

**Final Year Project Report**

**Project Name**



**Project Advisor: Dr. Asif Subhani**

**Submitted By: Abdul Raheem**

**Session**

**Fall 2022**

**University of Management and Technology**

**C-II Johar Town Lahore Pakistan**

# CONTENTS

---

CONTENTS.....	1
DEFINITIONS AND ACRONYMS.....	7
LIST OF FIGURES.....	8
LIST OF TABLES.....	11
<b>1. INTRODUCTION.....</b>	<b>13</b>
1.1 MOTIVATIONS.....	13
1.2 PROJECT OVERVIEW.....	13
1.3 PROBLEM STATEMENT.....	13
1.4 OBJECTIVES.....	14
<b>2. DOMAIN ANALYSIS.....</b>	<b>15</b>
2.1 CUSTOMER.....	15
2.2 STAKEHOLDERS.....	15
2.3 AFFECTED GROUPS WITH SOCIAL OR ECONOMIC IMPACT.....	15
2.4 DEPENDENCIES/ EXTERNAL SYSTEMS.....	15
2.5 REFERENCE DOCUMENTS.....	16
2.5.1 <i>Related Projects</i> .....	16
2.5.2 <i>Feature Comparison</i> .....	16
<b>3. REQUIREMENTS ANALYSIS.....</b>	<b>18</b>
3.1 REQUIREMENTS.....	18
3.2 LIST OF ACTORS.....	20
3.3 LIST OF USE CASES.....	20
3.4 SYSTEM USE CASE DIAGRAM.....	22
3.5 EXTENDED USE CASES.....	23
3.6 USER INTERFACES (MOCK SCREENS).....	62
<b>4. DATA FLOW DIAGRAM (OPTIONAL).....</b>	<b>69</b>
4.1 DATA FLOW DIAGRAM LEVEL 0.....	69
4.2 DATA FLOW DIAGRAM LEVEL 1.....	70
<b>5. SYSTEM DESIGN.....</b>	<b>71</b>
5.1 SYSTEM ARCHITECTURE DIAGRAM.....	71
5.2 CLASS DIAGRAM.....	72
5.3 SEQUENCE DIAGRAMS.....	73
5.4 COLLABORATION DIAGRAMS.....	82
5.5 ERD.....	91
5.6 DATA DICTIONARY.....	92
<b>6. IMPLEMENTATION DETAILS.....</b>	<b>95</b>
6.1 DEVELOPMENT SETUP.....	95
TOOL/TECHNOLOGY.....	95
DESCRIPTION.....	95

VS CODE.....	95
USED VS CODE IDE FOR CODING.....	95
POSTMAN.....	95
POSTMAN IS USE TO TEST API.....	95
GOOGLE CHROME.....	95
TO RUN WEB APPLICATION.....	95
MOZILLA FIREFOX.....	95
TO RUN WEB APPLICATION.....	95
MONGODB.....	95
NOSQL DATABASE.....	95
REACT JS.....	95
JS LIBRARY FOR FRONT END DEVELOPMENT.....	95
EXPRESS JS.....	95
JS FRAMEFORWORK FOR RESTFUL API DEVELOPMENT.....	95
NODE JS.....	95
JS ENVIROMENT.....	95
REDUX TOOLKIT.....	95
JS LIBRARY FOR STATE MANAGEMENT.....	95
JSON WEB TOKEN.....	95
FOR CREATING X-ACCESS TOKEN.....	95
MONGOOSE.....	95
JS LIBRARY FOR MONGODB.....	95
FONT AWESOME.....	95
CDN FOR FONTS AND ICONS.....	95
REACT-BOOTSTRAP.....	95
JS BOOTSTRAP LIBRARY FOR FRONTEND.....	95
6.2 DEPLOYMENT SETUP.....	95
6.3 ALGORITHMS.....	95
6.4 CONSTRAINTS.....	115
6.4.1 Assumptions.....	115
6.4.2 System constraints.....	115
6.4.3 Restrictions.....	115
6.4.4 Limitations.....	115
<b>7. TESTING.....</b>	<b>116</b>
7.1 EXTENDED TEST CASES.....	116
7.2 DECISION TABLE.....	146
7.2.1 Code snippet.....	146
7.2.2 Decision coverage table.....	146
7.3 TRACEABILITY MATRIX.....	146
7.3.1 RID vs UCID (requirements vs use cases).....	146
UCID/RID.....	146
R01.....	146
R02.....	146
R03.....	146

R04.....	146
R05.....	146
R06.....	146
R07.....	146
R08.....	146
R09.....	146
R10.....	146
R11.....	146
R12.....	146
R13.....	146
R14.....	146
R15.....	146
R16.....	146
R17.....	146
R18.....	146
R19.....	146
R20.....	146
R21.....	146
R22.....	146
R23.....	146
R24.....	146
R25.....	146
R26.....	146
UC-1.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
UC-2.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
UC-3.....	147
✓.....	147
✓.....	147

✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
UC-4.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
UC-5.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
UC-6.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
UC-7.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
UC-8.....	147
✓.....	147
✓.....	147
✓.....	147

✓.....	147
✓.....	147
✓.....	147
✓.....	147
UC-9.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
UC-10.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
UC-11.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
✓.....	147
UC-12.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
UC-13.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148

✓.....	148
✓.....	148
✓.....	148
UC-14.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
UC-15.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
UC-16.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
UC-17.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
UC-18.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148

✓.....	148
✓.....	148
UC-19.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148
✓.....	148



## DEFINITIONS AND ACRONYMS

---

Acronym	Definition
UMT	University of Management and Technology
POS	Point of Sale

**Table 1: table of acronyms and definitions**

## LIST OF FIGURES

---

Figure 1(3.4): Use Case Diagram	20
Figure 2(3.6): Manage User	60
Figure 3(3.6): Manage Account	60
Figure 4(3.6): Account Details	61
Figure 5(3.6): Manage Capital	61
Figure 6(3.6): Generate Report	62
Figure 7(3.6): Manage Payments	62
Figure 8(3.6): Manage Employee Request	63
Figure 9(3.6): Employee Request	63
Figure 10(3.6): Manage Purchase	64
Figure 11(3.6): Manage Department	64
Figure 12(3.6): Manage Attendance	65
Figure 13(3.6): Manage Backup	65
Figure 14(3.6): Manage Employee	66
Figure 15(3.6): Employee Details	66
Figure 16(3.6): Manage Production	67
Figure 17(3.6): View Production	67
Figure 18(3.6): Dashboard	68
Figure 19(4.1): DFD Level 0 Diagram	69
Figure 20(4.2): DFD Level 1 Diagram	70
Figure 21(5.1): System Architecture	71
Figure 22(5.2): Class Diagram	72
Figure 23(5.3): Authenticate User	73
Figure 24(5.3): Manage User	73
Figure 25(5.3): Manage Accounts	74
Figure 26(5.3): Manage Capital	74
Figure 27(5.3): Generate Report	75
Figure 28(5.3): Manage Expense	75
Figure 29(5.3): Manage Payments	76
Figure 30(5.3): Manage Employee Requests	76
Figure 31(5.3): Manage Purchases	77
Figure 32(5.3): Manage Production Requests	77
Figure 33(5.3): Manage Sales	78

Figure 34(5.3): Manage Products	78
Figure 35(5.3): Manage Raw Material	79
Figure 36(5.3): Manage Departments	79
Figure 37(5.3): Manage Backup	80
Figure 38(5.3): Manage Attendance	80
Figure 39(5.3): Manage Employees	81
Figure 40(5.3): Manage Salaries	81
Figure 41(5.3): Manage Production	82
Figure 42(5.4): Authenticate User	83
Figure 43(5.4): Manage User	83
Figure 44(5.4): Manage Accounts	84
Figure 45(5.4): Manage Capital	84
Figure 46(5.4): Generate Report	85
Figure 47(5.4): Manage Expense	85
Figure 48(5.4): Manage Payments	86
Figure 49(5.4): Manage Employee Request	86
Figure 50(5.4): Manage Purchases	87
Figure 51(5.4): Manage Production Request	87
Figure 52(5.4): Manage Sales	88
Figure 53(5.4): Manage Products	88
Figure 54(5.4): Manage Raw Material	89
Figure 55(5.4): Manage Department	89
Figure 56(5.4): Manage Backup	90
Figure 57(5.4): Manage Attendance	90
Figure 58(5.4): Manage Employee	91
Figure 59(5.4): Manage Salaries	91
Figure 60(5.4): Manage Production	92
Figure 61(5.5): ERD	93



## LIST OF TABLES

---

Table 1: table of acronyms and definitions	7
Table 2(2.5): Feature Comparison	14
Table 3(3.1): Requirement Categories	15
Table 4(3.1): List of Requirements	17
Table 5(3.5): UC-1 Authenticate	20
Table 6(3.5): UC-2 Manage Users	21
Table 7(3.5): UC-3 Manage Accounts	23
Table 8(3.5): UC-4 Manage Capital	27
Table 9(3.5): UC-5 Generate Reports	29
Table 10(3.5): UC-6 Manage Expense	30
Table 11(3.5): UC-7 Manage Payments	32
Table 12(3.5): UC-8 Manage Employee Requests	34
Table 13(3.5): UC-9 Manage Purchases	35
Table 14(3.5): UC-10 Manage Production Requests	39
Table 15(3.5): UC-11 Manage Sales	40
Table 16(3.5): UC-12 Manage Products	44
Table 17(3.5): UC-13 Manage Raw Material	45
Table 18(3.5): UC-14 Manage Departments	46
Table 19(3.5): UC-15 Manage Backup	46
Table 20(3.5): UC-16 Manage Attendance	47
Table 21(3.5): UC-17 Manage Employees	49
Table 22(3.5): UC-18 Manage Salaries	53
Table 23(3.5): UC-19 Manage Production	55
Table 24(5.6): Data Dictionary	93
Table 25(6.1): Development Setup	95
Table 26(7.1): Test Case 1 Login	116
Table 27(7.1): Test Case 2 Forget Password	117
Table 28(7.1): Test Case Manage Account	119
Table 29(7.1): Test Case 4 Manage Capital	123
Table 30(7.1): Test Case 6 Manage Expense	124
Table 31(7.1): Test Case 7 Manage Payments (Payments In)	125
Table 32(): Test Case 8 Manage Payments (Payments Out)	126
Table 33(7.1): Test Case 9 Manage Purchases	127

Table 34(7.1): Test Case 10 Add sale	130
Table 35(7.1): Test Case 11 Return Sale	133
Table 36(7.1): Test Case 12 Manage Products	134
Table 37(7.1): Test Case 13 Manage Raw Material	135
Table 38(7.1): Test Case 14 Manage Departments	136
Table 39(7.1): Test Case 15 Search Attendance	137
Table 40(7.1): Test Case 16 Marked Attendance	138
Table 41(7.1): Test Case 17 Add/Update Employees	139
Table 42(7.1): Test Case 18 Add/Update Production	143
Table 43(7.2): Decision Coverage Table	146
Table 44(7.3): RID vs UCID	146
Table 45(7.3): RID vs PID	148
Table 46(7.3): RID vs TID	148
Table 47(7.3): UCID vs TID	149

# 1. INTRODUCTION

---

## 1.1 Motivations

In this era of technology, almost everything is going online no matter if it is a student attendance system or an ultimate business solution. Everyone wants access to their data every minute. They want to stay updated about everything. So, what if we have an Integrated Business Solution that is accessible anytime anywhere? What if user is able to manage their business on the go?

So, if we have access to our business data anytime anywhere, we may work more efficiently and stay updated to current business status. We won't have to wait for our finance department to confirm if we can afford the product or not. We can automate most of our business load and keep track of every bit. We may reduce extra expenses and improve our business model by reading the stats of current business.

## 1.2 Project Overview

Currently, the main problem with traditional business models is no direct access to data. Some other problems are:

- Inefficiency
- Data Reporting
- Maintaining Records Manually

The goal of our project is to develop an efficient system that would be able to manage the business in more reliable and smart way. The users won't need to depend on other departments to stay updated about any transaction. We are actually developing:

1. Customer Relation Management
2. Book Keeping
3. Human Resource Management
4. Inventory Management
5. Point of Sale
6. Supply Chain Management

And then integrate these systems to form a single **Integrated Business Model**.

## 1.3 Problem Statement

In our traditional business scenario, we don't have a direct access to every data that we need to know. We need different departments for managing different accounts. Risk for mistakes is very high because the workflow is totally manual and time consuming.

- We don't know how much stock is left in our inventory without confirming from our Inventory Manager.
- We don't know what our financial condition is without our finance department.
- If there is even a single wrong entry in any transaction, the balance sheet would not be balanced anymore and you would have to find the mistake manually that could take hours, days or maybe months. Human may make mistake but Computer won't.
- You have to write every transaction manually, whether it is a product purchase invoice, or maintaining a customer's account.

## 1.4 Objectives

The main objective of this project is developing a system that would:

- Enhance the operational efficiency of business resources.
- Assist company owners to achieve their goals at a faster rate.
- Assist in knowing real time information about business field strategies.
- Enhance customer relation management.
- Increase quality of services.
- Shorten delivery time.
- Control Errors in transactions.
- Helps in better planning and coordination of business resources so as to achieve maximum profit.
- Improve accuracy rate of results.



## 2. DOMAIN ANALYSIS

---

### 2.1 Customer

RANKOLI is a Vanity manufacturing company in Johar Town, Lahore. They manufacture their products and sell them to dealers on credit. They have a production department, finance department, accountants and sales department. They need a software that can automate most of their transaction entries and that can generate some of the accounting reports like Balance Sheet, Profit Loss Statement etc.

### 2.2 Stakeholders

Stakeholder	Role in System
HR	Use human resource management tool for accessing data for employs and salaries. HR may insert new employs. HR can also update expenses.
Sales	This department is concerned with making sales and forecasting sales. They may use CRM data to approach the right customer with right product. Update or Insert entries in book keeping.
Account Management	Recording information directly related to specific customers. E.g. Credit Amount, Debit Amount, Contact Info
Finance	This department often uses CRM for invoicing purpose. They may check data from book keeping management for any debts or debit amount. Pay for the purchases. Keep record of capital/equity.
Business Owner/President	He can view stats and current status of company
Production	They may use the inventory management system to check for available raw material and stocks. Keep record or work in progress.
Delivery Team	They may use the system to retrieve customer information like address and contact information
Customer	They may use the system to fetch their records. E.g. Remaining Balance.

Table 1(2.2): List of stakeholders

### 2.3 Affected Groups with social or economic impact

- Business Owner/President
- Production Department
- Labor
- Sales Team
- Customers
- Finance Department
- Human Resource Department

### 2.4 Dependencies/ External Systems

- Internet
- Computer or Mobile Phone

- Web hosting or Dedicated Server
- Online Database
- Printe

## 2.5 Reference Documents

### 2.5.1 Related Projects

In order to develop RANKOLI IBM, we looked up several similar systems. Their details are given below:

1. Odoo  
Odoo is a suit of open source business apps that includes CRM, POS, Accounting, Supply Chain, Inventory System etc.
2. ERPNext  
ERPNext is free and open source integrated ERP software developed by Frappe Technologies PVT.

### 2.5.2 Feature Comparison

SR No.	Feature	Odoo	ERPNext	Remarks
1	Database	Postgresql	MariaDb	Postgresql offers more complex data-types and allows objects to inherit properties.
2	Invoicing	Yes	Yes	Available for both
3	Online Payment	Yes	No	Odoo Support VISA, American Express, UnionPay, Discover and many more online payment methods.
4	Inventory Module	Available	Available	Odoo have additional feature of QR Code Support
5	QR Code Support	Yes	No	QR Code Support is beneficial in many inventory management

**Table 2(2.5): Feature Comparison**



### 3. REQUIREMENTS ANALYSIS

---

#### 3.1 Requirements

The Requirements are based on the performance and functionalities the system must uphold in order to achieve objectives.

Table 4 contains the list of categories to which requirements belong. The requirements type given in table categorizes the requirements. So that it is easier for the student to identify the requirements by unique id given to each requirement.

RID	Requirement Type	Description
FR_1.x	Functional Requirement	Core requirements of business operations.
NFR_2.x	Non-Functional Requirement	Supporting requirements for functional requirements.
PR_3.x	Programing Requirement	Tools used to develop the web application.
DR_4.x	Data Requirement	Data storage and resources.

**Table 3(3.1): Requirement Categories**

RID	Description	Category	Attribute	Details & Boundary Constraints
FR_1.1	Authenticate the user to the system	Functional Requirement	Authenticate	
FR_1.2	Activate/Deactivate user and add role based access	Functional Requirement	User Management	
FR_1.3	Add, remove, update account and view account ledgers	Functional Requirement	Account Management	
FR_1.4	Add, remove and update capital	Functional Requirement	Capital Management	
FR_1.5	Generate various types of reports	Functional Requirement	Generate Report	
FR_1.6	Add, remove and update expense	Functional Requirement	Expense Management	

FR_1.7	Payment in/out	Functional Requirement	Payments Management	
FR_1.8	Accept/Decline employee creation request	Functional Requirement	Employee Request Management	
FR_1.9	Add, remove and update purchase/purchase returns	Functional Requirement	Purchases Management	
FR_1.10	Accept/Decline production add/update request	Functional Requirement	Production Request Management	
FR_1.11	Add, remove and update sale/sale return requests.	Functional Requirement	Sales Management	
FR_1.12	Add, remove and update products	Functional Requirement	Products Management	
FR_1.13	Add, remove and update raw material	Functional Requirement	Raw Material Management	
FR_1.14	Add, remove and update departments	Functional Requirement	Departments Management	
FR_1.15	Create/Restore Backup	Functional Requirement	Backup Management	
FR_1.16	Search attendance and keep record of check-in/check-out time	Functional Requirement	Attendance Management	
FR_1.17	Add, remove and update employee	Functional Requirement	Employee Management	
FR_1.18	Generate salary or pay salary to all/one	Functional Requirement	Salary Management	
FR_1.19	Start, update and finish production	Functional Requirement	Production Management	

NFR_2.1	Adjust component size according to screen size	Non Functional Requirement	Responsive Design	
NFR_2.2	Access system from anywhere over the internet	Non Functional Requirement	Online Access	
NFR_2.3	Passwords are hashed	Non Function Requirement	Credential Security	
NFR_2.4	Application don't need to be loaded multiple time on every submission	Non Function Requirement	Single Page Application	
NFR_2.5	User should be able to use application easily	Non Function Requirement	Easy Interface	
PR_3.1	MERN Stack	Programing Requirement	Technologies	
DR_4.1	Online MongoDB NoSQL database	Data Requirement	Storage Resource	

**Table 4(3.1): List of Requirements**

### 3.2 List of Actors

- **Administrator:** Manage system users and role based access.
- **Accounts Manager:** Manage accounts, payments and expense, capital and employee requests. Also responsible for generating financial reports.
- **HR Manager:** Manage attendance, employees and salaries.
- **Procurement Manager:** Manage purchases and production requests.
- **Production Manager:** Manage production of products.
- **Sales Manager:** Responsible for managing sales.
- **System Manager:** Manage products, raw material, departments and backup.

### 3.3 List of use cases

- **UC-1 Authenticate:** Authenticate the user to the system
- **UC-2 Manage Users:** Activate/Deactivate user and add role based access
- **UC-3 Manage Accounts:** Add, remove, update account and view account ledgers

- **UC-4 Manage Capital:** Add, remove and update capital
- **UC-5 Generate Reports:** Generate various types of reports
- **UC-6 Manage Expense:** Add, remove and update expense
- **UC-7 Manage Payments:** Payment in, Payment out
- **UC-8 Manage Employee Requests:** Accept/Decline employee creation request
- **UC-9 Manage Purchases:** Add, remove and update purchase/purchase returns
- **UC-10 Manage Production Requests:** Accept/Decline production add/update request
- **UC-11 Manage Sales:** Add, remove and update sale/sale return requests.
- **UC-12 Manage Products:** Add, remove and update products
- **UC-13 Manage Raw Material:** Add, remove and update raw material
- **UC-14 Manage Departments:** Add, remove and update departments
- **UC-15 Manage Backup:** Create/Restore Backup
- **UC-16 Manage Attendance:** Search attendance and keep record of check-in/check-out time
- **UC-17 Manage Employees:** Add, remove and update employee
- **UC-18 Manage Salaries:** Generate salary or pay salary to all/one
- **UC-19 Manage Production:** Start, update and finish production

### 3.4 System use case diagram

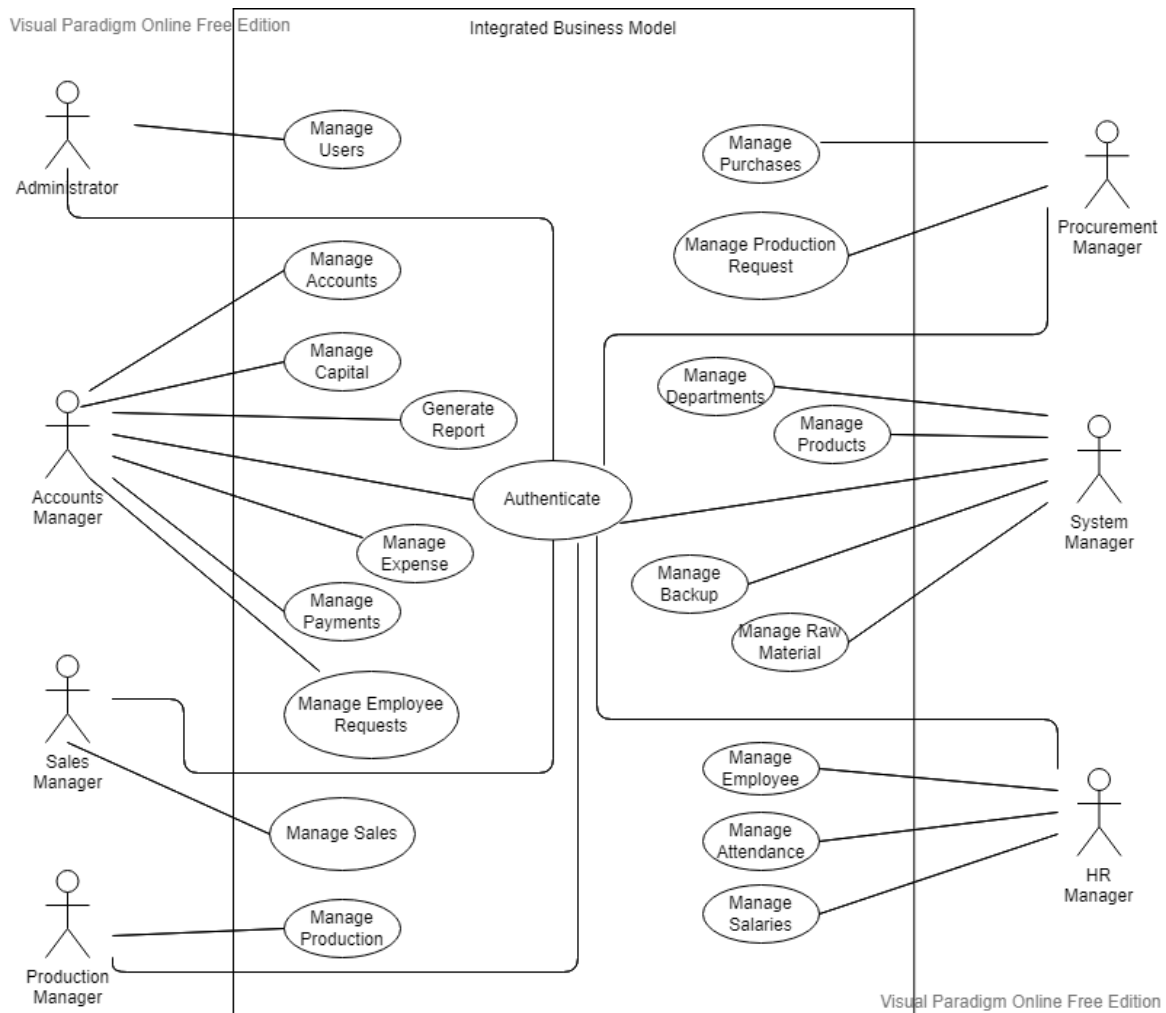


Figure 1(3.4): Use Case Diagram



### 3.5 Extended use cases

Table 5(3.5): UC-1 Authenticate

<b>Use Case ID:</b>	UC-1		
<b>Use Case Name:</b>	Authenticate		
<b>Created By:</b>	Abdul Raheem	<b>Last Updated By:</b>	Abdul Raheem
<b>Date Created:</b>	July 03, 2022	<b>Last Revision Date:</b>	Oct 10, 2022
<b>Actors:</b>	Accounts Manager, Administrator, HR Manager, Procurement Manager, Production Manager, Sales Manager, System Manager		
<b>Description:</b>	Actor would be able to login to the system or reset login password.		
<b>Trigger:</b>	Actor starts the application.		
<b>Preconditions:</b>	N.A		
<b>Post conditions:</b>	<b>Login</b> <ol style="list-style-type: none"> <li>1. System displays dashboard page.</li> <li>2. System displays message “Login Successful”.</li> </ol> <b>Forget Password</b> <ol style="list-style-type: none"> <li>1. System send password reset link to actor’s email address.</li> <li>2. System displays message “Password reset link sent to your email address”.</li> </ol>		
<b>Normal Flow:</b>	<b>Login</b> <ol style="list-style-type: none"> <li>1. Actor enters employee id.</li> <li>2. Actor enters password.</li> <li>3. Actor sends request to sign into the system.</li> <li>4. System displays dashboard page.</li> <li>5. System displays message “Login Successful”.</li> </ol> <b>Forget Password</b> <ol style="list-style-type: none"> <li>1. Actor selects forget password option.</li> <li>2. System displays reset account password page.</li> <li>3. Actor enters employee id.</li> <li>4. Actor sends request to reset password.</li> <li>5. System sends password link to actor’s email address.</li> <li>6. System displays message “Password reset link sent to your email address”.</li> </ol>		
<b>Alternative Flows:</b>	N.A		
<b>Exceptions:</b>	<b>Login</b> <ol style="list-style-type: none"> <li>1a. Employee id field is empty, <ol style="list-style-type: none"> <li>1. System displays message “Employee id is required”.</li> <li>2. Actor enters employee id.</li> <li>3. Use case resumes on step 2.</li> </ol> </li> <li>1b. Employee id field contains alphabet, <ol style="list-style-type: none"> <li>1. System displays message “Invalid employee id”.</li> <li>2. Actor enters valid employee id.</li> <li>3. Use case resumes on step 2.</li> </ol> </li> <li>1c. Employee id length is not equal to 13, <ol style="list-style-type: none"> <li>1. System displays message “Invalid employee id”.</li> <li>2. Actor enters valid employee id.</li> </ol> </li> </ol>		

