.

### ****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 for all roles (Project Manager, Contractor, etc.)
* Update user account information (e.g., permissions, contact details)
* Deactivate or delete user accounts
* Generate System Reports
* Define and manage roles and their associated permissions
* Approve project updates submitted by Project Managers

**Project Manager**:

* Create new projects
* Manage project details (e.g., budget, schedule, milestones)
* Assign tasks to specific contractors
* Monitor project budget
* Generate progress reports

**Contractor**:

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

**Site Engineer:**

* **Manage Daily Labor**
* **Prepare Report**

**Consultant:**

* **View 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].

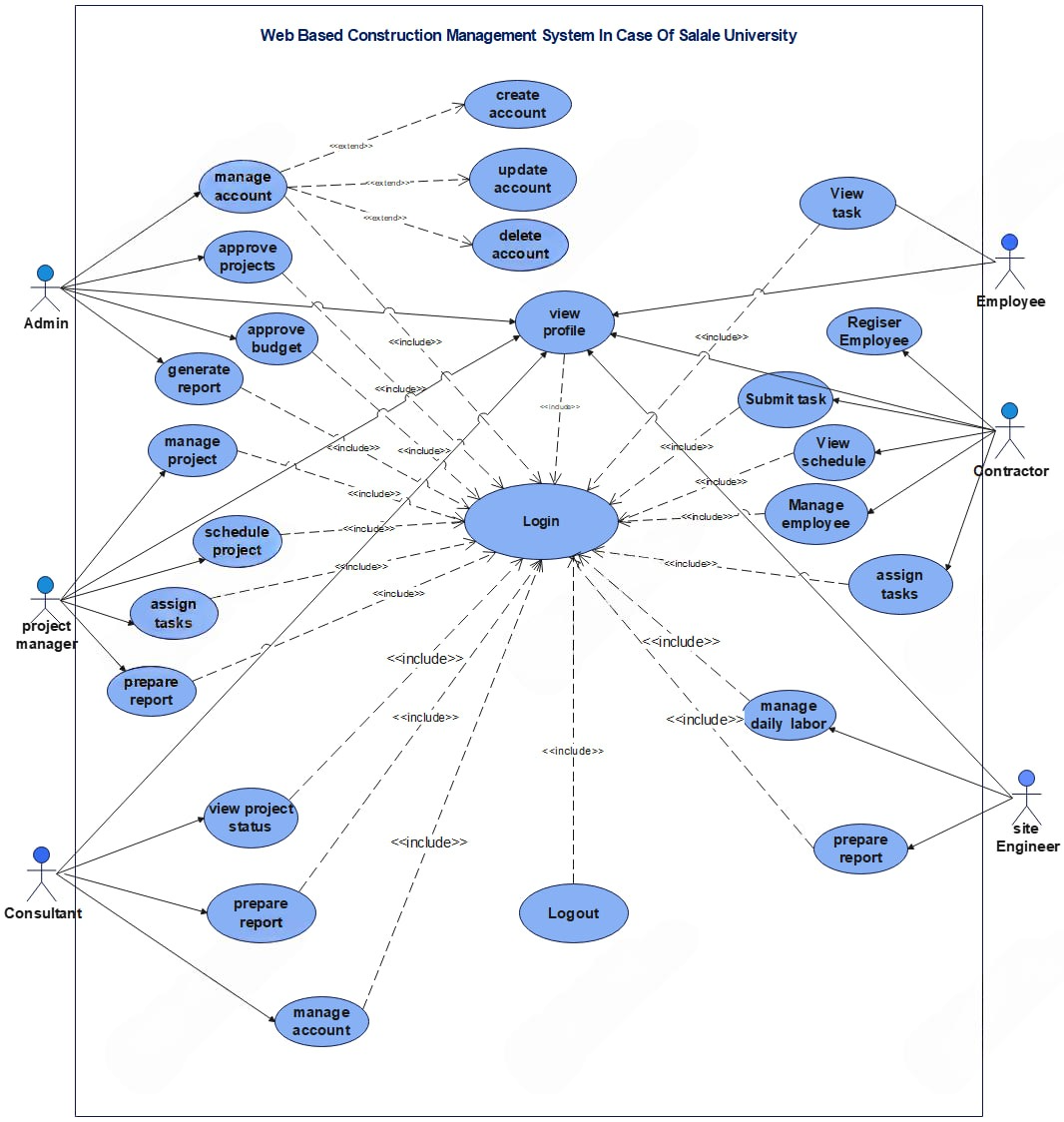


Figure 2. 3 Use case Diagram for the Proposed system

### 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. 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. 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. Use case description for Approve projects

|  |  |  |
| --- | --- | --- |
| Use case name | **Approve Projects** | |
| 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. Use case description for Approve budget

