Sri Lanka Institute of Information Technology

Construction Management System For
KSL Constructions (Pvt) Ltd

Project Report

Information Technology Project 2018

Project ID: **ITP-MLB-07**

Submitted by:

1.IT16165694 Nibraz M.M.
2.IT16170926 Abishalini K.
3.IT16170780 Thenuja S.
4.IT16165526 Wijeshinghe J.R.
5.IT16167124 Shangavie R.
6.IT16180444 Vanaja A.
7.IT16168800 Gunarathna P.M.S.S.B.
8.IT16010086 Samarawickrama Y.M.

Submitted to:

(Supervisor's signature)

……..

Name of the supervisor

18/05/2018
Date of submission

Abstract

KSL constructions is a major construction company located in Wallewatta, Colombo. The company is currently using a manual system to manage all contracts, client information and other management activities. Therefore, they have faced difficulties in handling their management sectors flawlessly. So their final decision to overcome those problems is move to a computerized system. So as a software development we have proposed to design a web based application to overcome their issues and make the management process a success. In developing the system, we are paying our attention toward accuracy, performance, security and most importantly efficiency. Our system will handle employees, sub-contractors, client databases. The system will be resulting increase the revenue of the company. The goal of the system is make the management process effective and easy to handle

Acknowledgement

I would like to express my deepest appreciation to all those who provided me the possibility to complete this project. A special gratitude I give to my group members and seniors who come with me from the start and till the end and shared resources that they got and gave suggestions and encouragement, helped me to coordinate my project especially in writing this document.

Furthermore I would also like to acknowledge with much appreciation the crucial role of the lecturer of ITP Ms. Janani Tharmaseelan, lecturer at the Sri Lanka Institute of Information Technology, who gave good explanations to done every single documents and project. Last but not least, many thanks go to the client of ours KSL Constructions whose have gave time and clear vision to us that helped the team in achieving the goal. I have to appreciate the guidance given by other supervisor as well as the panels especially in our project presentation that has improved our presentation skills thanks for their comment and advices.

Declaration

We declare that the this project report or part of it was not a copy of a document done by any organization, university any other institute or a previous student project group at SLIIT and was not copied from the Internet or other sources.

Project Details

Project Title	Construction Company Management System for KSL Constructions (Pvt) Ltd
Project ID	ITP_MLB_07

Group Members

Reg. No	Name	Signature
IT16165694	Nibraz M.M.	
IT16170926	Abishalini K.	
IT16170780	Thenuja S.	
IT16165526	Wijeshinghe J.R.	
IT16167124	Shangavie R.	
IT16180444	Vanaja A.	
IT16168800	Gunarathna P.M.S.S.B.	
IT16010086	Samarawickrama Y.M.	

Table of Contents

Abstract .....

ACKNOWLEDGEMENT ..... ERROR! BOOKMARK NOT DEFINED.

DECLARATION ..... ERROR! BOOKMARK NOT DEFINED.

TABLE OF CONTENTS V

LIST OF FIGURES VI

LIST OF TABLES 7

1. INTRODUCTION 9

1.1 Problem Statement 10

1.2 Product Scope 11

1.3 Project Report Structure 11

2. METHODOLOGY 12

2.1 Requirements and Analysis .....

2.2 Design .....

2.3 Implementation 65

2.4 Testing 66

3. EVALUATION 81

3.1 Assessment of the Project results 81

3.2 Lessons Learned 82

3.3 Future Work 82

4. CONCLUSION 82

5. REFERENCES 83

APPENDIX A: DESIGN DIAGRAMS 84

APPENDIX B: TEST RESULTS 85

APPENDIX C: SELECTED CODE LISTINGS 87

List of Figures

Figure 2.2.1 Class diagram for Supplier and Order Management System

Figure 2.2.2 Class diagram for Stock Management System

Figure 2.2.3 Class diagram for Safety Management System

Figure 2.2.4 Class diagram for Project Management System

Figure 2.2.5 Class diagram for Machine Management System

Figure 2.2.6 Class diagram for Employee Management System

Figure 2.2.7 Class diagram for Subcontractor and Client Management System

Figure 2.2.8 Class diagram for Account and Finance Management System

Figure 2.2.9 Sequence diagram for Supplier and Order Management System

Figure 2.2.10 Sequence diagram for Stock Management System

Figure 2.2.11 Sequence diagram for Stock Management System

Figure 2.2.12 Sequence diagram for Safety Management System

Figure 2.2.13 Sequence diagram for Safety Management System

Figure 2.2.14 Sequence diagram for Project Management System

Figure 2.2.15 Sequence diagram for Machine Management System

Figure 2.2.16 Sequence diagram for Employee Management System

Figure 2.2.17 Sequence diagram for Subcontractor and Client Management System

Figure 2.2.18 Sequence diagram for Account and Finance Management System

Figure 2.2.19 Use case diagram for Supplier and Order Management system

Figure 2.2.20 Use case diagram for Stock Management system

Figure 2.2.21 Use case diagram for Safety Management system

Figure 2.2.22 Use case diagram for Project Management system

Figure 2.2.23 Use case diagram for Machine Management system

Figure 2.2.24 Use case diagram for Employee Management system

Figure 2.2.25 Use case diagram for Subcontractor and Client Management system

Figure 2.2.26 Use case diagram for Account and Finance Management system

List of Tables

Table 1 Use Case Scenario - Handle Orders

Table 2 Use Case Scenario - Generate Report

Table 3 Use Case Scenario - Update Supplier

Table 4 Use Case Scenario - Delete Suppliers

Table 5 Use Case Scenario - Add project material request

Table 6 Use Case Scenario - Update materials details

Table 7 Use Case Scenario - Update materials details

Table 8 Use Case Scenario - Add supply details

Table 9 Use Case Scenario - Add training details

Table 10 Use Case Scenario - Modify /delete Training Details

Table 11 Use Case Scenario - Add accident details

Table 12 Use Case Scenario - Modify /delete Accident Details

Table 13 Use Case Scenario - Add Induction details

Table 14 Use Case Scenario - Modify /delete Induction Details

Table 15 Use Case Scenario - Add Risk Assessment details

Table 16 Use Case Scenario - Modify /delete Risk Assessment Details

Table 17 Use Case Scenario - Add Project Information

Table 18 Use Case Scenario - Update Project Information

Table 19 Use Case Scenario - Search Project Information

Table 20 Use Case Scenario - Change Engineers of Project

Table 21 Use Case Scenario - Assign Laborers for Project

Table 22 Use Case Scenario - Change Laborer of Project

Table 23 Use Case Scenario - Remove Engineers from Project

Table 24 Use Case Scenario - Remove Laborers from Project

Table 25 Use Case Scenario - Search Project Engineers

Table 26 Use Case Scenario - Search Project Laborers

Table 27 Use Case Scenario - Add Budget for Project

Table 28 Use Case Scenario - Delete Budget Details of Project

Table 29 Use Case Scenario - Update Budget Details of Project

Table 30 Use Case Scenario - Assign Engineer for Project

Table 31 Use Case Scenario - Generate Report

Table 32 Use Case Scenario – Add tool details

Table 33 Use Case Scenario – Update tool details

Table 34 Use Case Scenario – Delete tool details

Table 35 Use Case Scenario – Assign tools for projects

Table 36 Use Case Scenario – Add Damaged tools details

Table 37 Use Case Scenario – Add vehicle details

Table 38 Use Case Scenario – Update vehicle details

Table 39 Use Case Scenario – Delete vehicle details

Table 40 Use Case Scenario – Add vehicle details

Table 41 Use Case Scenario - Add Employee

Table 42 Use Case Scenario - Generate Report

Table 43 Use Case Scenario - Update Employee details

Table 44 Use Case Scenario - Delete Employee

Table 45 Use Case Scenario - Add Client

Table 46 Use Case Scenario - Update Client details

Table 47 Use Case Scenario - Delete Client

1. Introduction

KSL is a Construction Management company, which is located in Colombo, Sri Lanka. Construct building since 2000. It has 5 branches around in Colombo District. They had around 20,000 employees including Project Managers, Engineers, Stock manager, Accountant, Laborers.

Unfortunately, still they are using ledgers and hard copies for recording their data and analyzing. It is very difficult to handle because of its hugeness. Also they need to hire employees for the verification of manually entered information. Because of this manual system, they are facing so many problems such as,

- When calculating salaries for employees they may miss some employee's attendance. And if they need to view any details of employee or change any details or their attendance is very difficult.
- The finance manager is the person who is maintaining finance details of the whole factory. He should manage income of factory and expenses and he need to maintain payment of suppliers and transportation expenses. All these are handled manually through ledgers and receipts.
  Employee payroll is calculated manually and that would be time consuming also human error in calculating will lead to incorrect payment. This process is prone to human error and carries a major work to the staff. Also at certain intervals, our client requires great profit and loss report, balance sheet and liability and asserts reports to monitor the cash inflow and outflow of the company.
- When assigning Employees to the project sometimes the details will be duplicated. And maintaining the project details is complex when it was handling manually. Calculate the budget of the project also become difficult if we do the things manually.
- Storing the details of the product such as product name, type, unit price, quantity, etc. will become difficult when we record them manually. It's so complex to analyze whether the product is being received from client or supplier awhile we providing them to project.
- Our client's needs a system to store details of orders and produce invoice for orders. When the products in the order are damaged there should be a way to handle them. We need to supply orders to stock department to handle with projects. And there should be a way to handle supplier details in the system.
- When schedule the safety training need to get the project details as well as employee details for an induction. When the accident happened the details should be recorded under the project manager. Risk Assessment is registered before the project start on time.
- When we buy tools such as screwdriver, hammer, etc. and vehicles we need a proper way to store, update as well as delete them. There need to be a way to handle machines when they are damaged. And also we need to be able to check the warranty period of machines and be able to make decisions according to them.
- There should be a proper way to manage subcontractors and clients respectively in the system. We need to be able to reduce transaction cost when necessary.

To overcome from this problem, we are planning to design a solution called Construction Management System. By using our solution, they could get accurate results for each information they are entering. This proposed review will discuss the problems, our solution, benefits, objectives, procedures, budgets and system requirements.

1.2 Product Scope

Currently this company is running without any Computerized System. Therefore, they need an efficient system, which could run without any bugs. In addition, it should be user-friendly and time saving. When they expand their company, they need to consider about management staff. Because they needed trusted and talented person to handle company's important work like Calculating salaries for employees, Managing Finance etc. Even if they hired trusted person sometimes they could make some mistakes. Therefore, this project's main objective is providing Ease of access for their data and providing error free functionalities. This system requires less workers for maintenance. Therefore, it could replace many employees and this company can use them for other works. Also we have responsibility for securing this company's data, recover data from crashes, providing user friendly error messages, keeping logs for all events. This would increase system's reliability.

As construction system is doing bigger role in Sri Lankan economy, and there are many factories but most of them don't have a computerizes management system, our project would get a bigger business value in future.

Computerized system to hold & retrieve records is a major leap from physical data to computerized database makes more functionalities, which directly and indirectly assists the company's growth in many ways. Increased Create, Update and Delete functions that helps to manipulate data in a faster, responsive and time conserving way. Keeping track of historical data will be easier. Searching for data can be done hassle free. It also helps with data integrity and data security resulting in a more reliable data.

Safe and secured way to store financial records among other data of an organization are highly sensitive. The system with the help of security functionalities provided by the SQL database would be able to provide security for those data.

Generating automated reports are one of the essential part of a business. They help to foreshadow the growth and the future of the company. The system will have integrated with reporting functionalities, which would be able to generate the specific reports as required by the client such as monthly raw material stock report, Annual Financial Report etc.

1.3 Project Report Structure

This project report for the creation of the system for KSL Construction Company Management System is completely designed according to the standards of IEEE, document standards. It will give the reader a friendly environment to understand the content of it. If this project is used for further development in the future, this kind of document standard will also give a perfect approach to the development team also, to understand the system well and work of the modifications they want to achieve.

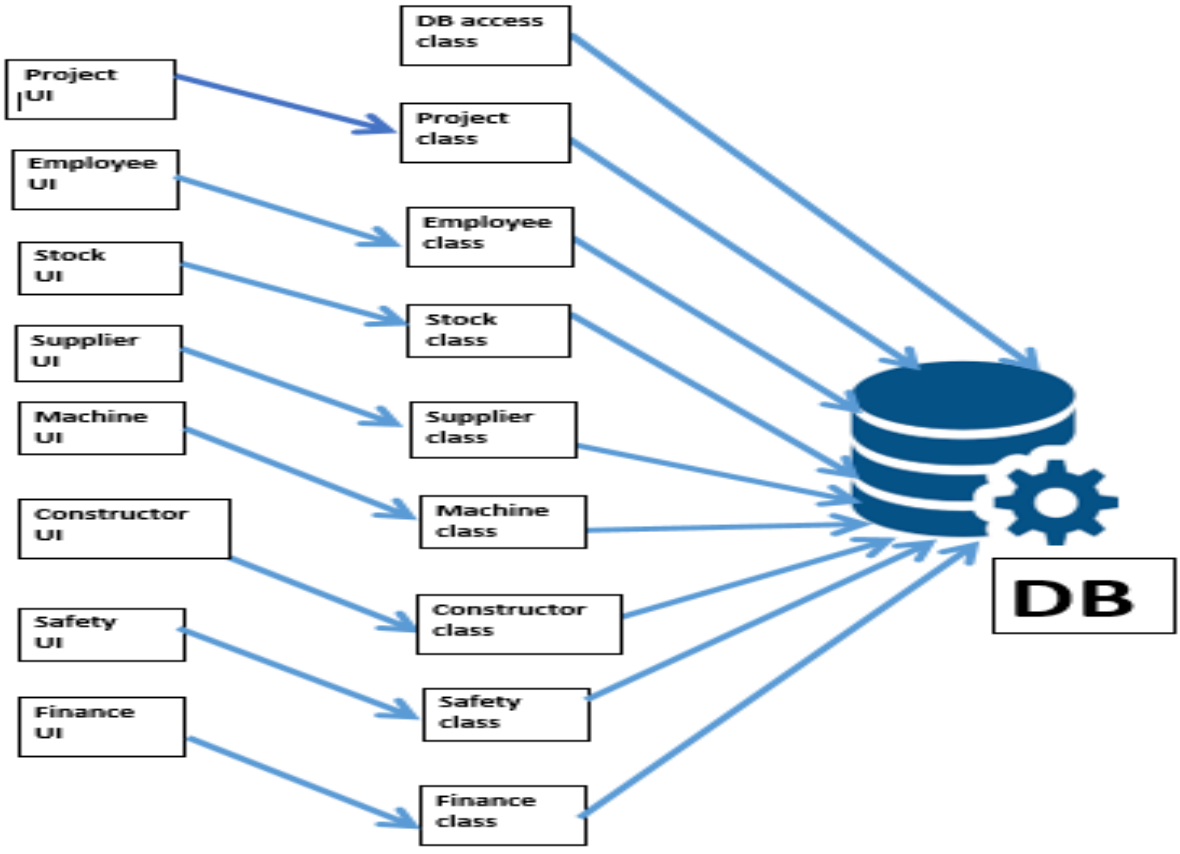
2. Methodology

2.1 Gathering requirements and analysis

In order to identify the problems and expectations of users, requirement gathering is very important part. By conducting one-on-one interviews and group interviews for several times we were able to gather user requirements. After requirements gathering we were able to identify main functionalities which are needed to be implemented

2.1.1 Requirement specification

Identify inputs, outputs and processes for the functions which are found at previous phase is done at requirement specification. And determining sub-functionalities also included in their phase. Along with functional requirements Non-functional requirements such as security, reliability also identified. Finally, we are able to create software requirements specification document(SRS). We have to discuss the SRS with client and get approval



2.2 Design and Implementation

By further clarifying the requirements in the SRS, specification on SRS will be transformed to be implemented using language “C#”.

Database Design and Implementation

First we identified the entities, attributes of entities and relationships between them. Identified data requirements are used to design Enhanced Entity Relationship(EER) diagram. This is the high level conceptual model of the system. As the next step EER diagram was mapped to the relational schema (Relational Database Design). Using normalization relational schema is refined. The final relational schema will be implemented using SQL.

Interface Design and Implementation

After analyzing the requirements, we identified there should be different user types to access the system in different user levels. We design the prototype for the system and implement it. This prototype is demonstrated to the client and get the feedback. Based on the prototype we created and the client feedback based on implementation of the real time system will be done. Each member individually implements the assigned part and those parts will be integrated later.

We have utilized a set of UML diagrams which were conceptualized. Given below are some of these diagrams.