	<p>3. Use case resumes on step 2.</p> <p>2a. Password field is empty,</p> <ol style="list-style-type: none"> <li>1. System displays message “Password field is required”.</li> <li>2. Actor enters password.</li> <li>3. Use case resumes on step 3.</li> </ol> <p>2b. Password length is smaller than 8 characters,</p> <ol style="list-style-type: none"> <li>1. System displays message “Password length can’t be less than 8 characters”.</li> <li>2. Actor enters valid password.</li> <li>3. Use case resumes on step 3.</li> </ol> <p><b>Forget Password</b></p> <p>3a. Employee id field is empty,</p> <ol style="list-style-type: none"> <li>1. System displays message “Employee id is required”.</li> <li>2. Actor enters employee id.</li> <li>3. Use case resumes on step 4.</li> </ol> <p>3b. Employee id field contains alphabet,</p> <ol style="list-style-type: none"> <li>1. System displays message “Invalid employee id”.</li> <li>2. Actor enters valid employee id.</li> <li>3. Use case resumes on step 4.</li> </ol> <p>3c. Employee id length is not equal to 13,</p> <ol style="list-style-type: none"> <li>1. System displays message “Invalid employee id”.</li> <li>2. Actor enters valid employee id.</li> <li>3. Use case resumes on step 4.</li> </ol>
<b>Includes:</b>	N.A
<b>Frequency of Use:</b>	Could be nearly continuous.
<b>Special Requirements:</b>	N.A
<b>Assumptions:</b>	N.A
<b>Notes and Issues:</b>	N.A

**Table 6(3.5): UC-2 Manage Users**

<b>Use Case ID:</b>	UC-2		
<b>Use Case Name:</b>	Manage Users		
<b>Created By:</b>	Abdul Raheem	<b>Last Updated By:</b>	Abdul Raheem
<b>Date Created:</b>	July 03, 2022	<b>Last Revision Date:</b>	Oct 10, 2022
<b>Actors:</b>	Administrator		
<b>Description:</b>	Actor would be able to activate/deactivate user account and grant role based access.		
<b>Trigger:</b>	Actor navigates to user management		
<b>Preconditions:</b>	N.A		

<b>Post conditions:</b>	<p><b>Activate User</b></p> <ol style="list-style-type: none"> <li>1. System updates employee account's active status.</li> <li>2. System sends login credentials to employee's email address.</li> <li>3. System displays message "User activated".</li> </ol> <p><b>Deactivate User</b></p> <ol style="list-style-type: none"> <li>1. System updates employee account's active status.</li> <li>2. System displays message "User deactivated".</li> </ol> <p><b>Grant Access</b></p> <ol style="list-style-type: none"> <li>1. System updates employee's access rights.</li> <li>2. System display message "Access Granted".</li> </ol>
<b>Normal Flow:</b>	<p><b>Activate User</b></p> <ol style="list-style-type: none"> <li>1. Actor navigates to user management.</li> <li>2. System displays list of all employees.</li> <li>3. Actor sends request to activate employee's account.</li> <li>4. System updates employee account's active status.</li> <li>5. System sends login credentials to employee's email address.</li> <li>6. System displays message "User activated".</li> </ol> <p><b>Deactivate User</b></p> <ol style="list-style-type: none"> <li>1. Actor navigates to user management.</li> <li>2. System displays list of all employees.</li> <li>3. Actor sends request to deactivate employee's account.</li> <li>4. System updates employee account's active status.</li> <li>5. System displays message "User deactivated".</li> </ol> <p><b>Grant Access</b></p> <ol style="list-style-type: none"> <li>1. Actor navigates to user management.</li> <li>2. System displays list of all employees.</li> <li>3. Actor selects manage access for an employee.</li> <li>4. System displays list of modules.</li> <li>5. Actor selects required access roles.</li> <li>6. Actor sends request to grant request.</li> <li>7. System updates employee's access rights.</li> <li>8. System display message "Access Granted".</li> </ol>
<b>Alternative Flows:</b>	N.A
<b>Exceptions:</b>	<p><b>Grant Access</b></p> <p>5. Actor didn't selected any access role,</p> <ol style="list-style-type: none"> <li>1. System displays message "At least 1 access role is required".</li> <li>2. Actor selects required access roles.</li> <li>3. Use case resumes on step 6.</li> </ol>
<b>Includes:</b>	N.A
<b>Frequency of Use:</b>	Could be nearly continuous.
<b>Special Requirements:</b>	N.A
<b>Assumptions:</b>	N.A
<b>Notes and Issues:</b>	N.A

**Table 7(3.5): UC-3 Manage Accounts**

<b>Use Case ID:</b>	UC-3		
<b>Use Case Name:</b>	Manage Account		
<b>Created By:</b>	Abdul Raheem	<b>Last Updated By:</b>	Abdul Raheem
<b>Date Created:</b>	July 03, 2022	<b>Last Revision Date:</b>	Oct 10, 2022
<b>Actors:</b>	Accounts Manager		
<b>Description:</b>	Actor would be able to add, remove and update accounts and view account ledgers.		
<b>Trigger:</b>	Actor navigates to account management		
<b>Preconditions:</b>	N.A		
<b>Post conditions:</b>	<b>Add Account</b> <ol style="list-style-type: none"> <li>1. System adds new account record.</li> <li>2. System displays message “Account Added”.</li> </ol> <b>Update Account</b> <ol style="list-style-type: none"> <li>1. System adds new account record.</li> <li>2. System displays message “Account Updated”.</li> </ol> <b>Remove Account</b> <ol style="list-style-type: none"> <li>1. System deletes account from records.</li> <li>2. System display message “Account Deleted”.</li> </ol> <b>View Account Ledger</b> <ol style="list-style-type: none"> <li>1. System display account ledgers.</li> </ol>		
<b>Normal Flow:</b>	<b>Add Account</b> <ol style="list-style-type: none"> <li>1. Actor navigates to account management.</li> <li>2. System displays list of all accounts</li> <li>3. Actor selects add account option.</li> <li>4. System displays add account form.</li> <li>5. Actor enters first name.</li> <li>6. Actor enters last name.</li> <li>7. Actor enters cnic number.</li> <li>8. Actor enters mobile number.</li> <li>9. Actor enters address.</li> <li>10. Actor enters email address.</li> <li>11. Actor enters emergency contact number.</li> <li>12. Actor selects account type.</li> <li>13. Actor enters opening balance.</li> <li>14. Actor selects opening balance type.</li> <li>15. Actor sends request to add account.</li> <li>16. System adds new account record.</li> <li>17. System displays message “Account Added”.</li> </ol> <b>Update Account</b> <ol style="list-style-type: none"> <li>1. Actor navigates to account management.</li> <li>2. System displays list of all accounts</li> <li>3. Actor selects edit account option from account list.</li> <li>4. System displays edit account form.</li> <li>5. Actor enters first name.</li> <li>6. Actor enters last name.</li> <li>7. Actor enters cnic number.</li> <li>8. Actor enters mobile number.</li> <li>9. Actor enters address.</li> <li>10. Actor enters email address.</li> <li>11. Actor enters emergency contact number.</li> </ol>		

	<ol style="list-style-type: none"> <li>12. Actor selects account type.</li> <li>13. Actor enters opening balance.</li> <li>14. Actor selects opening balance type.</li> <li>15. Actor sends request to update account.</li> <li>16. System updates account record.</li> <li>17. System displays message “Account Updated”.</li> </ol> <p><b>Remove Account</b></p> <ol style="list-style-type: none"> <li>1. Actor navigates to account management.</li> <li>2. System displays list of all accounts</li> <li>3. Actor sends request to delete account from account list.</li> <li>4. System deletes account from records.</li> <li>5. System display message “Account Deleted”.</li> </ol> <p><b>View Account Ledger</b></p> <ol style="list-style-type: none"> <li>1. Actor navigates to account management.</li> <li>2. System displays list of all accounts</li> <li>3. Actor selects account from the list.</li> <li>4. System display account information.</li> <li>5. Actor selects view ledger option.</li> <li>6. System display account ledgers.</li> </ol>
<b>Alternative Flows:</b>	N.A
<b>Exceptions:</b>	<p><b>Add Account</b></p> <p>5a. First name field is empty,</p> <ol style="list-style-type: none"> <li>1. System displays message “First name is required”.</li> <li>2. Actor enters first name.</li> <li>3. Use case resumes on step 6.</li> </ol> <p>5b. First name field contains number or special characters,</p> <ol style="list-style-type: none"> <li>1. System displays message “Invalid first name”.</li> <li>2. Actor enters first name.</li> <li>3. Use case resumes on step 6.</li> </ol> <p>6a. Last name field is empty,</p> <ol style="list-style-type: none"> <li>1. System displays message “Last name is required”.</li> <li>2. Actor enters last name.</li> <li>3. Use case resumes on step 7.</li> </ol> <p>6b. Last name field contains number or special characters,</p> <ol style="list-style-type: none"> <li>1. System displays message “Last name is required”.</li> <li>2. Actor enters last name.</li> <li>3. Use case resumes on step 7.</li> </ol> <p>7a. CNIC field is empty,</p> <ol style="list-style-type: none"> <li>1. System displays message “CNIC is required”.</li> <li>2. Actor enters CNIC.</li> <li>3. Use case resumes on step 8.</li> </ol> <p>7b. CNIC field contains alphabet or special characters,</p> <ol style="list-style-type: none"> <li>1. System displays message “CNIC is invalid”.</li> <li>2. Actor enters CNIC.</li> <li>3. Use case resumes on step 8.</li> </ol> <p>7c. CNIC length is not 13,</p> <ol style="list-style-type: none"> <li>1. System displays message “CNIC is invalid”.</li> <li>2. Actor enters CNIC.</li> <li>3. Use case resumes on step 8.</li> </ol> <p>8a. Mobile number field is empty,</p> <ol style="list-style-type: none"> <li>1. System displays message “Mobile number is required”.</li> <li>2. Actor enters mobile number.</li> </ol>

	<p>3. Use case resumes on step 9.</p> <p>8b. Mobile number field contains alphabet or special characters,</p> <ol style="list-style-type: none"> <li>1. System displays message “Mobile number is invalid”.</li> <li>2. Actor enters mobile number.</li> <li>3. Use case resumes on step 9.</li> </ol> <p>8c. Mobile number length is not 11,</p> <ol style="list-style-type: none"> <li>1. System displays message “Mobile number is invalid”.</li> <li>2. Actor enters mobile number.</li> <li>3. Use case resumes on step 9.</li> </ol> <p>9. Address field is empty,</p> <ol style="list-style-type: none"> <li>1. System displays message “Address is required”.</li> <li>2. Actor enters address.</li> <li>3. Use case resumes on step 10.</li> </ol> <p>10. Email address format is invalid,</p> <ol style="list-style-type: none"> <li>1. System displays message “Email address is invalid”.</li> <li>2. Actor enters email address.</li> <li>3. Use case resumes on step 11.</li> </ol> <p>11a. Emergency number field contains alphabet or special characters,</p> <ol style="list-style-type: none"> <li>1. System displays message “Emergency number is invalid”.</li> <li>2. Actor enters emergency number.</li> <li>3. Use case resumes on step 12.</li> </ol> <p>11b. Emergency number length is not 11,</p> <ol style="list-style-type: none"> <li>1. System displays message “Emergency number is invalid”.</li> <li>2. Actor enters emergency number.</li> <li>3. Use case resumes on step 12.</li> </ol> <p>12. Account type is not selected,</p> <ol style="list-style-type: none"> <li>1. System displays message “Account type is required”.</li> <li>2. Actor enters account type.</li> <li>3. Use case resumes on step 13.</li> </ol> <p>13a. Opening balance field is empty,</p> <ol style="list-style-type: none"> <li>1. System displays message “Opening balance is required”.</li> <li>2. Actor enters opening balance.</li> <li>3. Use case resumes on step 14.</li> </ol> <p>13b. Opening balance field contains alphabet or special characters,</p> <ol style="list-style-type: none"> <li>1. System displays message “Opening balance is invalid”.</li> <li>2. Actor enters opening balance.</li> <li>3. Use case resumes on step 14.</li> </ol> <p>14. Opening balance type is not selected,</p> <ol style="list-style-type: none"> <li>1. System displays message “Balance type is required”.</li> <li>2. Actor enters balance type.</li> <li>3. Use case resumes on step 15.</li> </ol> <p><b>Update Account</b></p> <p>5a. First name field is empty,</p> <ol style="list-style-type: none"> <li>1. System displays message “First name is required”.</li> <li>2. Actor enters first name.</li> <li>3. Use case resumes on step 6.</li> </ol> <p>5b. First name field contains number or special characters,</p> <ol style="list-style-type: none"> <li>1. System displays message “Invalid first name”.</li> <li>2. Actor enters first name.</li> <li>3. Use case resumes on step 6.</li> </ol> <p>6a. Last name field is empty,</p> <ol style="list-style-type: none"> <li>1. System displays message “Last name is required”.</li> </ol>
--	---

	<ul style="list-style-type: none"> <li>2. Actor enters last name.</li> <li>3. Use case resumes on step 7.</li> </ul>
	<ul style="list-style-type: none"> <li>6b. Last name field contains number or special characters, <ul style="list-style-type: none"> <li>1. System displays message “Last name is required”.</li> <li>2. Actor enters last name.</li> <li>3. Use case resumes on step 7.</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>7a. CNIC field is empty, <ul style="list-style-type: none"> <li>1. System displays message “CNIC is required”.</li> <li>2. Actor enters cnic.</li> <li>3. Use case resumes on step 8.</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>7b. CNIC field contains alphabet or special characters, <ul style="list-style-type: none"> <li>1. System displays message “CNIC is invalid”.</li> <li>2. Actor enters cnic.</li> <li>3. Use case resumes on step 8.</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>7c. CNIC length is not 13, <ul style="list-style-type: none"> <li>1. System displays message “CNIC is invalid”.</li> <li>2. Actor enters cnic.</li> <li>3. Use case resumes on step 8.</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>8a. Mobile number field is empty, <ul style="list-style-type: none"> <li>1. System displays message “Mobile number is required”.</li> <li>2. Actor enters mobile number.</li> <li>3. Use case resumes on step 9.</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>8b. Mobile number field contains alphabet or special characters, <ul style="list-style-type: none"> <li>1. System displays message “Mobile number is invalid”.</li> <li>2. Actor enters mobile number.</li> <li>3. Use case resumes on step 9.</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>8c. Mobile number length is not 11, <ul style="list-style-type: none"> <li>1. System displays message “Mobile number is invalid”.</li> <li>2. Actor enters mobile number.</li> <li>3. Use case resumes on step 9.</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>9. Address field is empty, <ul style="list-style-type: none"> <li>1. System displays message “Address is required”.</li> <li>2. Actor enters address.</li> <li>3. Use case resumes on step 10.</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>10. Email address format is invalid, <ul style="list-style-type: none"> <li>1. System displays message “Email address is invalid”.</li> <li>2. Actor enters email address.</li> <li>3. Use case resumes on step 11.</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>11a. Emergency number field contains alphabet or special characters, <ul style="list-style-type: none"> <li>1. System displays message “Emergency number is invalid”.</li> <li>2. Actor enters emergency number.</li> <li>3. Use case resumes on step 12.</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>11b. Emergency number length is not 11, <ul style="list-style-type: none"> <li>1. System displays message “Emergency number is invalid”.</li> <li>2. Actor enters emergency number.</li> <li>3. Use case resumes on step 12.</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>12. Account type is not selected, <ul style="list-style-type: none"> <li>1. System displays message “Account type is required”.</li> <li>2. Actor enters account type.</li> <li>3. Use case resumes on step 13.</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>13a. Opening balance field is empty, <ul style="list-style-type: none"> <li>1. System displays message “Opening balance is required”.</li> <li>2. Actor enters opening balance.</li> </ul> </li> </ul>

	3. Use case resumes on step 14. 13b. Opening balance field contains alphabet or special characters, 1. System displays message “Opening balance is invalid”. 2. Actor enters opening balance. 3. Use case resumes on step 14. 14. Opening balance type is not selected, 1. System displays message “Balance type is required”. 2. Actor enters balance type. 3. Use case resumes on step 15.  <b>Remove Account</b> 3. Selected account has entries in ledger, 1. System displays message “This account can’t be deleted”. 2. Use case ends here.
<b>Includes:</b>	N.A
<b>Frequency of Use:</b>	Could be nearly continuous.
<b>Special Requirements:</b>	N.A
<b>Assumptions:</b>	N.A
<b>Notes and Issues:</b>	N.A

**Table 8(3.5): UC-4 Manage Capital**

<b>Use Case ID:</b>	UC-4		
<b>Use Case Name:</b>	Manage Capital		
<b>Created By:</b>	Abdul Raheem	<b>Last Updated By:</b>	Abdul Raheem
<b>Date Created:</b>	July 03, 2022	<b>Last Revision Date:</b>	Oct 10, 2022
<b>Actors:</b>	Accounts Manager		
<b>Description:</b>	Actor would be add, remove and update capital.		
<b>Trigger:</b>	Actor navigates to capital management.		
<b>Preconditions:</b>	N.A		
<b>Post conditions:</b>	<b>Add Capital</b> <ol style="list-style-type: none"> <li>System adds a new capital record.</li> <li>System adds to entry to cash ledger debit.</li> <li>System adds to entry to account ledger credit.</li> <li>System displays message “Capital Added”.</li> </ol> <b>Update Capital</b> <ol style="list-style-type: none"> <li>System updates capital record.</li> <li>System updates entry to cash ledger debit.</li> <li>System updates entry to account ledger credit.</li> <li>System displays message “Capital Updated”.</li> </ol> <b>Remove Capital</b> <ol style="list-style-type: none"> <li>System deletes capital record.</li> <li>System displays message “Capital Deleted”.</li> </ol>		
<b>Normal Flow:</b>	<b>Add Capital</b> <ol style="list-style-type: none"> <li>Actor navigates to capital management.</li> <li>System displays list of investors.</li> <li>Actor selects add capital option.</li> <li>System displays add capital form.</li> <li>Actor selects investor.</li> <li>Actor enters capital amount.</li> </ol>		



	<ol style="list-style-type: none"> <li>7. Actor sends a request to add capital.</li> <li>8. System adds a new capital record.</li> <li>9. System adds to entry to cash ledger debit.</li> <li>10. System adds to entry to account ledger credit.</li> <li>11. System displays message “Capital Added”.</li> </ol> <p><b>Update Capital</b></p> <ol style="list-style-type: none"> <li>1. Actor navigates to capital management.</li> <li>2. System displays list of investors.</li> <li>3. Actor selects edit capital option from the list.</li> <li>4. System displays edit capital form.</li> <li>5. Actor selects investor.</li> <li>6. Actor enters capital amount.</li> <li>7. Actor sends a request to edit capital.</li> <li>8. System updates capital record.</li> <li>9. System updates entry to cash ledger debit.</li> <li>10. System updates entry to account ledger credit.</li> <li>11. System displays message “Capital Updated”.</li> </ol> <p><b>Remove Capital</b></p> <ol style="list-style-type: none"> <li>1. Actor navigates to capital management.</li> <li>2. System displays list of investors.</li> <li>3. Actor sends a request to delete capital from list.</li> <li>4. System deletes capital record.</li> <li>5. System displays message “Capital Deleted”.</li> </ol>
<b>Alternative Flows:</b>	N.A
<b>Exceptions:</b>	<p><b>Add Capital</b></p> <ol style="list-style-type: none"> <li>5. Account is not selected, <ol style="list-style-type: none"> <li>1. System displays message “Account is required”.</li> <li>2. Actor selects account.</li> <li>3. Use case resumes on step 6.</li> </ol> </li> <li>6a. Amount field is empty, <ol style="list-style-type: none"> <li>1. System displays message “Amount is required”.</li> <li>2. Actor enters amount.</li> <li>3. Use case resumes on step 7.</li> </ol> </li> <li>6b. Amount field contains special character or alphabet, <ol style="list-style-type: none"> <li>1. System displays message “Amount is invalid”.</li> <li>2. Actor enters amount.</li> <li>3. Use case resumes on step 7.</li> </ol> </li> </ol> <p><b>Update Capital</b></p> <ol style="list-style-type: none"> <li>5. Account is not selected, <ol style="list-style-type: none"> <li>1. System displays message “Account is required”.</li> <li>2. Actor selects account.</li> <li>3. Use case resumes on step 6.</li> </ol> </li> <li>6a. Amount field is empty, <ol style="list-style-type: none"> <li>1. System displays message “Amount is required”.</li> <li>2. Actor enters amount.</li> <li>3. Use case resumes on step 7.</li> </ol> </li> <li>6b. Amount field contains special character or alphabet, <ol style="list-style-type: none"> <li>1. System displays message “Amount is invalid”.</li> <li>2. Actor enters amount.</li> <li>3. Use case resumes on step 7.</li> </ol> </li> </ol>

<b>Includes:</b>	N.A
<b>Frequency of Use:</b>	Could be nearly continuous.
<b>Special Requirements:</b>	N.A
<b>Assumptions:</b>	N.A
<b>Notes and Issues:</b>	N.A

**Table 9(3.5): UC-5 Generate Reports**

<b>Use Case ID:</b>	UC-5		
<b>Use Case Name:</b>	Generate Reports		
<b>Created By:</b>	Abdul Raheem	<b>Last Updated By:</b>	Abdul Raheem
<b>Date Created:</b>	July 03, 2022	<b>Last Revision Date:</b>	Oct 10, 2022
<b>Actors:</b>	Accounts Manager		
<b>Description:</b>	Actor would be able to generate reports.		
<b>Trigger:</b>	Actor navigates to reports.		
<b>Preconditions:</b>	Actor is authenticated to application.		
<b>Post conditions:</b>	<ol style="list-style-type: none"> <li>1. System generates report.</li> <li>2. Report is downloaded to system storage.</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Actor navigates to reports.</li> <li>2. System displays generate report form.</li> <li>3. Actor selects report type.</li> <li>4. Actor selects start date.</li> <li>5. Actor selects end date.</li> <li>6. Actor sends request to generate report.</li> <li>7. System generates the report.</li> <li>8. Report is downloaded to system storage.</li> </ol>		
<b>Alternative Flows:</b>	N.A		
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>3. Report date is not selected, <ol style="list-style-type: none"> <li>1. System displays message "Report type is required".</li> <li>2. Actor selects report type.</li> <li>3. Use case resumes on step 4.</li> </ol> </li> <li>4. Start date is not selected, <ol style="list-style-type: none"> <li>1. System displays message "Start date is required".</li> <li>2. Actor selects start date.</li> <li>3. Use case resumes on step 5.</li> </ol> </li> <li>5. End date is not selected, <ol style="list-style-type: none"> <li>1. System displays message "End date is required".</li> <li>2. Actor selects end date.</li> <li>3. Use case resumes on step 6.</li> </ol> </li> </ol>		
<b>Includes:</b>	N.A		
<b>Frequency of Use:</b>	Could be nearly continuous.		
<b>Special Requirements:</b>	N.A		

<b>Assumptions:</b>	N.A
<b>Notes and Issues:</b>	N.A

**Table 10(3.5): UC-6 Manage Expense**

<b>Use Case ID:</b>	UC-6		
<b>Use Case Name:</b>	Manage Expense		
<b>Created By:</b>	Abdul Raheem	<b>Last Updated By:</b>	Abdul Raheem
<b>Date Created:</b>	July 03, 2022	<b>Last Revision Date:</b>	Oct 10, 2022
<b>Actors:</b>	Accounts Manager		
<b>Description:</b>	Actor would be able to add, view and delete expenses.		
<b>Trigger:</b>	Actor navigates to Expense Management.		
<b>Preconditions:</b>	Actor is authorized to the application.		
<b>Post conditions:</b>	<b>Add Expense</b> <ol style="list-style-type: none"> <li>1. System records new expense.</li> <li>2. System records new ledger entry for cash credit against amount of expense.</li> <li>3. System records new ledger entry for expense debit against amount of expense.</li> <li>4. System displays message “Expense Added”.</li> </ol> <b>Delete Expense</b> <ol style="list-style-type: none"> <li>1. System removes expense record.</li> <li>2. System removes cash ledger entry recorded against this expense.</li> <li>3. System removes expense ledger entry recorded against this expense.</li> <li>4. System displays message “Expense Deleted”.</li> </ol>		
<b>Normal Flow:</b>	<b>Add Expense</b> <ol style="list-style-type: none"> <li>1. Actor navigates to Expense Management.</li> <li>2. System display list of expenses.</li> <li>3. Actor selects add new expense.</li> <li>4. Actor enters expense title.</li> <li>5. Actor enters expense amount.</li> <li>6. Actor sends request to add new expense.</li> <li>7. System records new expense.</li> <li>8. System records new ledger entry for cash credit against amount of expense.</li> <li>9. System records new ledger entry for expense debit against amount of expense.</li> <li>10. System displays message “Expense Added”.</li> </ol> <b>Delete Expense</b> <ol style="list-style-type: none"> <li>1. Actor navigates to Expense Management.</li> <li>2. System display list of expenses.</li> <li>3. Actor deletes expense from the list.</li> <li>4. System asks for confirmation.</li> <li>5. Actor confirms expense deletion.</li> <li>6. System removes expense record.</li> <li>7. System removes cash ledger entry recorded against this expense.</li> <li>8. System removes expense ledger entry recorded against this expense.</li> <li>9. System displays message “Expense Deleted”.</li> </ol>		

<b>Alternative Flows:</b>	N.A
<b>Exceptions:</b>	<p><b>Add Expense</b></p> <p>4. Expense title is empty</p> <ol style="list-style-type: none"> <li>1. System displays message “Expense title is required”.</li> <li>2. Actor enters expense title.</li> <li>3. Use case resume on step 5</li> </ol> <p>5a. Expense amount is a negative number</p> <ol style="list-style-type: none"> <li>1. System display message “Expense amount should be greater than 0”.</li> <li>2. Actor enters valid expense amount.</li> <li>3. Use case resume on step 6</li> </ol> <p>5b. Expense amount field is empty</p> <ol style="list-style-type: none"> <li>1. System displays message “Expense amount is required”.</li> <li>2. Actor enters expense amount.</li> <li>3. Use case resume on step 6</li> </ol> <p>5c. Expense amount is not a number</p> <ol style="list-style-type: none"> <li>1. System displays message “Expense amount should be a number”.</li> <li>2. Actor enter valid expense amount.</li> <li>3. Use case resume on step 6</li> </ol> <p>7. System fail to record new expense record</p> <ol style="list-style-type: none"> <li>1. System tries again to add the record.</li> <li>2. Use case resume on step 8</li> </ol> <p>8. System fail to record new cash ledger entry</p> <ol style="list-style-type: none"> <li>1. System tries again to add the record.</li> <li>2. Use case resume on step 9</li> </ol> <p>9. System fail to record new expense ledger entry</p> <ol style="list-style-type: none"> <li>1. System tries again to add the record.</li> <li>2. Use case resume on step 10</li> </ol> <p><b>Delete Expense</b></p> <p>6. System fails to delete the expense record</p> <ol style="list-style-type: none"> <li>1. System tries again to delete the record.</li> <li>2. Use case resumes on step 7</li> </ol> <p>7. System fails to delete the cash ledger entry record</p> <ol style="list-style-type: none"> <li>1. System tries again to delete the record.</li> <li>2. Use case resumes on step 8</li> </ol> <p>8. System fails to delete the expense ledger entry record</p> <ol style="list-style-type: none"> <li>1. System tries again to delete the record.</li> <li>2. Use case resumes on step 9</li> </ol>
<b>Includes:</b>	N.A
<b>Frequency of Use:</b>	Could be nearly continuous.
<b>Special Requirements:</b>	Internet Connection
<b>Assumptions:</b>	N.A
<b>Notes and Issues:</b>	N.A