|  |  |  |
| --- | --- | --- |
| Use case Name | **Approve budget** | |
| Use Case Id | UC#04 | |
| Descriptions | This use case describes the process of approving a budget for a construction project in a web-based construction management system. The process involves a project manager or Admin reviewing and approving a submitted budget | |
| Actors | Admin | |
| Pre conditions | The users must be authenticated and logged into the system, and has the necessary permissions to Approve Budget. | |
| Basic course of Action: | Users Action | System Response |
| Step 1: The user selects the "Budgets" or "Approval" menu option.  Step 3: The user clicks on a specific budget entry to review its details.  Step 5: The user reviews the budget details and clicks the "Approve" button. | Step 2: Displays a list of submitted budgets awaiting approval.  Step 4: Loads and displays the budget details, including line items, total amounts.  Step 6: Prompts the user to confirm the approval action. |
| Alternative course of Action | Prompts the user to provide reasons for the modification request and sends the feedback to the project team. | |
| Post condition | The system logs if approved, rejected, or deferred where the budget remains in pending awaiting further action. | |

Table 2. Use case description for Generate report

|  |  |  |
| --- | --- | --- |
| Use case Name | **Prepare Report** | |
| Use Case Id | UC#05 | |
| Descriptions | This functionality allows authorized users to generate various reports, such as project progress, resource utilization, budget tracking, or employee productivity, to aid in construction project management. | |
| 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. 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. 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. Use case description for Prepare report

|  |  |  |
| --- | --- | --- |
| Use case name | **Prepare report** | |
| Use Case ID | UC#08 | |
| Actor | Project manager | |
| Description | Project Manager Prepare report to Admin. | |
| 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 "Prepare Report" option. | Step 5: System generates the report based on the selected template and entered data. |
| Step 4: Project Manager selects the desired report type (e.g., project progress report, resource utilization report, financial report). | Step 8: Use case end |
| Step-6 Project Manager reviews the generated report. |
| Step-7: Project Manager submits the report to the Admin |
| Alternative course of action | The system verify information is not correctly, the system displays error message as:   * Not successful * The system redirects to step 4 * Use case ends | |
| Post condition | The task assignment will be generated. | |

Table 2. 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. 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 (t**ask 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. 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. 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. 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. 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.