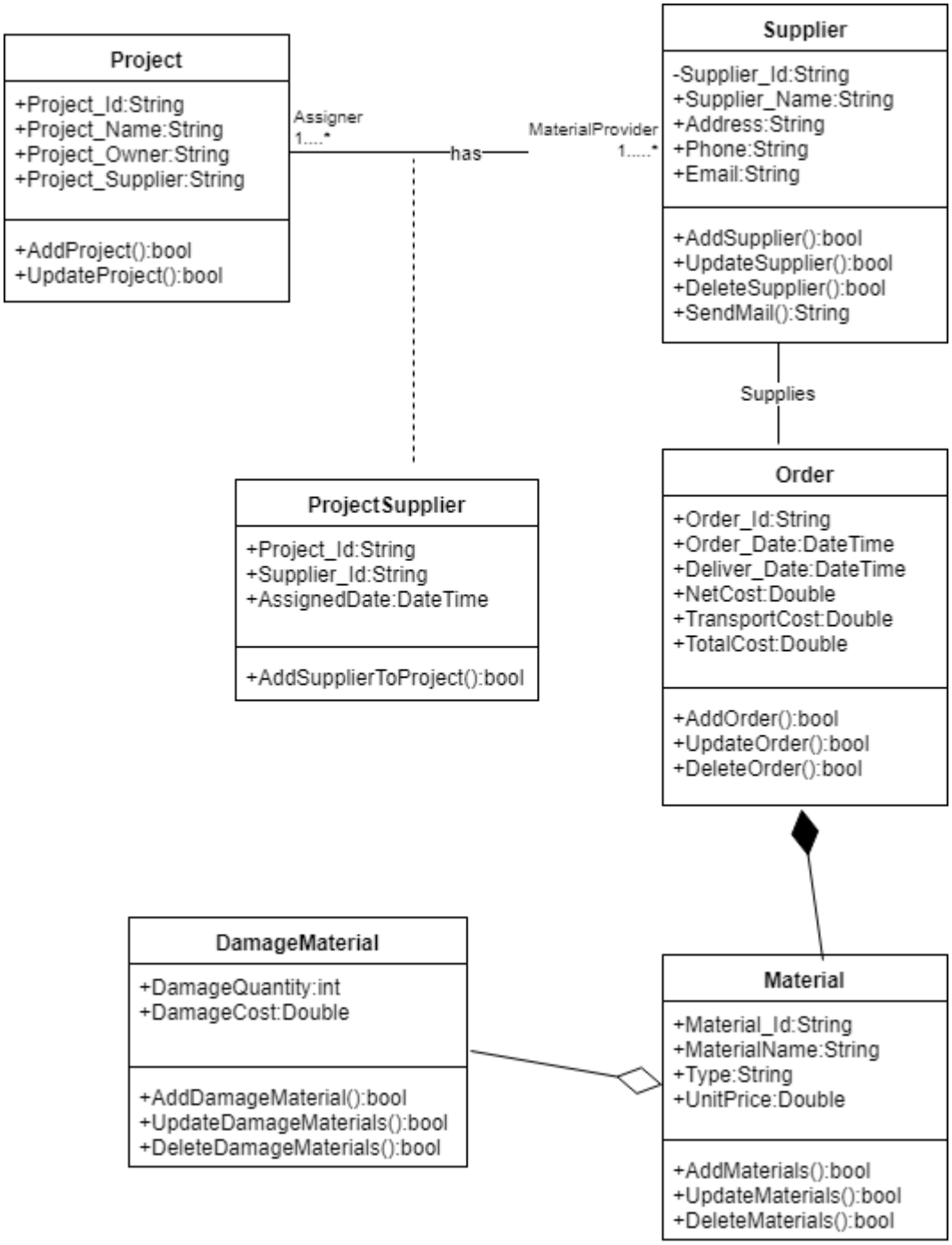


Figure 2.2.1 Class diagram for Supplier and Order Management System

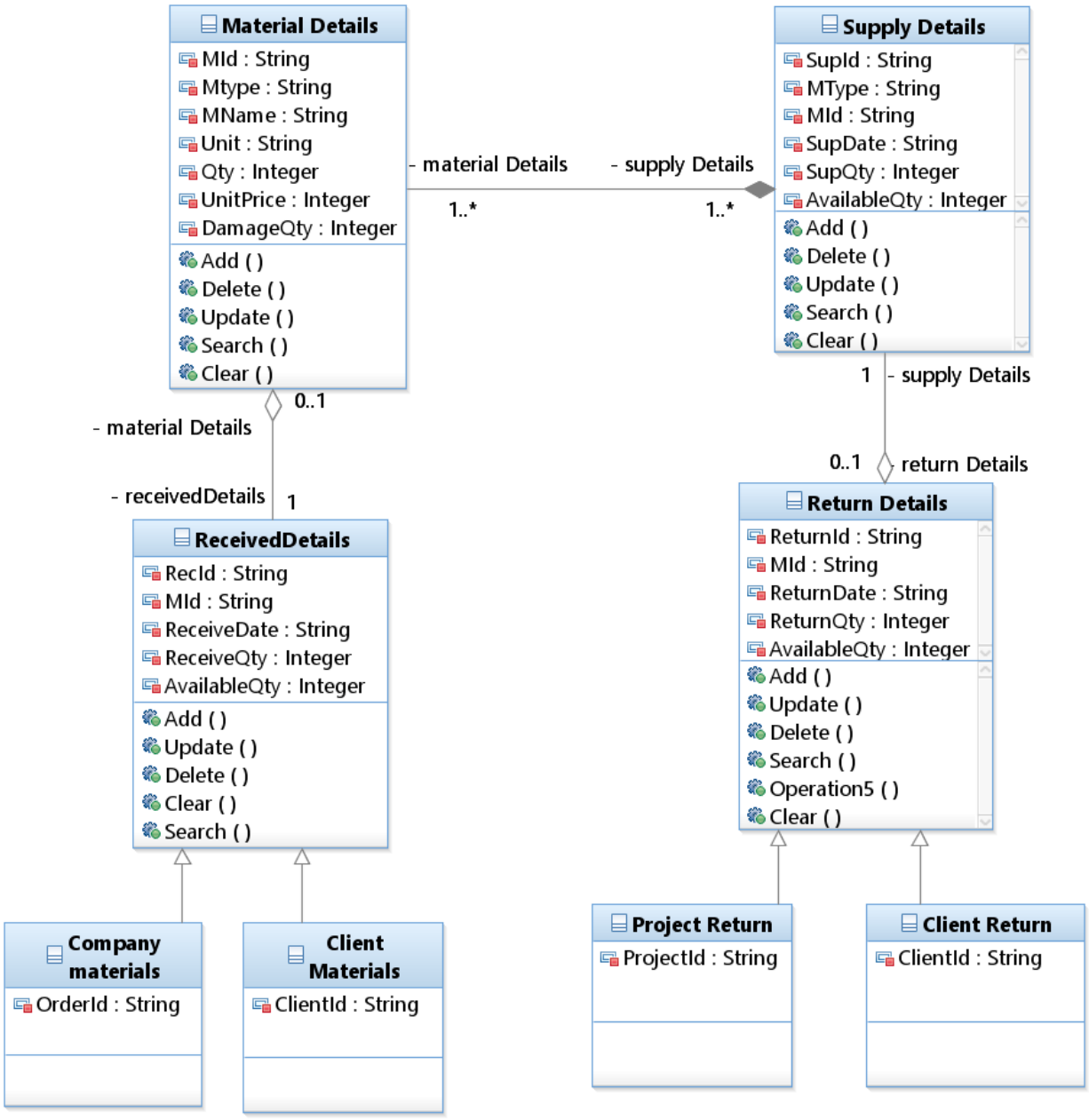


Figure 2.2.2 Class diagram for Stock Management System

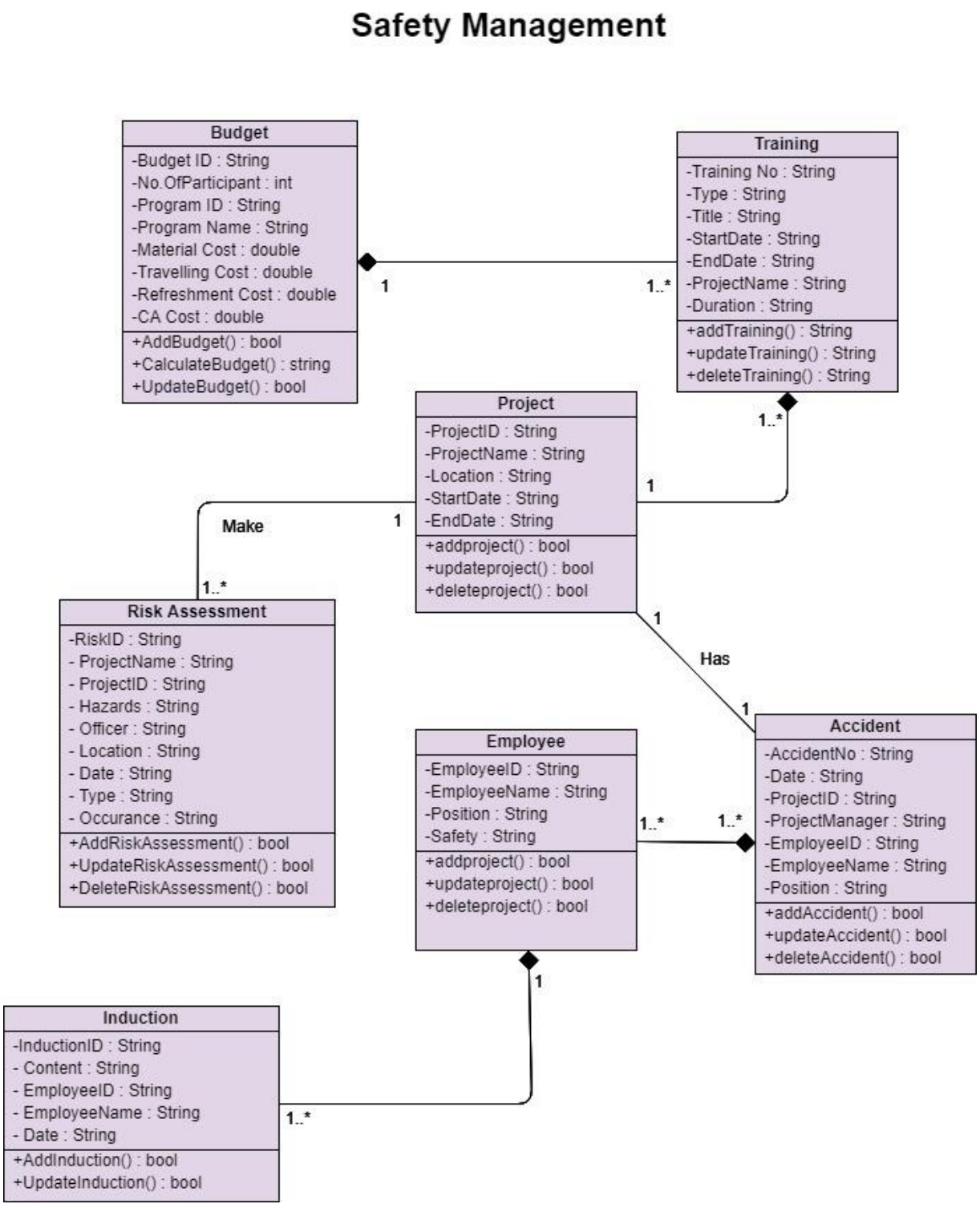


Figure 2.2.3 Class diagram for Safety Management System

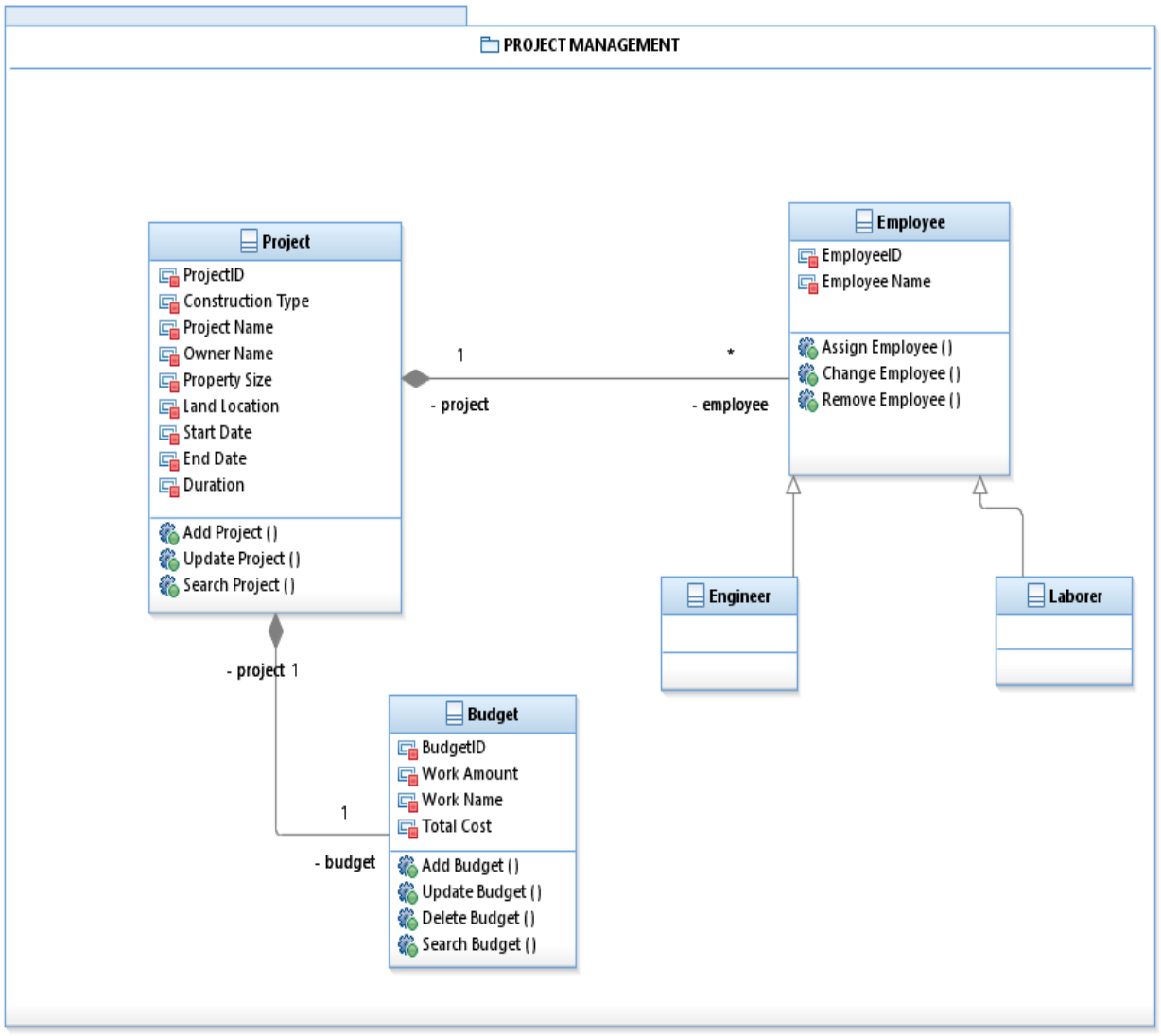


Figure 2.2.4 Class diagram for Project Management System

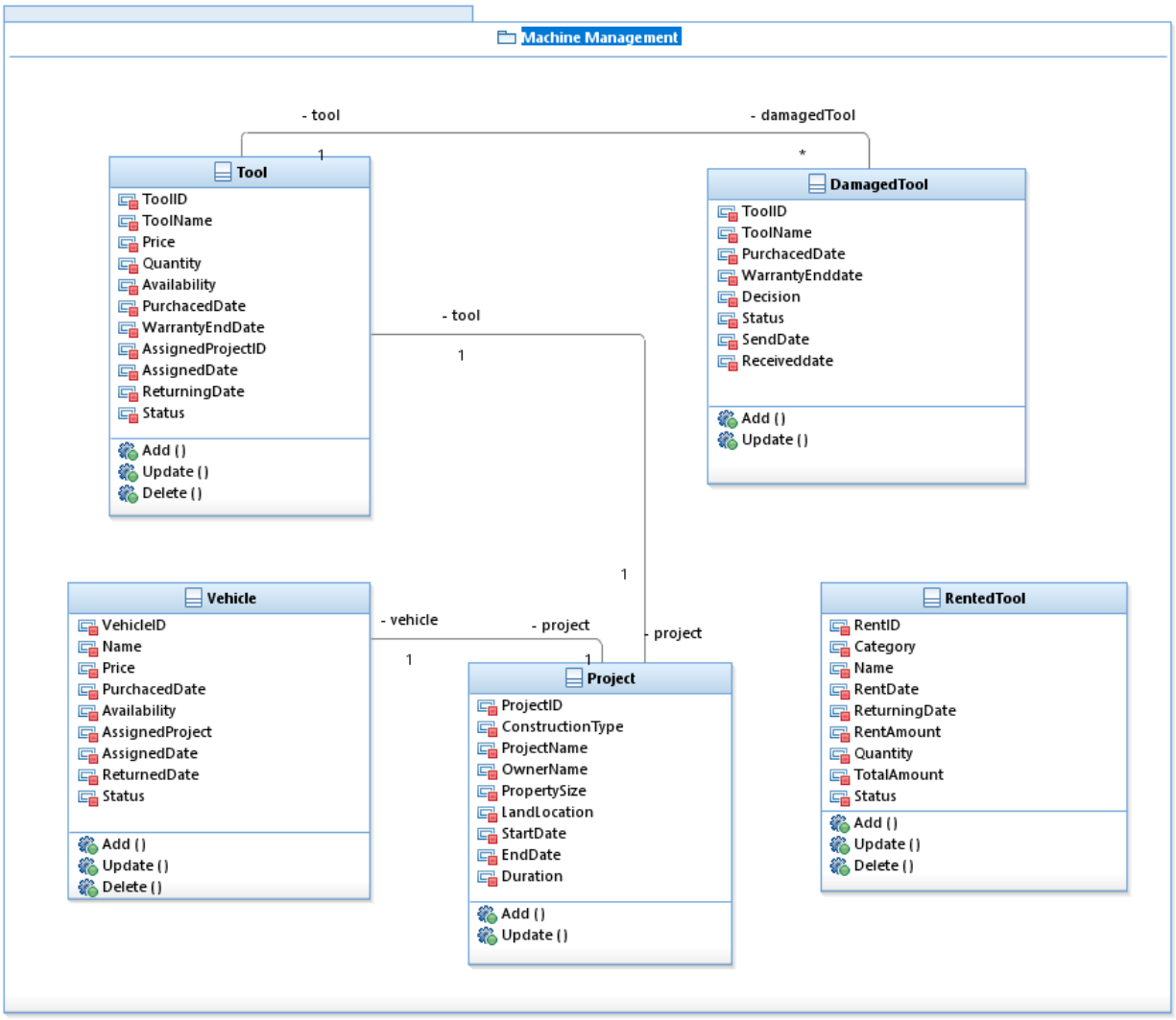


Figure 2.2.5 Class diagram for Machine Management System

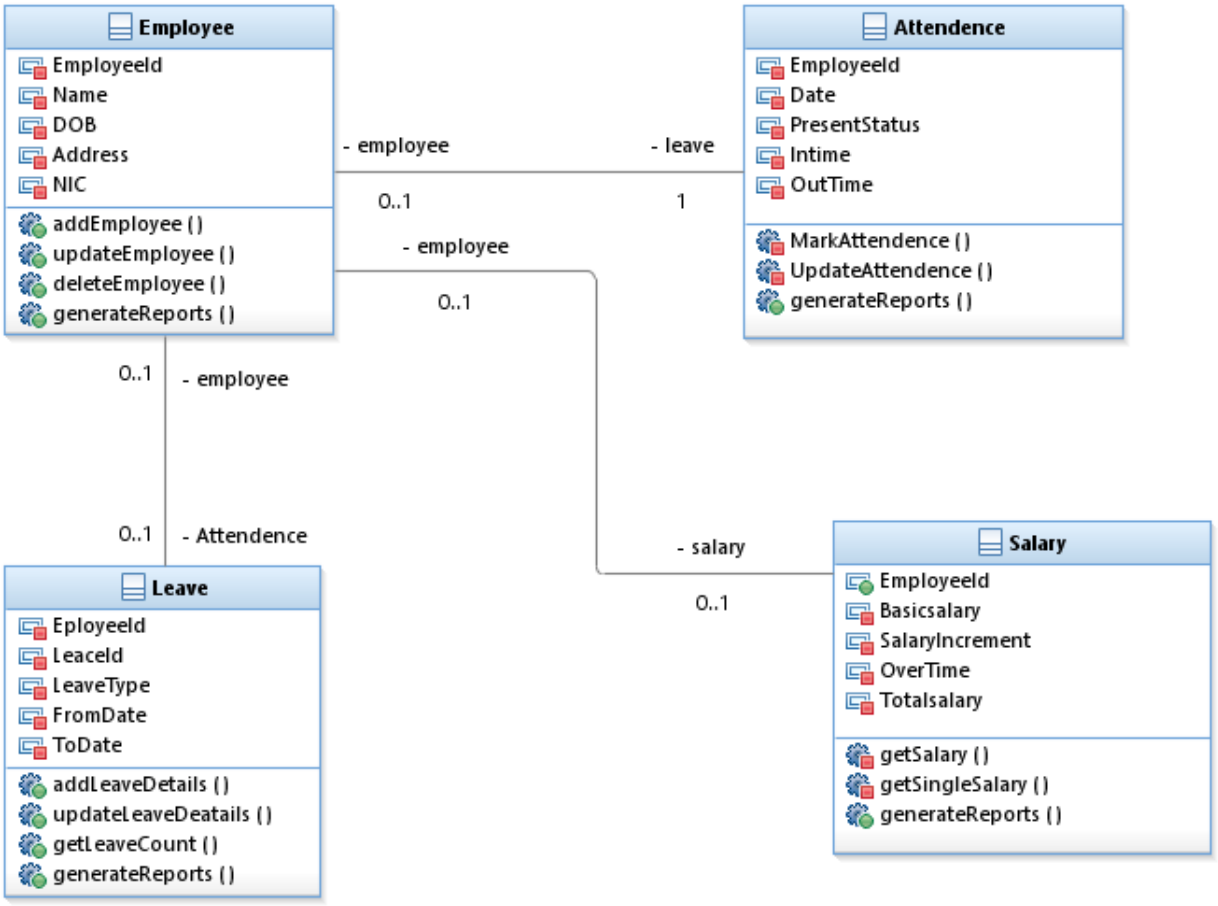


Figure 2.2.6 Class diagram for Employee Management System

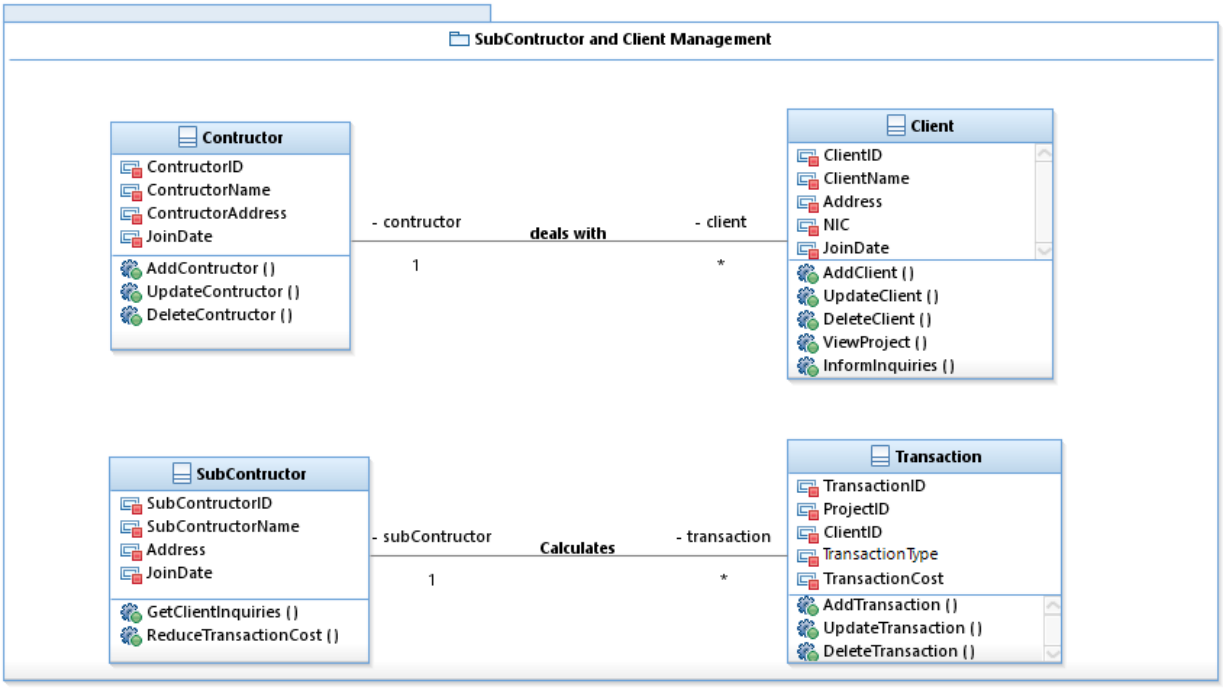


Figure 2.2.7 Class diagram for Subcontractor and Client Management System

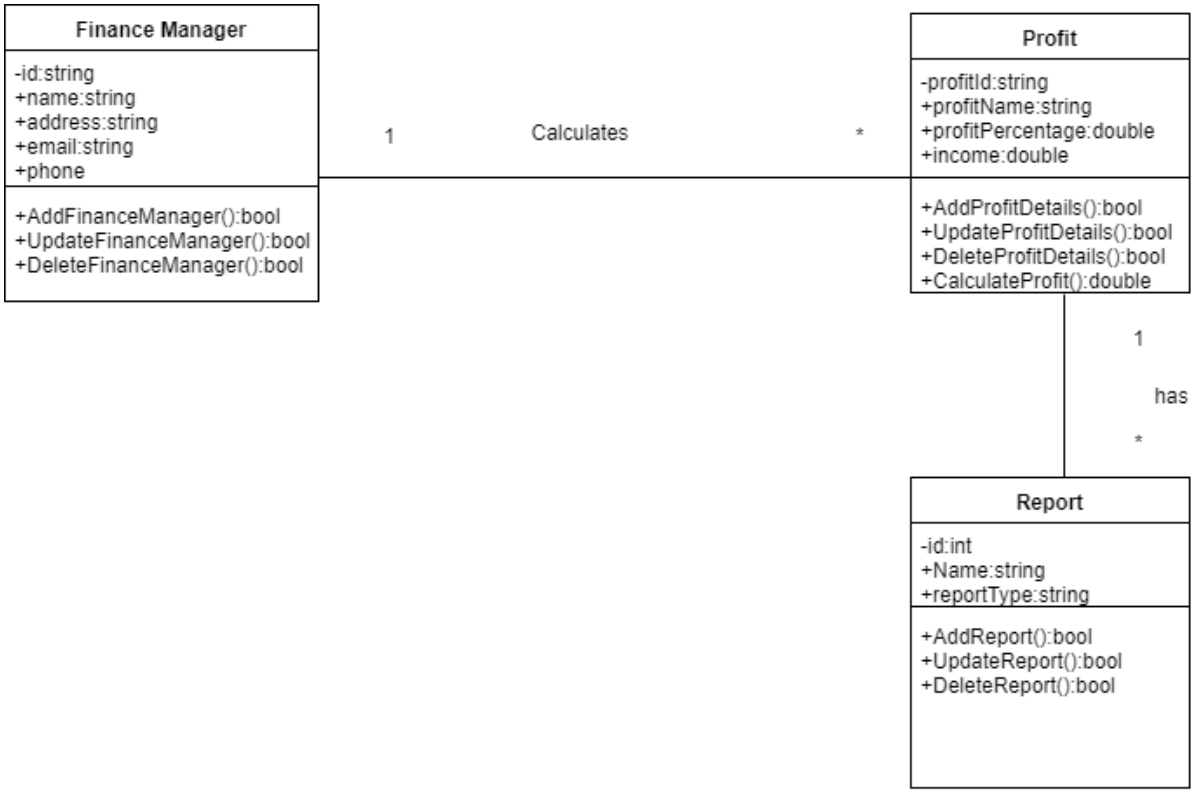


Figure 2.2.8 Class diagram for Account and Finance Management System

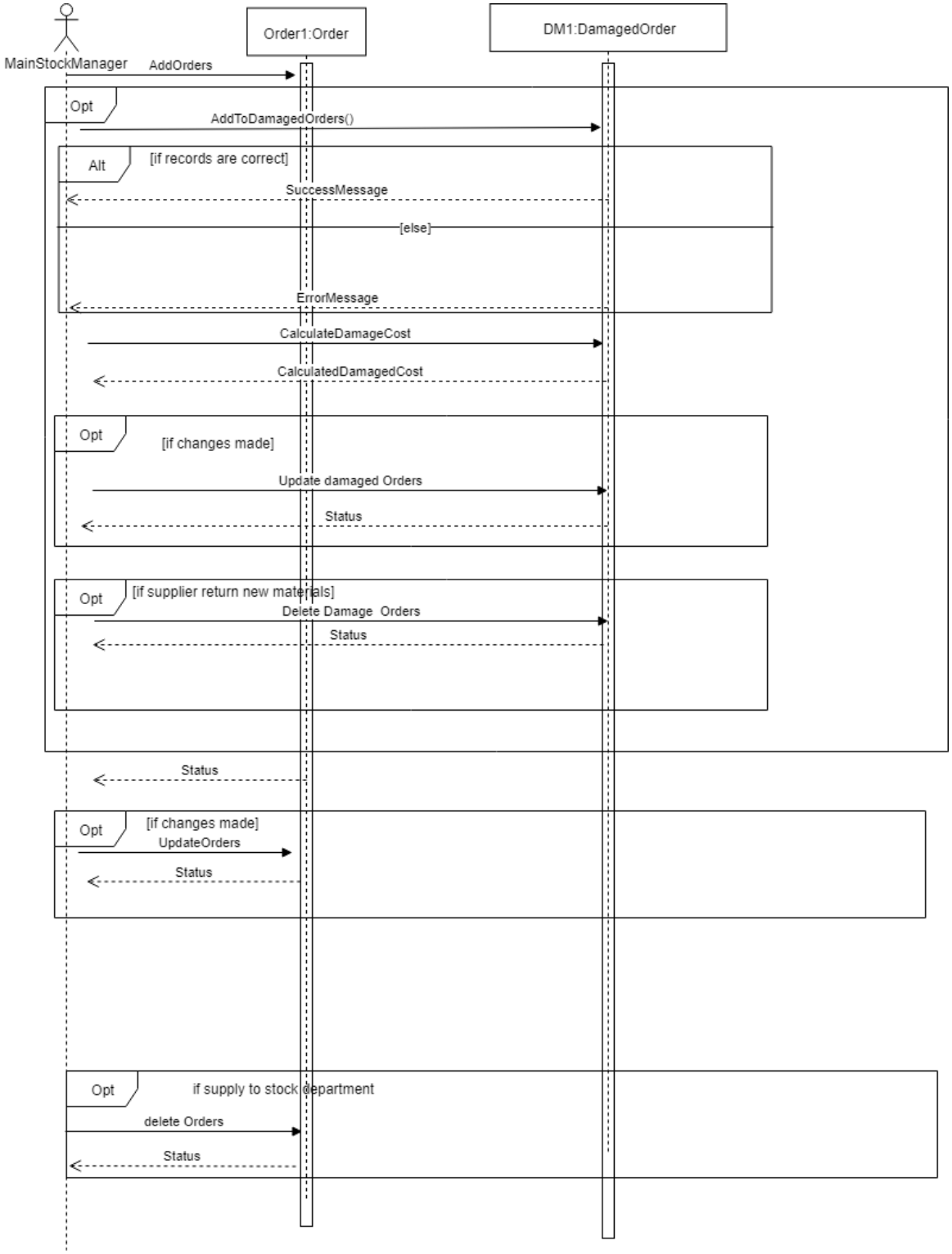


Figure 2.2.9 Sequence diagram for Supplier and Order Management System

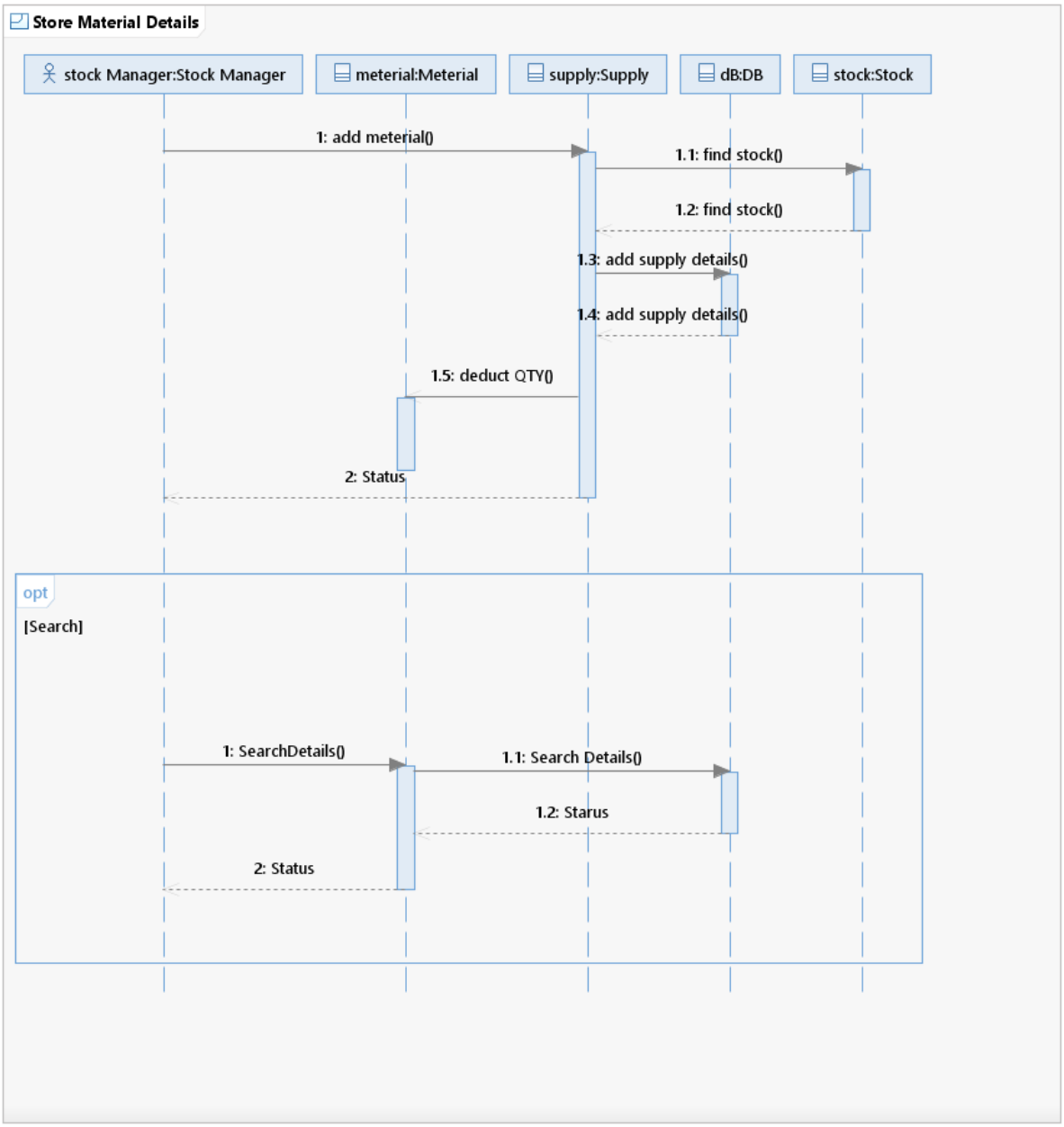


Figure 2.2.10 Sequence diagram for Stock Management System

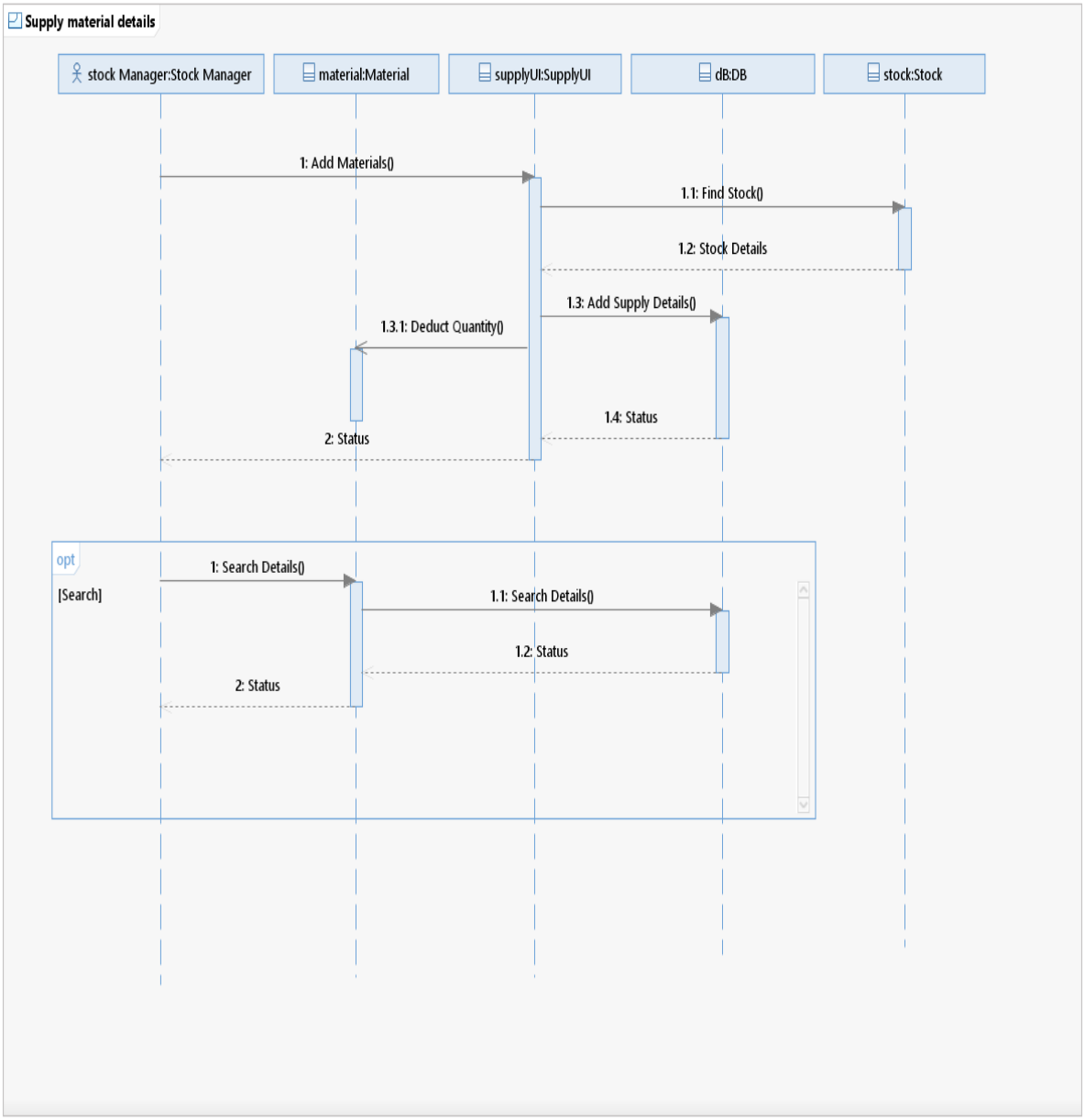


Figure 2.2.11 Sequence diagram for Stock Management System

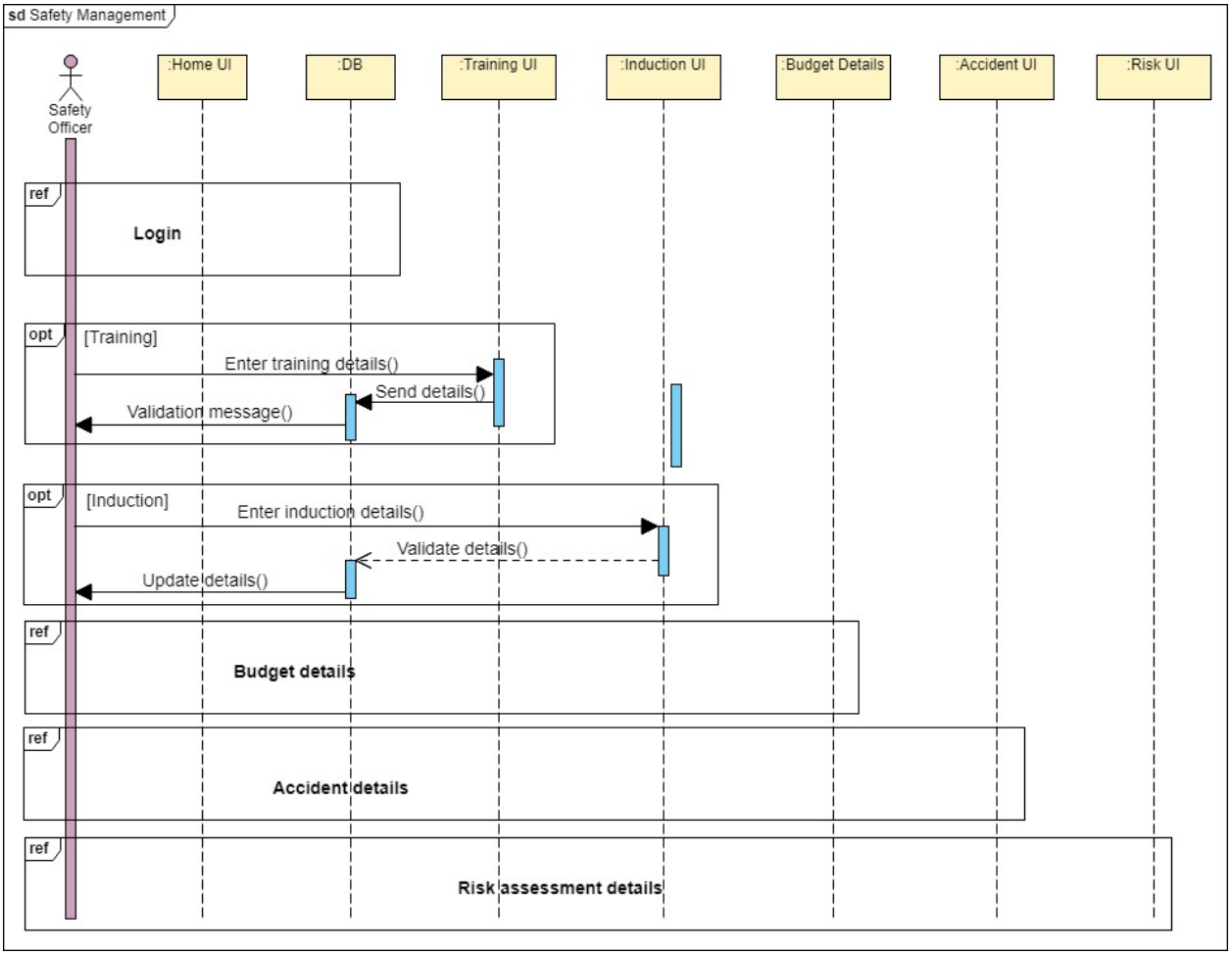


Figure 2.2.12 Sequence diagram for Stock Management System

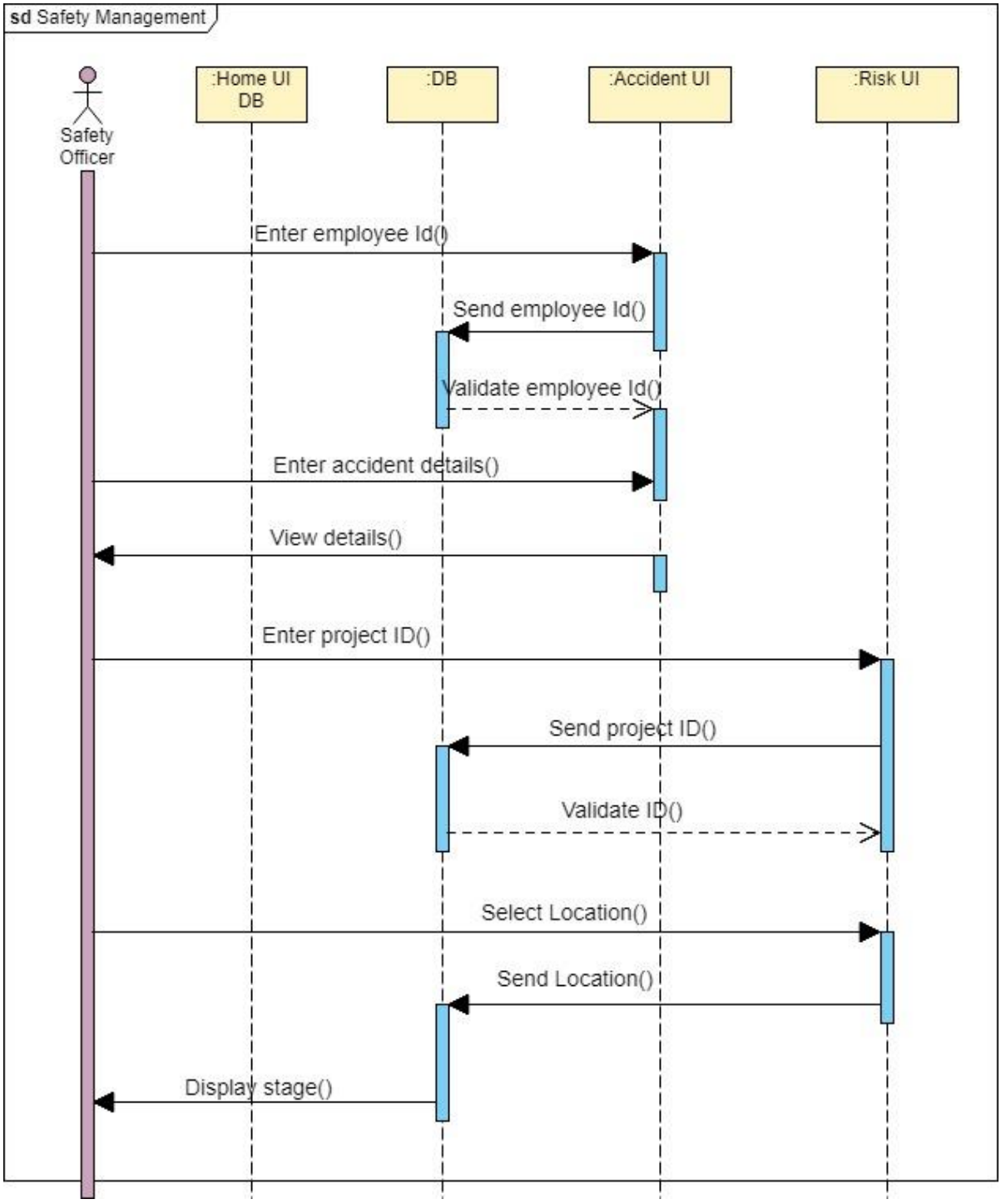


Figure 2.2.13 Sequence diagram for Safety Management System

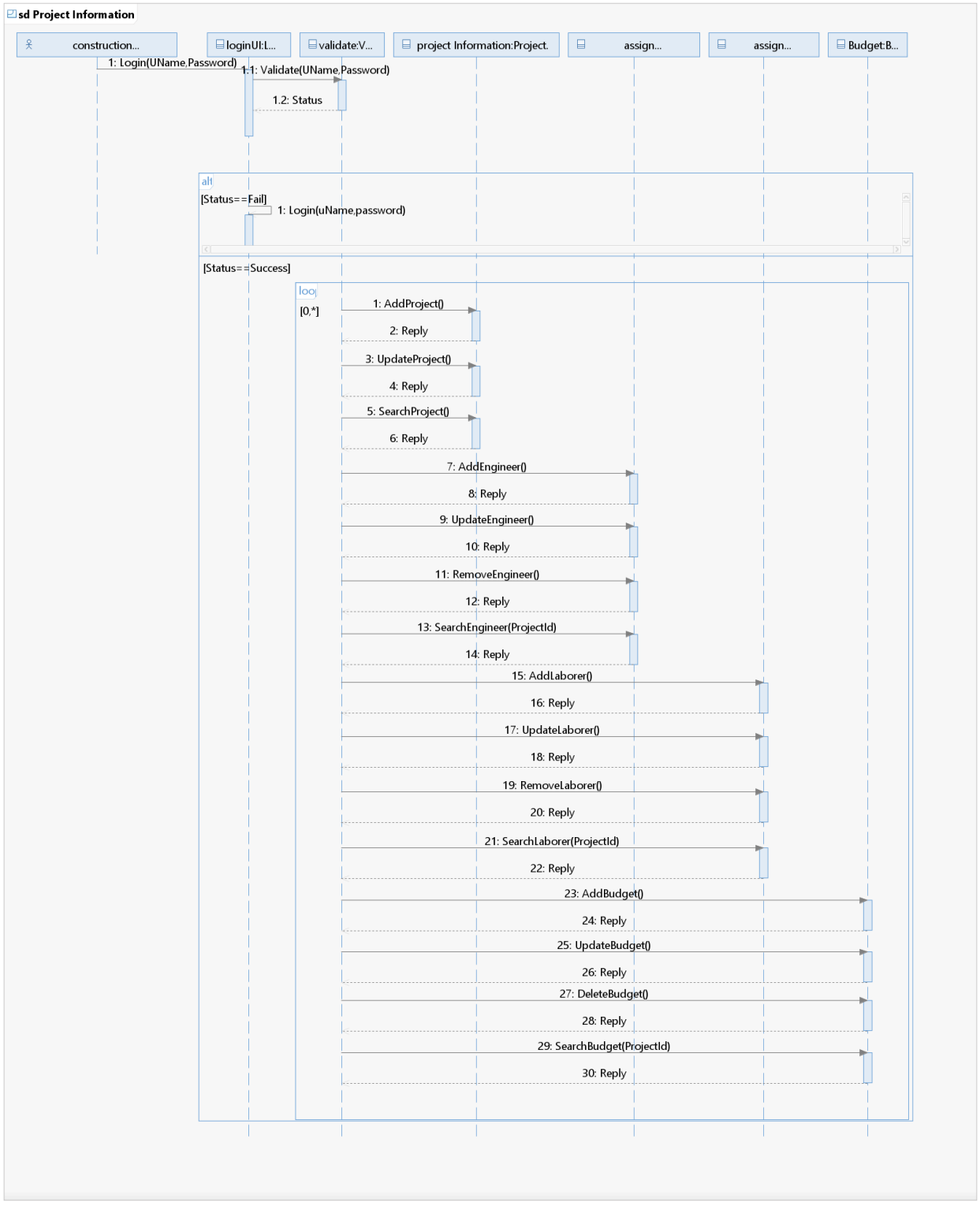


Figure 2.2.14 Sequence diagram for Project Management System

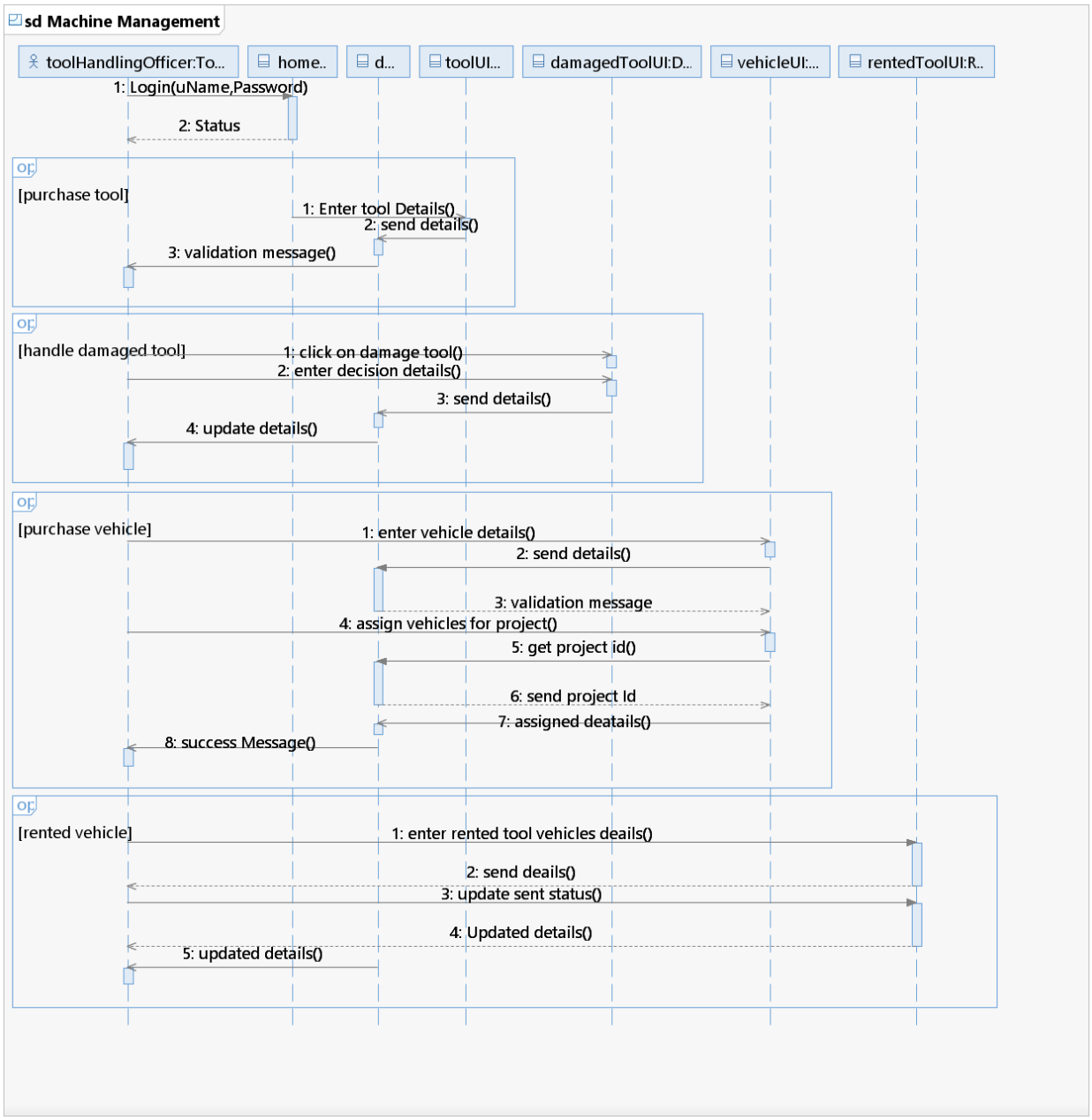


Figure 2.2.15 Sequence diagram for Machine Management System

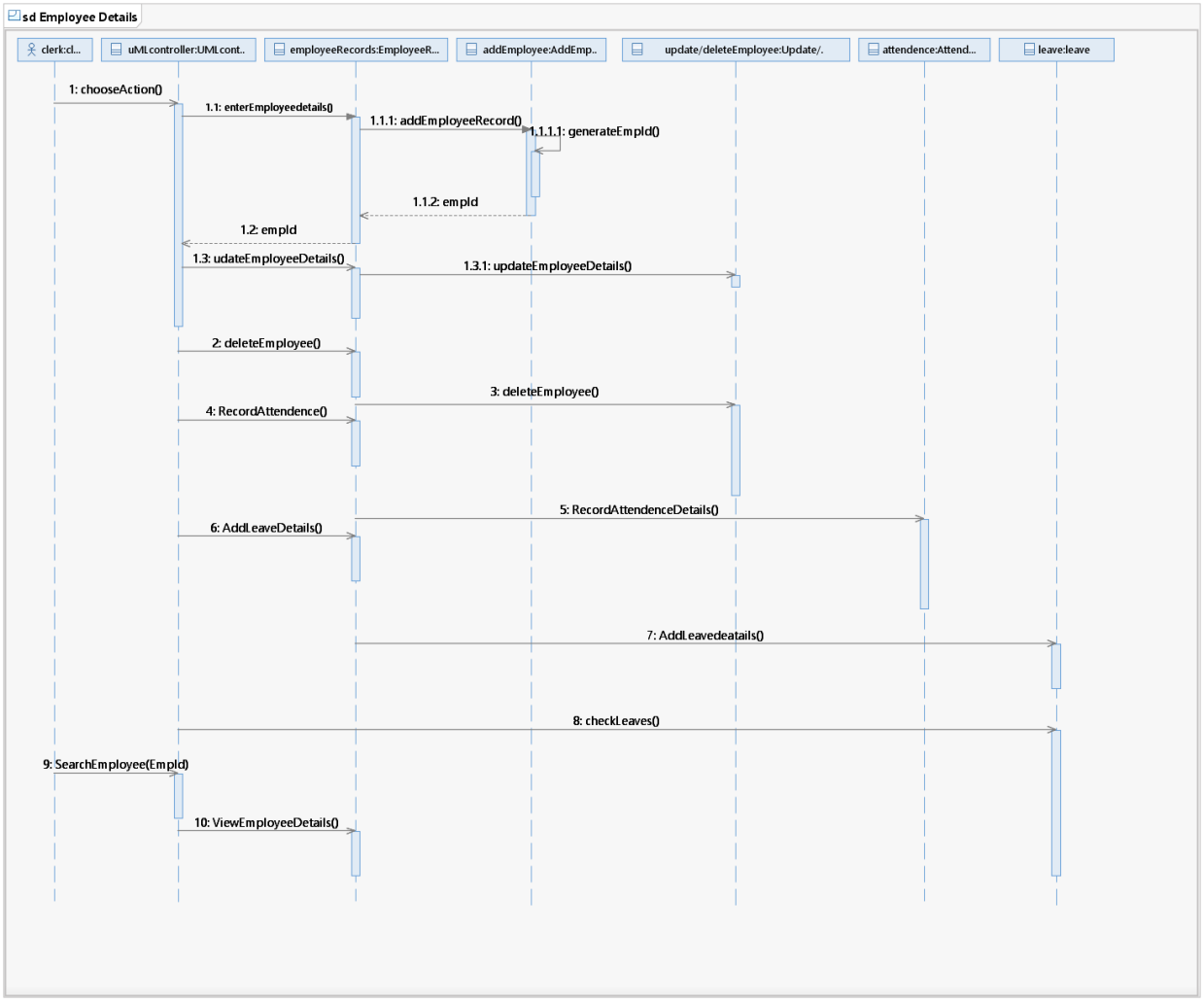


Figure 2.2.16 Sequence diagram for Employee Management System

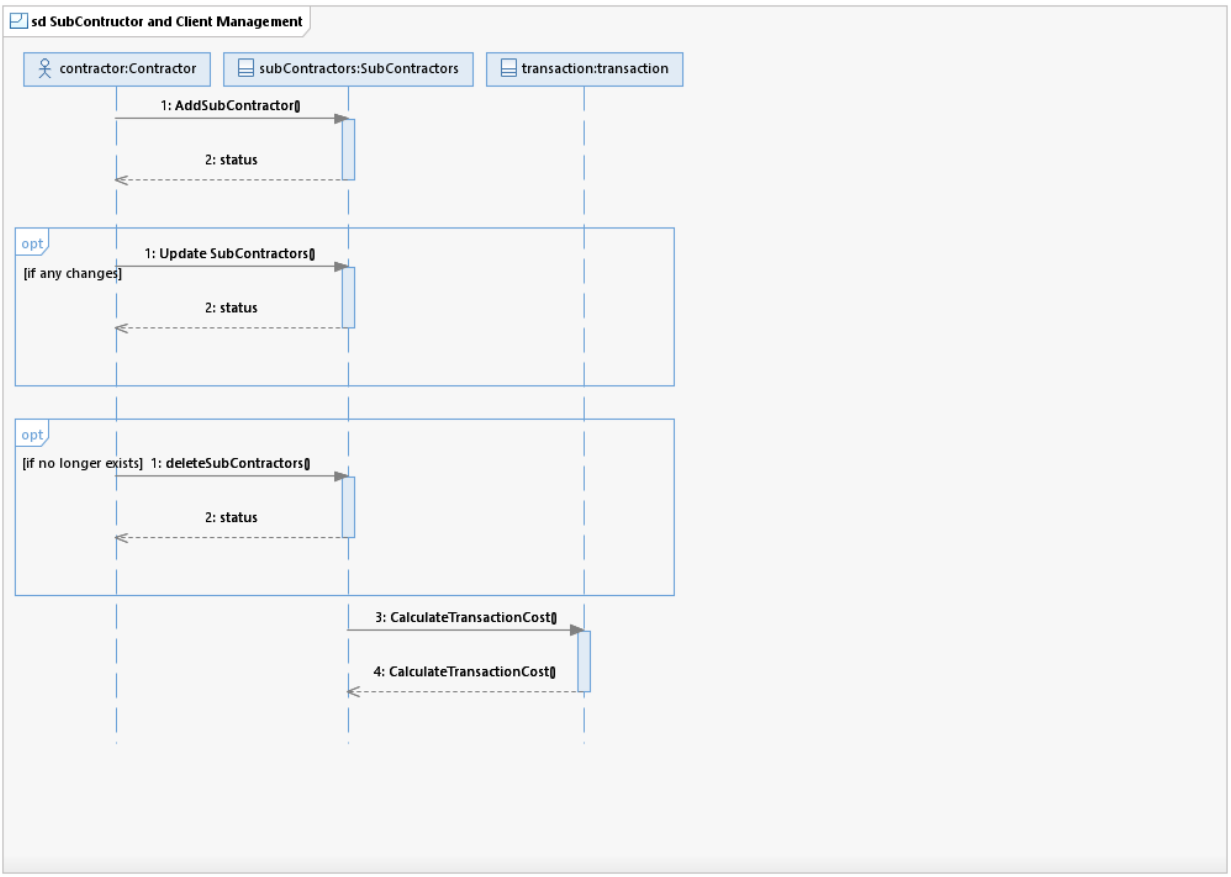


Figure 2.2.17 Sequence diagram for Subcontractor and client Management System

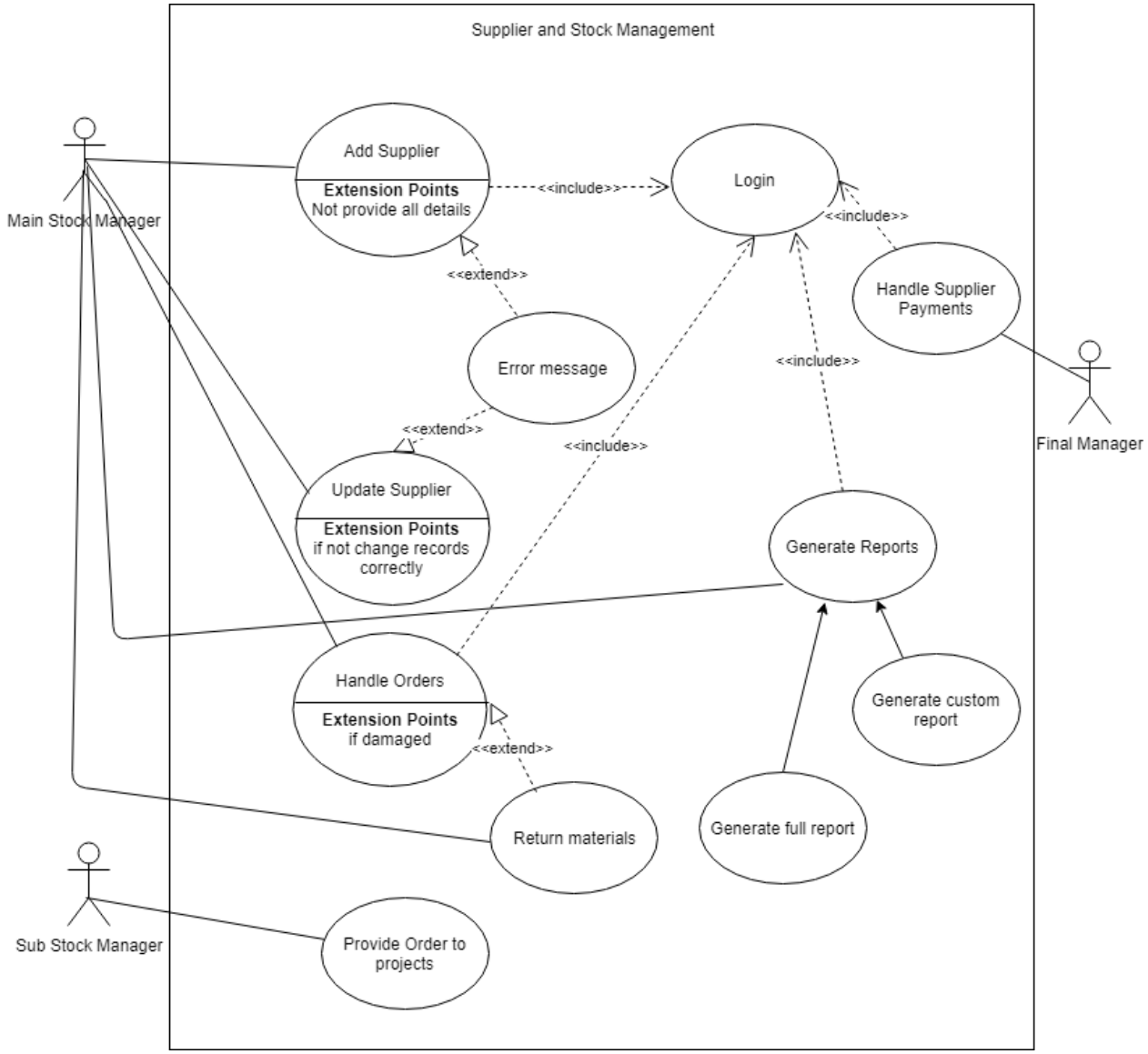


Figure 2.2.19 Use case diagram for Supplier and Order Management System

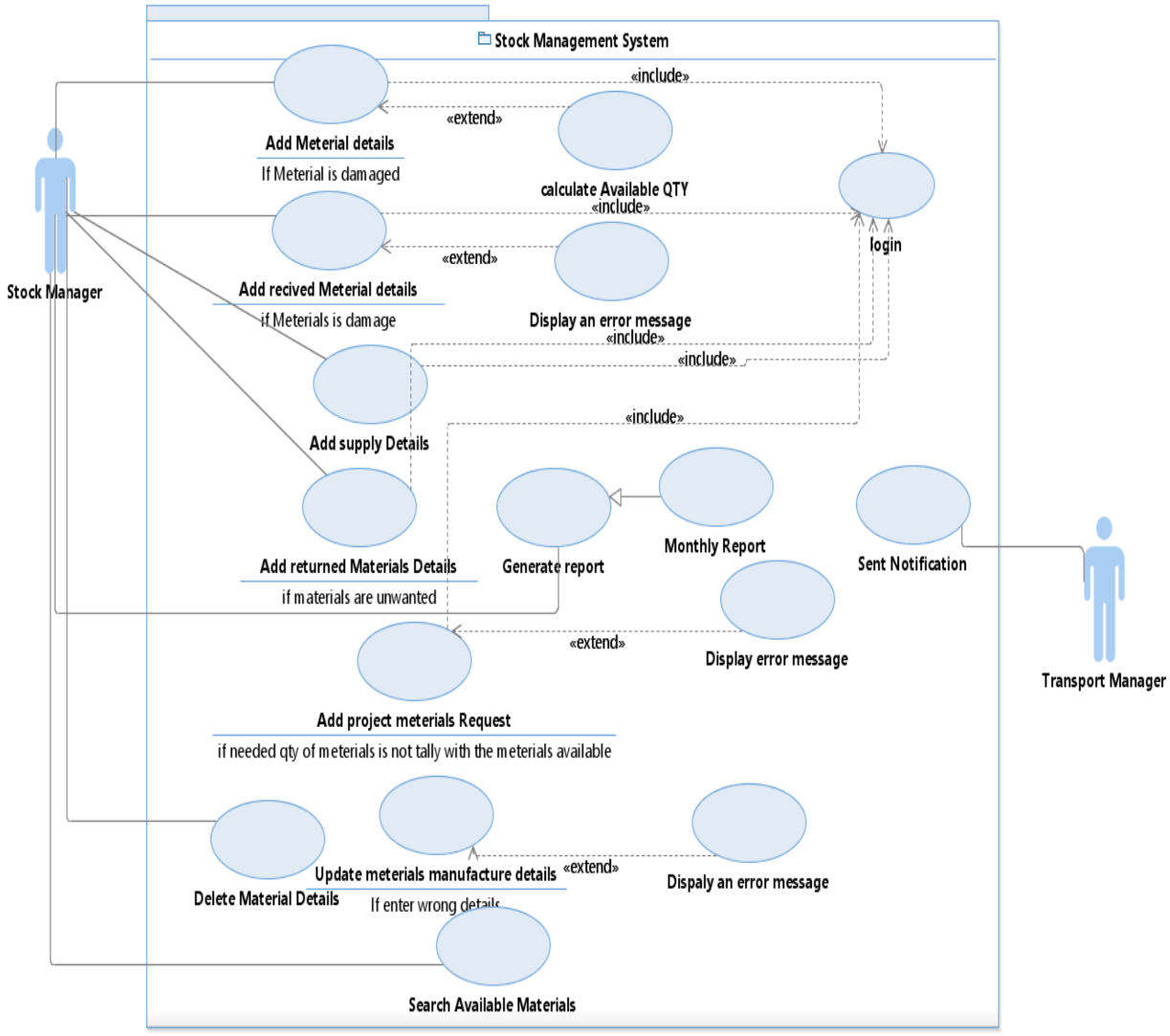


Figure 2.2.20 Use case diagram for Stock Management System

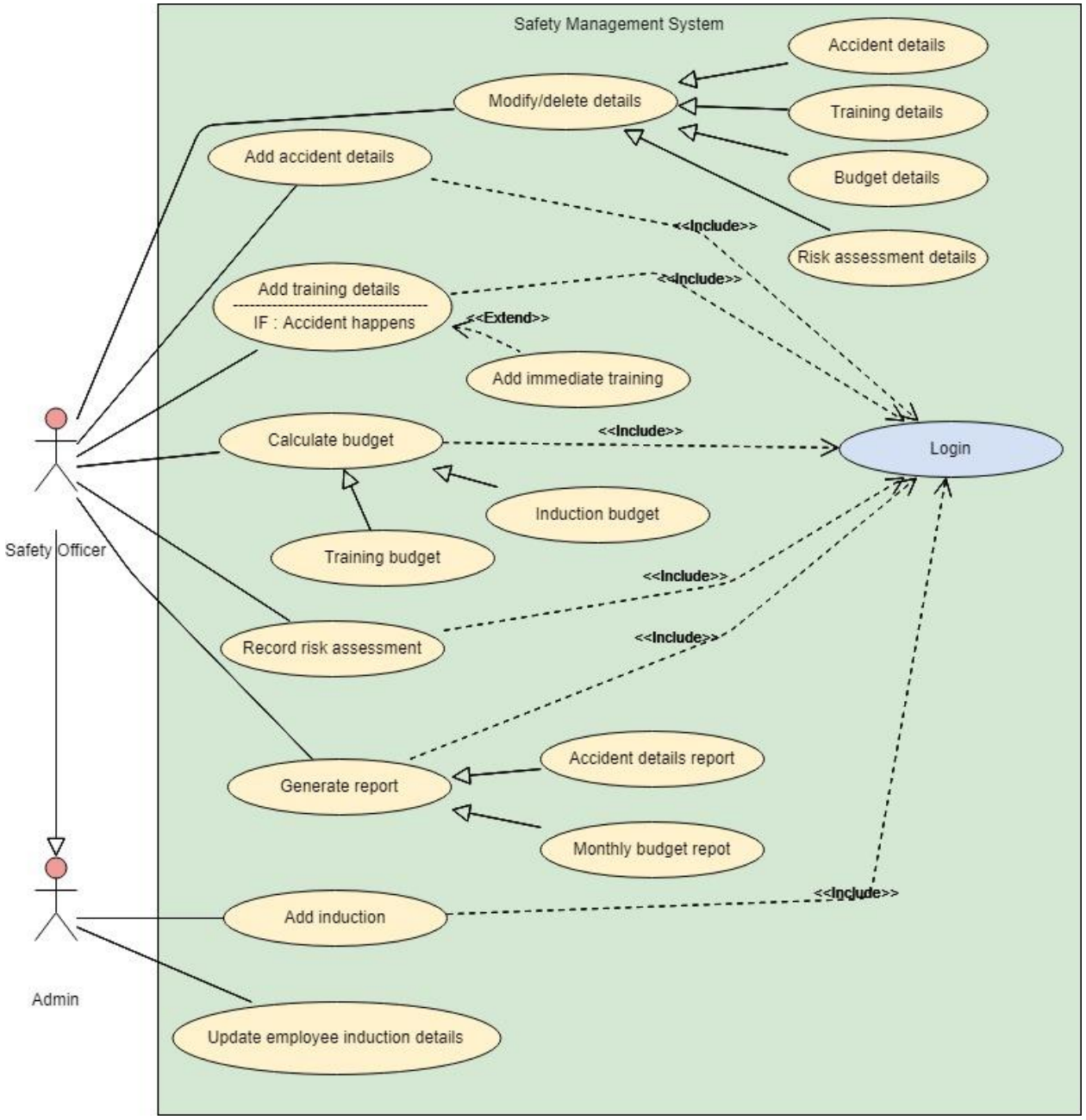


Figure 2.2.21 Use case diagram for Safety Management System

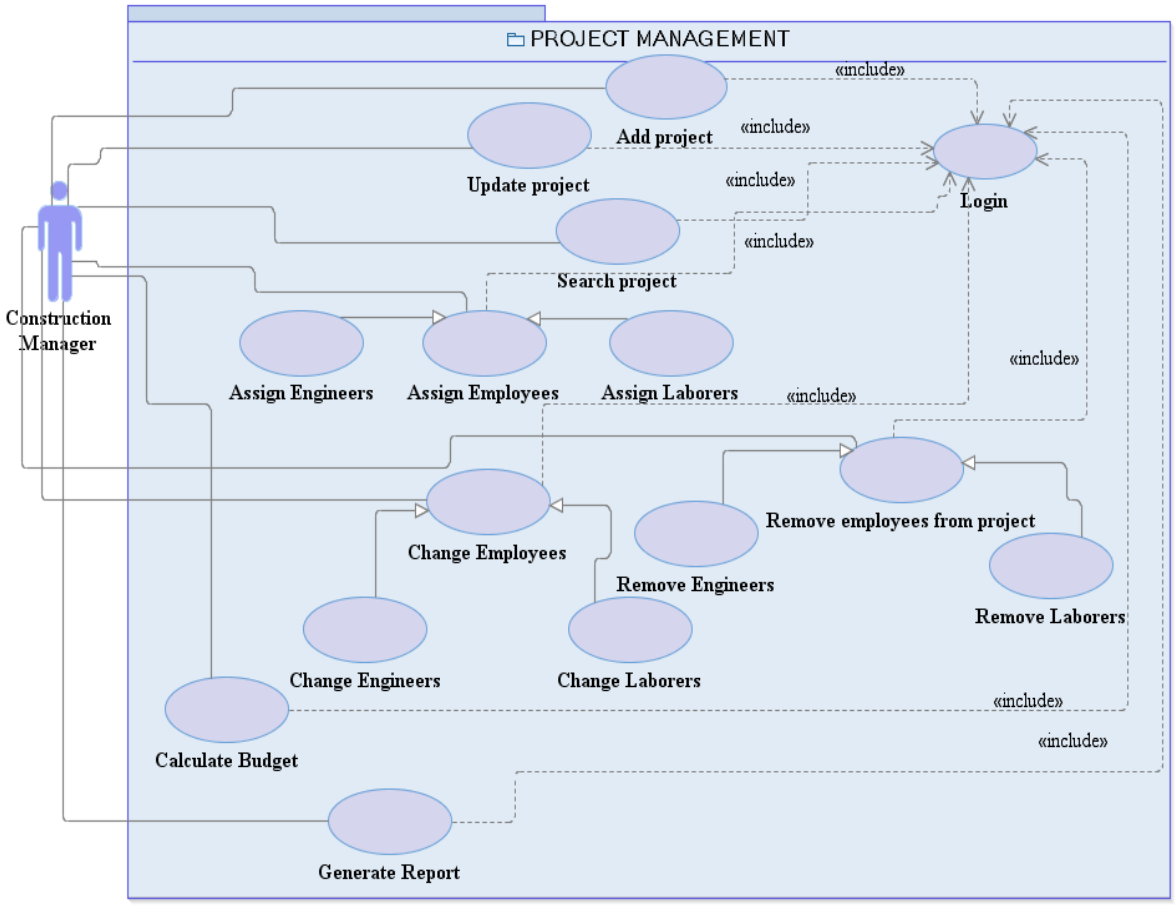


Figure 2.2.22 Use case diagram for Project Management System

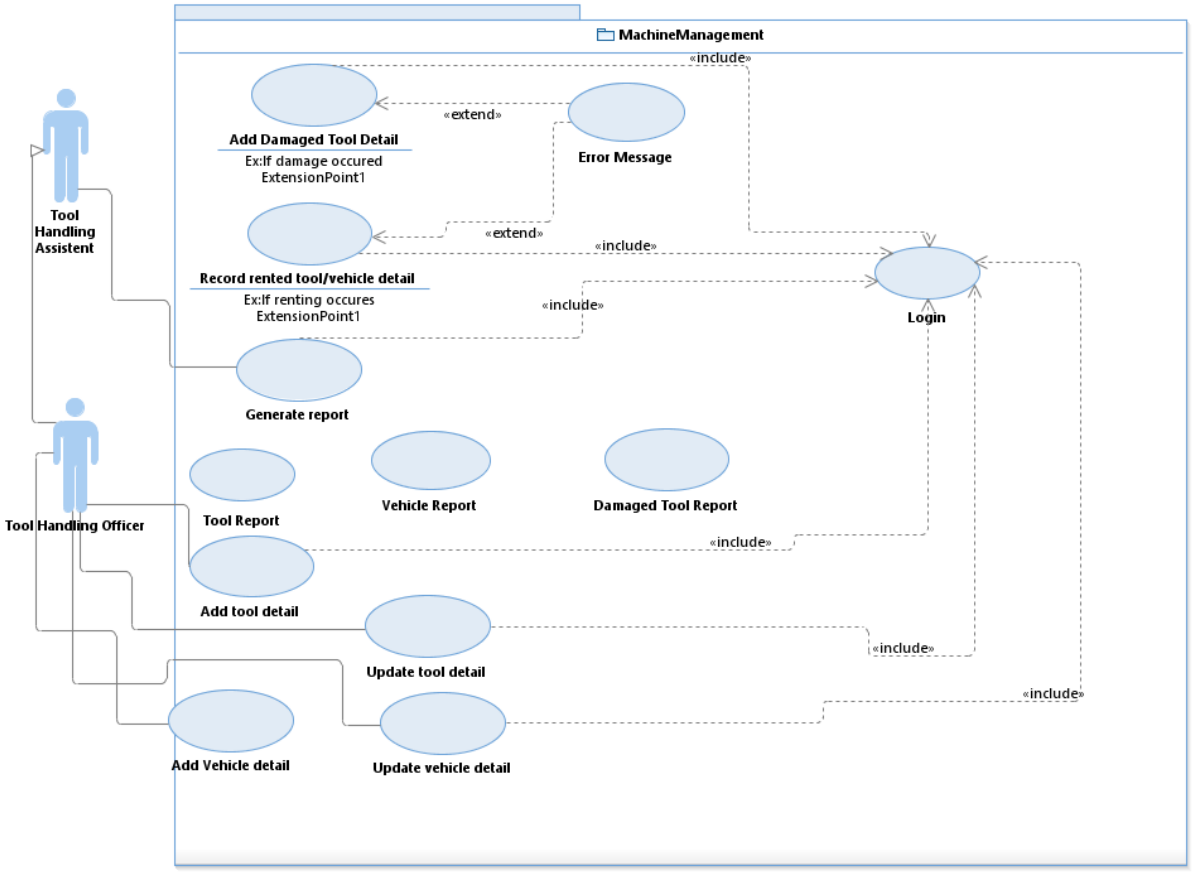


Figure 2.2.23 Use case diagram for Machine Management System

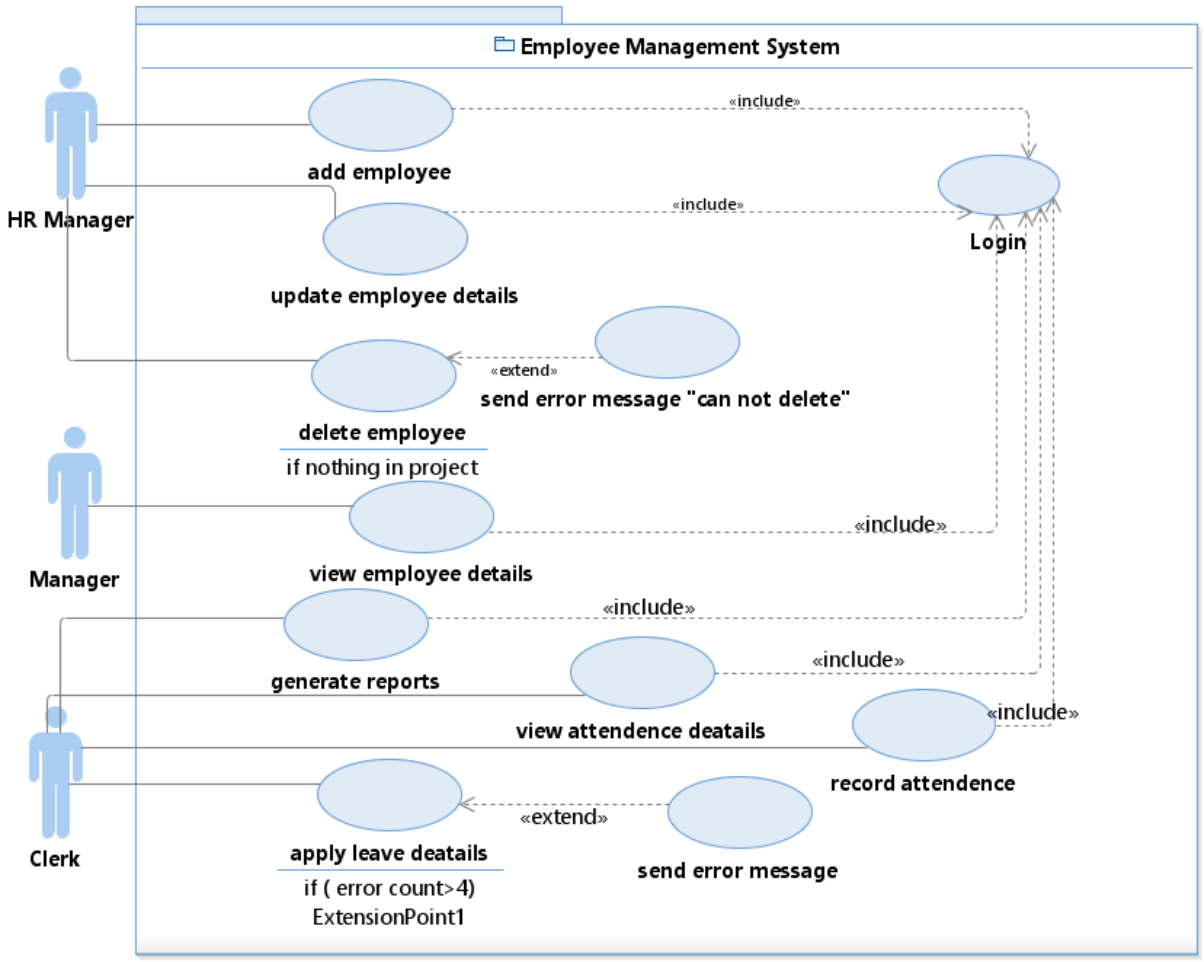


Figure 2.2.24 Use case diagram for Employee Management System

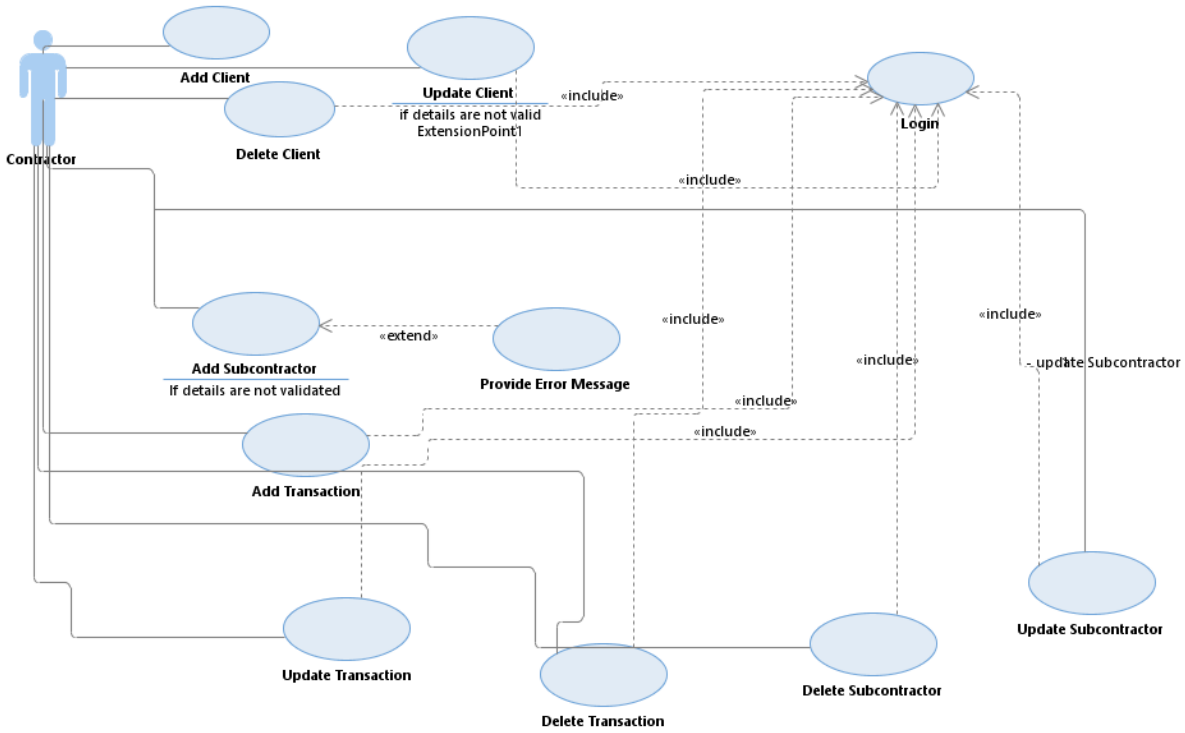


Figure 2.2.25 Use case diagram for Subcontractor and Client Management System

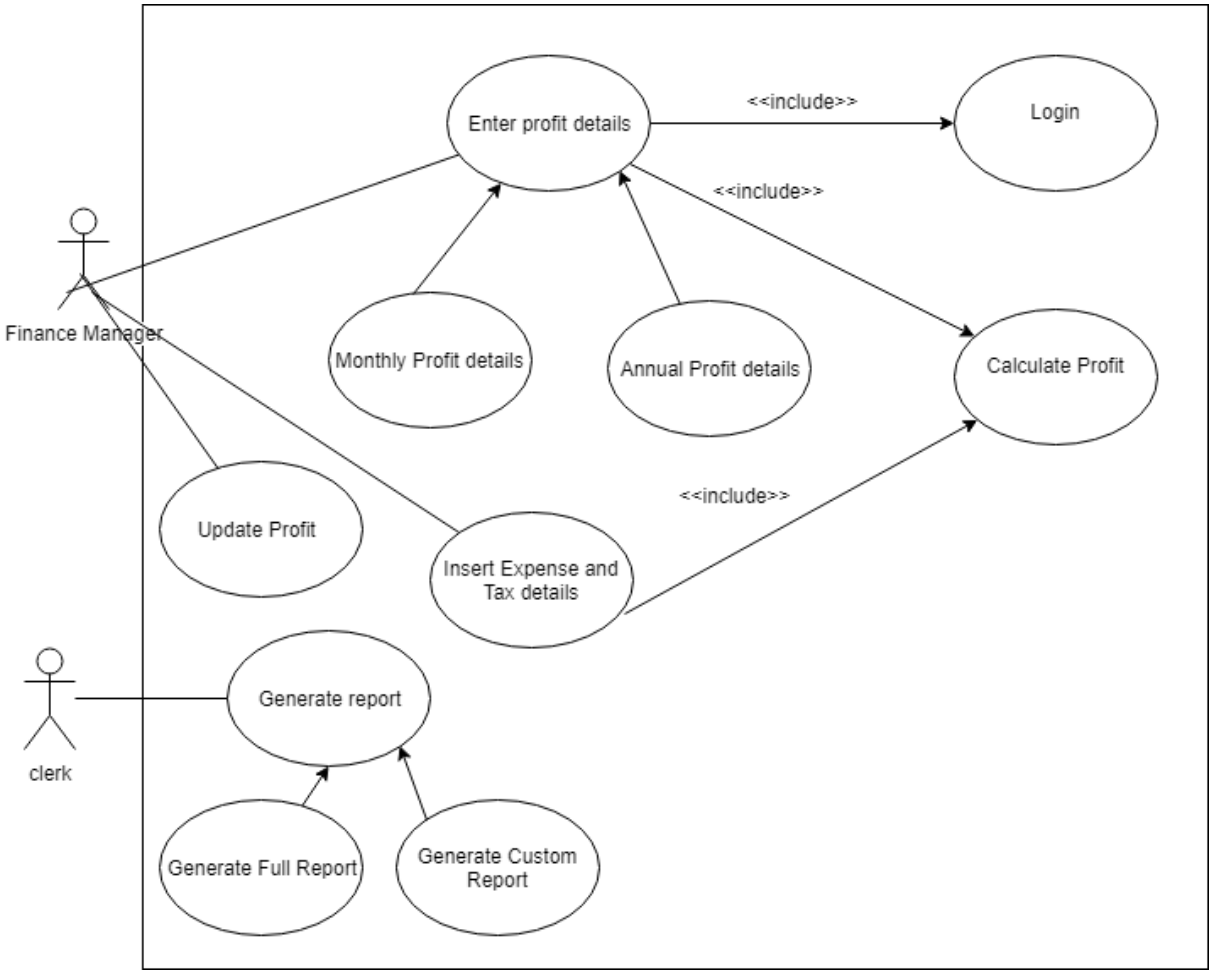


Figure 2.2.26 Use case diagram for Account and Finance Management System

Use Case Scenarios

- **Supplier and Order management system**

Use Case Name	Handle Orders
Pre-Condition	Main Stock Manager has logged into the system
Primary Actor	Main Stock Manager
Main Success Scenario	1.Use case begins when Main Stock Manager select the Handle Orders UI. 2.Insert records of order details 3.Click Add button 4.Terminate the use case
Extensions	2.a.If materials are damaged return to supplier

Table 1 Use Case Scenario - Handle Orders

Use Case Name	Generate Report
Pre-Condition	Main Stock Manager has logged into the system
Primary Actor	Main Stock Manager
Main Success Scenario	1.Use case begins when Main Stock Manager has logged into the system. 2.Select the option Generate report to get whole,custom reports. 3.Terminate the use case.

Table 2 Use Case Scenario - Generate Report

Use Case Name	Update Supplier
Pre-Condition	Main Stock Manager has logged into the system
Primary Actor	Main Stock Manager
Main Success Scenario	1.Use case begins when Main Stock Manager select the Update supplier UI. 2.Change the records of suppliers that you need to update. 3.Terminate the use case
Extensions	2.a.If changed records are not according to the validation criteria ,pop up error message.

Table 3 Use Case Scenario - Update Supplier

Use Case Name	Delete Suppliers
Pre-Condition	Main Stock Manager has logged into the system
Primary Actor	Main Stock Manager
Main Success Scenario	1.Use case begins when Main Stock Manager select the Delete Suppliers UI. 2.Delete records of specified Supplier 3.Click delete button 4.Terminate the use case

Table 4 Use Case Scenario - Delete Suppliers

- **Stock management system**

Use Case name	**Add project material request**
Pre-condition	Stock manager has logged into the system
Primary actor	Stock manger