**Table 11(3.5): UC-7 Manage Payments**

<b>Use Case ID:</b>	UC-7		
<b>Use Case Name:</b>	Manage Payments		
<b>Created By:</b>	Abdul Raheem	<b>Last Updated By:</b>	Abdul Raheem
<b>Date Created:</b>	July 03, 2022	<b>Last Revision Date:</b>	Oct 14, 2022
<b>Actors:</b>	Accounts Manager		
<b>Description:</b>	Actor would be able to payment in and payment out		
<b>Trigger:</b>	Actor navigates to Payment Management.		
<b>Preconditions:</b>	Account Manager is authorized to the application.		
<b>Post conditions:</b>	<p><b>Payment In</b></p> <ol style="list-style-type: none"> <li>1. System updates payment record.</li> <li>2. System records new ledger entry for cash account debit against received amount.</li> <li>3. System records new ledger entry for customer account credit against received amount.</li> <li>4. System display message “Payment Received”.</li> </ol> <p><b>Payment Out</b></p> <ol style="list-style-type: none"> <li>1. System updates payment record.</li> <li>2. System records new ledger entry for cash account credit against received amount.</li> <li>3. System records new ledger entry for vendor account debit against received amount.</li> <li>4. System display message “Payment Paid”.</li> </ol>		
<b>Normal Flow:</b>	<p><b>Payment In</b></p> <ol style="list-style-type: none"> <li>1. Actor navigates to Payment Management.</li> <li>2. Actor selects customer account.</li> <li>3. Actor selects sale invoice.</li> <li>4. System displays receivable amount.</li> <li>5. Actor enters amount received.</li> <li>6. Actor sends request to update receivable payment.</li> <li>7. System updates payment record.</li> <li>8. System records new ledger entry for cash account debit against received amount.</li> <li>9. System records new ledger entry for customer account credit against received amount.</li> <li>10. System display message “Payment Received”.</li> </ol> <p><b>Payment Out</b></p> <ol style="list-style-type: none"> <li>1. Actor navigates to Payment Management.</li> <li>2. Actor selects vendor account.</li> <li>3. Actor selects purchase invoice.</li> <li>4. System displays payable amount.</li> <li>5. Actor enters amount paid.</li> <li>6. Actor sends request to update payable payment.</li> <li>7. System updates payment record.</li> <li>8. System records new ledger entry for cash account credit against received amount.</li> <li>9. System records new ledger entry for vendor account debit against received amount.</li> <li>10. System display message “Payment Paid”.</li> </ol>		

<b>Alternative Flows:</b>	N.A
<b>Exceptions:</b>	<p><b>Payment In</b></p> <ol style="list-style-type: none"> <li>2. Customer account not selected. <ol style="list-style-type: none"> <li>1. System display message “Customer account is required”.</li> <li>2. Actor selects customer account.</li> <li>3. Use case resumes on step 3.</li> </ol> </li> <li>3. Sale invoice not selected. <ol style="list-style-type: none"> <li>1. System display message “Sale invoice is required”.</li> <li>2. Actor selects sale invoice.</li> <li>3. Use case resumes on step 4.</li> </ol> </li> <li>5a. Amount field is empty. <ol style="list-style-type: none"> <li>1. System display message “Amount is required”.</li> <li>2. Actor enters valid amount.</li> <li>3. Use case resume on step 6.</li> </ol> </li> <li>5b. Amount field contains alphabet. <ol style="list-style-type: none"> <li>1. System display message “Please enter amount in numbers only”.</li> <li>2. Actor enters valid amount.</li> <li>3. Use case resumes on step 6.</li> </ol> </li> <li>5c. Amount is greater than receivable amount. <ol style="list-style-type: none"> <li>1. System display message “Amount can’t be greater than receivable amount”.</li> <li>2. Actor enters valid amount.</li> <li>3. Use case resumes on step 6.</li> </ol> </li> <li>7. System failed to update payment record. <ol style="list-style-type: none"> <li>1. System tries again to update payment.</li> <li>2. Use case resumes on step 8.</li> </ol> </li> <li>8. System failed to record new ledger entry for cash account. <ol style="list-style-type: none"> <li>1. System tries again to record new ledger entry for cash account.</li> <li>2. Use case resumes on step 9.</li> </ol> </li> <li>9. System failed to record new ledger entry for customer account. <ol style="list-style-type: none"> <li>1. System tries again to record new ledger entry for customer account.</li> <li>2. Use case resumes on step 10.</li> </ol> </li> </ol> <p><b>Payment Out</b></p> <ol style="list-style-type: none"> <li>2. Vendor account not selected. <ol style="list-style-type: none"> <li>1. System displays message “Vendor account is required”.</li> <li>2. Actor selects vendor account.</li> <li>3. Use case resumes on step 3.</li> </ol> </li> <li>3. Purchase invoice not selected. <ol style="list-style-type: none"> <li>1. System displays message “Purchase invoice is required”.</li> <li>2. Actor selects purchase invoice.</li> <li>3. Use case resumes on step 4.</li> </ol> </li> <li>5a. Amount field is empty. <ol style="list-style-type: none"> <li>1. System display message “Amount is required”.</li> <li>2. Actor enters valid amount.</li> <li>3. Use case resume on step 6.</li> </ol> </li> <li>5b. Amount field contains alphabet. <ol style="list-style-type: none"> <li>1. System display message “Please enter amount in numbers only”.</li> <li>2. Actor enters valid amount.</li> <li>3. Use case resumes on step 6.</li> </ol> </li> <li>5c. Amount is greater than payable amount. <ol style="list-style-type: none"> <li>1. System display message “Amount can’t be greater than payable amount”.</li> <li>2. Actor enters valid amount.</li> </ol> </li> </ol>

	3. Use case resumes on step 6. 7. System failed to update payment record. 1. System tries again to update payment. 2. Use case resumes on step 8. 8. System failed to record new ledger entry for cash account. 1. System tries again to record new ledger entry for cash account. 2. Use case resumes on step 9. 9. System failed to record new ledger entry for customer account. 1. System tries again to record new ledger entry for customer account. 2. Use case resumes on step 10.
<b>Includes:</b>	N.A
<b>Frequency of Use:</b>	Could be nearly continuous.
<b>Special Requirements:</b>	Internet Connection
<b>Assumptions:</b>	N.A
<b>Notes and Issues:</b>	N.A

**Table 12(3.5): UC-8 Manage Employee Requests**

<b>Use Case ID:</b>	UC-8		
<b>Use Case Name:</b>	Manage Employee Requests		
<b>Created By:</b>	Abdul Raheem	<b>Last Updated By:</b>	Abdul Raheem
<b>Date Created:</b>	July 03, 2022	<b>Last Revision Date:</b>	Oct 10, 2022
<b>Actors:</b>	Accounts Manager		
<b>Description:</b>	Actor would be able to approve or reject requests.		
<b>Trigger:</b>	Actor navigates to Employee Requests.		
<b>Preconditions:</b>	Account Manager is authorized to the application.		
<b>Post conditions:</b>	1. System record new employee details. 2. System display message "Request Approved".		
<b>Normal Flow:</b>	1. Actor navigates to Account Requests. 2. System displays all employee requests. 3. Actor views the request. 4. Actor approves the employee creation request. 5. System asks for confirmation to approve request. 6. Actor confirms request approval. 7. System records new employee details. 8. System displays message "Request Approved".		
<b>Alternative Flows:</b>	N.A		
<b>Exceptions:</b>	4. Actor rejects employee creation request 1. System updates request status to "Rejected". 2. Use case ends here. 7. System failed to record new employee details. 1. System tries again to record employee details. 2. Use case resumes on step 8.		
<b>Includes:</b>	N.A		
<b>Frequency of Use:</b>	Could be nearly continuous.		
<b>Special Requirements:</b>	Internet Connection		
<b>Assumptions:</b>	N.A		

<b>Notes and Issues:</b>	N.A
--------------------------	-----

**Table 13(3.5): UC-9 Manage Purchases**

<b>Use Case ID:</b>	UC-13		
<b>Use Case Name:</b>	Manage Purchases		
<b>Created By:</b>	M. Haris Siddiqui	<b>Last Updated By:</b>	Abdul Raheem
<b>Date Created:</b>	July 03, 2022	<b>Last Revision Date:</b>	Oct 10, 2022
<b>Actors:</b>	Procurement manager		
<b>Description:</b>	Actor would be able to add, view purchase and add, view purchase return.		
<b>Trigger:</b>	Actor navigates to sale management.		
<b>Preconditions:</b>	Actor is authorized to the application.		
<b>Post conditions:</b>	<p><b>Add Purchase</b></p> <ol style="list-style-type: none"> <li>1. System record new purchase</li> <li>2. System record new ledger entry for cash account credit</li> <li>3. System record new ledger entry for raw material account debit</li> <li>4. System record new ledger entry for payable account debit</li> <li>5. System record new ledger entry for vendor account credit</li> <li>6. System display message "Purchase added successfully"</li> </ol> <p><b>View Purchase</b></p> <ol style="list-style-type: none"> <li>1. System display purchase details</li> </ol> <p><b>Return Purchase</b></p> <ol style="list-style-type: none"> <li>1. System add new return record</li> <li>2. System add new ledger entry for raw material account credit</li> <li>3. System add new ledger entry for cash account debit</li> <li>4. System add new ledger entry for vendor account credit</li> <li>5. System add new ledger entry for payable account debit</li> <li>6. System display message "Purchase return added"</li> </ol> <p><b>View Invoice</b></p> <ol style="list-style-type: none"> <li>1. System display invoice details</li> </ol>		
<b>Normal Flow:</b>	<p><b>Add Sale</b></p> <ol style="list-style-type: none"> <li>1. Actor navigate to purchase management</li> <li>2. System display list of all purchases</li> <li>3. Actor select add new purchase</li> <li>4. Actor select the vendor</li> <li>5. Actor enter the paid amount</li> <li>6. Actor select the raw material</li> <li>7. Actor enter the raw material quantity</li> <li>8. Actor enter per unit price of raw material</li> <li>9. Actor send request to add raw material in list</li> </ol> <p>Actor repeats step 6-9 until indicates done</p> <ol style="list-style-type: none"> <li>10. Actor enter expense title</li> <li>11. Actor enter expense cost</li> <li>12. Actor send request to add expense in list</li> </ol> <p>Actor repeats step 10-12 until indicates done</p> <ol style="list-style-type: none"> <li>13. Actor send request to add new purchase</li> <li>14. System record new purchase</li> <li>15. System record new ledger entry for cash account credit</li> <li>16. System record new ledger entry for raw material account debit</li> <li>17. System record new ledger entry for payable account debit</li> <li>18. System record new ledger entry for vendor account credit</li> <li>19. System display message "Purchase added successfully"</li> </ol>		



	<p><b>View Purchase</b></p> <ol style="list-style-type: none"> <li>1. Actor navigate to purchase management</li> <li>2. Actor send request to view purchase details</li> <li>3. System display purchase details</li> </ol> <p><b>Return Sale</b></p> <ol style="list-style-type: none"> <li>1. Actor navigate to purchase management</li> <li>2. Actor select purchase return</li> <li>3. Actor select item from item list</li> <li>4. Actor enter return quantity</li> <li>5. Actor send a request to return item</li> </ol> <p>Actor repeats step 3-5 until indicates done</p> <ol style="list-style-type: none"> <li>6. Actor send a request to add new return</li> <li>7. System add new return record</li> <li>8. System add new ledger entry for raw material account credit</li> <li>9. System add new ledger entry for cash account debit</li> <li>10. System add new ledger entry for vendor account credit</li> <li>11. System add new ledger entry for payable account debit</li> <li>12. System display message "Purchase return added"</li> </ol> <p><b>View Invoice</b></p> <ol style="list-style-type: none"> <li>1. Actor navigate to purchase management</li> <li>2. Actor send a request to view invoice</li> <li>3. System display invoice details</li> </ol>
<b>Alternative Flows:</b>	N.A
<b>Exceptions:</b>	<p><b>Add Purchase</b></p> <p>4. Vendor field is not selected</p> <ol style="list-style-type: none"> <li>1. System display message "Vendor field is required"</li> <li>2. Actor select vendor</li> <li>3. Use case resumes on step 5</li> </ol> <p>5a. Paid amount field is empty</p> <ol style="list-style-type: none"> <li>1. System display message "Paid amount field is required"</li> <li>2. Actor enter paid amount</li> <li>3. Use case resumes on step 6</li> </ol> <p>5b. Paid amount field contains alphabet</p> <ol style="list-style-type: none"> <li>1. System display message "Paid amount is invalid"</li> <li>2. Actor enter valid paid amount</li> <li>3. Use case resumes on step 6</li> </ol> <p>5c. Paid amount is a negative number</p> <ol style="list-style-type: none"> <li>1. System display message "Paid amount is invalid"</li> <li>2. Actor enter valid paid amount</li> <li>3. Use case resumes on step 6</li> </ol> <p>6. Raw material field is not selected</p> <ol style="list-style-type: none"> <li>1. System display message "Raw material field is required"</li> <li>2. Actor select product</li> <li>3. Use case resumes on step 7</li> </ol> <p>7a. Raw material quantity field is empty</p> <ol style="list-style-type: none"> <li>1. System display message "Raw material quantity field is required"</li> </ol>

	<ol style="list-style-type: none"> <li>2. Actor enter raw material quantity</li> <li>3. Use case resumes on step 8</li> </ol>
	<p>7b. Raw material quantity field contains alphabet</p> <ol style="list-style-type: none"> <li>1. System display message “Raw material quantity is invalid”</li> <li>2. Actor enter valid raw material quantity</li> <li>3. Use case resumes on step 8</li> </ol>
	<p>7c. Raw material quantity is a negative number</p> <ol style="list-style-type: none"> <li>1. System display message “Raw material quantity is invalid”</li> <li>2. Actor enter valid raw material quantity</li> <li>3. Use case resumes on step 8</li> </ol>
	<p>8a. Per unit price field is empty</p> <ol style="list-style-type: none"> <li>1. System display message “Per unit price field is required”</li> <li>2. Actor enter per unit price</li> <li>3. Use case resumes on step 9</li> </ol>
	<p>8b. Per unit price field contains alphabet</p> <ol style="list-style-type: none"> <li>1. System display message “Per unit price is invalid”</li> <li>2. Actor enter valid per unit price</li> <li>3. Use case resumes on step 9</li> </ol>
	<p>8c. Per unit price is a negative number</p> <ol style="list-style-type: none"> <li>1. System display message “Per unit price is invalid”</li> <li>2. Actor enter valid per unit price</li> <li>3. Use case resumes on step 9</li> </ol>
	<p>10. Expense title field is empty</p> <ol style="list-style-type: none"> <li>1. System display message “Expense title field is required”</li> <li>2. Actor enter expense title</li> <li>3. Use case resume on step 11</li> </ol>
	<p>11a. Expense cost field is empty</p> <ol style="list-style-type: none"> <li>1. System display message “Expense cost field is required”</li> <li>2. Actor enter Expense cost</li> <li>3. Use case resumes on step 12</li> </ol>
	<p>11b. Expense cost field contains alphabet</p> <ol style="list-style-type: none"> <li>1. System display message “Expense cost is invalid”</li> <li>2. Actor enter valid Expense cost</li> <li>3. Use case resumes on step 12</li> </ol>
	<p>11c. Expense cost is a negative number</p> <ol style="list-style-type: none"> <li>1. System display message “Expense cost is invalid”</li> <li>2. Actor enter valid Expense cost</li> <li>3. Use case resumes on step 12</li> </ol>
	<p>14. System failed to add new record</p> <ol style="list-style-type: none"> <li>1. System tries again to add record</li> </ol>

	<ol style="list-style-type: none"> <li>2. Use case resumes on step 15</li> </ol> <p>15. System failed to add new cash ledger entry</p> <ol style="list-style-type: none"> <li>1. System tries again to add record</li> <li>2. Use case resumes on step 16</li> </ol> <p>16. System failed to add new raw material ledger entry</p> <ol style="list-style-type: none"> <li>1. System tries again to add record</li> <li>2. Use case resumes on step 17</li> </ol> <p>17. System failed to add new payable ledger entry</p> <ol style="list-style-type: none"> <li>1. System tries again to add record</li> <li>2. Use case resumes on step 18</li> </ol> <p>18. System failed to add new vendor ledger entry</p> <ol style="list-style-type: none"> <li>1. System tries again to add record</li> <li>2. Use case resumes on step 19</li> </ol> <p><b>Return Sale</b></p> <p>3. Item field is not selected</p> <ol style="list-style-type: none"> <li>1. System display message “Item field is required”</li> <li>2. Actor select item from the list</li> <li>3. Use case resumes on step 4</li> </ol> <p>4a. Return quantity field is empty</p> <ol style="list-style-type: none"> <li>1. System display message “Return quantity field is required”</li> <li>2. Actor enter return quantity</li> <li>3. Use case resumes on step 5</li> </ol> <p>4b. Return quantity field contains alphabet</p> <ol style="list-style-type: none"> <li>1. System display message “Return quantity is invalid”</li> <li>2. Actor enter valid return quantity</li> <li>3. Use case resumes on step 5</li> </ol> <p>4c. Return quantity is a negative number</p> <ol style="list-style-type: none"> <li>1. System display message “Return quantity is invalid”</li> <li>2. Actor enter valid return quantity</li> <li>3. Use case resumes on step 5</li> </ol> <p>7. System failed to add new record</p> <ol style="list-style-type: none"> <li>1. System tries again to add record</li> <li>2. Use case resumes on step 8</li> </ol> <p>8. System failed to add new cash ledger entry</p> <ol style="list-style-type: none"> <li>1. System tries again to add record</li> <li>2. Use case resumes on step 9</li> </ol> <p>9. System failed to add new raw material ledger entry</p> <ol style="list-style-type: none"> <li>1. System tries again to add record</li> <li>2. Use case resumes on step 10</li> </ol>
--	--

	10. System failed to add new payable ledger entry <ol style="list-style-type: none"> <li>1. System tries again to add record</li> <li>2. Use case resumes on step 11</li> </ol> 11. System failed to add new vendor ledger entry <ol style="list-style-type: none"> <li>1. System tries again to add record</li> <li>2. Use case resumes on step 12</li> </ol>
<b>Includes:</b>	N.A
<b>Frequency of Use:</b>	Could be nearly continuous.
<b>Special Requirements:</b>	Internet Connection
<b>Assumptions:</b>	N.A
<b>Notes and Issues:</b>	N.A

**Table 14(3.5): UC-10 Manage Production Requests**

<b>Use Case ID:</b>	UC8		
<b>Use Case Name:</b>	Manage Production Requests		
<b>Created By:</b>	M. Zeeshan Fiaz	<b>Last Updated By:</b>	Abdul Raheem
<b>Date Created:</b>	July 03, 2022	<b>Last Revision Date:</b>	Oct 10, 2022
<b>Actors:</b>	Procurement Manager		
<b>Description:</b>	Actor would be able to approve or reject requests.		
<b>Trigger:</b>	Actor navigates to Production Requests.		
<b>Preconditions:</b>	Procurement Manager is authorized to the application.		
<b>Post conditions:</b>	<ol style="list-style-type: none"> <li>1. System record new production details.</li> <li>2. System records a ledger for the raw materials credit against the request.</li> <li>3. System records a ledger for the production debit against the request.</li> <li>4. System display message “Request Approved”.</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Actor navigates to Production Requests.</li> <li>2. System displays all add production requests.</li> <li>3. Actor views the request.</li> <li>4. Actor approves the add product request.</li> <li>5. System asks for confirmation to approve request.</li> <li>6. Actor confirms request approval.</li> <li>7. System records new production details.</li> <li>8. System records new ledger entry for the raw materials account credit against the request.</li> <li>9. System records new ledger entry for the product account debit against the request.</li> <li>10. System displays message “Request Approved”.</li> </ol>		
<b>Alternative Flows:</b>	N.A		
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>4. Actor rejects production request. <ol style="list-style-type: none"> <li>1. System updates request status to “Rejected”.</li> <li>2. Use case ends here.</li> </ol> </li> <li>7. System failed to record new production details. <ol style="list-style-type: none"> <li>1. System tries again to record production details.</li> <li>2. Use case resumes on step 8.</li> </ol> </li> <li>8. System failed to create a ledger for raw materials account.</li> </ol>		

	<ol style="list-style-type: none"> <li>1. System tries again to record new ledger entry for raw materials account.</li> <li>2. Use Case resume on step 9.</li> </ol> 9. System failed to create a ledger for product account. <ol style="list-style-type: none"> <li>1. System tries again to record new ledger entry for product account.</li> <li>2. Use case resumes on step 10.</li> </ol>
<b>Includes:</b>	N.A
<b>Frequency of Use:</b>	Could be nearly continuous.
<b>Special Requirements:</b>	Internet Connection
<b>Assumptions:</b>	N.A
<b>Notes and Issues:</b>	N.A

**Table 15(3.5): UC-11 Manage Sales**

<b>Use Case ID:</b>	UC-11		
<b>Use Case Name:</b>	Manage Sales		
<b>Created By:</b>	M. Haris Siddiqui	<b>Last Updated By:</b>	Abdul Raheem
<b>Date Created:</b>	July 03, 2022	<b>Last Revision Date:</b>	Oct 10, 2022
<b>Actors:</b>	Sale managers		
<b>Description:</b>	Actor would be able to add, view sale and add, view sale return.		
<b>Trigger:</b>	Actor navigates to sale management.		
<b>Preconditions:</b>	Actor is authorized to the application.		
<b>Post conditions:</b>	<ol style="list-style-type: none"> <li>1. <b>Add Sale</b></li> <li>2. System record new sale</li> <li>3. System record new ledger entry for cash account debit</li> <li>4. System record new ledger entry for product account credit</li> <li>5. System record new ledger entry for receivable account debit</li> <li>6. System record new ledger entry for customer account credit</li> <li>7. System display message "Sale added successfully"</li> </ol> <b>View Sale</b> <ol style="list-style-type: none"> <li>1. System display sale details</li> </ol> <b>Return Sale</b> <ol style="list-style-type: none"> <li>1. System add new return record</li> <li>2. System add new ledger entry for product account debit</li> <li>3. System add new ledger entry for cash account credit</li> <li>4. System add new ledger entry for customer account debit</li> <li>5. System add new ledger entry for receivable account credit</li> <li>6. System display message "Sale return added"</li> </ol> <b>View Invoice</b> <ol style="list-style-type: none"> <li>1. System display invoice details</li> </ol>		
<b>Normal Flow:</b>	<b>Add Sale</b> <ol style="list-style-type: none"> <li>1. Actor navigate to sale management</li> <li>2. System display list of all sales</li> <li>3. Actor select add new sale</li> <li>4. Actor select the customer</li> <li>5. Actor enter the paid amount</li> <li>6. Actor select the product</li> <li>7. Actor enter the product quantity</li> <li>8. Actor enter per unit price of product</li> <li>9. Actor send request to add product in list</li> </ol> Actor repeats step 6-9 until indicates done		

	<ul style="list-style-type: none"> <li>10. Actor enter expense title</li> <li>11. Actor enter expense cost</li> <li>12. Actor send request to add expense in list</li> <li>Actor repeats step 10-12 until indicates done</li> <li>13. Actor send request to add new sale</li> <li>14. System record new sale</li> <li>15. System record new ledger entry for cash account debit</li> <li>16. System record new ledger entry for product account credit</li> <li>17. System record new ledger entry for receivable account debit</li> <li>18. System record new ledger entry for customer account credit</li> <li>19. System display message “Sale added successfully”</li> </ul> <p><b>View Sale</b></p> <ul style="list-style-type: none"> <li>1. Actor navigate to sale management</li> <li>2. Actor send request to view sale details</li> <li>3. System display sale details</li> </ul> <p><b>Return Sale</b></p> <ul style="list-style-type: none"> <li>1. Actor navigate to sale management</li> <li>2. Actor select sale return</li> <li>3. Actor select item from item list</li> <li>4. Actor enter return quantity</li> <li>5. Actor send a request to return item</li> <li>Actor repeats step 3-5 until indicates done</li> <li>6. Actor send a request to add new return</li> <li>7. System add new return record</li> <li>8. System add new ledger entry for product account debit</li> <li>9. System add new ledger entry for cash account credit</li> <li>10. System add new ledger entry for customer account debit</li> <li>11. System add new ledger entry for receivable account credit</li> <li>12. System display message “Sale return added”</li> </ul> <p><b>View Invoice</b></p> <ul style="list-style-type: none"> <li>1. Actor navigate to sale management</li> <li>2. Actor send a request to view invoice</li> <li>3. System display invoice details</li> </ul>
<b>Alternative Flows:</b>	N.A
<b>Exceptions:</b>	<p><b>Add Sale</b></p> <ul style="list-style-type: none"> <li>4. Customer field is not selected <ul style="list-style-type: none"> <li>1. System display message “Customer field is required”</li> <li>2. Actor select customer</li> <li>3. Use case resumes on step 5</li> </ul> </li> <li>5a. Paid amount field is empty <ul style="list-style-type: none"> <li>1. System display message “Paid amount field is required”</li> <li>2. Actor enter paid amount</li> <li>3. Use case resumes on step 6</li> </ul> </li> <li>5b. Paid amount contains alphabet <ul style="list-style-type: none"> <li>1. System display message “Paid amount is invalid”</li> <li>2. Actor enter valid paid amount</li> <li>3. Use case resumes on step 6</li> </ul> </li> <li>5c. Paid amount is a negative number</li> </ul>

	<ol style="list-style-type: none"> <li>1. System display message “Paid amount is invalid”</li> <li>2. Actor enter valid paid amount</li> <li>3. Use case resumes on step 6</li> </ol> <p>6. Product field is not selected</p> <ol style="list-style-type: none"> <li>1. System display message “Product field is required”</li> <li>2. Actor select product</li> <li>3. Use case resumes on step 7</li> </ol> <p>7a. Product quantity field is empty</p> <ol style="list-style-type: none"> <li>1. System display message “Product quantity field is required”</li> <li>2. Actor enter product quantity</li> <li>3. Use case resumes on step 8</li> </ol> <p>7b. Product quantity contains alphabet</p> <ol style="list-style-type: none"> <li>1. System display message “Product quantity is invalid”</li> <li>2. Actor enter valid product quantity</li> <li>3. Use case resumes on step 8</li> </ol> <p>7c. Product quantity is a negative number</p> <ol style="list-style-type: none"> <li>1. System display message “Product quantity is invalid”</li> <li>2. Actor enter valid product quantity</li> <li>3. Use case resumes on step 8</li> </ol> <p>8a. Per unit price field is empty</p> <ol style="list-style-type: none"> <li>1. System display message “Per unit price field is required”</li> <li>2. Actor enter per unit price</li> <li>3. Use case resumes on step 9</li> </ol> <p>8b. Per unit price contains alphabet</p> <ol style="list-style-type: none"> <li>1. System display message “Per unit price is invalid”</li> <li>2. Actor enter valid per unit price</li> <li>3. Use case resumes on step 9</li> </ol> <p>8c. Per unit price is a negative number</p> <ol style="list-style-type: none"> <li>1. System display message “Per unit price is invalid”</li> <li>2. Actor enter valid per unit price</li> <li>3. Use case resumes on step 9</li> </ol> <p>10. Expense title field is empty</p> <ol style="list-style-type: none"> <li>1. System display message “Expense title field is required”</li> <li>2. Actor enter expense title</li> <li>3. Use case resume on step 11</li> </ol> <p>11a. Expense cost field is empty</p> <ol style="list-style-type: none"> <li>1. System display message “Expense cost field is required”</li> <li>2. Actor enter Expense cost</li> <li>3. Use case resumes on step 12</li> </ol> <p>11b. Expense cost contains alphabet</p> <ol style="list-style-type: none"> <li>1. System display message “Expense cost is invalid”</li> </ol>
--	---