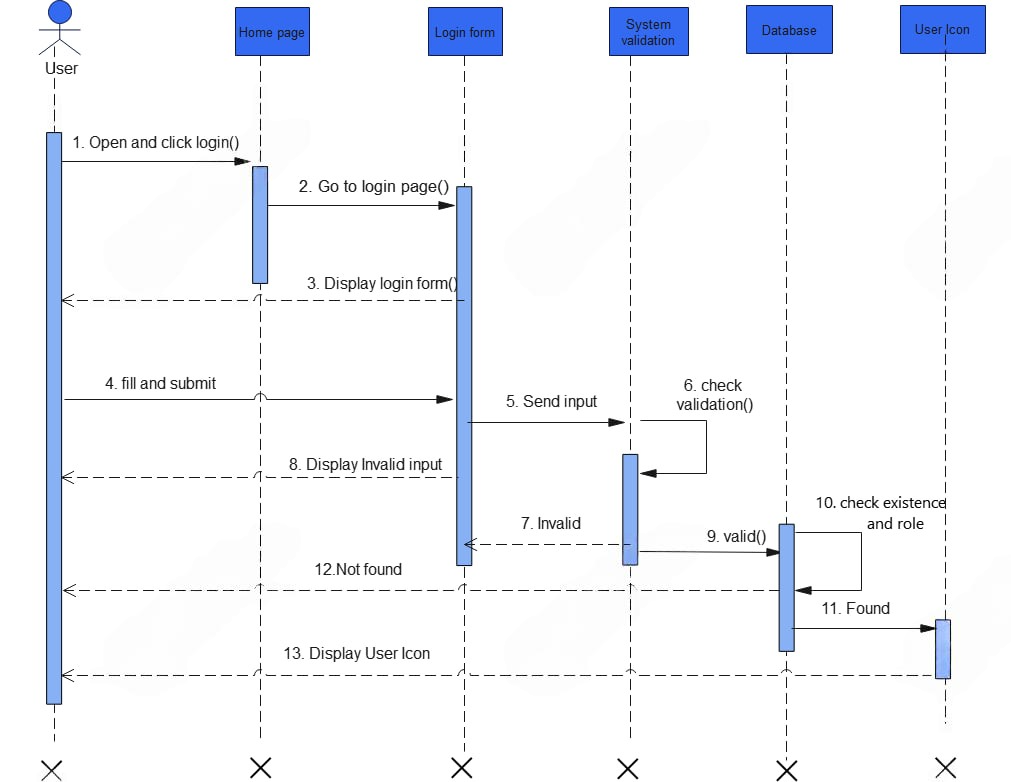


Figure 2. Sequence diagram for Login

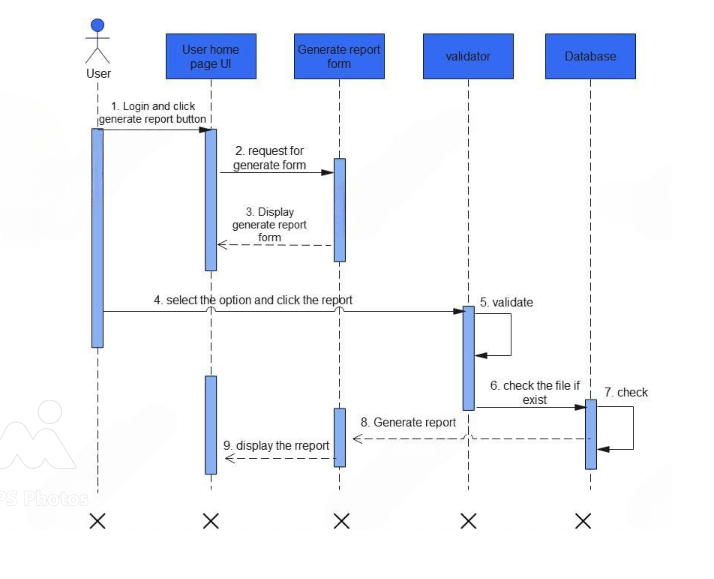


Figure 2. Sequence diagram for Generate report



Figure 2. Sequence diagram for Registration

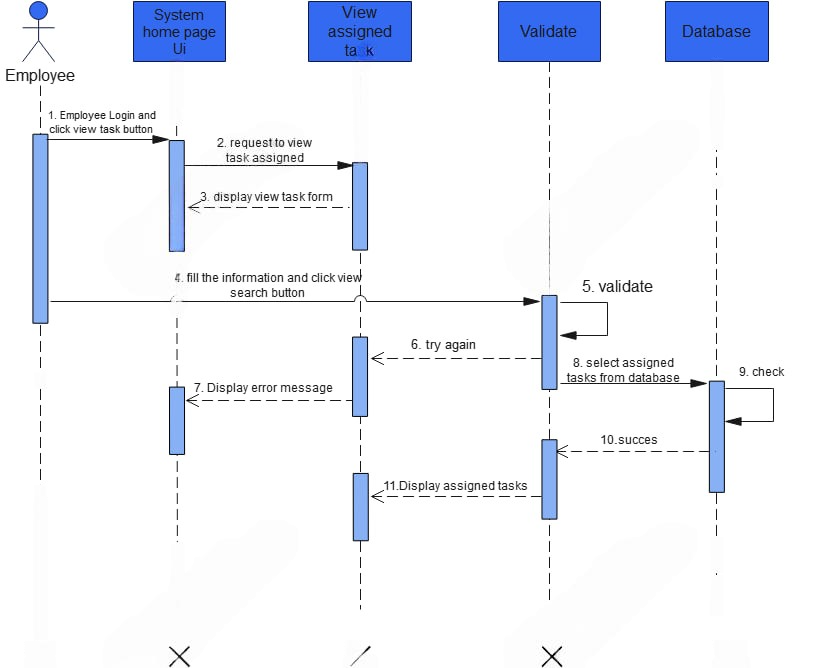


Figure 2. Sequence diagram for view task

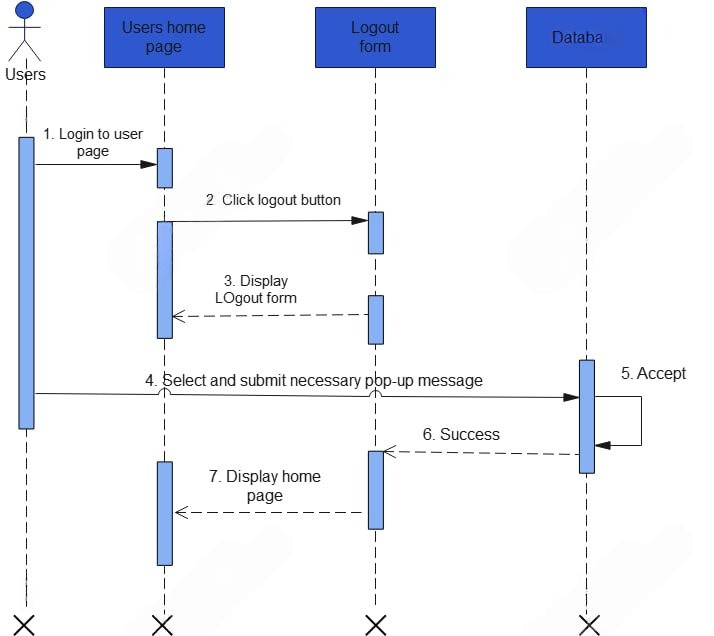


Figure 2. sequence diagram for logout

### 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

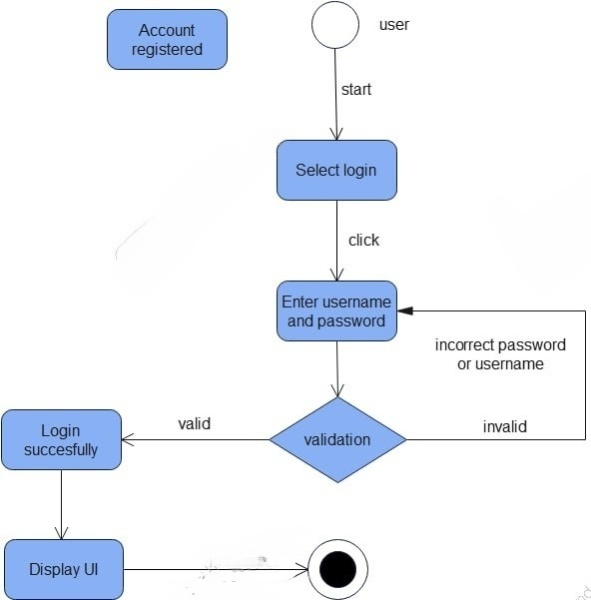


Figure 2. State chart diagram for Login