Main Success scenario	1. Use case begins when the stock manager selects the supply order 2. Clicks the add button 3. Insert the records of wanted materials. 4. Supplied materials details are recorded in the database. 5. Terminate the use case
Extensions	3. a. If the needed quantity of materials is not tally with the materials available, an error message is displayed. 3.a.1) Delay the supply

Table 5 Use Case Scenario - **Add project material request**

Use case name	**Update materials details**
Pre- condition	Stock manage has logged into the system. Inserted material details are available.
Primacy actor	Stock manager
Main success scenario	1. Stock manager selects the materials details. 2.click the data grid view 3. Update the material details are recoded in the database.
Extension	3.a.IF the entered details are false then display an error message

Table 6 Use Case Scenario - **Update materials details**

Use Case name	**Generate report**
Pre-condition	Stock manager has logged into the system
Primary actor	Stock manager
Main Success scenario	1. Use case begins when the stock manager receives the details of the monthly issued stock materials 2. Select the option to generate report of issued stock 3. Terminate the use

Table 7 Use Case Scenario - **Update materials details**

Use case name	**Add supply details**
Pre-condition	Stock manager has logged into the system
Primary actor	Stock manager
Main success scenario	1.require: login 2. Use case begins when the user clicks supplied materials button. 3. supplied materials. Interface displayed. 4. User selects the add order details. 5. Display all the details to be entered. 6. User enters the supplied materials details. 7.manager clicks the add button 8. New record is added to the project supplied table.

Table 8 Use Case Scenario - **Add supply details**

- **Safety management system**

Use Case Name	Add training details
Preconditions	Admin must already logged in to the system
Primary Actor	Admin
Main Flow of Events	1. The use case begins when admin click on the ***Add Training Details*** option. 2. User enter training details. 3. System validate details. 4. Valid data will add to database. 5. System end with display status.
Extensions	3a. if details not valid display error message.

Table 9 Use Case Scenario - Add training details

Use Case Name	Modify /delete Training Details
Preconditions	Admin already logged in to the system
Primary Actor	Admin
Main Flow of Events	1. The use case begins when admin click on the ***Modify/delete training Details*** option. 2. Modify /delete training details based on training id. 3. When modifying system validate details. 4. . System end with display status.
Extensions	2a.when Modifying, if details are not provided in proper manner display error message.

Table 10 Use Case Scenario - Modify /delete Training Details

Use Case Name	Add accident details
Preconditions	Admin must already logged in to the system
Primary Actor	Admin
Main Flow of Events	1. The use case begins when admin click on the ***Add Accident Details*** option. 2. User enter accident details. 3. System validate details. 4. Valid data will add to database. 5. System end with display status.
Extensions	3a. if details not valid display error message.

Table 11 Use Case Scenario - Add accident details

Use Case Name	Modify /delete Accident Details
Preconditions	Admin already logged in to the system
Primary Actor	Admin
Main Flow of Events	1. The use case begins when admin click on the ***Modify/delete accident Details*** option. 2. Modify /delete accident details based on accident id. 3. When modifying system validate details. 4. . System end with display status.
Extensions	2a.when Modifying, if details are not provided in proper manner display error message.

Table 12 Use Case Scenario - Modify /delete Accident Details

Use Case Name	Add Induction details
Preconditions	Admin must already logged in to the system
Primary Actor	Admin
Main Flow of Events	1. The use case begins when admin click on the ***Add induction Details*** option. 2. User enter induction details. 3. System validate details. 4. Valid data will add to database. 5. System end with display status.
Extensions	3a. if details not valid display error message.

Table 13 Use Case Scenario - Add Induction details

Use Case Name	Modify /delete Induction Details
Preconditions	Admin already logged in to the system
Primary Actor	Admin
Main Flow of Events	1. The use case begins when admin click on the ***Modify/delete induction Details*** option. 2. Modify /delete induction details based on induction id. 3. When modifying system validate details. 4. . System end with display status.