	<ol style="list-style-type: none"> <li>2. Actor enter valid Expense cost</li> <li>3. Use case resumes on step 12</li> </ol> <p>11c. Expense cost is a negative number</p> <ol style="list-style-type: none"> <li>1. System display message “Expense cost is invalid”</li> <li>2. Actor enter valid Expense cost</li> <li>3. Use case resumes on step 12</li> </ol> <p>14. System failed to add new record</p> <ol style="list-style-type: none"> <li>1. System tries again to add record</li> <li>2. Use case resumes on step 15</li> </ol> <p>15. System failed to add new cash ledger entry</p> <ol style="list-style-type: none"> <li>1. System tries again to add record</li> <li>2. Use case resumes on step 16</li> </ol> <p>16. System failed to add new product ledger entry</p> <ol style="list-style-type: none"> <li>1. System tries again to add record</li> <li>2. Use case resumes on step 17</li> </ol> <p>17. System failed to add new receivable ledger entry</p> <ol style="list-style-type: none"> <li>1. System tries again to add record</li> <li>2. Use case resumes on step 18</li> </ol> <p>18. System failed to add new customer ledger entry</p> <ol style="list-style-type: none"> <li>1. System tries again to add record</li> <li>2. Use case resumes on step 19</li> </ol> <p><b>Return Sale</b></p> <p>3. Item field is not selected</p> <ol style="list-style-type: none"> <li>1. System display message “Item field is required”</li> <li>2. Actor select item from the list</li> <li>3. Use case resumes on step 4</li> </ol> <p>4a. Return quantity field is empty</p> <ol style="list-style-type: none"> <li>1. System display message “Return quantity field is required”</li> <li>2. Actor enter return quantity</li> <li>3. Use case resumes on step 5</li> </ol> <p>4b. Return quantity contains alphabet</p> <ol style="list-style-type: none"> <li>1. System display message “Return quantity is invalid”</li> <li>2. Actor enter valid return quantity</li> <li>3. Use case resumes on step 5</li> </ol> <p>4c. Return quantity is a negative number</p> <ol style="list-style-type: none"> <li>1. System display message “Return quantity is invalid”</li> <li>2. Actor enter valid return quantity</li> <li>3. Use case resumes on step 5</li> </ol> <p>7. System failed to add new record</p>
--	---



	<ol style="list-style-type: none"> <li>1. System tries again to add record</li> <li>2. Use case resumes on step 8</li> </ol> <p>8. System failed to add new cash ledger entry</p> <ol style="list-style-type: none"> <li>1. System tries again to add record</li> <li>2. Use case resumes on step 9</li> </ol> <p>9. System failed to add new product ledger entry</p> <ol style="list-style-type: none"> <li>1. System tries again to add record</li> <li>2. Use case resumes on step 10</li> </ol> <p>10. System failed to add new receivable ledger entry</p> <ol style="list-style-type: none"> <li>1. System tries again to add record</li> <li>2. Use case resumes on step 11</li> </ol> <p>11. System failed to add new customer ledger entry</p> <ol style="list-style-type: none"> <li>1. System tries again to add record</li> <li>2. Use case resumes on step 12</li> </ol>
<b>Includes:</b>	N.A
<b>Frequency of Use:</b>	Could be nearly continuous.
<b>Special Requirements:</b>	Internet Connection
<b>Assumptions:</b>	N.A
<b>Notes and Issues:</b>	N.A

**Table 16(3.5): UC-12 Manage Products**

<b>Use Case ID:</b>	UC-12		
<b>Use Case Name:</b>	Manage Products		
<b>Created By:</b>	Abdul Raheem	<b>Last Updated By:</b>	Abdul Raheem
<b>Date Created:</b>	July 03, 2022	<b>Last Revision Date:</b>	Oct 10, 2022
<b>Actors:</b>	System Manager		
<b>Description:</b>	Actor would be able to add, remove and update products.		
<b>Trigger:</b>	Actor navigates to product management.		
<b>Preconditions:</b>	Actor is authorized to the application.		
<b>Post conditions:</b>	<ol style="list-style-type: none"> <li>1. System records a new product.</li> <li>2. System display message “Product added successfully”.</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Actor navigates to product management.</li> <li>2. System displays the list of products.</li> <li>3. Actor enters product title.</li> <li>4. Actor enters product code.</li> <li>5. Actor sends the request to add product.</li> <li>6. System asks for confirmation to add product.</li> <li>7. Actor confirms the request.</li> <li>8. System records a new product.</li> <li>9. System display message “Product added successfully”.</li> </ol>		
<b>Alternative Flows:</b>	N.A		

<b>Exceptions:</b>	3. Product title is empty. <ol style="list-style-type: none"> <li>1. System displays message “Product title is required”.</li> <li>2. Actor enters valid product title.</li> <li>3. Use case resumes on step 4.</li> </ol> 4a. Product code is empty. <ol style="list-style-type: none"> <li>1. System displays message “Product code is required”</li> <li>2. Actor enters valid product.</li> <li>3. Use case resumes on step 5.</li> </ol> 4b. Product code is already in record. <ol style="list-style-type: none"> <li>1. System displays message “Product is already in record”</li> <li>2. Actor enter valid product code</li> <li>3. Use case resumes on step 5.</li> </ol>
<b>Includes:</b>	N.A
<b>Frequency of Use:</b>	Could be nearly continuous.
<b>Special Requirements:</b>	Internet Connection
<b>Assumptions:</b>	N.A
<b>Notes and Issues:</b>	N.A

**Table 17(3.5): UC-13 Manage Raw Material**

<b>Use Case ID:</b>	UC-13		
<b>Use Case Name:</b>	Manage Raw Material		
<b>Created By:</b>	Abdul Raheem	<b>Last Updated By:</b>	Abdul Raheem
<b>Date Created:</b>	July 03, 2022	<b>Last Revision Date:</b>	Oct 10, 2022
<b>Actors:</b>	System Manager		
<b>Description:</b>	Actor would be able to add, remove and update raw material.		
<b>Trigger:</b>	Actor navigates to raw material management.		
<b>Preconditions:</b>	Actor is authorized to the application.		
<b>Post conditions:</b>	<ol style="list-style-type: none"> <li>1. System records a new raw material.</li> <li>2. System display message “Raw material added successfully”.</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Actor navigates to raw material management.</li> <li>2. System displays the list of all raw materials.</li> <li>3. Actor enters raw material title.</li> <li>4. Actor sends the request to add raw material.</li> <li>5. System asks for confirmation to add raw material.</li> <li>6. Actor confirms the request.</li> <li>7. System records a new raw material.</li> <li>8. System displays message “Raw material added successfully”.</li> </ol>		
<b>Alternative Flows:</b>	N.A		
<b>Exceptions:</b>	3a. Raw Martial title is empty. <ol style="list-style-type: none"> <li>1. System displays message “Raw Martial title is required”.</li> <li>2. Actor enters valid raw martial title.</li> <li>4. Use case resumes on step 4.</li> </ol> 3b. Raw Martial title is already in record. <ol style="list-style-type: none"> <li>1. System displays message “Raw Martial title is already in record”</li> <li>2. Actor enters valid raw martial title</li> <li>3. Use case resumes on step 4.</li> </ol>		
<b>Includes:</b>	N.A		
<b>Frequency of Use:</b>	Could be nearly continuous.		

<b>Special Requirements:</b>	Internet Connection
<b>Assumptions:</b>	N.A
<b>Notes and Issues:</b>	N.A

**Table 18(3.5): UC-14 Manage Departments**

<b>Use Case ID:</b>	UC-14		
<b>Use Case Name:</b>	Manage Departments		
<b>Created By:</b>	Abdul Raheem	<b>Last Updated By:</b>	Abdul Raheem
<b>Date Created:</b>	July 03, 2022	<b>Last Revision Date:</b>	Oct 10, 2022
<b>Actors:</b>	System Manager		
<b>Description:</b>	Actor would be able to add, remove and update departments.		
<b>Trigger:</b>	Actor navigates to department management.		
<b>Preconditions:</b>	Actor is authorized to the application.		
<b>Post conditions:</b>	<ol style="list-style-type: none"> <li>1. System records a new department.</li> <li>2. System display message "Department added successfully".</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Actor navigates to department management.</li> <li>2. System displays the list of all departments.</li> <li>3. Actor enters department title.</li> <li>4. Actor sends a request to add department.</li> <li>5. System asks for confirmation.</li> <li>6. Actor confirms the request.</li> <li>7. System records a new department.</li> <li>8. System displays message "Department added successfully".</li> </ol>		
<b>Alternative Flows:</b>	N.A		
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>3a. Department title is empty. <ol style="list-style-type: none"> <li>1. System displays message "Department title is required".</li> <li>2. Actor enters valid department title.</li> <li>3. Use case resumes on step 4.</li> </ol> </li> <li>3b. Department title is already record <ol style="list-style-type: none"> <li>1. System displays message "Department is already in record".</li> <li>2. Actor enters valid department title.</li> <li>3. Use case resumes on step 4.</li> </ol> </li> </ol>		
<b>Includes:</b>	N.A		
<b>Frequency of Use:</b>	Could be nearly continuous.		
<b>Special Requirements:</b>	Internet Connection		
<b>Assumptions:</b>	N.A		
<b>Notes and Issues:</b>	N.A		

**Table 19(3.5): UC-15 Manage Backup**

<b>Use Case ID:</b>	UC9		
<b>Use Case Name:</b>	Manage Backup		
<b>Created By:</b>	Abdul Raheem	<b>Last Updated By:</b>	Abdul Raheem
<b>Date Created:</b>	July 03, 2022	<b>Last Revision Date:</b>	Oct 10, 2022
<b>Actors:</b>	System Manager		
<b>Description:</b>	Actor would be able to create or restore backup.		

<b>Trigger:</b>	Actor navigates to product management.
<b>Preconditions:</b>	Actor is authorized to the application.
<b>Post conditions:</b>	<b>Create Backup</b> <ol style="list-style-type: none"> <li>1. System generates backup.</li> <li>2. System display message “Backup created successfully”</li> </ol> <b>Restore Backup</b> <ol style="list-style-type: none"> <li>1. System restore backup.</li> <li>2. System display message “Backup restored successfully”</li> </ol>
<b>Normal Flow:</b>	<b>Create Backup</b> <ol style="list-style-type: none"> <li>1. Actor navigates to backup management.</li> <li>2. Actor sends a request to create backup.</li> <li>3. System asks for confirmation to create backup.</li> <li>4. Actor confirms backup request.</li> <li>5. System generates backup.</li> <li>6. System display message “Backup created successfully”</li> </ol> <b>Restore Backup</b> <ol style="list-style-type: none"> <li>1. Actor navigates to backup management.</li> <li>2. Actor selects the backup file.</li> <li>3. Actor sends a request to restore backup.</li> <li>4. System asks for confirmation to restore backup.</li> <li>5. Actor confirms backup request.</li> <li>6. System restores backup.</li> <li>7. System displays message “Backup restored successfully”</li> </ol>
<b>Alternative Flows:</b>	N.A
<b>Exceptions:</b>	<b>Create Backup</b> <ol style="list-style-type: none"> <li>5. System failed to generate backup. <ol style="list-style-type: none"> <li>1. System tries again to generate backup.</li> <li>2. Use case resume on step 6.</li> </ol> </li> </ol> <b>Restore Backup</b> <ol style="list-style-type: none"> <li>6. System failed to restore backup. <ol style="list-style-type: none"> <li>1. System tries again to restore backup.</li> <li>2. Use case resume on step 7.</li> </ol> </li> </ol>
<b>Includes:</b>	N.A
<b>Frequency of Use:</b>	Could be nearly continuous.
<b>Special Requirements:</b>	Internet Connection
<b>Assumptions:</b>	N.A
<b>Notes and Issues:</b>	N.A

**Table 20(3.5): UC-16 Manage Attendance**

<b>Use Case ID:</b>	UC-16		
<b>Use Case Name:</b>	Manage Attendance		
<b>Created By:</b>	Abdul Raheem	<b>Last Updated By:</b>	Abdul Raheem
<b>Date Created:</b>	July 03, 2022	<b>Last Revision Date:</b>	Oct 10, 2022
<b>Actors:</b>	HR Manager		
<b>Description:</b>	Actor would be able to add, view and search attendance.		
<b>Trigger:</b>	Actor navigates to Attendance Management.		

<b>Preconditions:</b>	Actor is authorized to the application.
<b>Post conditions:</b>	<b>Search Attendance</b> <ol style="list-style-type: none"> <li>1. System display selected date attendance.</li> </ol> <b>Add Attendance</b> <ol style="list-style-type: none"> <li>1. System records attendance details with current time.</li> <li>2. System displays message “Attendance marked”.</li> </ol>
<b>Normal Flow:</b>	<b>Search Attendance</b> <ol style="list-style-type: none"> <li>1. Actor navigates to attendance management.</li> <li>2. System will display current date’s attendance.</li> <li>3. Actor selects date to search attendance.</li> <li>4. Actor sends request to search for attendance.</li> <li>5. System display selected date attendance.</li> </ol> <b>Mark Attendance</b> <ol style="list-style-type: none"> <li>1. Actor navigates to attendance management.</li> <li>2. System will display current date’s attendance.</li> <li>3. Actor enters employee id.</li> <li>4. Actor sends request to mark attendance.</li> <li>5. System gets check-in/check-out time.</li> <li>6. System records attendance details with current time.</li> <li>7. System displays message “Attendance marked”.</li> </ol>
<b>Alternative Flows:</b>	N.A
<b>Exceptions:</b>	<b>Search Attendance</b> <ol style="list-style-type: none"> <li>3. Date not selected. <ol style="list-style-type: none"> <li>1. System display message “Date is required”.</li> <li>2. HR Manager selects date.</li> <li>3. Use case resumes at step 4.</li> </ol> </li> </ol> <b>Mark Attendance</b> <ol style="list-style-type: none"> <li>3a. Employee ID field is empty. <ol style="list-style-type: none"> <li>1. System display message “Employee ID is required”.</li> <li>2. HR Manager enters employee id.</li> <li>3. Use case resume on step 4.</li> </ol> </li> <li>3b. Employee ID length is not 13. <ol style="list-style-type: none"> <li>1. System display message “Invalid Employee ID”.</li> <li>2. HR Manager enters valid employee id.</li> <li>3. Use case resume on step 4.</li> </ol> </li> <li>3c. Employee ID field contains alphabet. <ol style="list-style-type: none"> <li>1. System display message “Invalid Employee ID”.</li> <li>2. HR Manager enters valid employee id.</li> <li>3. Use case resume on step 4.</li> </ol> </li> <li>5. System failed to record attendance details. <ol style="list-style-type: none"> <li>1. System tries again to record attendance.</li> <li>2. Use case resume on step 6.</li> </ol> </li> </ol>
<b>Includes:</b>	N.A

<b>Frequency of Use:</b>	Could be nearly continuous.
<b>Special Requirements:</b>	Internet Connection
<b>Assumptions:</b>	N.A
<b>Notes and Issues:</b>	N.A

**Table 21(3.5): UC-17 Manage Employees**

<b>Use Case ID:</b>	UC-17		
<b>Use Case Name:</b>	Manage Employees		
<b>Created By:</b>	Abdul Raheem	<b>Last Updated By:</b>	Abdul Raheem
<b>Date Created:</b>	July 03, 2022	<b>Last Revision Date:</b>	Oct 10, 2022
<b>Actors:</b>	HR Manager		
<b>Description:</b>	Actor would be able to add, remove and update employees.		
<b>Trigger:</b>	Actor navigates to employee management		
<b>Preconditions:</b>	N.A		
<b>Post conditions:</b>	<b>Add Employee</b> <ol style="list-style-type: none"> <li>1. System adds new account record.</li> <li>2. System displays message “Add account request initiated”.</li> </ol> <b>Update Employee</b> <ol style="list-style-type: none"> <li>1. System adds new account record.</li> <li>2. System displays message “Account update request initiated”.</li> </ol> <b>Remove Employee</b> <ol style="list-style-type: none"> <li>1. System deletes account from records.</li> <li>2. System display message “Account delete request initiated”.</li> </ol>		
<b>Normal Flow:</b>	<b>Add Employee</b> <ol style="list-style-type: none"> <li>1. Actor navigates to employee management.</li> <li>2. System displays list of all employees.</li> <li>3. Actor selects add employee option.</li> <li>4. System displays add employee form.</li> <li>5. Actor enters first name.</li> <li>6. Actor enters last name.</li> <li>7. Actor enters CNIC number.</li> <li>8. Actor enters mobile number.</li> <li>9. Actor enters address.</li> <li>10. Actor enters email address.</li> <li>11. Actor enters emergency contact number.</li> <li>12. Actor selects department.</li> <li>13. Actor enters salary.</li> <li>14. Actor sends request to add employee.</li> <li>15. System adds new employee request record.</li> <li>16. System displays message “Add employee request initiated”.</li> </ol> <b>Update Employee</b> <ol style="list-style-type: none"> <li>1. Actor navigates to employee management.</li> <li>2. System displays list of all employees.</li> <li>3. Actor selects edit employee option from employee list.</li> <li>4. System displays edit employee form.</li> <li>5. Actor enters first name.</li> </ol>		

	<ol style="list-style-type: none"> <li>6. Actor enters last name.</li> <li>7. Actor enters CNIC number.</li> <li>8. Actor enters mobile number.</li> <li>9. Actor enters address.</li> <li>10. Actor enters email address.</li> <li>11. Actor enters emergency contact number.</li> <li>12. Actor selects department.</li> <li>13. Actor enters salary.</li> <li>14. Actor sends request to update employee.</li> <li>15. System records update employee request.</li> <li>16. System displays message "Employee update request initiated".</li> </ol> <p><b>Remove Employee</b></p> <ol style="list-style-type: none"> <li>1. Actor navigates to employee management.</li> <li>2. System displays list of all employee.</li> <li>3. Actor sends request to delete employee from employee list.</li> <li>4. System deletes employee from records.</li> <li>5. System display message "Employee delete request initiated".</li> </ol>
<b>Alternative Flows:</b>	N.A
<b>Exceptions:</b>	<p><b>Add Employee</b></p> <p>5a. First name field is empty,</p> <ol style="list-style-type: none"> <li>1. System displays message "First name is required".</li> <li>2. Actor enters first name.</li> <li>3. Use case resumes on step 6.</li> </ol> <p>5b. First name field contains number or special characters,</p> <ol style="list-style-type: none"> <li>1. System displays message "Invalid first name".</li> <li>2. Actor enters first name.</li> <li>3. Use case resumes on step 6.</li> </ol> <p>6a. Last name field is empty,</p> <ol style="list-style-type: none"> <li>1. System displays message "Last name is required".</li> <li>2. Actor enters last name.</li> <li>3. Use case resumes on step 7.</li> </ol> <p>6b. Last name field contains number or special characters,</p> <ol style="list-style-type: none"> <li>1. System displays message "Last name is required".</li> <li>2. Actor enters last name.</li> <li>3. Use case resumes on step 7.</li> </ol> <p>7a. CNIC field is empty,</p> <ol style="list-style-type: none"> <li>1. System displays message "CNIC is required".</li> <li>2. Actor enters CNIC.</li> <li>3. Use case resumes on step 8.</li> </ol> <p>7b. CNIC field contains alphabet or special characters,</p> <ol style="list-style-type: none"> <li>1. System displays message "CNIC is invalid".</li> <li>2. Actor enters CNIC.</li> <li>3. Use case resumes on step 8.</li> </ol> <p>7c. CNIC length is not 13,</p> <ol style="list-style-type: none"> <li>1. System displays message "CNIC is invalid".</li> <li>2. Actor enters CNIC.</li> <li>3. Use case resumes on step 8.</li> </ol> <p>8a. Mobile number field is empty,</p> <ol style="list-style-type: none"> <li>1. System displays message "Mobile number is required".</li> <li>2. Actor enters mobile number.</li> <li>3. Use case resumes on step 9.</li> </ol> <p>8b. Mobile number field contains alphabet or special characters,</p>

	<ol style="list-style-type: none"> <li>1. System displays message “Mobile number is invalid”.</li> <li>2. Actor enters mobile number.</li> <li>3. Use case resumes on step 9.</li> </ol> <p>8c. Mobile number length is not 11,</p> <ol style="list-style-type: none"> <li>1. System displays message “Mobile number is invalid”.</li> <li>2. Actor enters mobile number.</li> <li>3. Use case resumes on step 9.</li> </ol> <p>9. Address field is empty,</p> <ol style="list-style-type: none"> <li>1. System displays message “Address is required”.</li> <li>2. Actor enters address.</li> <li>3. Use case resumes on step 10.</li> </ol> <p>10. Email address format is invalid,</p> <ol style="list-style-type: none"> <li>1. System displays message “Email address is invalid”.</li> <li>2. Actor enters email address.</li> <li>3. Use case resumes on step 11.</li> </ol> <p>11a. Emergency number field contains alphabet or special characters,</p> <ol style="list-style-type: none"> <li>1. System displays message “Emergency number is invalid”.</li> <li>2. Actor enters emergency number.</li> <li>3. Use case resumes on step 12.</li> </ol> <p>11b. Emergency number length is not 11,</p> <ol style="list-style-type: none"> <li>1. System displays message “Emergency number is invalid”.</li> <li>2. Actor enters emergency number.</li> <li>3. Use case resumes on step 12.</li> </ol> <p>12. Department is not selected,</p> <ol style="list-style-type: none"> <li>1. System displays message “Department is required”.</li> <li>2. Actor enters department.</li> <li>3. Use case resumes on step 13.</li> </ol> <p>13a. Salary field is empty,</p> <ol style="list-style-type: none"> <li>1. System displays message “Salary is required”.</li> <li>2. Actor enters salary.</li> <li>3. Use case resumes on step 14.</li> </ol> <p>13b. Salary field contains alphabet or special characters,</p> <ol style="list-style-type: none"> <li>1. System displays message “Salary is invalid”.</li> <li>2. Actor enters salary.</li> <li>3. Use case resumes on step 14.</li> </ol> <p><b>Update Employee</b></p> <p>5a. First name field is empty,</p> <ol style="list-style-type: none"> <li>1. System displays message “First name is required”.</li> <li>2. Actor enters first name.</li> <li>3. Use case resumes on step 6.</li> </ol> <p>5b. First name field contains number or special characters,</p> <ol style="list-style-type: none"> <li>1. System displays message “Invalid first name”.</li> <li>2. Actor enters first name.</li> <li>3. Use case resumes on step 6.</li> </ol> <p>6a. Last name field is empty,</p> <ol style="list-style-type: none"> <li>1. System displays message “Last name is required”.</li> <li>2. Actor enters last name.</li> <li>3. Use case resumes on step 7.</li> </ol> <p>6b. Last name field contains number or special characters,</p> <ol style="list-style-type: none"> <li>1. System displays message “Last name is required”.</li> <li>2. Actor enters last name.</li> <li>3. Use case resumes on step 7.</li> </ol> <p>7a. CNIC field is empty,</p>
--	--



	<ol style="list-style-type: none"> <li>1. System displays message “CNIC is required”.</li> <li>2. Actor enters CNIC.</li> <li>3. Use case resumes on step 8.</li> </ol> <p>7b. CNIC field contains alphabet or special characters,</p> <ol style="list-style-type: none"> <li>1. System displays message “CNIC is invalid”.</li> <li>2. Actor enters CNIC.</li> <li>3. Use case resumes on step 8.</li> </ol> <p>7c. CNIC length is not 13,</p> <ol style="list-style-type: none"> <li>1. System displays message “CNIC is invalid”.</li> <li>2. Actor enters CNIC.</li> <li>3. Use case resumes on step 8.</li> </ol> <p>8a. Mobile number field is empty,</p> <ol style="list-style-type: none"> <li>1. System displays message “Mobile number is required”.</li> <li>2. Actor enters mobile number.</li> <li>3. Use case resumes on step 9.</li> </ol> <p>8b. Mobile number field contains alphabet or special characters,</p> <ol style="list-style-type: none"> <li>1. System displays message “Mobile number is invalid”.</li> <li>2. Actor enters mobile number.</li> <li>3. Use case resumes on step 9.</li> </ol> <p>8c. Mobile number length is not 11,</p> <ol style="list-style-type: none"> <li>1. System displays message “Mobile number is invalid”.</li> <li>2. Actor enters mobile number.</li> <li>3. Use case resumes on step 9.</li> </ol> <p>9. Address field is empty,</p> <ol style="list-style-type: none"> <li>1. System displays message “Address is required”.</li> <li>2. Actor enters address.</li> <li>3. Use case resumes on step 10.</li> </ol> <p>10. Email address format is invalid,</p> <ol style="list-style-type: none"> <li>1. System displays message “Email address is invalid”.</li> <li>2. Actor enters email address.</li> <li>3. Use case resumes on step 11.</li> </ol> <p>11a. Emergency number field contains alphabet or special characters,</p> <ol style="list-style-type: none"> <li>1. System displays message “Emergency number is invalid”.</li> <li>2. Actor enters emergency number.</li> <li>3. Use case resumes on step 12.</li> </ol> <p>11b. Emergency number length is not 11,</p> <ol style="list-style-type: none"> <li>1. System displays message “Emergency number is invalid”.</li> <li>2. Actor enters emergency number.</li> <li>3. Use case resumes on step 12.</li> </ol> <p>12. Department is not selected,</p> <ol style="list-style-type: none"> <li>1. System displays message “Department is required”.</li> <li>2. Actor enters department.</li> <li>3. Use case resumes on step 13.</li> </ol> <p>13a. Salary field is empty,</p> <ol style="list-style-type: none"> <li>1. System displays message “Salary is required”.</li> <li>2. Actor enters salary.</li> <li>3. Use case resumes on step 14.</li> </ol> <p>13b. Salary field contains alphabet or special characters,</p> <ol style="list-style-type: none"> <li>1. System displays message “Salary is invalid”.</li> <li>2. Actor enters salary.</li> <li>3. Use case resumes on step 14.</li> </ol> <p><b>Remove Employee</b></p>
--	--

	3. Selected account has entries in ledger, <ol style="list-style-type: none"> <li>1. System displays message “This employee can’t be deleted”.</li> <li>2. Use case ends here.</li> </ol>
<b>Includes:</b>	N.A
<b>Frequency of Use:</b>	Could be nearly continuous.
<b>Special Requirements:</b>	N.A
<b>Assumptions:</b>	N.A
<b>Notes and Issues:</b>	N.A