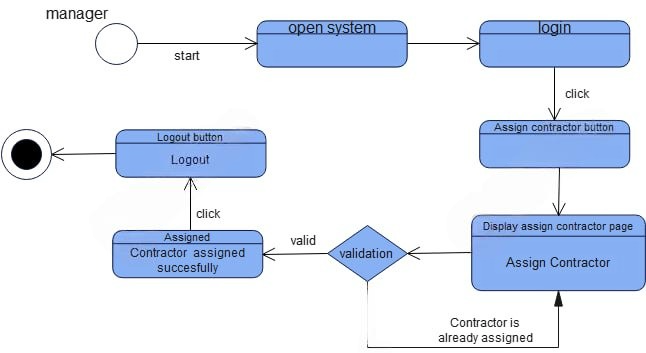


Figure 2. state chart diagram for Assign contractor

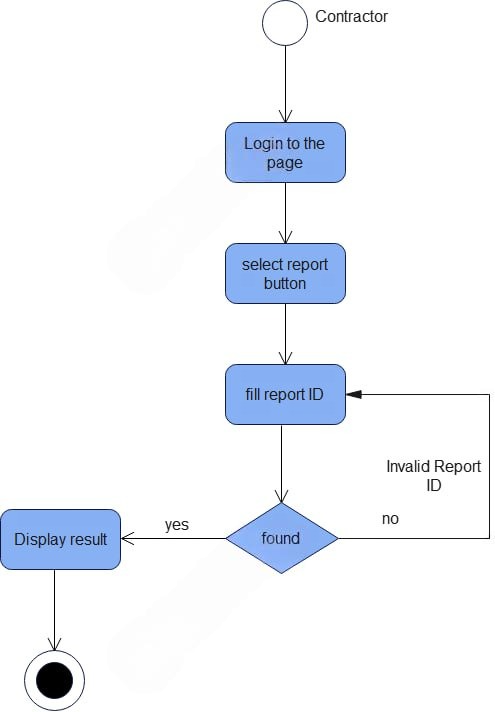


Figure 2. state chart diagram for Generate report

### 

### 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

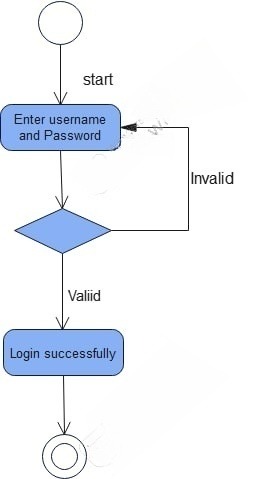


Figure 2. Activity diagram for Login

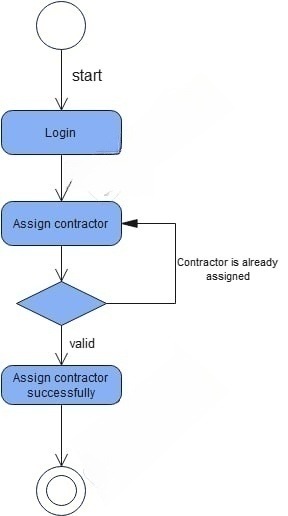


Figure 2. Activity diagram for Assign contractor

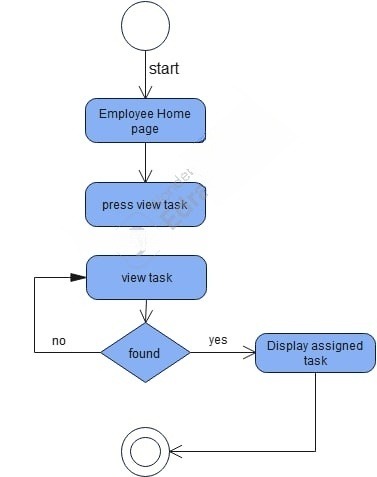


Figure 2. Activity diagram for View task