Extensions	2a.when Modifying, if details are not provided in proper manner display error message.

Table 14 Use Case Scenario - Modify /delete Induction Details

Use Case Name	Add Risk Assessment details
Preconditions	Admin must already logged in to the system
Primary Actor	Admin
Main Flow of Events	1.The use case begins when admin click on the ***Add risk assessment Details*** option 2. User enter risk details. 3. System validate details. 4. Valid data will add to database. 5. System end with display status.
Extensions	3a. if details not valid display error message.

Table 15 Use Case Scenario - Add Risk Assessment details

Use Case Name	Modify /delete Risk Assessment Details
Preconditions	Admin already logged in to the system
Primary Actor	Admin
Main Flow of Events	1. The use case begins when admin click on the ***Modify/delete risk assessment Details*** option. 2. Modify /delete risk details based on risk id. 3. When modifying system validate details. 4. System end with display status.
Extensions	2a.when Modifying, if details are not provided in proper manner display error message.

Table 16 Use Case Scenario - Modify /delete Risk Assessment Details

- **Project management system**

Use Case Scenario for Project Management

Use Case ID	PM-1	
Use Case Name	Add Project Information	
Description	Adding new project information to the system	
Primary Actor	Construction Manager	
Pre-Condition	User must already logged in to the system	
Main Flow	**Step**	**Action**
	1	The use case begins when user selects ADD PROJECT INFORMATION button.
	2	System loads project information form.
	3	User enters project details.
	4	User selects SAVE button.
	5	The use case ends when system displays success message.
Post-Condition	Newly added project will be available in the system	
Extension	**Step**	**Action**
	4.1	If user did not fill some fields then system provide an alert message to fill them.

Use Case ID	PM-2	
Use Case Name	Update Project Information	
Primary Actor	Construction Manager	
Pre-Condition	User must already logged in to the system	
Main Flow	**Step**	**Action**
	1	The use case begins when user selects ADD PROJECT INFORMATION button.
	2	System loads project information form.
	3	User selects project information from table to update.
	4	User updates project name, owner name and duration.
	5	User selects UPDATE button.
	6	The use case ends when system displays success message
Post-Conditions	Updated project information will be available in the system	
Extensions	**Step**	**Action**
	5.1	If user did not fill some fields then system provide an alert message to fill them.

Table 17 Use Case Scenario - Add Project Information

Table 18 Use Case Scenario - Update Project Information

Use Case ID	PM-3	
Use Case Name	Search Project Information	
Primary Actor	Construction Manager	
Pre-Condition	User must already logged in to the system	
Main Flow	**Step**	**Action**
	1	The use case begins when user selects ADD PROJECT INFORMATION button.
	2	System loads project information form.

	3	User search project by Project ID , Location , Construction Type, Start Date or End Date
	4	User type the value to search and selects search button
	5	The use case ends when system display the information related to search in table.
Extensions	**Step**	**Action**
	4.1	If project information is not found for relevant search then system provide an error message.

Table 19 Use Case Scenario - Search Project Information

Use Case ID	PM-4	
Use Case Name	Change Engineers of Project	
Primary Actor	Construction Manager	
Pre-Condition	User must already logged in to the system	
Main Flow	**Step**	**Action**
	1	The use case begins when user selects ASSIGN ENGINEER button.
	2	System loads assign engineer form.
	3	User select information from table that they want to update.
	4	User can change engineers, add or remove 2nd engineer to or from project.
	5	User selects UPDATE button.
	6	The use case ends when system displays success message
Extensions	**Step**	**Action**
	5.1	If user did not fill some fields then system provide an alert message to fill them.

Table 20 Use Case Scenario - Change Engineers of Project

Use Case ID	PM-5	
Use Case Name	Assign Laborers For Project	
Primary Actor	Construction Manager	
Pre-Condition	User must already logged in to the system	
Main Flow	**Step**	**Action**
	1	The use case begins when user selects ASSIGN LABORERS button.
	2	System loads assign laborer form.

	3	User selects project id and laborers ids for a project
	4	User selects SAVE button.
	5	The use case ends when system displays success message
Extensions	**Step**	**Action**