**Table 22(3.5): UC-18 Manage Salaries**

<b>Use Case ID:</b>	UC-18		
<b>Use Case Name:</b>	Manage Salary		
<b>Created By:</b>	Abdul Raheem	<b>Last Updated By:</b>	Abdul Raheem
<b>Date Created:</b>	July 03, 2022	<b>Last Revision Date:</b>	Oct 10, 2022
<b>Actors:</b>	HR Manager		
<b>Description:</b>	Actor would be able to generate and pay salaries.		
<b>Trigger:</b>	Actor navigates to Salary Management.		
<b>Preconditions:</b>	Actor is authorized to the application.		
<b>Post conditions:</b>	<b>Generate salaries</b> <ol style="list-style-type: none"> <li>1. System display pending salaries of all employees.</li> <li>2. System display message “Salaries Generated”.</li> </ol> <b>Pay salary to all</b> <ol style="list-style-type: none"> <li>1. System records new ledgers entries for employee account debit against salary.</li> <li>2. System records new ledgers entries for cash account credit against salary.</li> <li>3. System updates all employees’ salary status.</li> <li>4. System display message “Salary paid to all employees”.</li> </ol> <b>Pay salary to one</b> <ol style="list-style-type: none"> <li>1. System records new ledger entry for employee account debit against salary.</li> <li>2. System records new ledger entry for cash account credit against salary.</li> <li>3. System updates employee salary status.</li> <li>4. System display message “Salary paid”.</li> </ol>		
<b>Normal Flow:</b>	<b>Generate salaries</b> <ol style="list-style-type: none"> <li>1. Actor navigates to salary management.</li> <li>2. Actor sends a request to generate salaries.</li> <li>3. System fetches salary details.</li> <li>4. System displays pending salaries of all employees.</li> <li>5. System displays message “Salaries Generated”.</li> </ol> <b>Pay salary to all</b> <ol style="list-style-type: none"> <li>1. Actor navigates to salary management.</li> <li>2. Actor sends a request to pay all.</li> <li>3. System records new ledger entries for employee account debit against salary.</li> <li>4. System records new ledger entries for cash account credit against salary.</li> </ol>		

	<ol style="list-style-type: none"> <li>5. System updates all employees' salary status.</li> <li>6. System displays message "Salary paid to all employees".</li> </ol> <p><b>Pay salary to one</b></p> <ol style="list-style-type: none"> <li>1. Actor navigates to salary management.</li> <li>2. Actor sends a request to pay salary of single employee.</li> <li>3. System records new ledger entry for employee account debit against salary.</li> <li>4. System records new ledger entry for cash account credit against salary.</li> <li>5. System updates employee salary status.</li> <li>6. System displays message "Salary paid".</li> </ol>
<b>/Alternative Flows:</b>	N.A
<b>Exceptions:</b>	<p><b>Generate salaries</b></p> <ol style="list-style-type: none"> <li>3. System failed to fetch salary details. <ol style="list-style-type: none"> <li>1. System tries again to fetch salary details.</li> <li>2. Use case resumes at step 4.</li> </ol> </li> </ol> <p><b>Pay salary to all</b></p> <ol style="list-style-type: none"> <li>3. System failed to add new ledger entry for employee accounts <ol style="list-style-type: none"> <li>1. System tries again to add new ledger entry for employee accounts.</li> <li>2. Use case resumes at step 4.</li> </ol> </li> <li>4. System failed to add new ledger entry for cash account <ol style="list-style-type: none"> <li>1. System tries again to add new ledger entry for cash account.</li> <li>2. Use case resumes at step 5.</li> </ol> </li> <li>5. System failed to update salary status <ol style="list-style-type: none"> <li>1. System tries again to update salary status.</li> <li>2. Use case resumes at step 6.</li> </ol> </li> </ol> <p><b>Pay salary to one</b></p> <ol style="list-style-type: none"> <li>3. System failed to add new ledger entry for employee account <ol style="list-style-type: none"> <li>1. System tries again to add new ledger entry for employee account.</li> <li>2. Use case resumes at step 4.</li> </ol> </li> <li>4. System failed to add new ledger entry for cash account <ol style="list-style-type: none"> <li>1. System tries again to add new ledger entry for cash account.</li> <li>2. Use case resumes at step 5.</li> </ol> </li> <li>5. System failed to update salary status <ol style="list-style-type: none"> <li>1. System tries again to update salary status.</li> <li>2. Use case resumes at step 6.</li> </ol> </li> </ol>
<b>Includes:</b>	N.A
<b>Frequency of Use:</b>	Could be nearly continuous.
<b>Special Requirements:</b>	Internet Connection
<b>Assumptions:</b>	N.A
<b>Notes and Issues:</b>	N.A

**Table 23(3.5): UC-19 Manage Production**

<b>Use Case ID:</b>	UC19		
<b>Use Case Name:</b>	Manage Production		
<b>Created By:</b>	Abu-Bakar Rehan	<b>Last Updated By:</b>	Abdul Raheem
<b>Date Created:</b>	July 03, 2022	<b>Last Revision Date:</b>	Oct 10, 2022
<b>Actors:</b>	Production manager		
<b>Description:</b>	Actor would be able to add, view, update and finish production.		
<b>Trigger:</b>	Actor navigates to production management.		
<b>Preconditions:</b>	Actor is authorized to the application.		
<b>Post conditions:</b>	<p><b>Add Production</b></p> <ol style="list-style-type: none"> <li>1. System records new production request.</li> <li>2. System displays message “Production request initiated”</li> </ol> <p><b>Update Production</b></p> <ol style="list-style-type: none"> <li>1. System records new production update request.</li> <li>2. System displays message “Production update request initiated”</li> </ol> <p><b>Finish Production</b></p> <ol style="list-style-type: none"> <li>1. System adds new product(s) to records.</li> <li>2. System displays message “Product(s) added to inventory”</li> </ol>		
<b>Normal Flow:</b>	<p><b>Add Production</b></p> <ol style="list-style-type: none"> <li>1. Actor navigates to production management.</li> <li>2. Actor selects add production option.</li> <li>3. Actor selects product code.</li> <li>4. Actor enters quantity of product.</li> <li>5. Actor selects raw material.</li> <li>6. Actor enters quantity of raw material.</li> <li>7. Actor sends request to add item.</li> </ol> <p>Actor repeats step 5-7 until indicates done.</p> <ol style="list-style-type: none"> <li>8. Actor enters expense title.</li> <li>9. Actor enters expense cost.</li> <li>10. Actor sends request to add expense.</li> </ol> <p>Actor repeats step 8-10 until indicates done.</p> <ol style="list-style-type: none"> <li>11. Actor sends request to start production.</li> <li>12. System records new production request.</li> <li>13. System displays message “Production request initiated”</li> </ol> <p><b>Update Production</b></p> <ol style="list-style-type: none"> <li>1. Actor navigates to production management.</li> <li>2. Actor selects update production option.</li> <li>3. Actor selects product code.</li> <li>4. Actor enters quantity of product.</li> <li>5. Actor selects raw material.</li> <li>6. Actor selects quantity of raw material.</li> <li>7. Actor sends request to add item.</li> </ol> <p>Actor repeats step 5-7 until indicates done.</p> <ol style="list-style-type: none"> <li>8. Actor enters expense title.</li> <li>9. Actor enters expense cost.</li> <li>10. Actor sends request to add expense.</li> </ol> <p>Actor repeats step 8-10 until indicates done.</p> <ol style="list-style-type: none"> <li>11. Actor sends request to update production.</li> <li>12. System records new production update request.</li> <li>13. System displays message “Production update request initiated”</li> </ol>		

	<b>Finish Production</b> <ol style="list-style-type: none"> <li>1. Actor navigates to production management.</li> <li>2. Actor sends request to finish production.</li> <li>3. System adds new product(s) to records.</li> <li>4. System displays message “Product(s) added to inventory”</li> </ol>
<b>Alternative Flows:</b>	N.A
<b>Exceptions:</b>	<b>Add Production</b> <ol style="list-style-type: none"> <li>3. Product code is empty. <ol style="list-style-type: none"> <li>1. System displays message “Product code is required”</li> <li>2. Actor selects product code.</li> <li>3. Use case resumes on step 4.</li> </ol> </li> <li>4a. Product quantity is empty. <ol style="list-style-type: none"> <li>1. System displays message “Product quantity is required”</li> <li>2. Actor enters product quantity.</li> <li>3. Use case resumes on step 5.</li> </ol> </li> <li>4b. Product quantity field contains alphabet. <ol style="list-style-type: none"> <li>1. System displays message “Product quantity is invalid”</li> <li>2. Actor enters valid product quantity.</li> <li>3. Use case resumes on step 5.</li> </ol> </li> <li>4c. Product quantity is a negative number. <ol style="list-style-type: none"> <li>1. System displays message “Product quantity is invalid”.</li> <li>2. Actor enters valid product quantity.</li> <li>3. Use case resumes on step 5.</li> </ol> </li> <li>5. Raw material is empty. <ol style="list-style-type: none"> <li>1. System displays message “Raw material is required”.</li> <li>2. Actor selects raw material.</li> <li>3. Use case resumes on step 6.</li> </ol> </li> <li>6a. Raw material quantity is empty. <ol style="list-style-type: none"> <li>1. System displays message “Raw material quantity is required”.</li> <li>2. Actor enters Raw material quantity.</li> <li>3. Use case resumes on step 7.</li> </ol> </li> <li>6b. Raw material quantity field contains alphabet. <ol style="list-style-type: none"> <li>1. System displays message “Raw material quantity is invalid”.</li> <li>2. Actor enters valid raw material quantity.</li> <li>3. Use case resumes on step 7.</li> </ol> </li> <li>6c. Raw material quantity is non-positive. <ol style="list-style-type: none"> <li>1. System displays message “Raw material quantity is invalid”</li> <li>2. Actor enters valid raw material quantity.</li> <li>3. Use case resumes on step 7.</li> </ol> </li> <li>8. Expense title is empty. <ol style="list-style-type: none"> <li>1. System displays message “Expense title is required”</li> <li>2. Actor enters expense title.</li> <li>3. Use case resumes on step 9.</li> </ol> </li> <li>9a. Expense cost is empty. <ol style="list-style-type: none"> <li>1. System displays message “Expense cost is required”</li> <li>2. Actor enters valid expense cost.</li> <li>3. Use case resumes on step 10.</li> </ol> </li> <li>9b. Expense cost field contains alphabet. <ol style="list-style-type: none"> <li>1. System displays message “Expense cost is invalid”</li> <li>2. Actor enters valid expense cost.</li> <li>3. Use cases resume on step 10.</li> </ol> </li> <li>9c. Expense cost is a negative number.</li> </ol>

	<ol style="list-style-type: none"> <li>1. System displays message “Expense cost is invalid”</li> <li>2. Actor enters valid expense cost.</li> <li>3. Use case resumes on step 10.</li> </ol> <p>12. System failed to add new record.</p> <ol style="list-style-type: none"> <li>1. System tries again to add new record</li> <li>2. Use case resumes on step 13.</li> </ol> <p><b>Update Production</b></p> <p>3. Product code is empty.</p> <ol style="list-style-type: none"> <li>1. System displays message “Product code is required”</li> <li>2. Actor selects product code.</li> <li>3. Use case resumes on step 4.</li> </ol> <p>4a. Product quantity is empty.</p> <ol style="list-style-type: none"> <li>1. System displays message “Product quantity is required”</li> <li>2. Actor enters product quantity.</li> <li>3. Use case resumes on step 5.</li> </ol> <p>4b. Product quantity field contains alphabet.</p> <ol style="list-style-type: none"> <li>1. System displays message “Product quantity is invalid”</li> <li>2. Actor enters valid product quantity.</li> <li>3. Use case resumes on step 5.</li> </ol> <p>4c. Product quantity is a negative number.</p> <ol style="list-style-type: none"> <li>1. System displays message “Product quantity is invalid”</li> <li>2. Actor enters valid product quantity.</li> <li>3. Use case resumes on step 5.</li> </ol> <p>5. Raw material is empty.</p> <ol style="list-style-type: none"> <li>1. System displays message “Raw material is required”</li> <li>2. Actor selects raw material.</li> <li>3. Use case resumes on step 6.</li> </ol> <p>6a. Raw material quantity is empty.</p> <ol style="list-style-type: none"> <li>1. System displays message “Raw material quantity is required”</li> <li>2. Actor enters Raw material quantity.</li> <li>3. Use case resumes on step 7.</li> </ol> <p>6b. Raw material quantity field contains alphabet.</p> <ol style="list-style-type: none"> <li>1. System displays message “Raw material quantity is invalid”</li> <li>2. Actor enters valid raw material quantity.</li> <li>3. Use case resumes on step 7.</li> </ol> <p>6c. Raw material quantity is a negative number.</p> <ol style="list-style-type: none"> <li>1. System displays message “Raw material quantity is invalid”</li> <li>2. Actor enters valid raw material quantity.</li> <li>3. Use case resumes on step 7.</li> </ol> <p>8. Expense title is empty.</p> <ol style="list-style-type: none"> <li>1. System displays message “Expense title is required”</li> <li>2. Actor enters expense title.</li> <li>3. Use case resumes on step 9.</li> </ol> <p>9a. Expense cost is empty.</p> <ol style="list-style-type: none"> <li>1. System displays message “Expense cost is required”</li> <li>2. Actor enters valid expense cost.</li> <li>3. Use case resumes on step 10.</li> </ol> <p>9b. Expense cost field contains alphabet.</p> <ol style="list-style-type: none"> <li>1. System displays message “Expense cost is invalid”</li> <li>2. Actor enters valid expense cost.</li> <li>3. Use case resume on step 10.</li> </ol> <p>9c. Expense cost is a negative number.</p> <ol style="list-style-type: none"> <li>1. System displays message “Expense cost is invalid”</li> </ol>
--	---

	<ol style="list-style-type: none"> <li>2. Actor enters valid expense cost.</li> <li>3. Use case resumes on step 10.</li> </ol> <ol style="list-style-type: none"> <li>12. System failed to add new record. <ol style="list-style-type: none"> <li>1. System tries again to add new record</li> <li>2. Use case resumes on step 13.</li> </ol> </li> </ol> <p><b>Finish Production</b></p> <ol style="list-style-type: none"> <li>3. System failed to add new record. <ol style="list-style-type: none"> <li>1. System tries again to add new record</li> <li>2. Use case resumes on step 4.</li> </ol> </li> </ol>
<b>Includes:</b>	N.A
<b>Frequency of Use:</b>	Could be nearly continuous.
<b>Special Requirements:</b>	Internet Connection
<b>Assumptions:</b>	N.A
<b>Notes and Issues:</b>	N.A

### 3.6 User interfaces (mock screens)

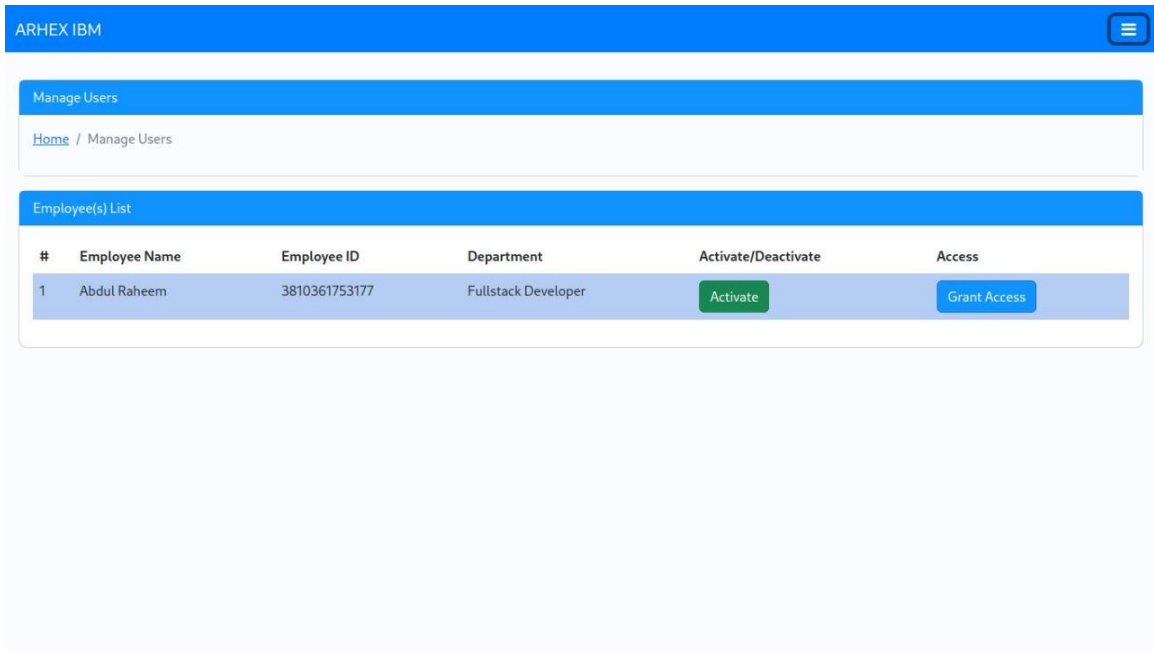


Figure 2(3.6): Manage User

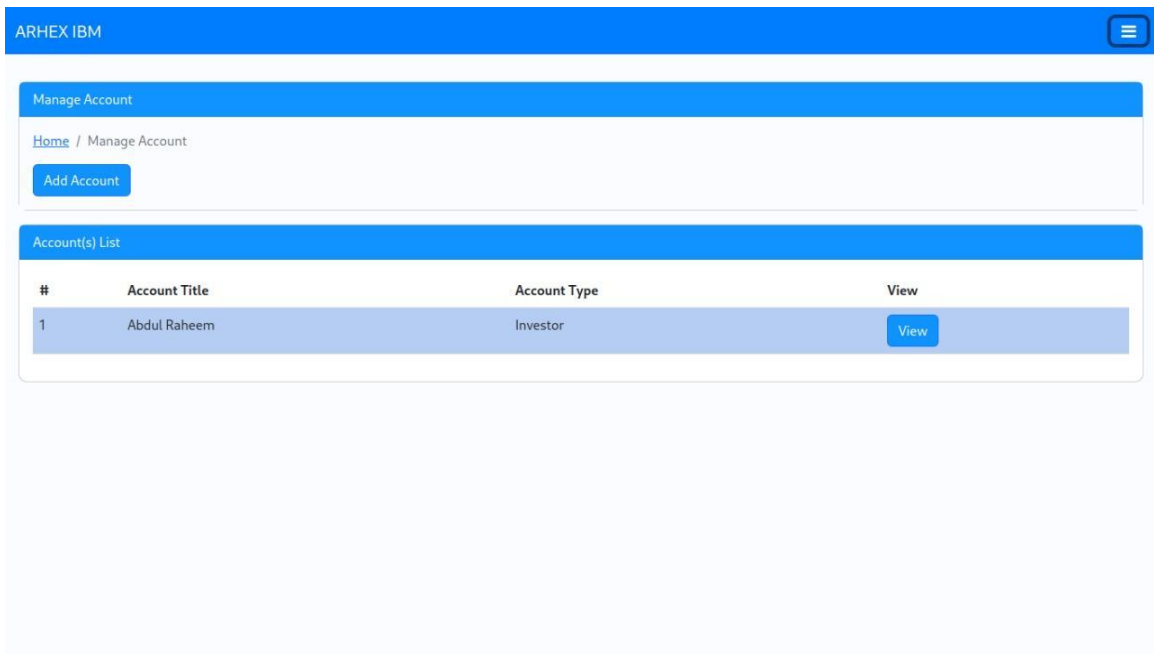



Figure 3(3.6): Manage Account



Account Details



Abdul Raheem

Investor

Update

Deactivate

Account Information

Type	Investor
CNIC	3810361753177
Email Address	Null
Mobile	03134386826
Emergency Contact	Null
Address	House No. 69, Farooq e Azam street, High Court Society, Johar Town, Lahore

Ledgers

Description	Date	Account	Dr	Cr
Capital Ammount added	Dec 23, 2022	Cash	1500	0
Capital Ammount added	Dec 23, 2022	Capital	0	1500

Total: 0

**Figure 4(3.6): Account Details**

ARHEX IBM

Manage Capitals

[Home](#) / Manage Capitals
 

Add Capital

Capitals List

#	Account Title	Amount	Delete
---	---------------	--------	--------

**Figure 5(3.6): Manage Capital**

ARHEX IBM

Generate Reports

[Home](#) / Generate Reports

Generate new report

Report Type  
 Select Report Type

Start Date  
 mm/dd/yyyy

End Date  
 mm/dd/yyyy

Generate

**Figure 6(3.6): Generate Report**

ARHEX IBM

Manage Payments

[Home](#) / Manage Payments

Payment In

Invoice  
 Select Sale Invoice

Receivable  
 Receivable Amount

Amount  
 Enter Amount

Update

Payment Out

Invoice  
 Select Purchase Invoice

Payable  
 Payable Amount

Amount  
 Enter Amount

Update

**Figure 7(3.6): Manage Payments**

ARHEX IBM

Manage Employee Requests

Home / Manage Employee Requests

Request(s) List

#	Employee Name	Department	Request Type	View
1	Abdul Raheem	Fullstack Developer	Update	View

**Figure 8(3.6): Manage Employee Request**

ARHEX IBM

Manage Purchase

Home / Manage Purchase

Add Purchase

Purchase(s) List

#	Invoice #	Vendor	Date	View/Return
1	312	Abdul Raheem	Jan 2, 2022	View

ARHEX IBM

Manage Backup

Home / Manage Backup

Create Backup

Date

Today's Date

Backup

Restore Backup

Restore File

Choose File

No file chosen

Restore

ARHEX IBM

Manage Employee


[Home](#) / Manage Employee

Add Employee

Employee(s) List

#	Employee Name	Employee ID	View
1	Abdul Raheem	3810361753177	<a href="#">View</a>

Employee Details



Abdul Raheem  
Investor

[Update](#) [Deactivate](#)

Employee Information

Type	Investor
CNIC	3810361753177
Email Address	Nil
Mobile	03134386826
Emergency Contact	Nil
Address	House No. 69, Farooq e Azam street, High Court Society, Johar Town, Lahore

Ledgers

Description	Date	Employee	Dr	Cr
Capital Ammount added	Dec 23, 2022	Cash	1500	0
Capital Ammount added	Dec 23, 2022	Capital	0	1500
Total: 0				

ARHEX IBM

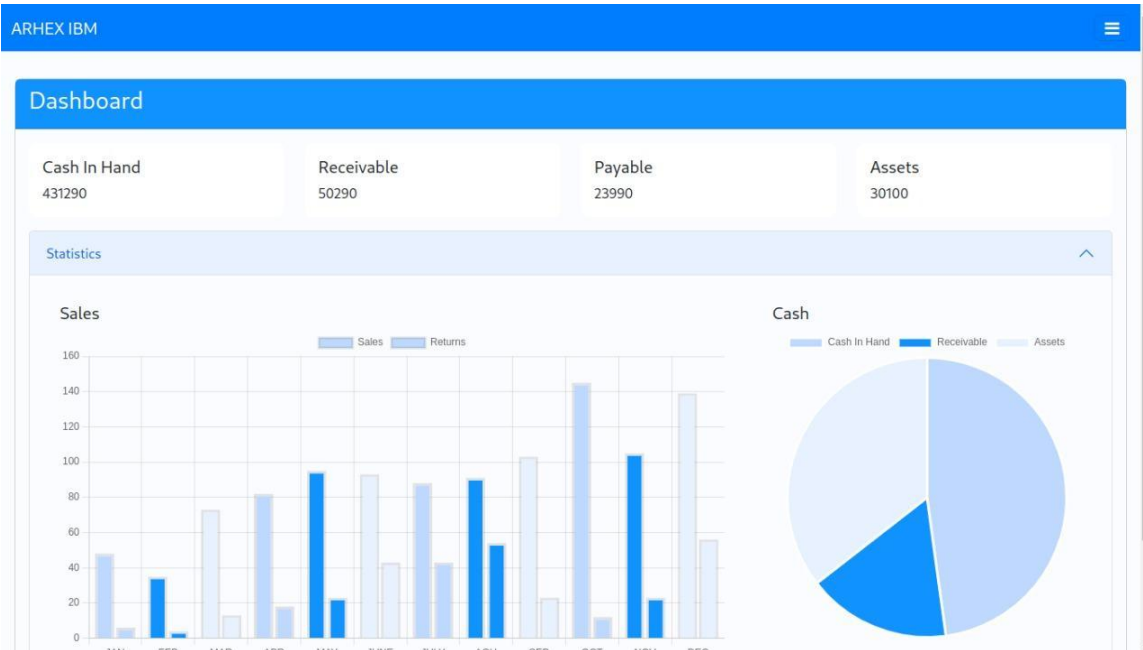
Manage Productions

[Home](#) / Manage Productions

Start New Production

Production(s) In Progress

#	PID	Product Code	Quantity	Date	View/Update
1	24	RAN-U-786	3	Jan 2, 2022	<a href="#">View</a>



View Production

Update Production

Add Raw Material

Raw Material

Select Item

Available Quantity

Available Quantity

Quantity

Enter Quantity

Add

Add Expense

Expense Title

Expense Title

Expense Cost

Expense Cost

Add

Update

Finish Production

PID: 24

Customer Details

Product Code: RAN-U-786

Quantity: 30

Items

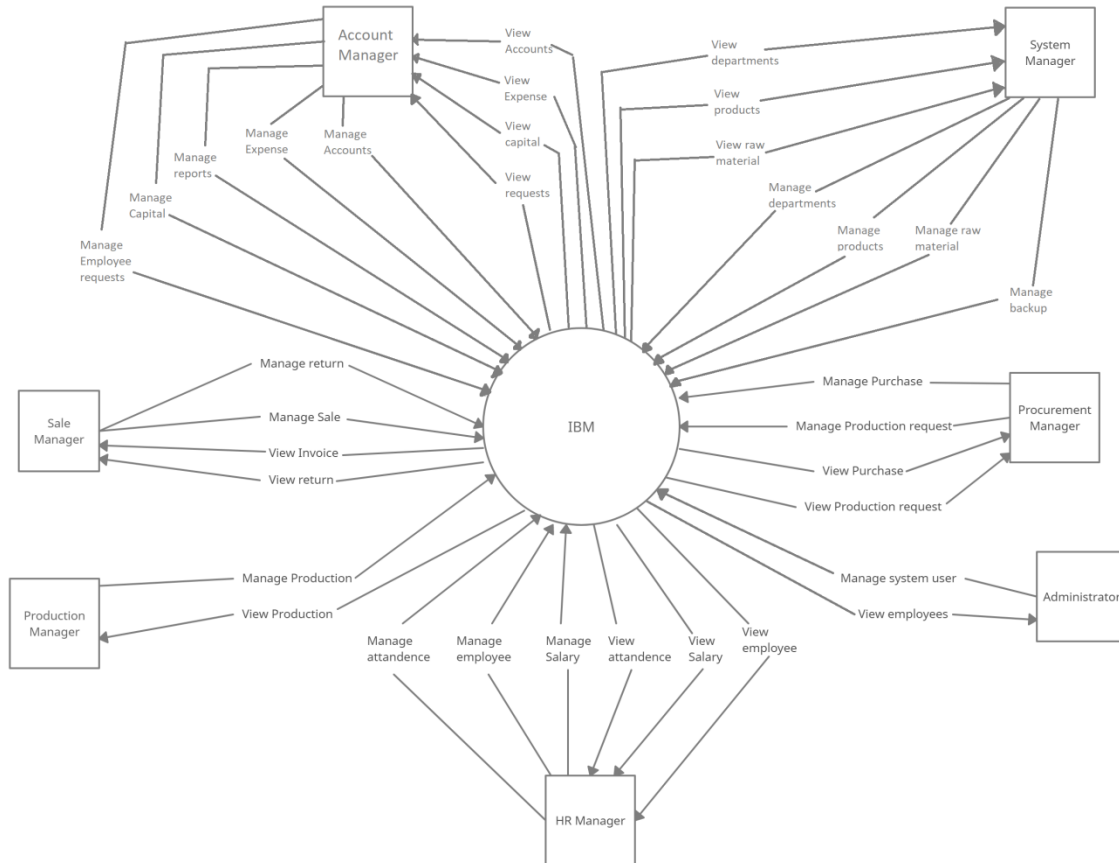
#	Item Name	QTY	Delete
1	Paint (White)	3	Delete
2	Paint (Black)	2	Delete
3	Spray	1	Delete