## 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].

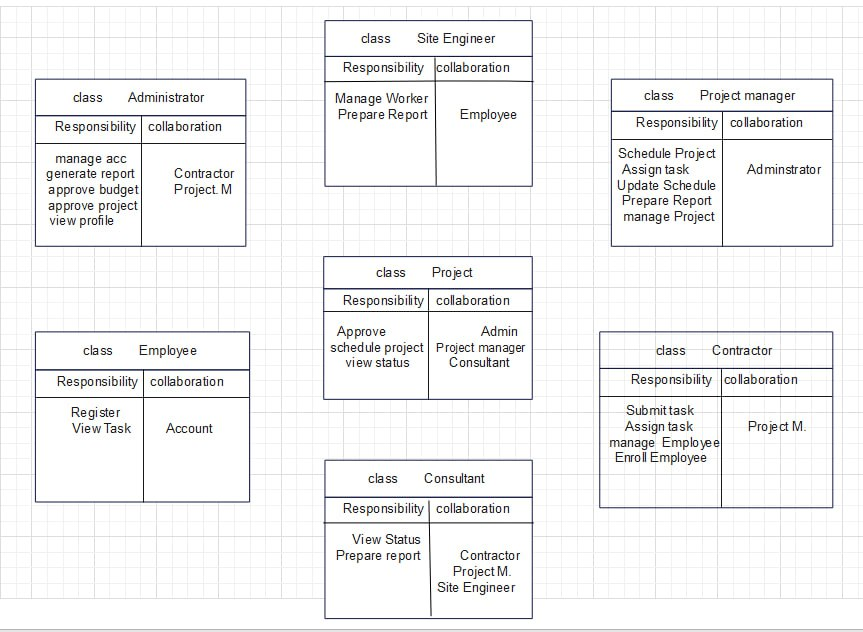


Figure 2. Key Abstraction with CRC analysis

## 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.

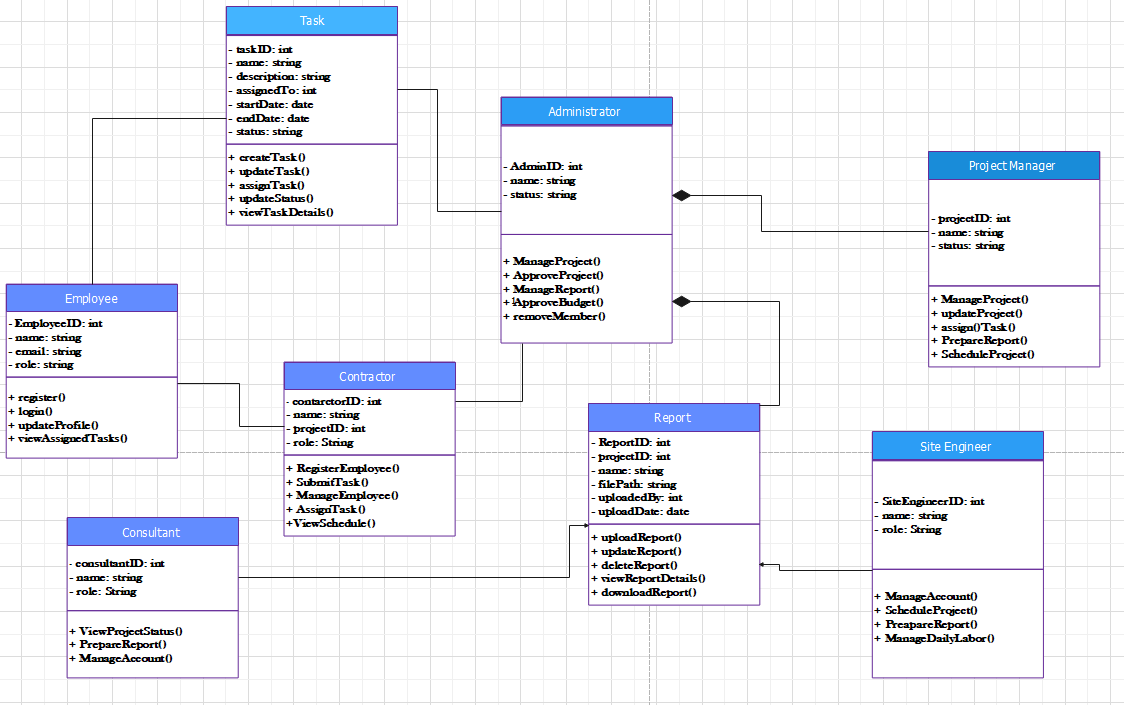


Figure 2. Conceptual Model of class diagram

## 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.

A web-based Construction Management System offers a significant improvement by providing a centralized platform for managing all aspects of construction projects. Key benefits include:

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

## 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.

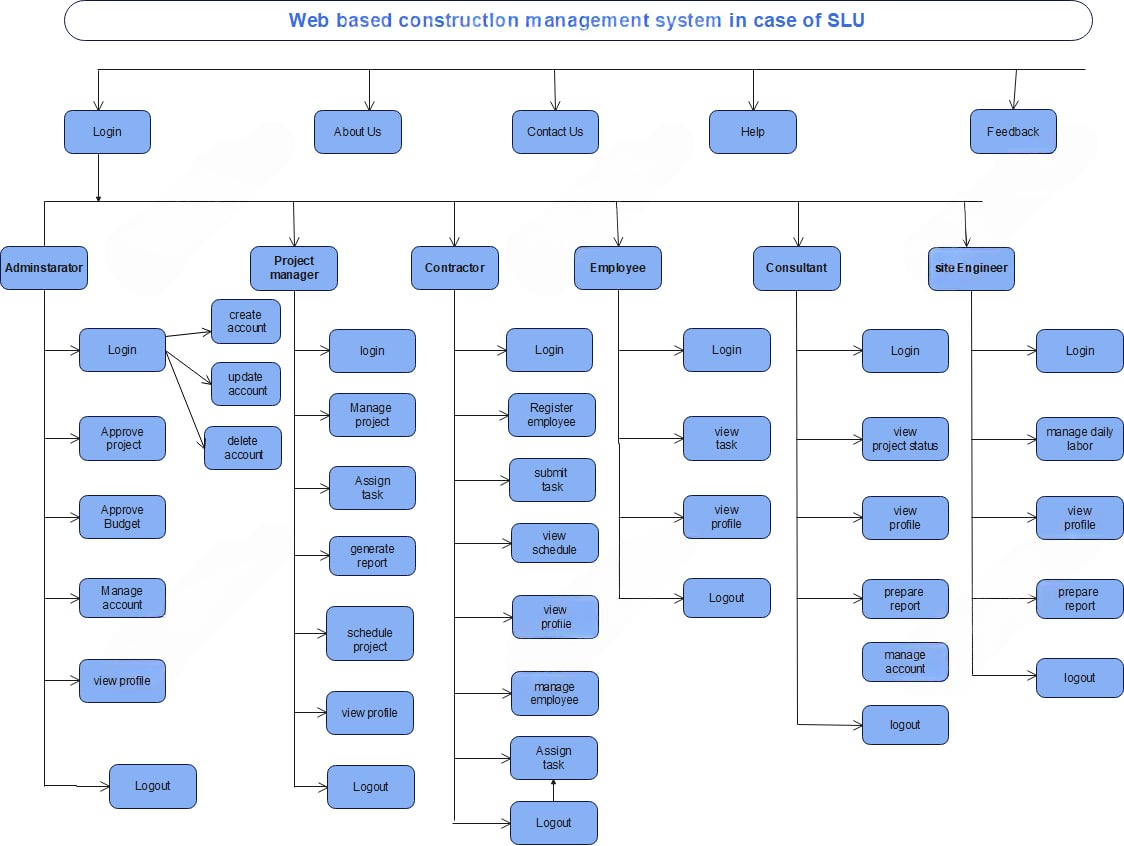


Figure 2. User Interface Prototype

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.

1. 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.
* Managing Growing Data Volume: As projects scale, the volume of data (e.g., project details, financial reports, employee information) will also increase. The system must handle large amounts of data efficiently.