	3.1	If user select laborer id without select project id then system provide an alert message to select project id.
	4.1	If user did not fill some fields then system provide an alert message to fill them.

Table 21 Use Case Scenario - Assign Laborers for Project

Use Case ID	PM-6	
Use Case Name	Change Laborer of Project	
Primary Actor	Construction Manager	
Pre-Condition	User must already logged in to the system	
Main Flow	**Step**	**Action**
	1	The use case begins when user selects ASSIGN LABORER button.
	2	System loads assign laborer form.
	3	User select information from table that they want to update.
	4	User can change laborer, add or remove 2nd laborer to or from project.
	5	User selects UPDATE button.
	6	The use case ends when system displays success message
Extensions	**Step**	**Action**
	5.1	If user did not fill some fields then system provide an alert message to fill them.

Table 22 Use Case Scenario - Change Laborer of Project

Use Case ID	PM-7	
Use Case Name	Remove Engineers from Project	
Primary Actor	Construction Manager	
Pre-Condition	User must already logged in to the system	
Main Flow	**Step**	**Action**

	1	The use case begins when user selects ASSIGN ENGINEER button.
	2	System loads assign engineer form.
	3	User select information from table that they want to remove.
	4	User selects DELETE button.
	5	The use case ends when system displays success message.

Table 23 Use Case Scenario - Remove Engineers from Project

Use Case ID	PM-8	
Use Case Name	Remove Laborers from Project	
Primary Actor	Construction Manager	
Pre-Condition	User must already logged in to the system	
Main Flow	**Step**	**Action**
	1	The use case begins when user selects ASSIGN LABORER button.
	2	System loads assign laborer form.
	3	User select information from table that they want to remove.
	4	User selects DELETE button.
	5	The use case ends when system displays success message.

Table 24 Use Case Scenario - Remove Laborers from Project

Use Case ID	PM-9	
Use Case Name	Search Project Engineers	
Primary Actor	Construction Manager	
Pre-Condition	User must already logged in to the system	
Main Flow	**Step**	**Action**

	1	The use case begins when user selects ASSIGN ENGINEER button.
	2	System loads assign engineer form.
	3	User enters project id and select search button.
	4	The use case ends when system displays information relevant to search in table.

Table 25 Use Case Scenario - Search Project Engineers

Use Case ID	PM-10	
Use Case Name	Search Project Laborers	
Primary Actor	Construction Manager	
Pre-Condition	User must already logged in to the system	
Main Flow	**Step**	**Action**
	1	The use case begins when user selects ASSIGN LABORER button.
	2	System loads assign laborer form.
	3	User enters project id and select search button.
	4	The use case ends when system displays information relevant to search in table.

Table 26 Use Case Scenario - Search Project Laborers

Use Case ID	PM-11	
Use Case Name	Add Budget For Project	
Primary Actor	Construction Manager	
Pre-Condition	User must already logged in to the system	
Main Flow	**Step**	**Action**
	1	The use case begins when user selects BUDGET CALCULATION button.

	2	System loads budget calculation form.
	3	User enters amount details.
	4	User selects SAVE button.
	6	The use case ends when system displays success message.
Extension	**Step**	**Action**
	4.1	If user did not fill some fields then system provide an alert message to fill them.

Table 27 Use Case Scenario - Add Budget for Project

Use Case ID	PM-12	
Use Case Name	Delete Budget Details of Project	
Primary Actor	Construction Manager	
Pre-Condition	User must already logged in to the system	
Main Flow	**Step**	**Action**
	1	The use case begins when user selects BUDGET CALCULATION button.
	2	System loads budget calculation form.
	3	User selects information from table to delete.
	4	User selects DELETE button.
	6	The use case ends when system displays success message.

Table 28 Use Case Scenario - Delete Budget Details of Project

Use Case ID	PM-13	
Use Case Name	Update Budget Details of Project	
Primary Actor	Construction Manager	
Pre-Condition	User must already logged in to the system	
Main Flow	**Step**	**Action**

	1	The use case begins when user selects BUDGET CALCULATION button.
	2	System loads budget calculation form.
	3	User selects information from table to update.
	4	User selects UPDATE button.
	6	The use case ends when system displays success message.
Extension	**Step**	**Action**
	4.1	If user did not fill some fields then system provide an alert message to fill them.

Table 29 Use Case Scenario - Update Budget Details of Project