Other Expense

#	Expense Title	Cost	Delete
1	Freight Charges	300	Delete

## 4. DATA FLOW DIAGRAM (OPTIONAL)

### 4.1 Data Flow Diagram Level 0



**Figure 19(4.1): DFD Level 0 Diagram**

## 4.2 Data Flow Diagram Level 1

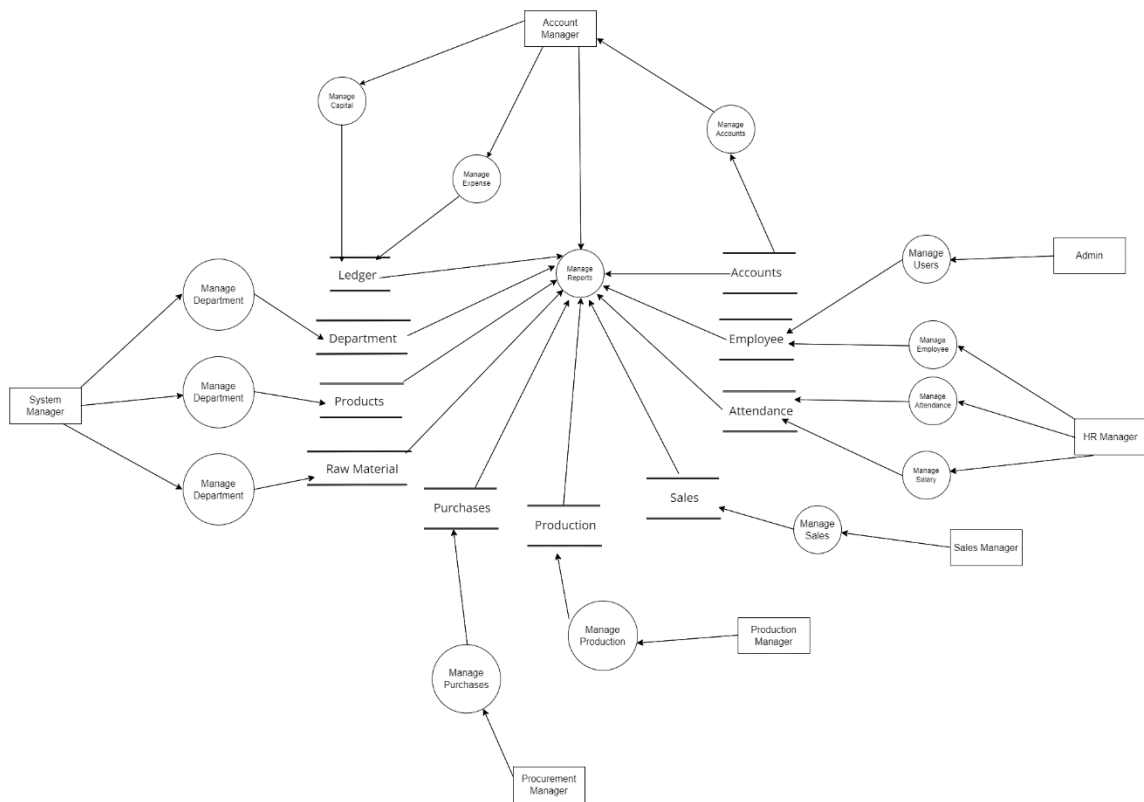


Figure 20(4.2): DFD Level 1 Diagram

## 5. SYSTEM DESIGN

---

### 5.1 System Architecture Diagram

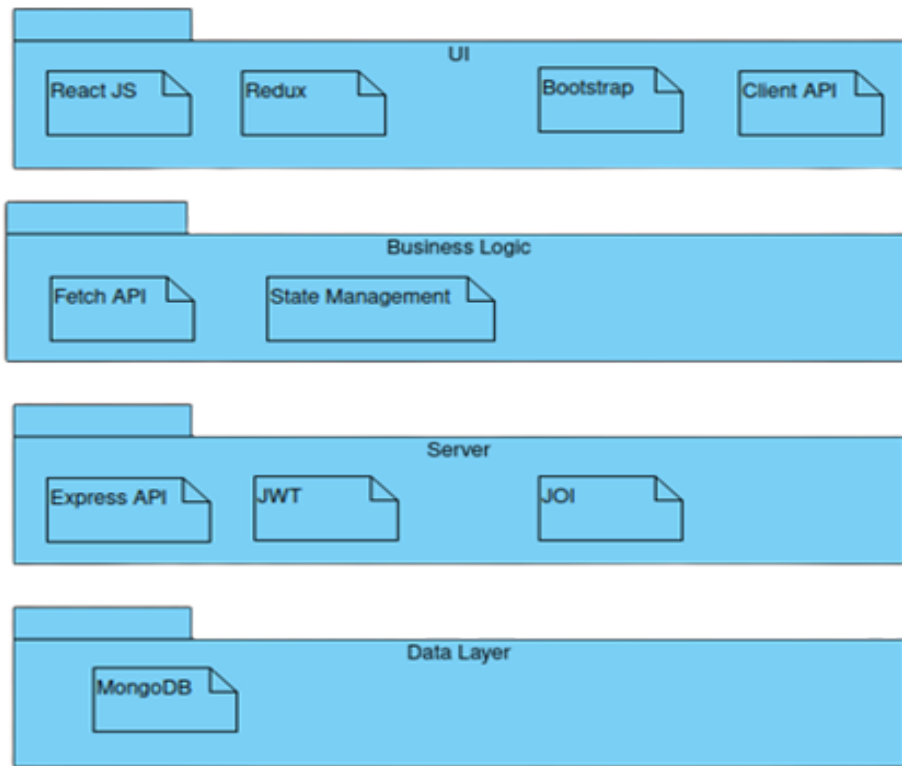
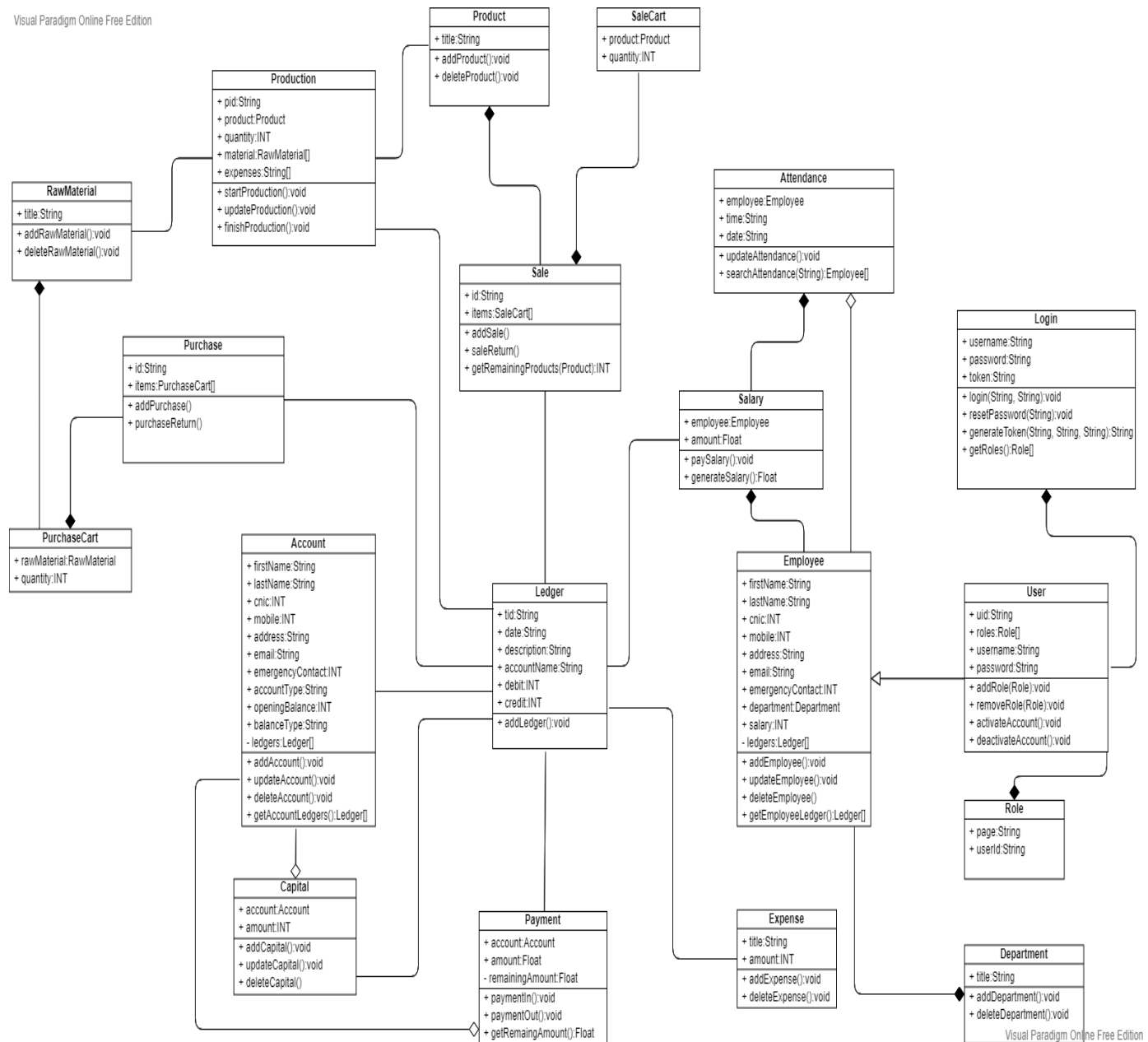


Figure 21(5.1): System Architecture



## 5.2 Class Diagram

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

## 5.3 Sequence Diagrams

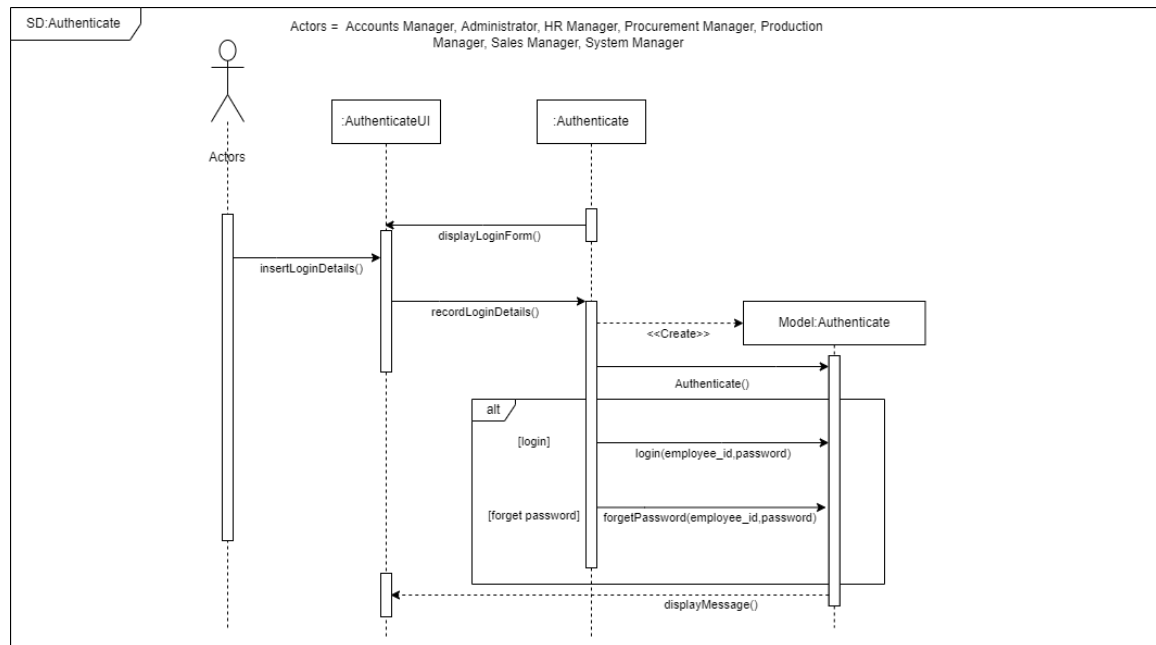


Figure 23(5.3): Authenticate User

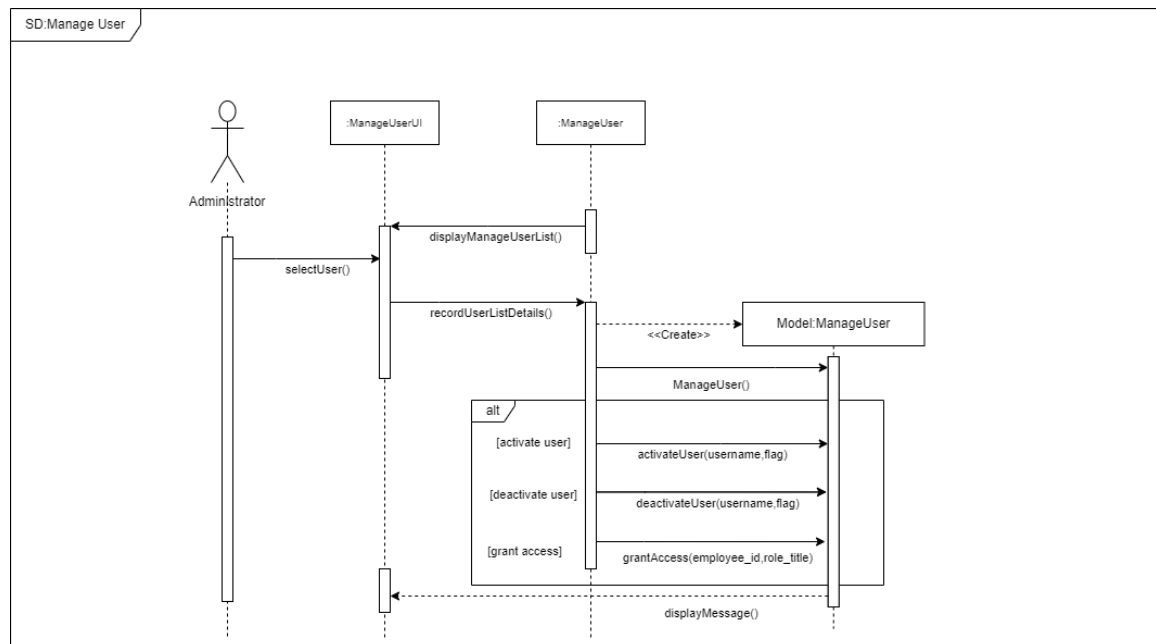


Figure 24(5.3): Manage User

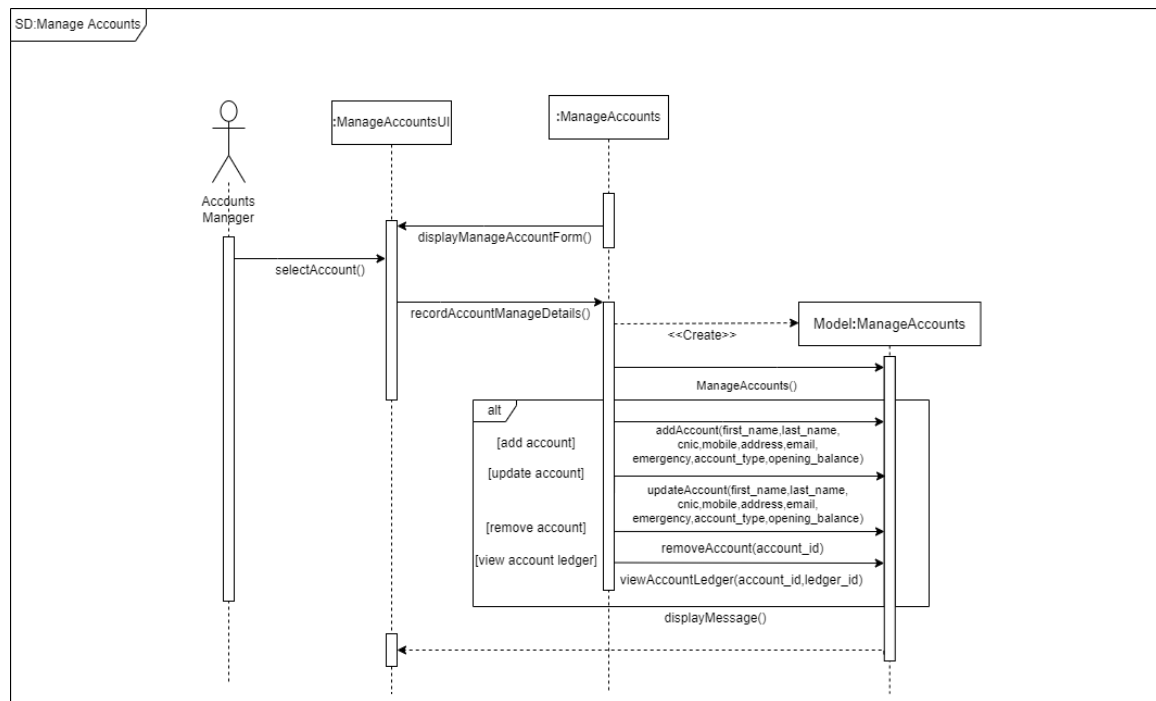
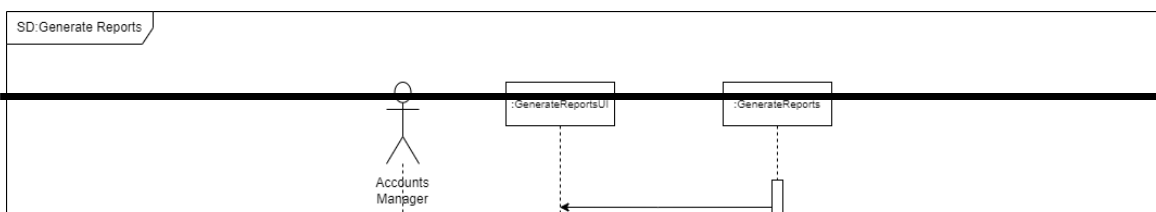
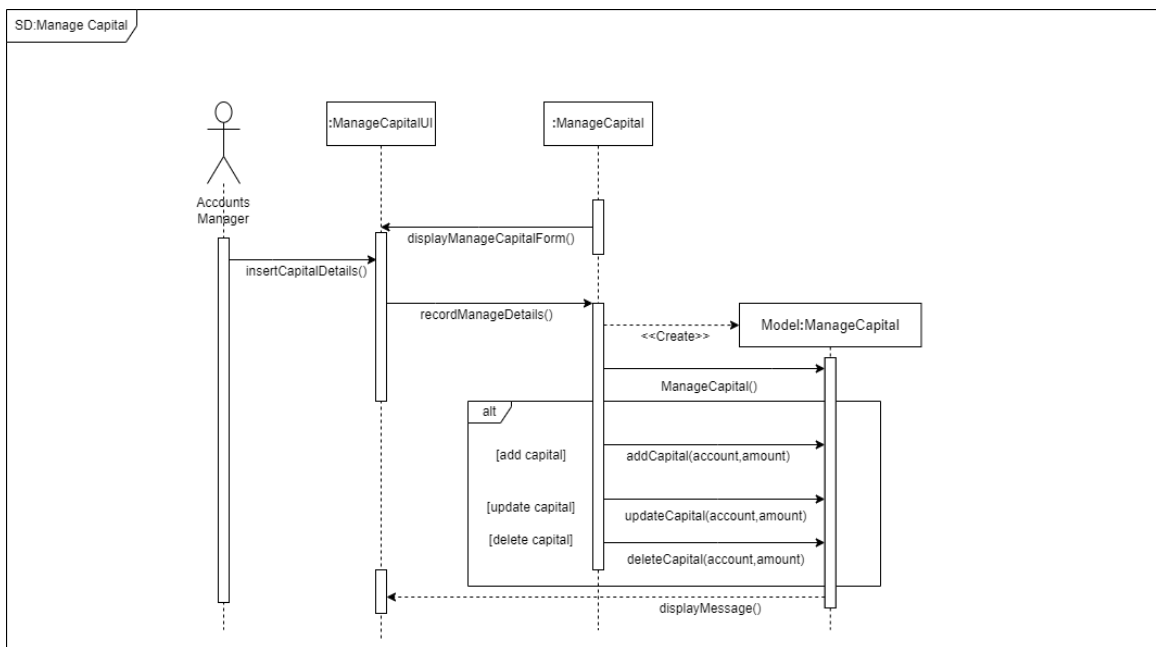
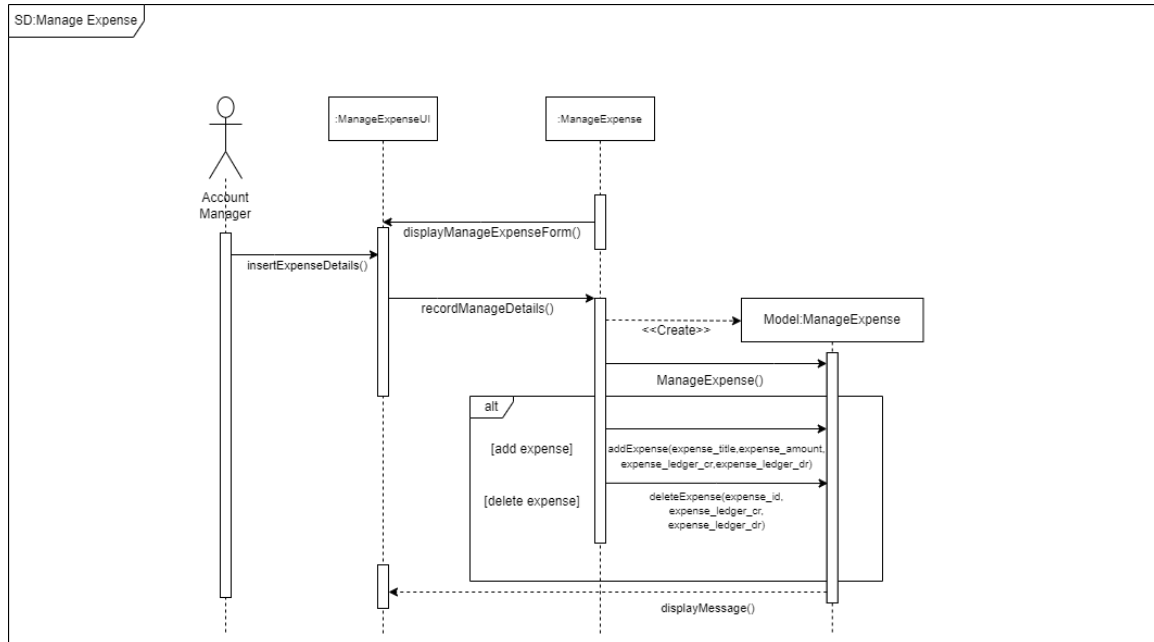