## 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. 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.

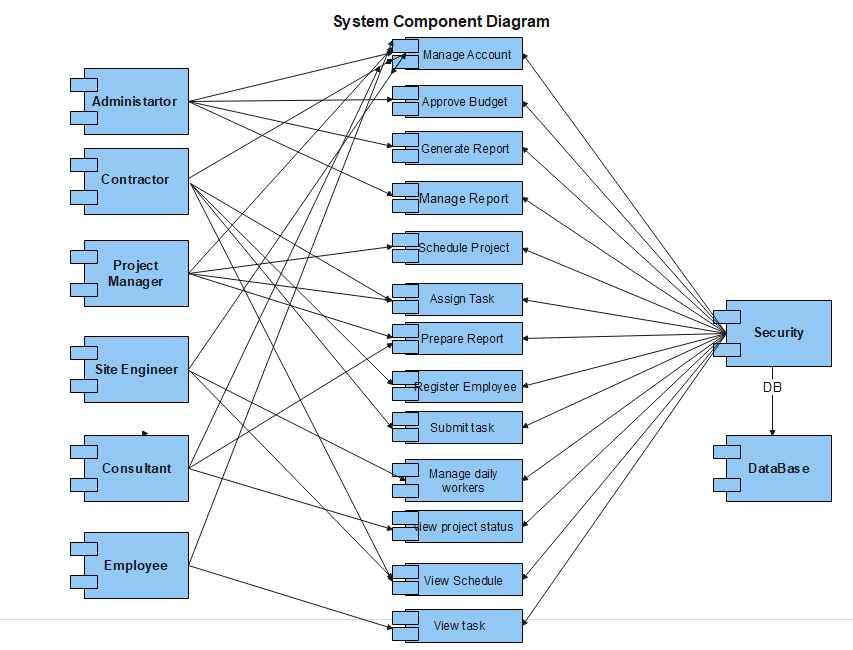


Figure 3. Component Diagram

### 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].

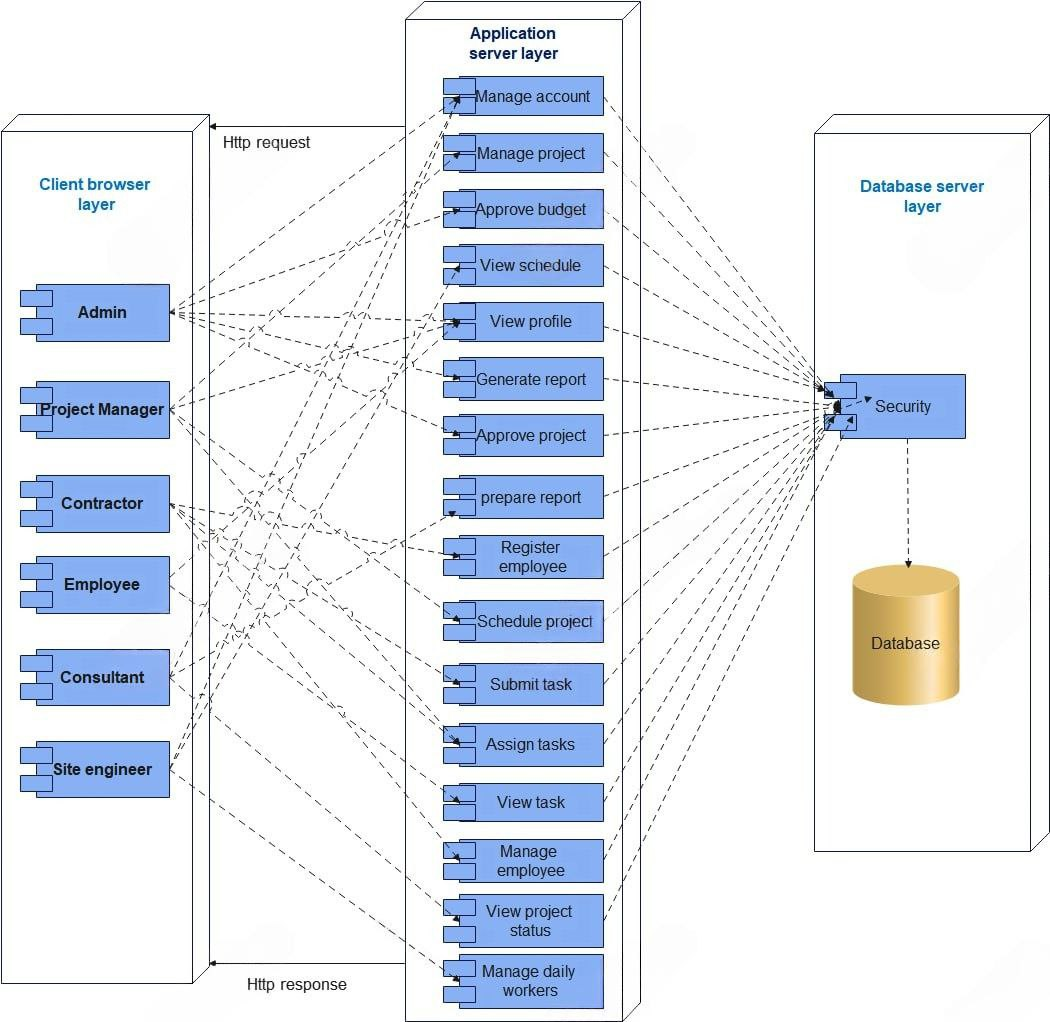


Figure 3. Deployment Diagram

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

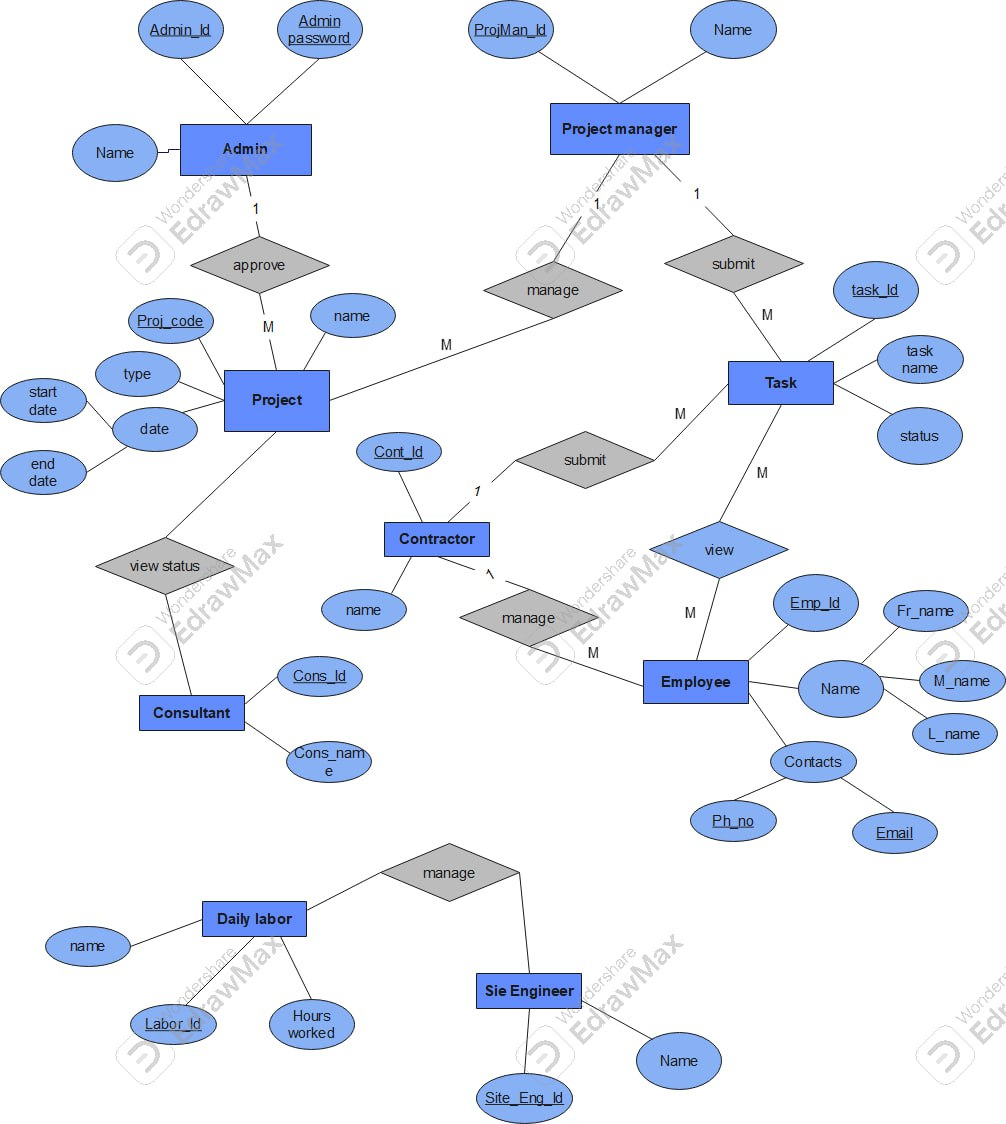
This relationship model defines data elements and their relationships within a system. It creates a conceptual database design and provides a simple, user-friendly data view.

Figure 3. 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

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:**

* Full access to account management (create, update, delete accounts).
* Approve projects and budgets.
* Generate reports.

**Project Manager:**

* Access to project and task management (manage, schedule projects, assign tasks).
* View project status.
* Prepare reports.

**Consultant:**

* Prepare reports.
* Manage own account information.

**Contractor:**

* Register employees.
* View tasks and schedules.
* Manage employees.
* Assign tasks to daily labor.

**Site Engineer:**

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