Use Case ID	PM-14	
Use Case Name	Assign Engineer For Project	
Primary Actor	Construction Manager	
Pre-Condition	User must already logged in to the system	
Main Flow	**Step**	**Action**
	1	The use case begins when user selects ASSIGN ENGINEER button.
	2	System loads assign engineer form.
	3	User selects project id and engineer ids for a project
	4	User selects SAVE button.
	6	The use case ends when system displays success message.
Extension	**Step**	**Action**
	4.1	If user did not fill some fields then system provide an alert message to fill them.

Table 30 Use Case Scenario - Assign Engineer for Project

Use Case ID	PM-15
Use Case Name	Generate Report

Primary Actor	Construction Manager	
Pre-Condition	User must already logged in to the system	
Main Flow	**Step**	**Action**
	1	The use case begins when user selects GENERATE REPORTS button.
	2	System loads generate reports form.
	3	User click relevant report button.
	4	The use case ends when system displays report.
Extension	**Step**	**Action**
	3.1	If user selects Full Report button, it generates full detailed report.
	3.2	If user search location or construction type and select report button, it generates customized report.

Table 31 Use Case Scenario - Generate Report

- **Machine management system**

Use Case ID	MM-01	
Use Case Name	Add Purchased tool details	
Primary Actor	Tool management officer	
Pre-Condition	User must already logged in to the system	
Main Flow	**Step**	**Action**
	1	The use case begins when user selects TOOL MANAGEMENT button.
	2	System loads Tool management form.
	3	User enters the relevant tool details and press ADD button.
	4	The use case ends when system displays Insertion success message to the user.
Extension	**Step**	**Action**
	3.1	If user did not fill some fields then system provide an alert message to fill them.

	3.2	If user enters a warranty end date greater than the purchase date, display an error message.

Table 32 Use Case Scenario – Add tool details

Use Case ID	MM-02	
Use Case Name	Update Purchased tool details	
Primary Actor	Tool management officer	
Pre-Condition	User must already logged in to the system	
Main Flow	**Step**	**Action**
	1	The use case begins when user selects TOOL MANAGEMENT button.
	2	System loads Tool management form.
	3	User updates the relevant tool details and press UPDATE button.
	4	The use case ends when system displays Update success message to the user.
Extension	**Step**	**Action**
	3.1	If user did not fill some fields then system provide an alert message to fill them.

Table 33 Use Case Scenario – Update tool details

Use Case ID	MM-03	
Use Case Name	Delete Purchased tool details	
Primary Actor	Tool management officer	
Pre-Condition	User must already logged in to the system	
Main Flow	**Step**	**Action**
	1	The use case begins when user selects TOOL MANAGEMENT button.
	2	System loads Tool management form.
	3	User deletes the relevant tool details and press DELETE button.
	4	The use case ends when system displays Deletion success message to the user.

Table 34 Use Case Scenario – Delete tool details

Use Case ID	MM-04	
Use Case Name	Assign tools for projects	
Primary Actor	Tool management officer	
Pre-Condition	User must already logged in to the system	
Main Flow	**Step**	**Action**
	1	The use case begins when user selects TOOL MANAGEMENT button.
	2	System loads Tool management form.
	3	User makes the availability to Not Available.
	4	User selects the project ID and assigning date and then the Update button.
	5	The use case ends when system displays Update success message to the user.
Extension	**Step**	**Action**
	3.1	If user did not make the availability to not available, then the project ID, assigning date and returning date columns will not be enabled.

Table 35 Use Case Scenario – Assign tools for projects

Use Case ID	MM-05	
Use Case Name	Add Damaged tools details.	
Primary Actor	Tool management officer	
Pre-Conditions	User must already logged in to the system, a tool must be damaged	
Main Flow	**Step**	**Action**
	1	The use case begins when user selects DAMAGED TOOL MANAGEMENT button.
	2	System loads Damaged Tool management form.
	3	User selects the damaged from the list of the tools.
	4	User selects the decision (send for repair or discard)
	5	User inserts the details and click on the ADD button.
	6	The use case ends when system displays Add success message to the user.
Extension	**Step**	**Action**
	4.1	If user selects Discard option, the selected tool detail will be deleted from the table.

Table 36 Use Case Scenario – Add Damaged tools details.