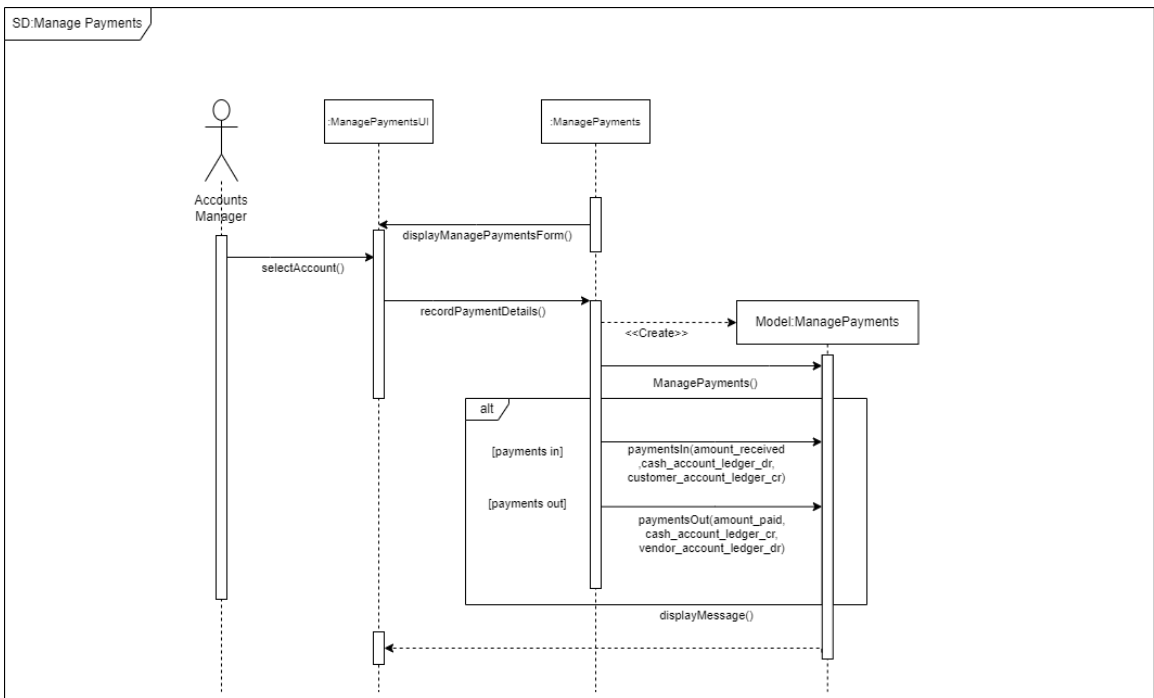
Figure 25(5.3): Manage Accounts



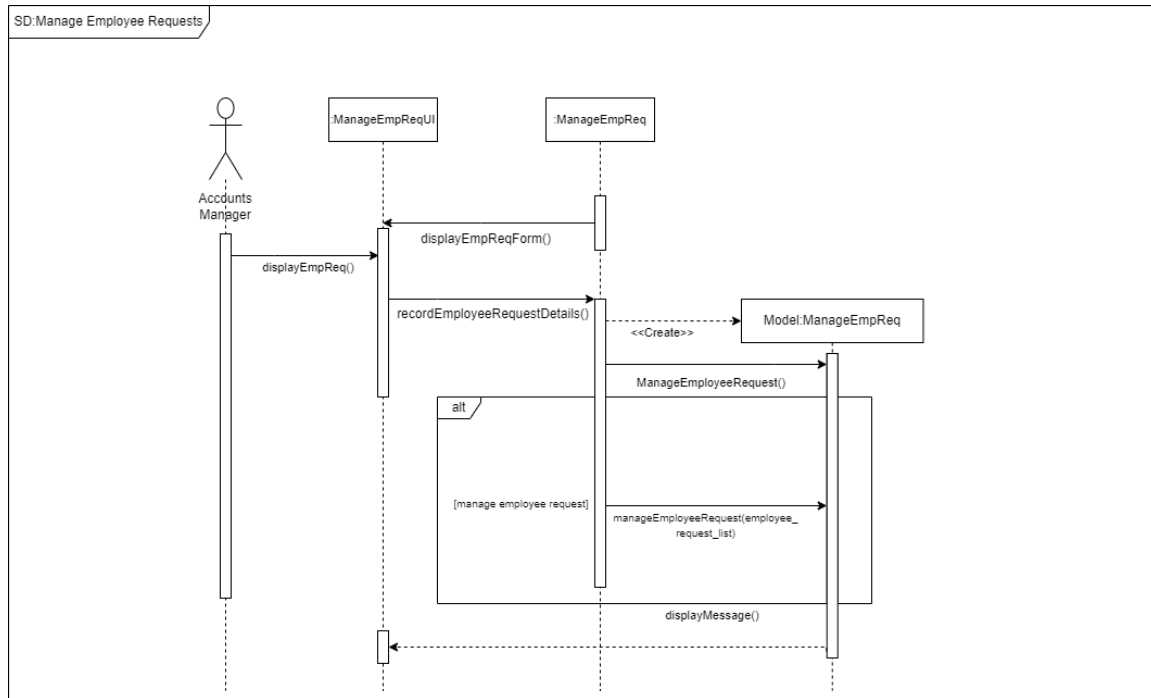
**Figure 27(5.3): Generate Report**



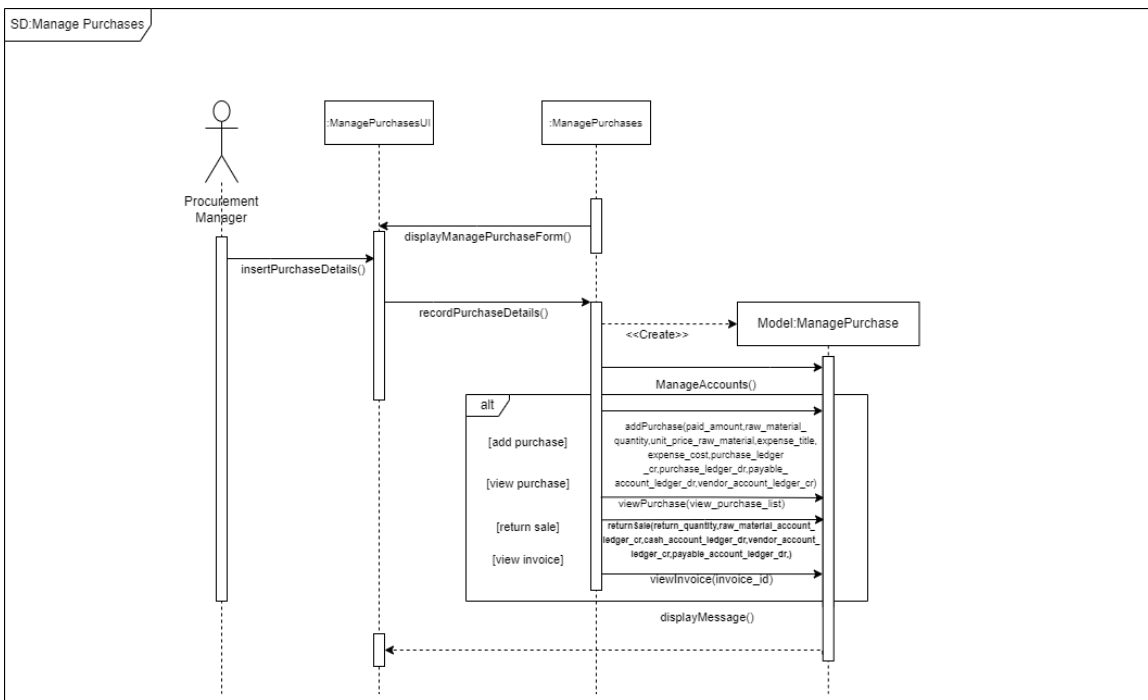
**Figure 28(5.3): Manage Expense**

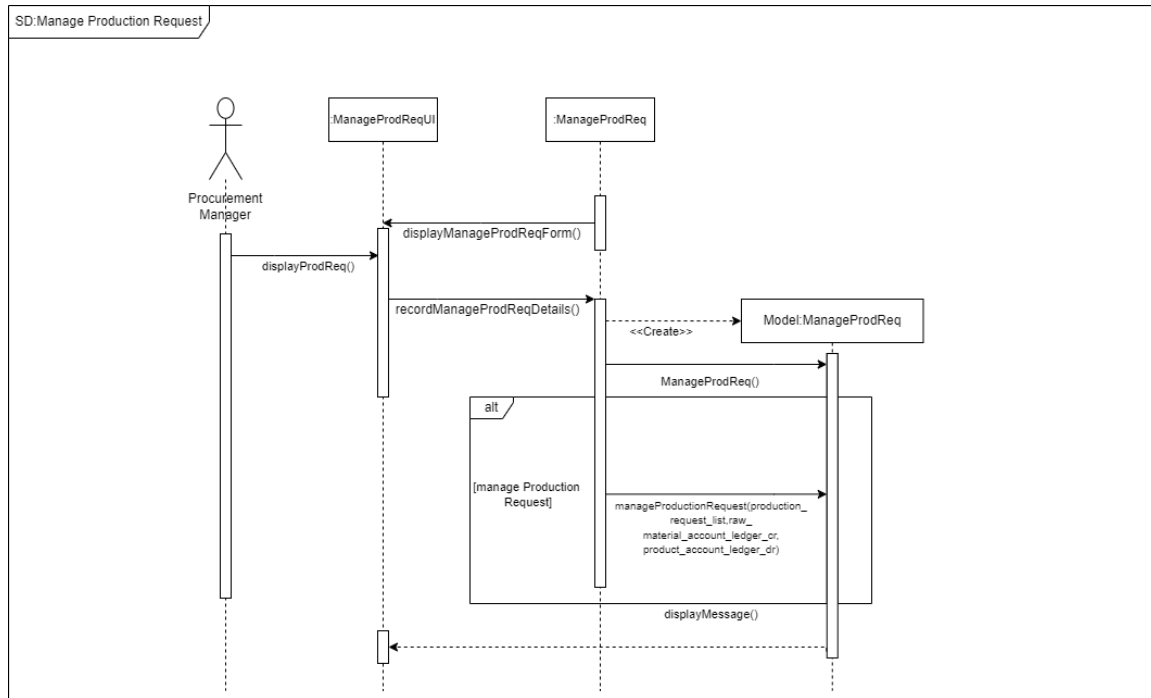


**Figure 29(5.3): Manage Payments**

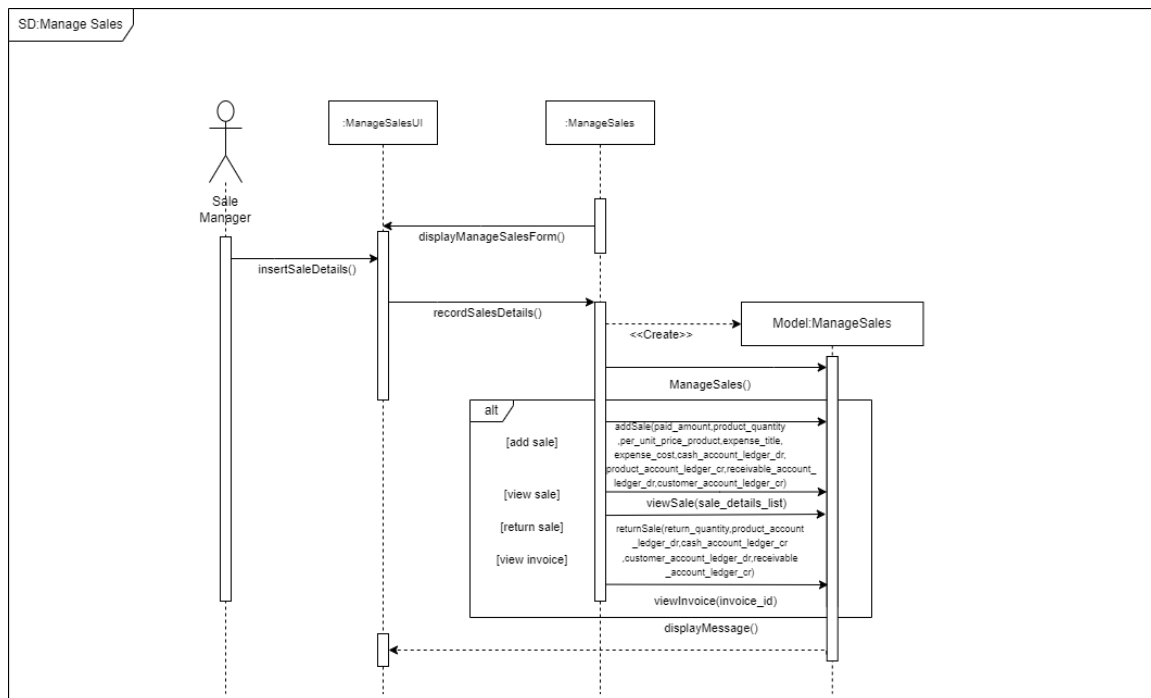


**Figure 30(5.3): Manage Employee Requests**

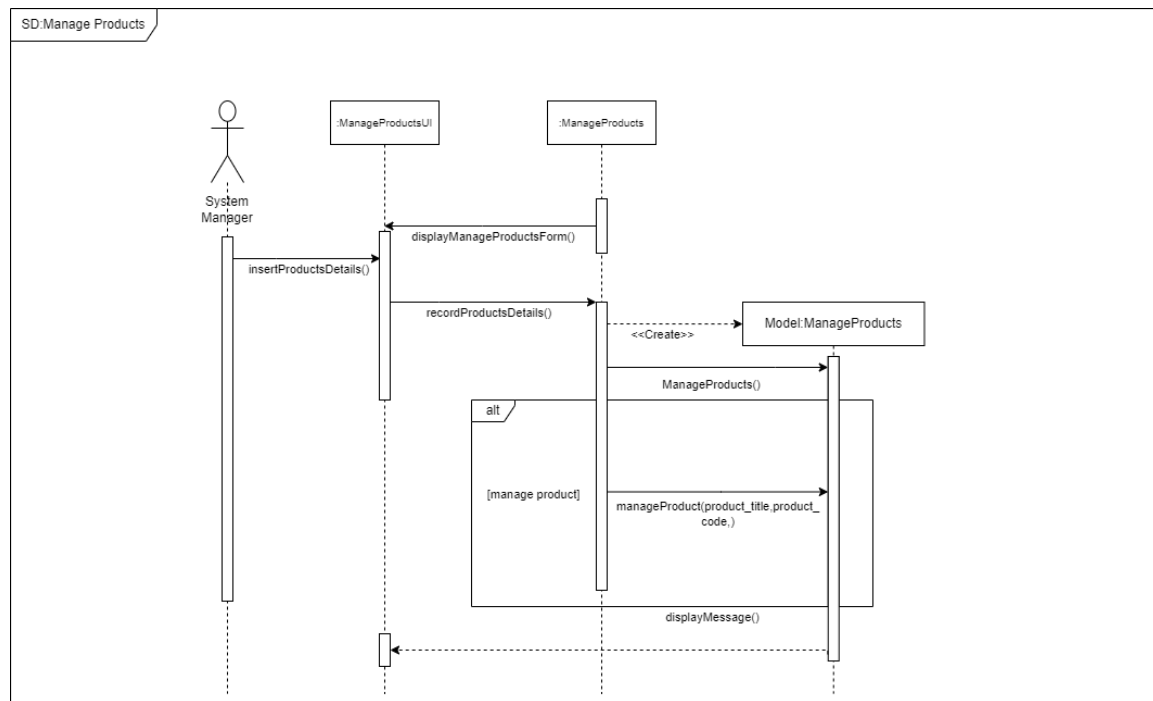




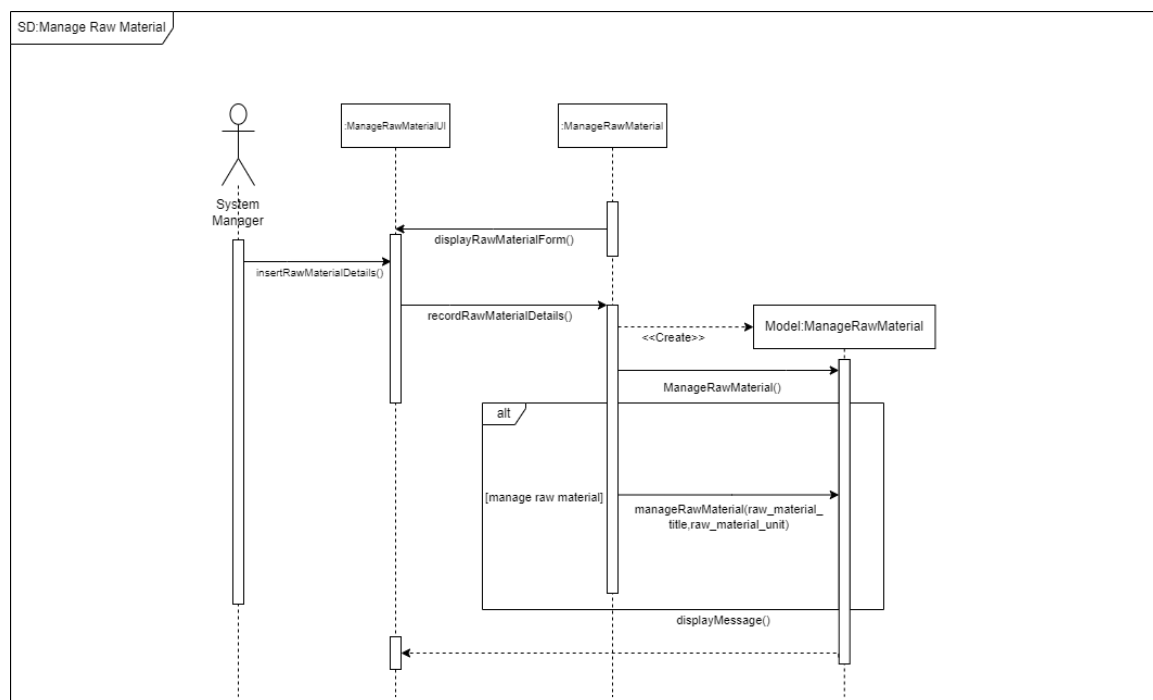
**Figure 32(5.3): Manage Production Requests**