**Employee:**

* View assigned tasks.
* Submit completed tasks.

Table 3. Activity Control

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Actors | | | | | |
| Function | Admin | Project  manager | contractor | Employee | Consultant | Site Engineer |
| Approve project |  |  |  |  |  |  |
| Login |  |  |  |  |  |  |
| Generate report |  |  |  |  |  |  |
| Approve budget |  |  |  |  |  |  |
| Update user account |  |  |  |  |  |  |
| Manage project |  |  |  |  |  |  |
| Schedule project |  |  |  |  |  |  |
| Assign task |  |  |  |  |  |  |
| View task |  |  |  |  |  |  |
| View profile |  |  |  |  |  |  |
| Register employee |  |  |  |  |  |  |
| Submit task |  |  |  |  |  |  |
| View schedule |  |  |  |  |  |  |
| logout |  |  |  |  |  |  |
| Manage account |  |  |  |  |  |  |
| View project status |  |  |  |  |  |  |
| Manage daily labor |  |  |  |  |  |  |

# 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]. |
| [11] | [Online].Available:https://search.brave.com/search?q=conceptual+modeling+class+diagram&source=  desktop&summary=1&conversation=a935b9ad3c988c809994ed. [Accessed 6 Janauary 2025]. |

# APPENDIX

Interview Question

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?