Use Case ID	MM-06	
Use Case Name	Add Vehicle details.	
Primary Actor	Tool management officer	
Pre-Conditions	User must already logged in to the system, a tool must be damaged	
Main Flow	**Step**	**Action**
	1	The use case begins when user selects VEHICLE MANAGEMENT button.
	2	System loads Vehicle management form.
	3	User enters the relevant vehicle details and press ADD button.
	4	The use case ends when system displays Insertion success message to the user.
Extension	**Step**	**Action**
	3.1	If user did not fill some fields then system provide an alert message to fill them.

Table 37 Use Case Scenario – Add vehicle details.

Use Case ID	MM-07	
Use Case Name	Update Vehicle details	
Primary Actor	Tool management officer	
Pre-Condition	User must already logged in to the system	
Main Flow	**Step**	**Action**
	1	The use case begins when user selects VEHICLE MANAGEMENT button.
	2	System loads Vehicle management form.
	3	User updates the relevant vehicle details and press UPDATE button.
	4	The use case ends when system displays Update success message to the user.
Extension	**Step**	**Action**
	3.1	If user did not fill some fields then system provide an alert message to fill them.

Table 38 Use Case Scenario – Update vehicle details.

Use Case ID	MM-08	
Use Case Name	Delete Vehicle details	
Primary Actor	Tool management officer	
Pre-Condition	User must already logged in to the system	
Main Flow	**Step**	**Action**
	1	The use case begins when user selects VEHICLE MANAGEMENT button.
	2	System loads Vehicle management form.
	3	User deletes the relevant vehicle details and press DELETE button.
	4	The use case ends when system displays Deletion success message to the user.

Table 39 Use Case Scenario – Delete vehicle details.

Use Case ID	MM-09	
Use Case Name	Add Rented tool details.	
Primary Actor	Tool management officer	
Pre-Conditions	User must already logged in to the system, a tool must be damaged	
Main Flow	**Step**	**Action**
	1	The use case begins when user selects RENTED TOOL MANAGEMENT button.
	2	System loads Rented tool & vehicle management form.
	3	User enters the relevant tool/vehicle details and press ADD button.
	4	The use case ends when system displays Insertion success message to the user.
Extension	**Step**	**Action**
	3.1	If user did not fill some fields then system provide an alert message to fill them.

Table 40 Use Case Scenario – Add vehicle details.

- **Employee Management**

Use Case Name	Add Employee
Pre-Condition	HR Manager needs to login into the system
Primary Actor	HR Manager
Main Success Scenario	1. Use case begins when HR Manager select the Add Employee UI. 2.Insert records of Employee details 3.Click Add button 4.Terminate the use case
Extensions	2.a.If records are not validated generate error message

Table 41 Use Case Scenario - Add Employee

Use Case Name	Generate Report
Pre-Condition	Clerk has logged into the system
Primary Actor	Clerk
Main Success Scenario	1. Use case begins when Clerk has logged into the system. 2.Select the option Generate report to get whole, custom reports. 3. Terminate the use case.

Table 42 Use Case Scenario - Generate Report

Use Case Name	Update Employee details
Pre-Condition	HR Manager has logged into the system
Primary Actor	HR Manager
Main Success Scenario	1.Use case begins when HR Manager select the Update Employee UI. 2.Change the records of Employee that you need to update. 3.Terminate the use case
Extensions	2.a.If changed records are not according to the validation criteria ,pop up error message.

Table 43 Use Case Scenario - Update Employee details

Use Case Name	Delete Employee
Pre-Condition	HR Manager has logged into the system
Primary Actor	HR Manager
Main Success Scenario	1.Use case begins when HR Manager select the Delete Employee UI. 2.Delete records of specified Employee 3.Click delete button 4.Terminate the use case

Table 44 Use Case Scenario - Delete Employee

- **Subcontractor and Client management system**

Use Case Name	Add Client
Pre-Condition	Contractor needs to login into the system
Primary Actor	Contractor
Main Success Scenario	1. Use case begins when Contractor select the Add Client UI. 2.Insert records of Client details 3.Click Add button 4.Terminate the use case
Extensions	2.a.If records are not validated generate error message

Table 45 Use Case Scenario - Add Client

Use Case Name	Update Client details
Pre-Condition	Contractor has logged into the system
Primary Actor	Contractor
Main Success Scenario	1. Use case begins when Contractor select the Update Client UI. 2. Change the records of Client that you need to update. 3.Terminate the use case
Extensions	2. a. If changed records are not according to the validation criteria, pop up error message.

Table 46 Use Case Scenario - Update Client details

Use Case Name	Delete Client
Pre-Condition	Contractor has logged into the system
Primary Actor	Contractor
Main Success Scenario	1. Use case begins when Contractor select the Delete Client UI. 2.Delete records of specified Client 3.Click delete button 4.Terminate the use case

Table 47 Use Case Scenario - Delete Client

2.3 Implementation

The application was implemented using C# language which is contain more features and system access Sql server database which consist of the capability of producing results table efficient. Following code was implemented commonly to gathering objects from the database and to avoid Redundant of code. Part belongs to making connection with the database, was reduced by creating data returning methods. It provided the gateway of windows form to access system database. By calling function can easily access database.

```
class DBconnection
  {
    public static SqlConnection NewCon;
        public static string ConStr = "Data Source=THENUJA;Initial
      Catalog=KSL;Integrated Security=True";
    public static SqlConnection GetConnection()
    {
      NewCon = new SqlConnection(ConStr);
      return NewCon;
    }
  }
```

2.4 Testing

Once the implementation phase is finished, those individual modules are tested well before the integration. Each module is error free system will start to integrate all module. . Each integrated module was tested to make sure that it is working properly. Next the fully integrated system was tested again until the logical and syntax error are eliminates. Finally the system was tested for higher and accurate performance.

Supplier and Order Management System		
Input	*Output*	*Test Results*
Save the data without inserting value to text fields	*Error message will be display*	✓
Save data inserting value to text fields	*Relevant data save in to related database*	✓
Update the data without inserting value to text fields	*Error message will be display*	✓
Update data inserting value to text fields	*Relevant data update in to database*	✓
Delete any empty text field	*Error message will be display*	✓
Delete selected field value to any input fields	*Relevant data delete from database*	✓
Search a Product or any other related information	*Relevant result will be display*	✓
Search non related information	*Result will not display.*	✓
All input text field validated according their input characters	*Error message will be display.*	✓
Type non- related characters on any input field	*Error message will be appear display*	✓

Table 2.4.1 Validation testing for Supplier and Order management system

Test Case ID	TEST-CASE-TC001
Test Case Summary	Add Order details
Test Plan	1.Login as a Main Stock Manager 2.Go to Order interface 3. Enter all related details in correct format. 4.Click 'Add' button to Add the Order
Expected Output	A successful message 'Successfully added'
Test Date	**Order ID**– O101 **Supplier ID**– S101 **Project ID**– KSL101 **Order Date**– 24-03-2018 **Deliver Date** – 28-03-2018 **Net Cost** – Rs.25000.00 **TransportCost**Rs.5000.00 **Total Cost** – Rs.30000.00
Outcome	A successful message 'Successfully added'
Test Status	Passed

Table 2.4.2 Data-driven testing for Supplier and Order management system

Stock Management System		
Input	*Output*	*Test Results*
Save the data without inserting value to text fields	*Error message will be display*	✓
Save data inserting value to text fields	*Relevant data save in to related database*	✓
Update the data without inserting value to text fields	*Error message will be display*	✓
Update data inserting value to text fields	*Relevant data update in to database*	✓
Delete any empty text field	*Error message will be display*	✓
Delete selected field value to any input fields	*Relevant data delete from database*	✓

Search a Product or any other related information	*Relevant result will be display*	✓
Search non related information	*Result will not display.*	✓
All input text field validated according their input characters	*Error message will be display.*	✓
Type non- related characters on any input field	*Error message will be appear display*	✓

Table 2.4.3 Validation testing for Stock management system

Test Case ID	TEST-CASE-TC002
Test Case Summary	Add Supplied Materials Details
Test Plan	1.Login as an Assistant Stock Manager 2.Go to Supplied Material interface 3. Enter all related details in correct format. 4.Click 'Add' button to Add the Materials
Expected Output	A successful message 'Successfully added'
Test Date	**Supplied ID**– SP101 **Project ID**– KSL101 **Material ID**– MD01 **Supply Date**– 24-03-2018 **Quantity** – 55 **Type** – Bulk **Supply Quantity** -20
Outcome	A successful message 'Successfully added'
Test Status	Passed

Table 2.4.4 Data-driven testing for Stock management system

Safety Management System		
Input	*Output*	*Test Results*
Save the data without inserting value to text fields	*Error message will be display*	✓
Save data inserting value to text fields	*Relevant data save in to related database*	✓
Update the data without inserting value to text fields	*Error message will be display*	✓
Update data inserting value to text fields	*Relevant data update in to database*	✓
Delete any empty text field	*Error message will be display*	✓
Delete selected field value to any input fields	*Relevant data delete from database*	✓
Search a Product or any other related information	*Relevant result will be display*	✓
Search non related information	*Result will not display.*	✓
All input text field validated according their input characters	*Error message will be display.*	✓
Type non- related characters on any input field	*Error message will be appear display*	✓

Table 2.4.5 Validation testing for Safety management system

Test Case ID	TEST-CASE-TC003
Test Case Summary	Add Induction details
Test Plan	1.Login as an Admin 2.Go to Induction interface 3. Enter all related details in correct format. 4.Click ‘Add’ button to Add the Induction
Expected Output	A successful message ‘Successfully added’
Test Date	**Induction ID**– IN_101 **Employee ID**– E101 **Employee Name** – Thenuja **Position**– 24-03-2018 **Start Date** – 28-03-2018 **End Date** – 29-03-2018 **Content –**Managing Fire **Status-** Done
Outcome	A successful message ‘Successfully added’
Test Status	Passed

Table 2.4.6 Data-driven testing for Safety management system

Project Management System		
Input	*Output*	*Test Results*
Save the data without inserting value to text fields	*Error message will be display*	✓
Save data inserting value to text fields	*Relevant data save in to related database*	✓
Update the data without inserting value to text fields	*Error message will be display*	✓
Update data inserting value to text fields	*Relevant data update in to database*	✓
Delete any empty text field	*Error message will be display*	✓
Delete selected field value to any input fields	*Relevant data delete from database*	✓
Search a Product or any other related information	*Relevant result will be display*	✓
Search non related information	*Result will not display.*	✓
All input text field validated according their input characters	*Error message will be display.*	✓
Type non- related characters on any input field	*Error message will be appear display*	✓

Table 2.4.7 Validation testing for Project management system

Test Case ID	TEST-CASE-TC004
Test Case Summary	Add Project details
Test Plan	1.Login as a Construction Manager 2.Go to Add Project interface 3. Enter all related details in correct format. 4.Click 'Save' button to Add the Project details
Expected Output	A successful message 'Information are successfully added'
Test Date	**Project ID**- KSL 101 **Construction Type-** HOUSE **Project Name –** KSLHOUSE101 **Owner Name –** Raj Kumar **Property Size(In perches) –** 10 **Location of the land -** Negombo **Start Date** – 04/06/2018 **End Date** – 05/07/2019 **Duration –** 1year/s 1month/s
Outcome	A successful message 'Information are successfully added'
Test Status	Passed

Table 2.4.8 Data-driven testing for Project management system

Machine Management System		
Input	*Output*	*Test Results*
Save the data without inserting value to text fields	*Error message will be display*	✓
Save data inserting value to text fields	*Relevant data save in to related database*	✓
Update the data without inserting value to text fields	*Error message will be display*	✓
Update data inserting value to text fields	*Relevant data update in to database*	✓
Delete any empty text field	*Error message will be display*	✓
Delete selected field value to any input fields	*Relevant data delete from database*	✓
Search a Product or any other related information	*Relevant result will be display*	✓
Search non related information	*Result will not display.*	✓
All input text field validated according their input characters	*Error message will be display.*	✓
Type non- related characters on any input field	*Error message will be appear display*	✓

Table 2.4.9 Validation testing for Machine management system

Test Case ID	TEST-CASE-TC005
Test Case Summary	Add Rented machine details
Test Plan	1.Login as an Admin 2.Go to Rented machine interface 3. Enter all related details in correct format. 4.Click 'Add' button to Add the Rented machine
Expected Output	A successful message 'Successfully added'
Test Date	**Rent ID**– IN_101 **Category**– Tool **Name** – Hammer **Rented Date**– 24-03-2018 **Return Date** – 28-03-2018 **Rent Amount** – 3000 **Quantity –**10 **Status-** sent
Outcome	A successful message 'Successfully added'
Test Status	Passed

Table 2.4.10 Data-driven testing for Machine management system

Employee Management System		
Input	*Output*	*Test Results*
Save the data without inserting value to text fields	*Error message will be display*	✓
Save data inserting value to text fields	*Relevant data save in to related database*	✓
Update the data without inserting value to text fields	*Error message will be display*	✓
Update data inserting value to text fields	*Relevant data update in to database*	✓
Delete any empty text field	*Error message will be display*	✓
Delete selected field value to any input fields	*Relevant data delete from database*	✓
Search a Product or any other related information	*Relevant result will be display*	✓
Search non related information	*Result will not display.*	✓
All input text field validated according their input characters	*Error message will be display.*	✓
Type non- related characters on any input field	*Error message will be appear display*	✓

Table 2.4.11 Validation testing for Employee management system

Test Case ID	TEST-CASE-TC006
Test Case Summary	Add Employee details
Test Plan	1.Login as an Admin 2.Go to Employee interface 3. Enter all related details in correct format. 4.Click 'Add' button to Add the Employee
Expected Output	A successful message 'Successfully added'
Test Date	**Employee ID**– E101 **Employee Name** – Thenuja **Position**– Project Manager **NIC** – 975311341V **DOB** – 29-03-1997 **Safety-** Done **Contact Number –** 0774494665 **Address -**Malabe
Outcome	A successful message 'Successfully added'
Test Status	Passed

Table 2.4.12 Data-driven testing for Employee management system

Subcontractor and Client Management System		
Input	*Output*	*Test Results*
Save the data without inserting value to text fields	*Error message will be display*	✓
Save data inserting value to text fields	*Relevant data save in to related database*	✓
Update the data without inserting value to text fields	*Error message will be display*	✓
Update data inserting value to text fields	*Relevant data update in to database*	✓
Delete any empty text field	*Error message will be display*	✓
Delete selected field value to any input fields	*Relevant data delete from database*	✓
Search a Product or any other related information	*Relevant result will be display*	✓
Search non related information	*Result will not display.*	✓
All input text field validated according their input characters	*Error message will be display.*	✓
Type non- related characters on any input field	*Error message will be appear display*	✓

Table 2.4.13 Validation testing for Subcontractor and Client management system

Test Case ID	TEST-CASE-TC007
Test Case Summary	Add Client details
Test Plan	1.Login as an Contract Manager 2.Go to Client interface 3. Enter all related details in correct format. 4.Click ‘Add’ button to Add the Client
Expected Output	A successful message ‘Successfully added’
Test Date	**Client ID**– C101 **Client Name** – Shangavie **Contact Number –** 0774494665 **Address** -Malabe
Outcome	A successful message ‘Successfully added’
Test Status	Passed

Table 2.4.14 Data-driven testing for Subcontractor and Client management system

Account and Finance Management System		
Input	*Output*	*Test Results*
Save the data without inserting value to text fields	*Error message will be display*	✓
Save data inserting value to text fields	*Relevant data save in to related database*	✓
Update the data without inserting value to text fields	*Error message will be display*	✓
Update data inserting value to text fields	*Relevant data update in to database*	✓
Delete any empty text field	*Error message will be display*	✓
Delete selected field value to any input fields	*Relevant data delete from database*	✓
Search a Product or any other related information	*Relevant result will be display*	✓
Search non related information	*Result will not display.*	✓
All input text field validated according their input characters	*Error message will be display.*	✓
Type non- related characters on any input field	*Error message will be appear display*	✓

Table 2.4.15 Validation testing for Account and Finance management system

Test Case ID	TEST-CASE-TC008
Test Case Summary	Add Induction details
Test Plan	1.Login as an Admin 2.Go to Induction interface 3. Enter all related details in correct format. 4.Click 'Add' button to Add the Induction
Expected Output	A successful message 'Successfully added'
Test Date	**Induction ID**– IN_101 **Employee ID**– E101 **Employee Name** – Thenuja **Position**– 24-03-2018 **Start Date** – 28-03-2018 **End Date** – 29-03-2018 **Content –**Managing Fire **Status-** Done
Outcome	A successful message 'Successfully added'
Test Status	Passed

Table 2.4.16 Data-driven testing for Account and Finance management system

3. Evaluation

3.1 Assessment of the Project results

One of the major and initial step for marking out the entrance for the project is formation of a group with eight members that hold different aspects and achievements with Knowledge in programming with specified skills and talents. With some regular discussions those aspects and talents were identified and differentiation was reduced. Several discussions with group leader and team members along with the client discussions made is easier to identify user requirements.

After conducting several questionnaires with the client exact scope of the system's outline for the team purposes was emended. Thereafter reporting and attending service was analyzed until a proper hierarchy developed. Some of the brainstorming and mind mapping sessions among the group members aided in analysis of requirements accurately to the top most level with relevant achievement. The project was aimed to provide the efficient solutions to the external users and improve the productivity of internal users.

Our Supervisor Ms. Janani Tharmaseelan provided us with the relevant guidelines throughout the development process. Several presentations were scheduled at several milestones such as 'Proposal Presentation', 'Prototype Presentation' and 'Final Presentation' to assess the progress or the development of the project. Most of the steps that we took were recorded in the documents that we have created and those reports were evaluated by our Supervisor Ms. Janani Tharmaseelan . She advised us to do the modifications as necessary. The presentations mentioned above gave us the chance to present the development of the project time to time and those works were observed by a panel of lecturers and those observations resulted us to follow the standards of the project as much as we could.

3.2 Lessons Learned

There were lots of lessons that we learned from throughout the time we spent for developing this project. Most importantly we learned to work according to a time schedule and how to manage the time to take the maximum benefit of the time we had.

We faced lot of inconveniences while engaging with the project development and those incidents gave us the experiences of facing the problems and finding the appropriate solutions as necessary. It is very hard to develop a first rate software without the given time period, because we were not well experienced. But at the end of this project we have learnt many things about the industry and how to face problems and how to overcome those problems easily.

We faced many difficulties during coding phase and project delivery. Since we have grouped together, we were able to overcome those difficulties. Implementation and discover age of user friendly interfaces with the aid of new technology was in success. During the phase of implementation new techniques of coding, faster and accurate methods of interacting with the database were learned.

Working as group was aided to learn sharing knowledge, thoughts and ideas, each individual possessed a positive attitude by working together. Also the team spirit has been developed.

Identifying this problematic situation clearly and step by step approach of discovering the best solution were some various challenges and the team really totally was developed. We gained experience in applying our knowledge to achieve goals in practical situations.

This will help us immensely in developing software solution in our professional work in the future. When we develop a new software in the future, all of us will have confidence level far greater than before. Therefore we will be able to perform as real software engineers.

3.3 Future Work

The work in the future that we have to engage should be done with lessons we learned from the project work that we have done so far. The steps that we have to be taken should be included with a higher standard than the tasks we did in the past. We should be able to reduce the errors that we have identified all through the time we spent for the project and that should be included with the experiences we gathered but with a high standard.

When the time passes, our client's requirements can be changed. They might need more functionalities, new designs, new concepts or change existing functions that are not necessary. In such situations if they wanted us to develop and improve the current system we hope to consider their requirements and make the relevant changes as the system should be more usable and accurate for the client which fulfill their expected tasks. Apart from client requirements, if there are any changes in the system environment or inside the system, we will adapt the system to compatible with those changes as the system should be properly work according the available environments and functionalities.

Furthermore if there are any unexpected errors occur while using our system due to system failures or other reasons, we will be responsible of developing and handling those cases whenever they requested.

4. Conclusion

Construction Company management system standalone application comes with the centralized and large database. It also give a simple interface to user to handle the system easily. It make sure that the data access is reliable and efficient also overcome from the existing system. Here are solutions that proposed by our system.

When number of data increased the number of files also increased therefore no need to worry about those things our system come up with a centralized database so don't need to worry about storage. Calculations is most time consuming work and should be careful with those things they need to calculate like find total product, total no of employees.

These are easy to find in a minute by system. It's easy to record large amount of data in correct format without error and also can retrieve when it's needed. Searching of a specific employee details or a specific product details can be done less than a minute. No need to worry about data duplication. With our system Simple data duplication can be done in a minute. No need to talk about data loss or data misplaced whenever a data or a set of data needed it can be retrieve Since the existing manual system do not consists of a proper report generation, with this new

system it was able to give a solution to generate realistic reports, which would help the management to get practical decisions and ideas related to the process. Here the system would eliminate out all the unauthorized access which eventually leads to a high security. This is authenticated by user login. User friendly interfaces that were implemented allow quick navigation. This is mostly done by the tabs applied here. Therefore whole process can be easily understood. Validation processes that were implemented do not allow the user to enter wrong information leading data inconsistency.

All the data entered can be modified and viewed on a specified way that they are stored inside a centralized database. System is highly efficient hence it increases the productivity and quality along with the realistic decisions that are to be customized and let the management to produce reports included with them.

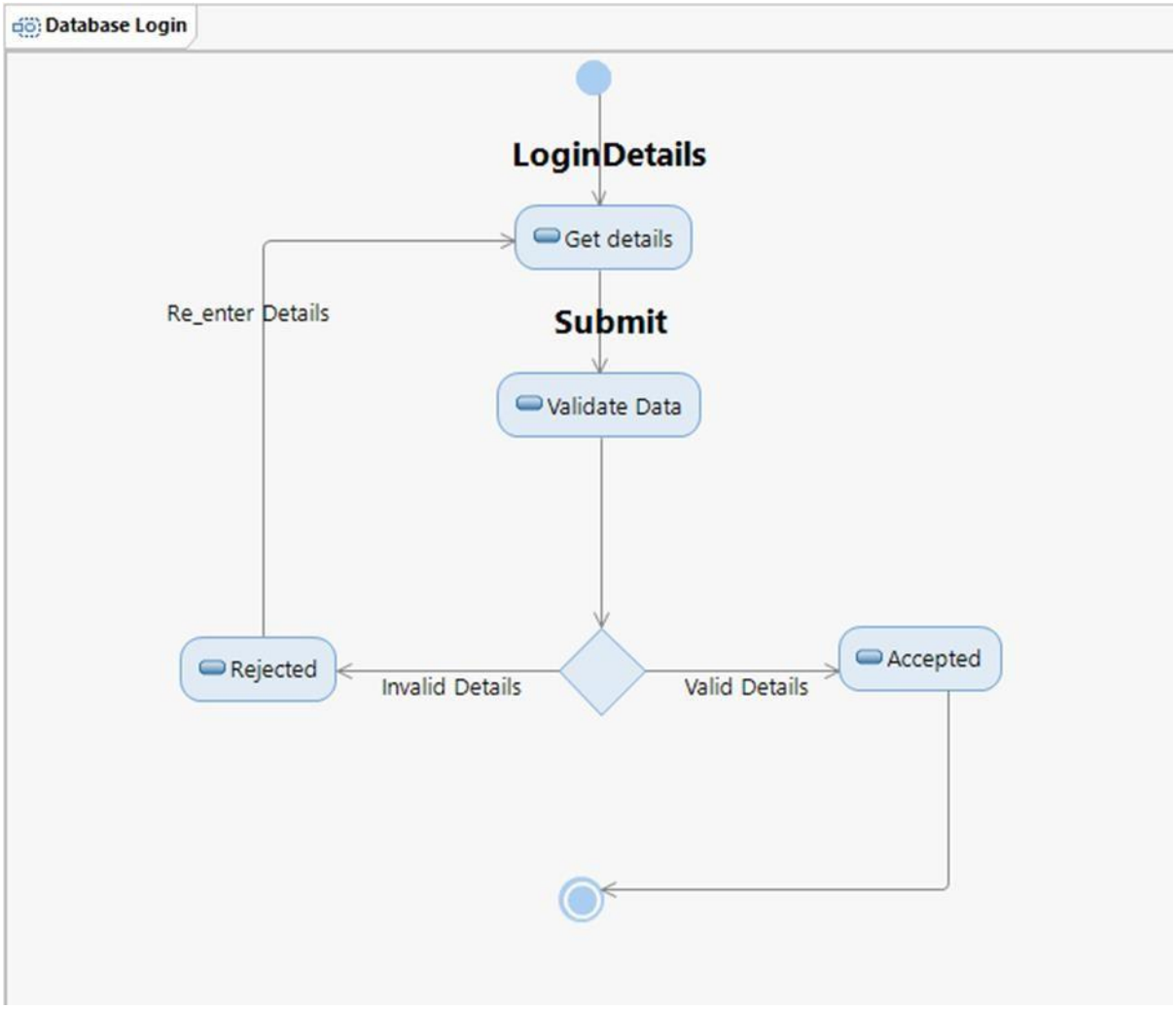
5. References

[1] Microsoft Corporation, "SQL :: Microsoft SQL 5.7 Reference Manual," dev.mysql.com, August 9 2017. [online]. Available:https://docs.microsoft.com/en-us/sql/t-sql/language-reference?view=sql-server-2017 [Accessed:March.30, 2017].

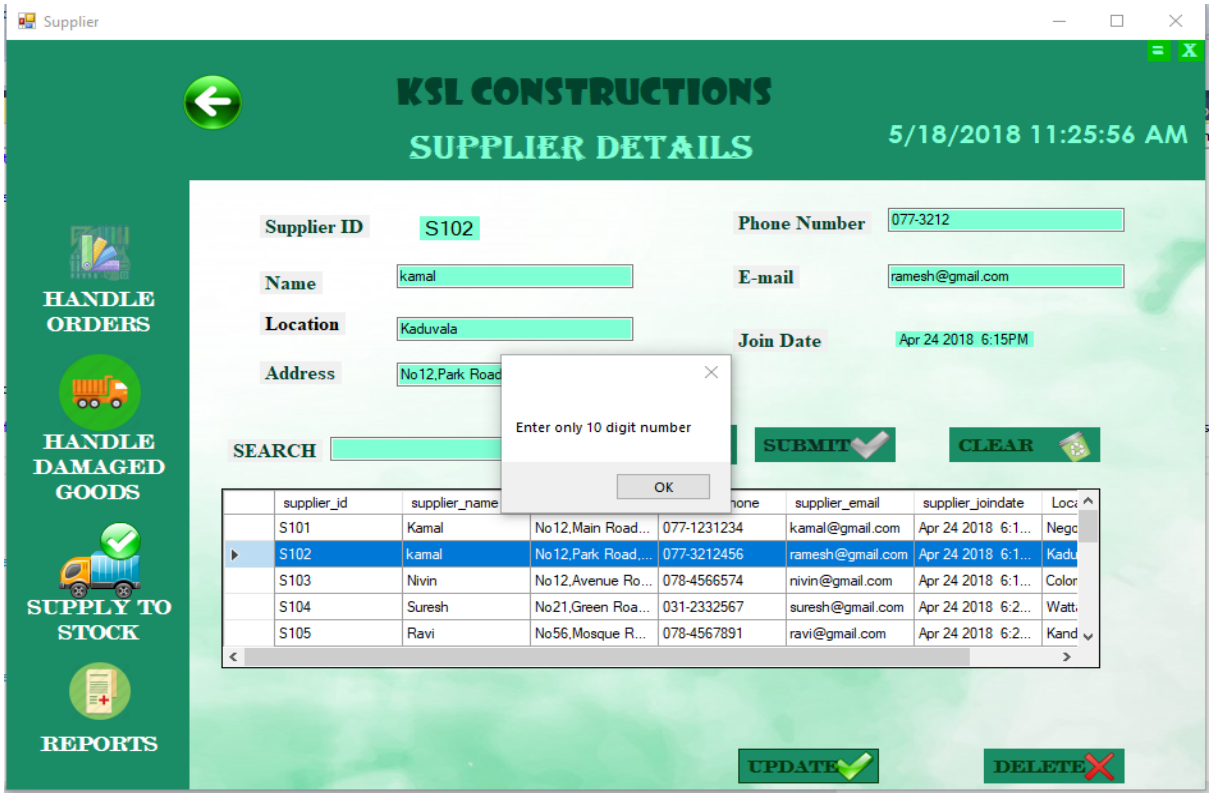
[2] #91 Adam Ralph, NServiceBus, Microservices and SOA, et al., "C#," docs.microsoft.com, March 12 2018. [online].Available: https://nodogmapodcast.bryanhogan.net/91-adam-ralph-nservicebus-microservices-and-soa/ [Accessed:March.20, 2018].

[3] KSL,"Construction Company Management", presented at Construction Management Systems Conference, Colombo, Sri Lanka, 2018 <Include a list of references done in the IEEE referencing style>

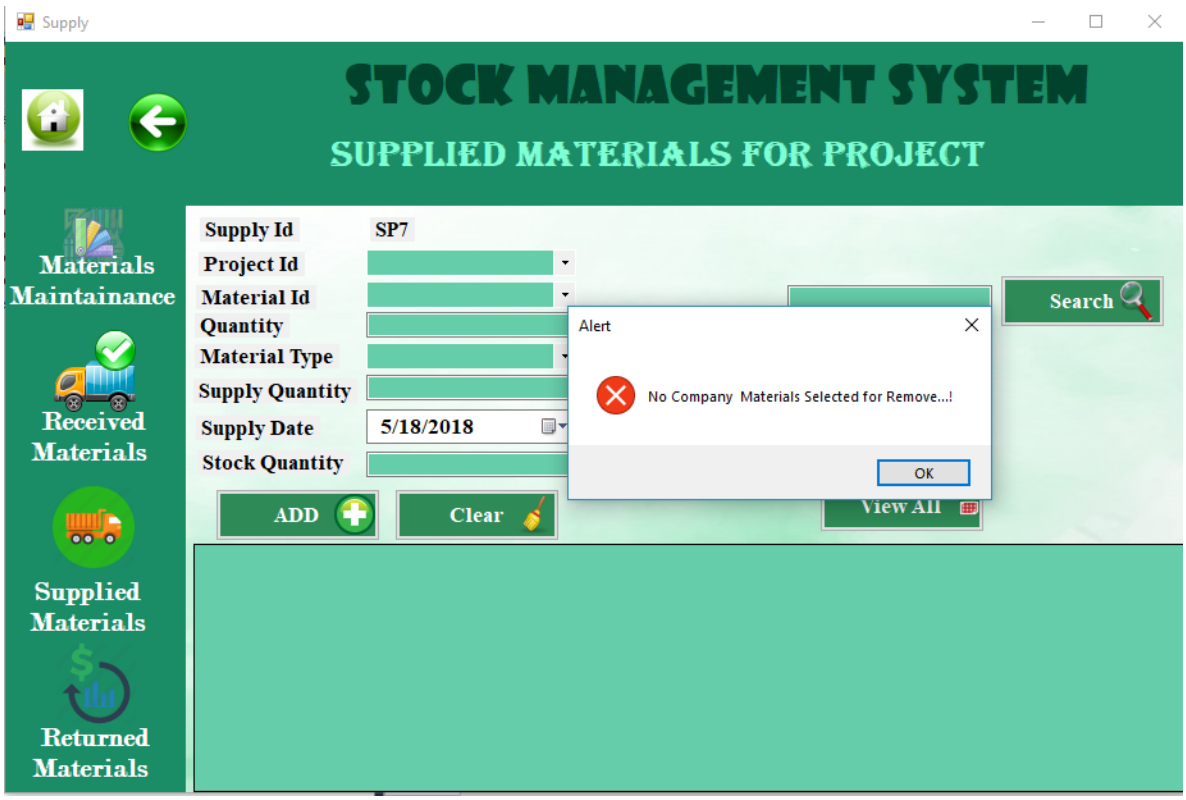
Appendix A: Design Diagrams



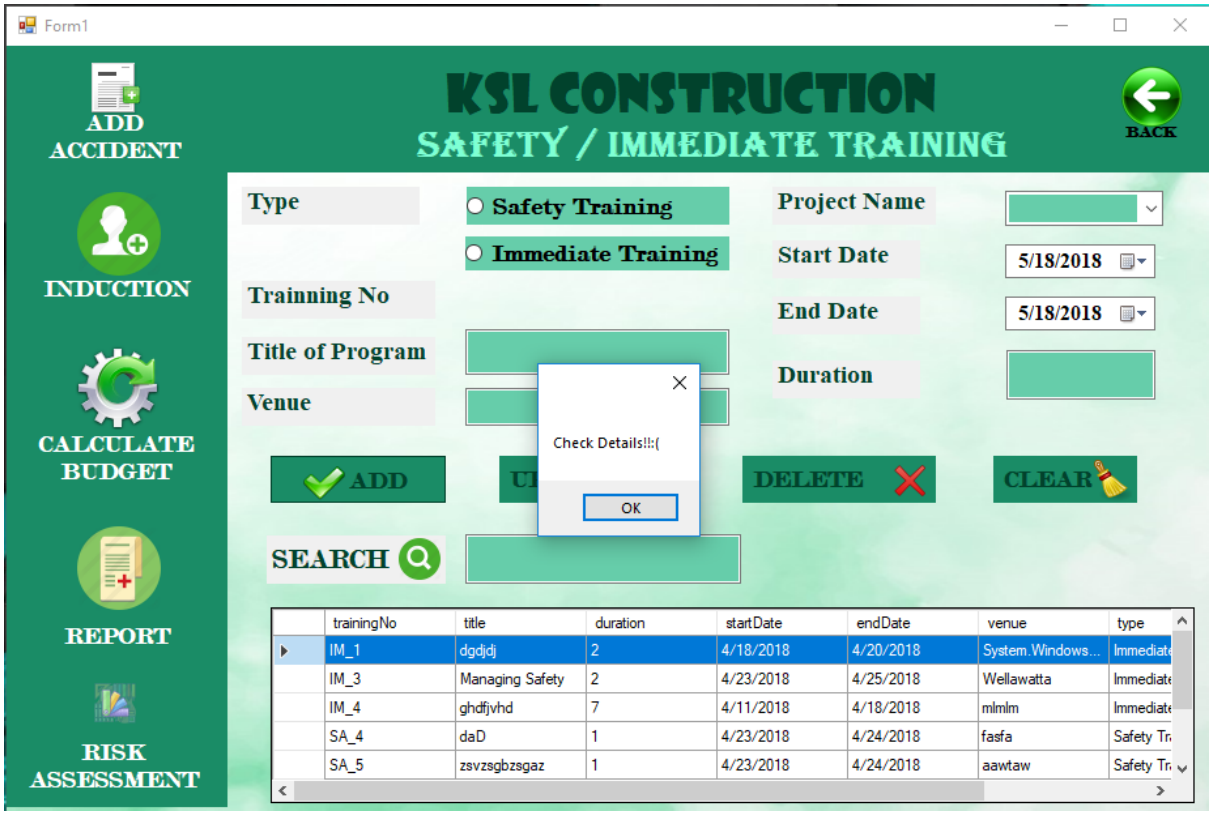
Appendix B: Test Results



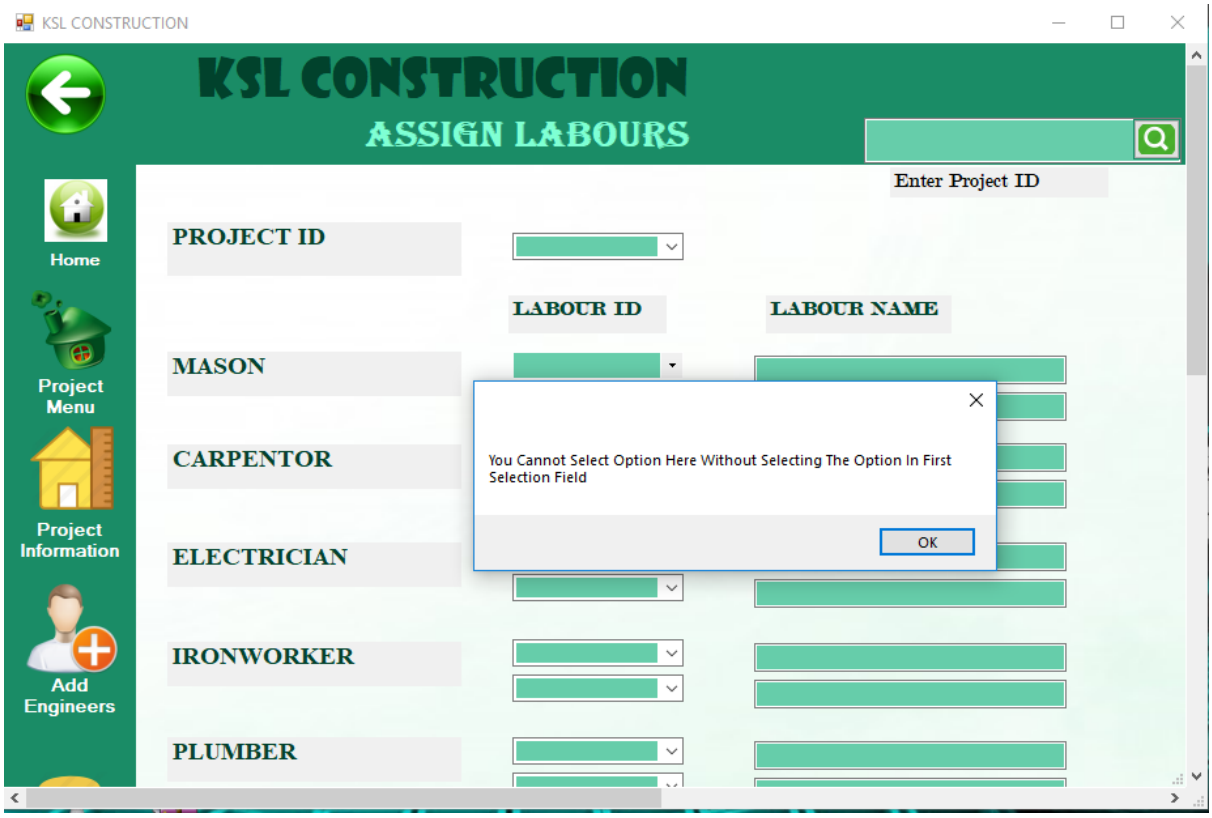
Supplier and Order Management System



Stock Management System



Safety Management System



Project Management System

Appendix C: Selected Code Listings

Auto generate ID

```
public void AutoID_Generate_Induct()
        {
            con = DBconnection.GetConnection();
            con.Open();
            string query = "SELECT COUNT(InductionID) FROM Induction";
            SqlCommand com = new SqlCommand(query, con);
            int i = Convert.ToInt32(com.ExecuteScalar());
            con.Close();
            i++;
            la_InductID.Text = Induct_Prefix + i.ToString();
        }
```

Load Combo box

```
public void load_cmb_EmpID()
        {
            try
            {
                cmbEmpID.Items.Clear();
                con = DBconnection.GetConnection();
                con.Open();
                SqlCommand command = con.CreateCommand();
                command.CommandType = CommandType.Text;
                command.CommandText = "SELECT DISTINCT EmployeeId FROM Employee WHERE
status=1 AND Safety = 'NOT DONE'";
                command.ExecuteNonQuery();
                DataTable dt = new DataTable();
                SqlDataAdapter da = new SqlDataAdapter(command);
                da.Fill(dt);
```

```
                foreach (DataRow dr in dt.Rows)
                {
                    cmbEmpID.Items.Add(dr["EmployeeId"].ToString());
                }
                con.Close();

            }
            catch (SqlException x)
            {
                MessageBox.Show(x.Message);
            }
        }
```

Auto fill text box

```
private void emp_SelectedIndexChanged(object sender, EventArgs e)
        {
            try
            {
                con = DBconnection.GetConnection();
                con.Open();
                string query = "SELECT EmployeeName , Position  FROM Employee WHERE
EmployeeId ='" + cmbEmpID.Text + "'";
                SqlCommand com = new SqlCommand(query, con);
                com.ExecuteNonQuery();
                SqlDataReader dr;
                dr = com.ExecuteReader();
                while (dr.Read())
                {
                    string eName = dr["EmployeeName"].ToString();
                    txtEmpName.Text = eName;

                    string eGrade = dr["Position"].ToString();
                    txtEmpGrade.Text = eGrade;
                }
                dr.Close();
                con.Close();
            }
            catch (SqlException x)
            {
                MessageBox.Show(x.Message);
            }
        }
```

Validation

- Allow only characters enter the text box

```
private void txtHazards_KeyPress(object sender, KeyPressEventArgs e)
        {
            e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar ==
(char)Keys.Space || e.KeyChar == (char)Keys.Back || e.KeyChar ==
(char)Keys.Delete);
        }
```

- Date Validation

```
private void dtp_projectStartDate_ValueChanged(object sender, EventArgs e)
        {
            DateTime val = dtp_projectStartDate.Value;
            DateTime now = DateTime.Today.Date;
            if (val > now)
            {
                dtp_projectStartDate.Value = DateTime.Today.Date;
                MessageBox.Show("Not allowed to select!!:|");
            }
        }
```

- Empty text field validation

```
private void txtTrainingContent_Leave(object sender, EventArgs e)
        {
            if (string.IsNullOrEmpty(txtTrainingContent.Text))
            {
                epTrainingContent.Icon = Properties.Resources.err;
                epTrainingContent.SetError(txtTrainingContent, "Training content not
empty");
            }
            else
            {
                epTrainingContent.Icon = Properties.Resources.ok1;
                epTrainingContent.SetError(txtTrainingContent, "OK");
            }
        }
```

WITHOUT YOUR INVOLVEMENT YOU CANNOT SUCCEED

WITH YOUR INVOLVEMENT YOU CANNOT FAIL

-Dr.A.P.J.Abdulkalam-

THANK YOU