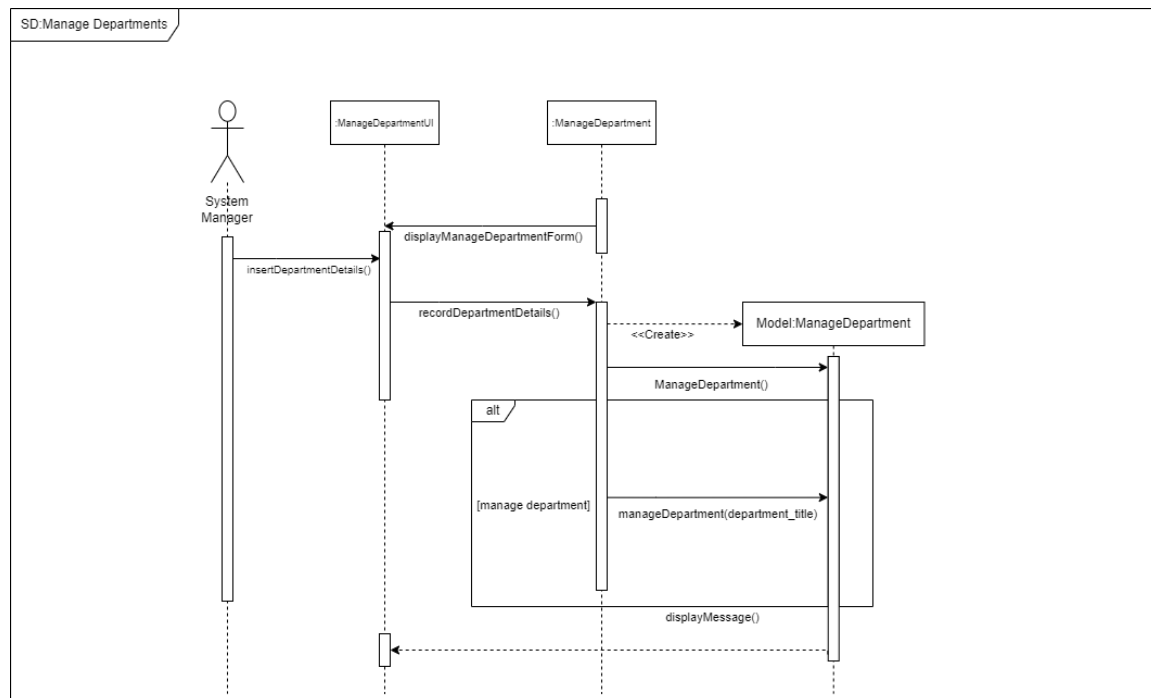
**Figure 33(5.3): Manage Sales**



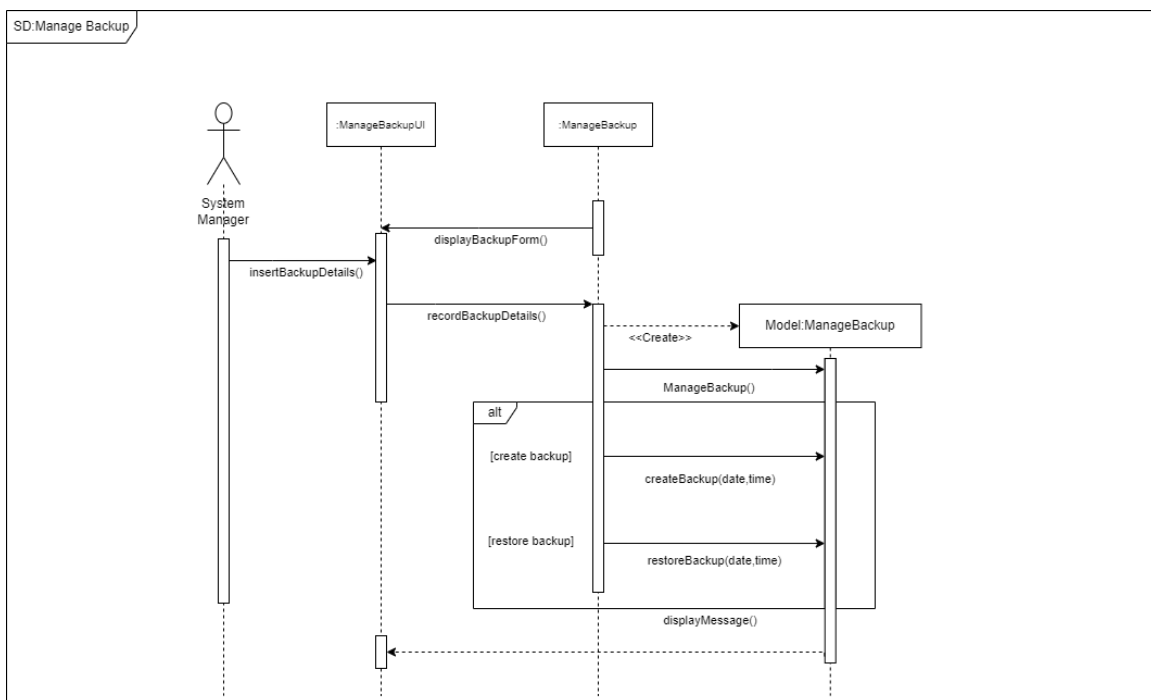
**Figure 34(5.3): Manage Products**



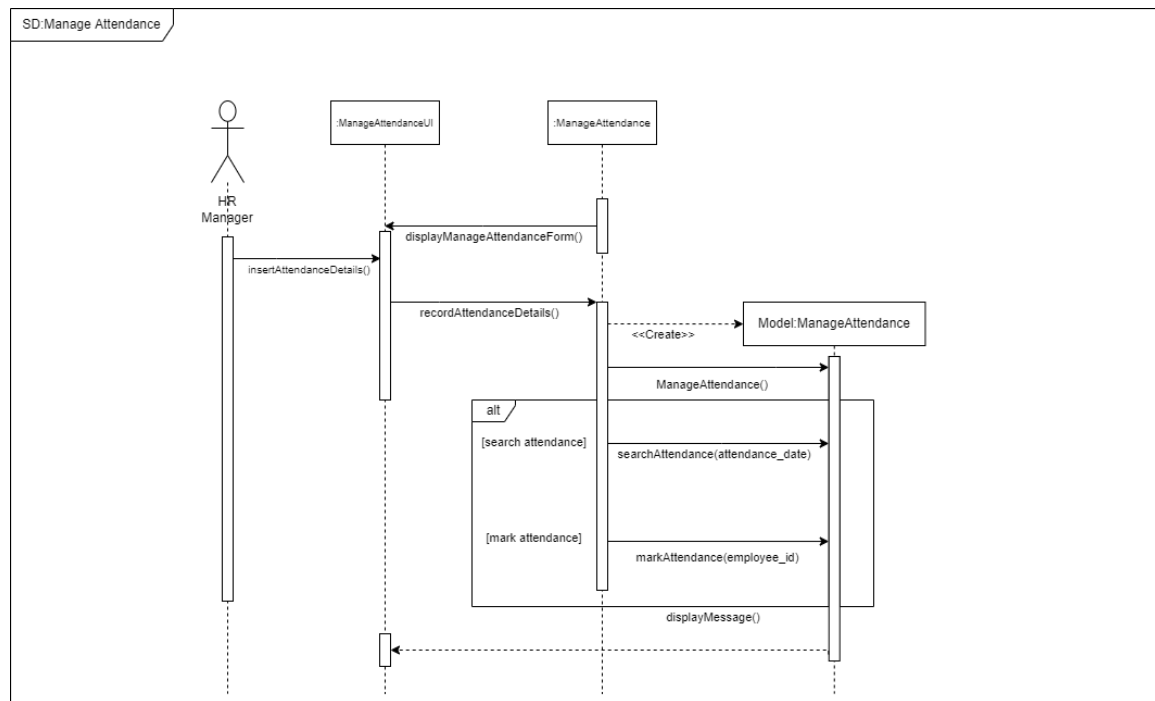
**Figure 35(5.3): Manage Raw Material**



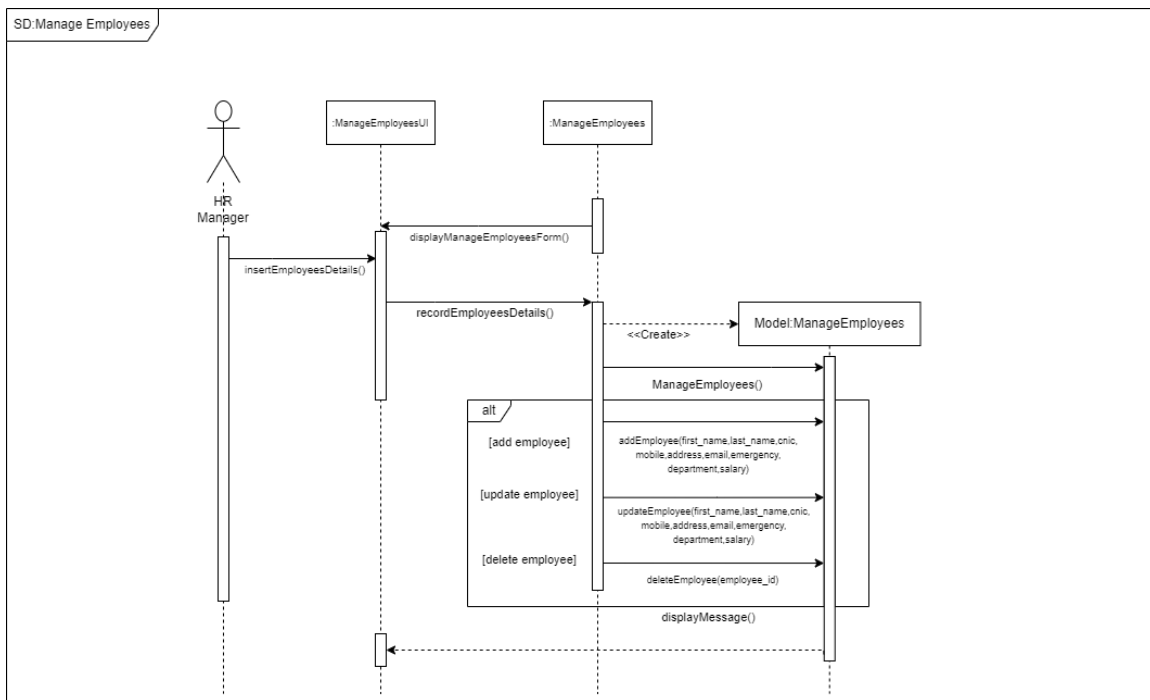
**Figure 36(5.3): Manage Departments**

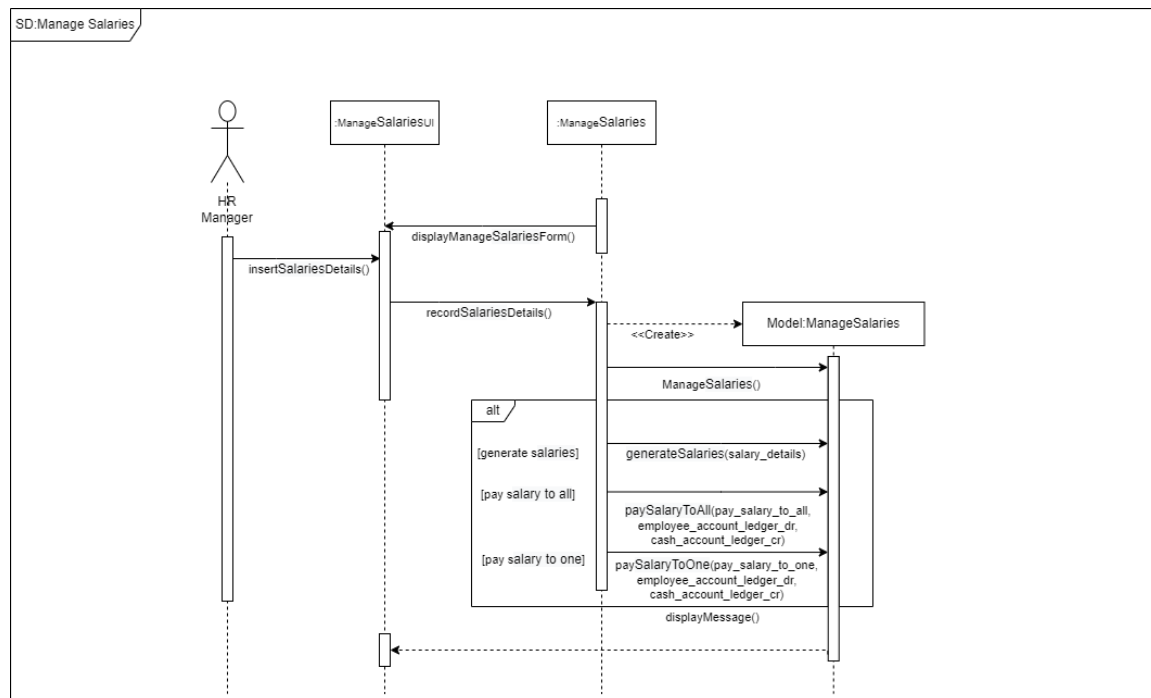




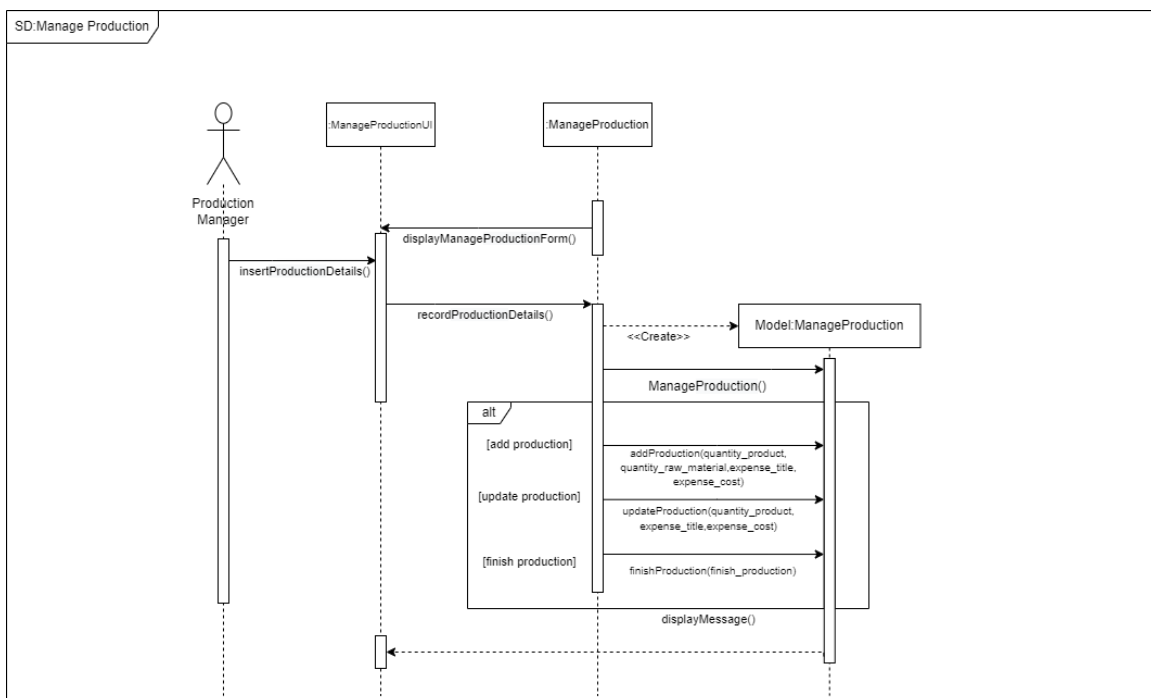


**Figure 38(5.3): Manage Attendance**





**Figure 40(5.3): Manage Salaries**



## 5.4 Collaboration Diagrams

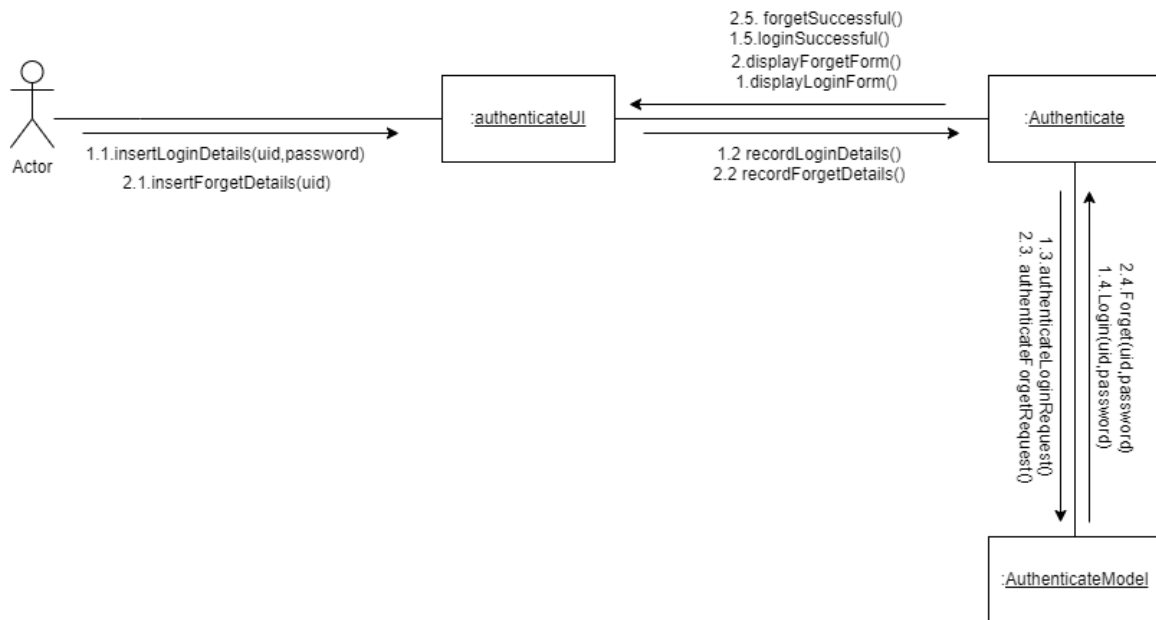
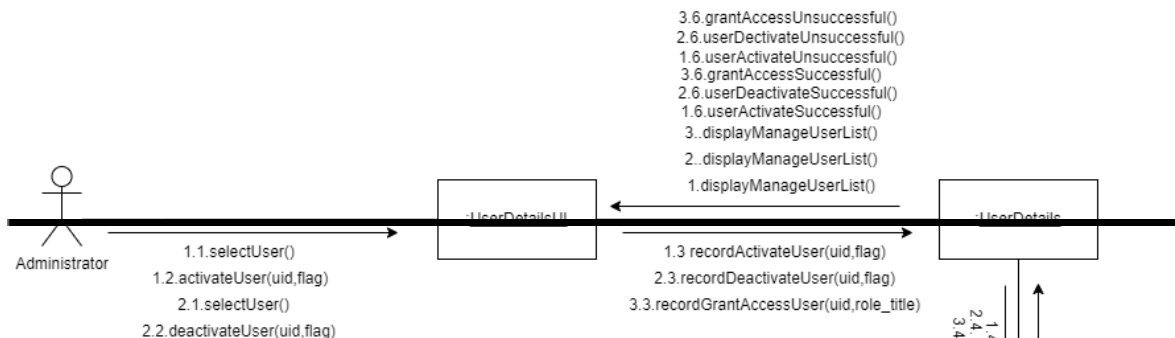
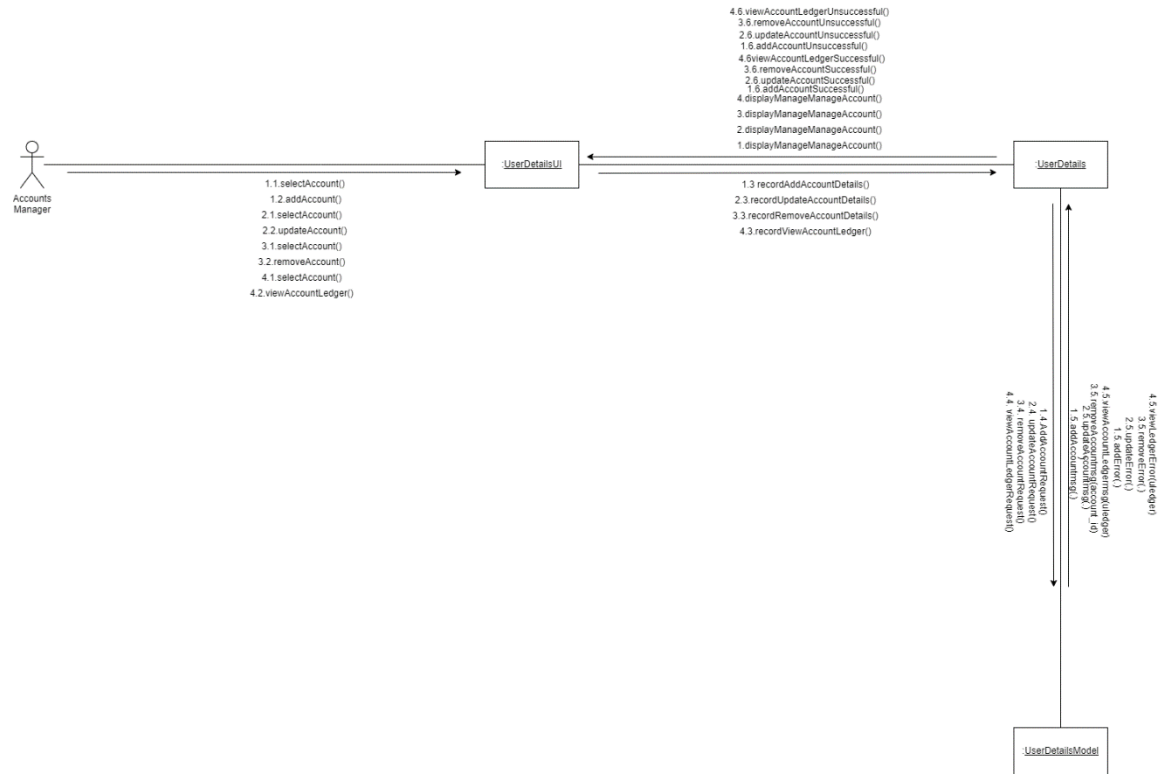


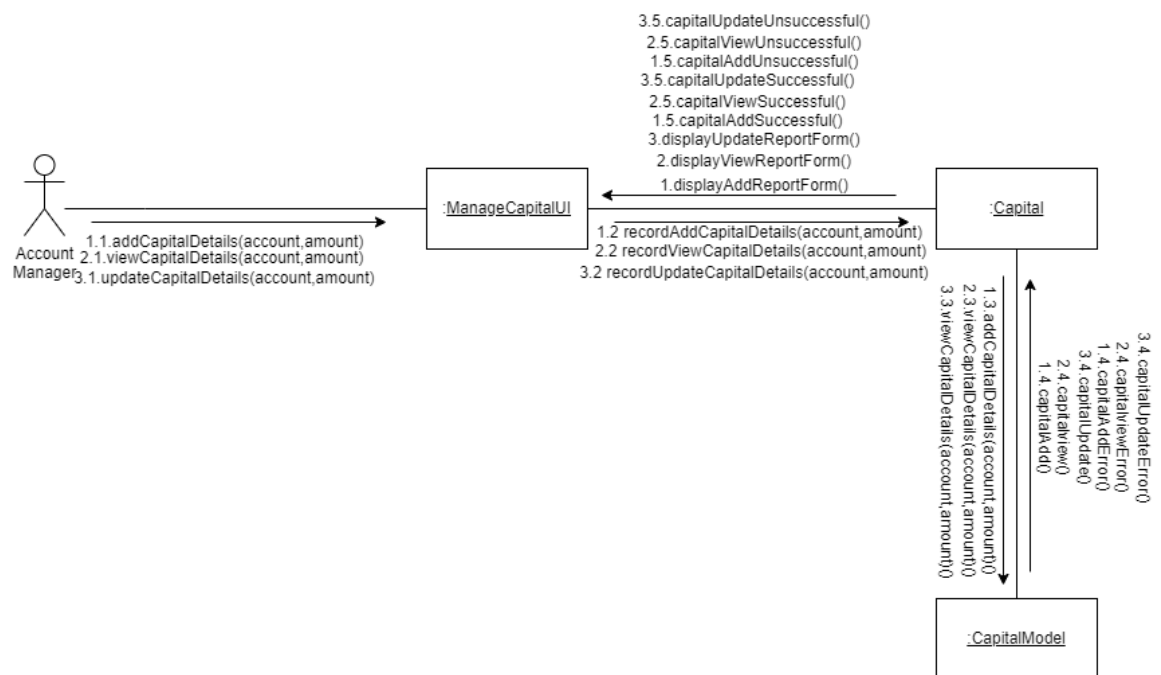
Figure 42(5.4): Authenticate User



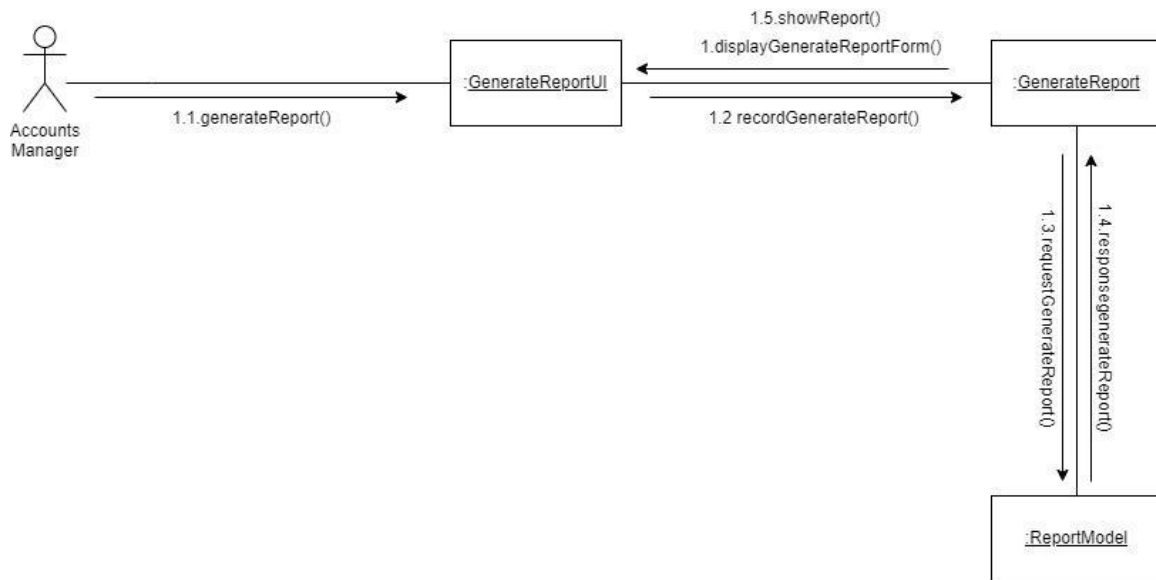
**Figure 43(5.4): Manage User**



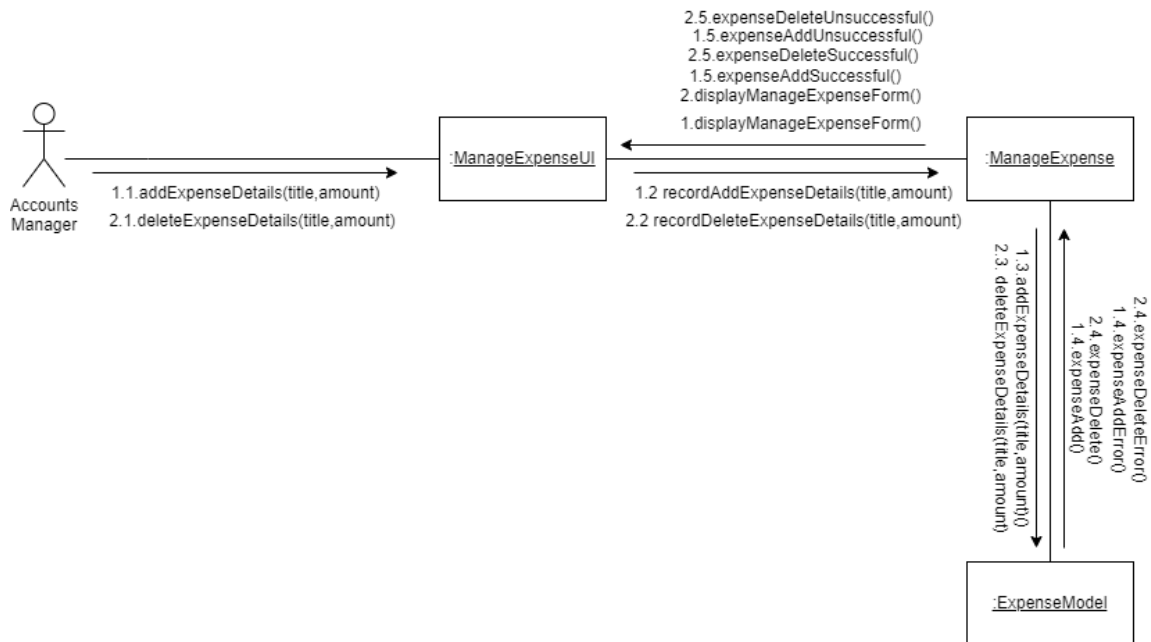
**Figure 44(5.4): Manage Accounts**



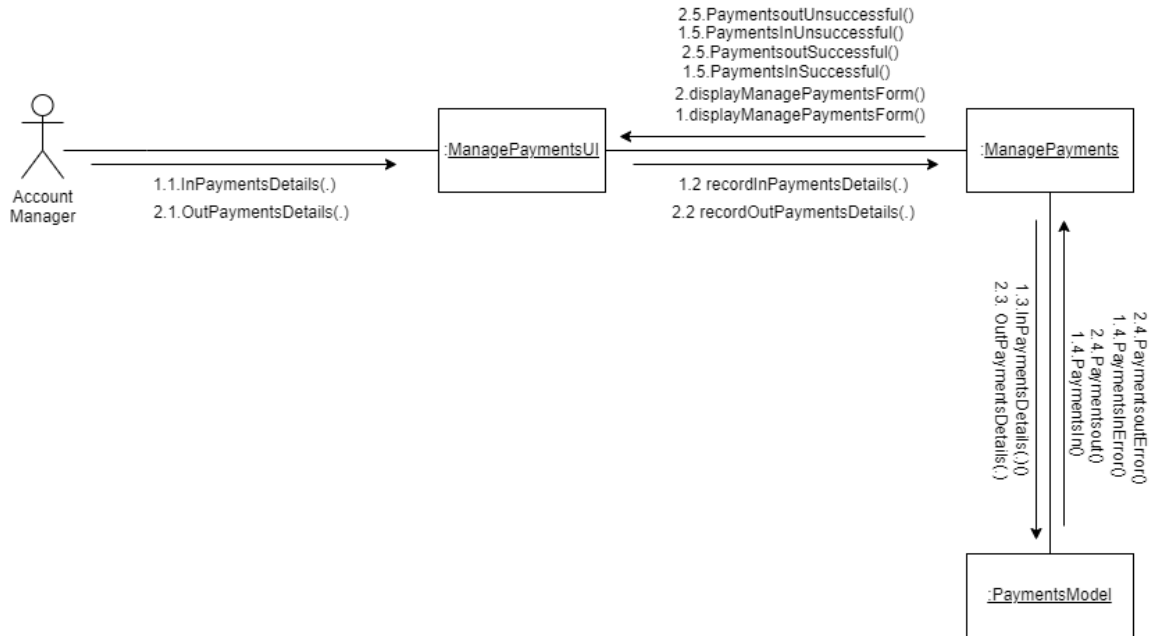
**Figure 45(5.4): Manage Capital**



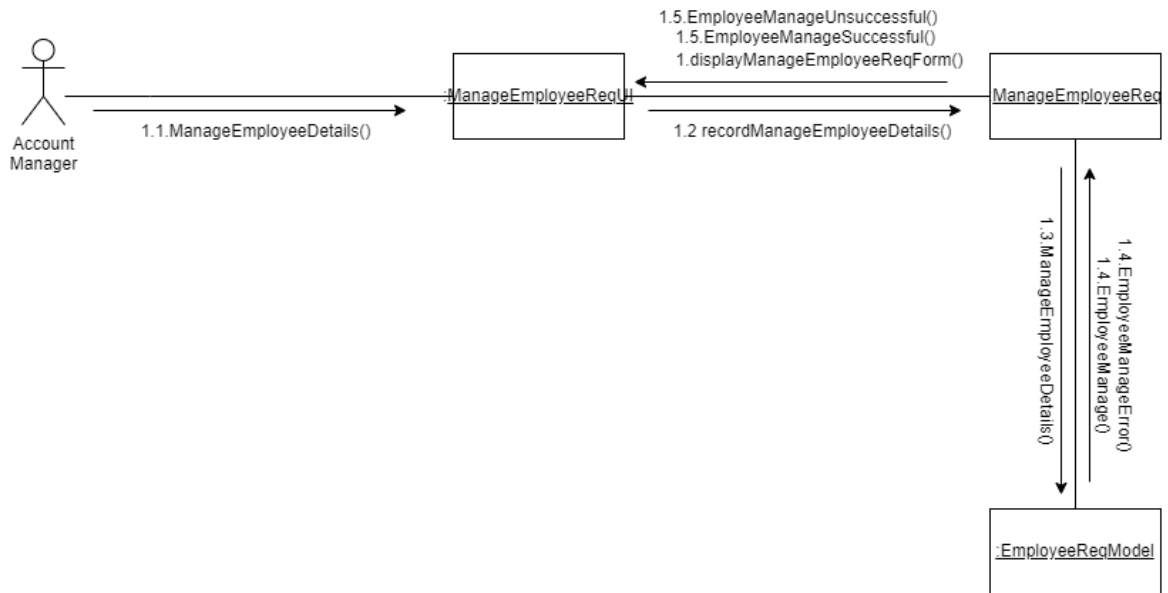
**Figure 46(5.4): Generate Report**



**Figure 47(5.4): Manage Expense**



**Figure 48(5.4): Manage Payments**



**Figure 49(5.4): Manage Employee Request**

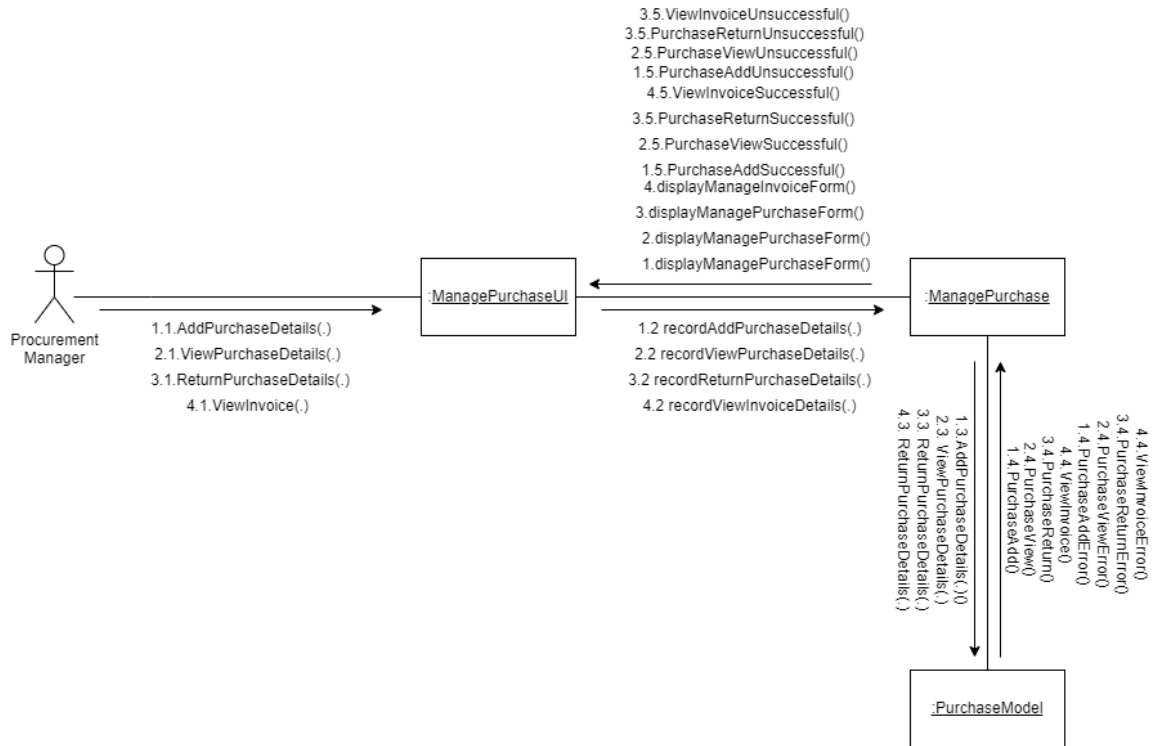


Figure 50(5.4): Manage Purchases

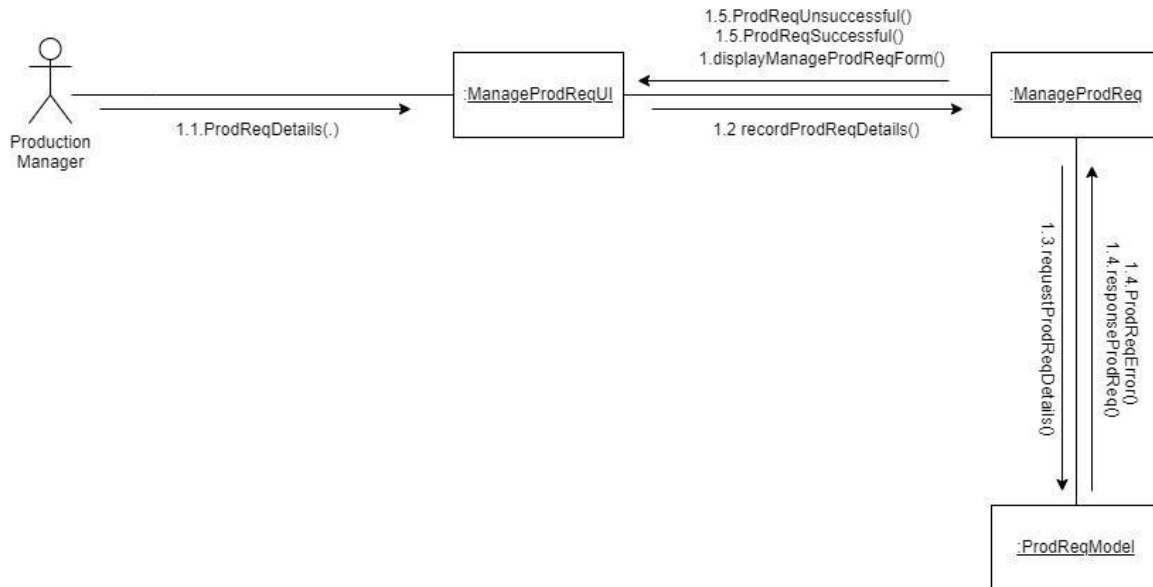
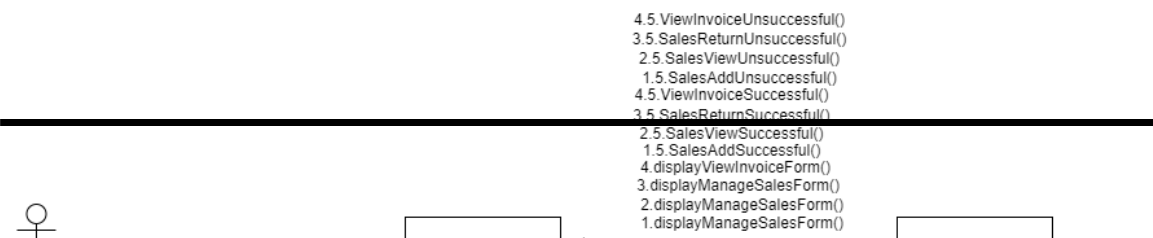
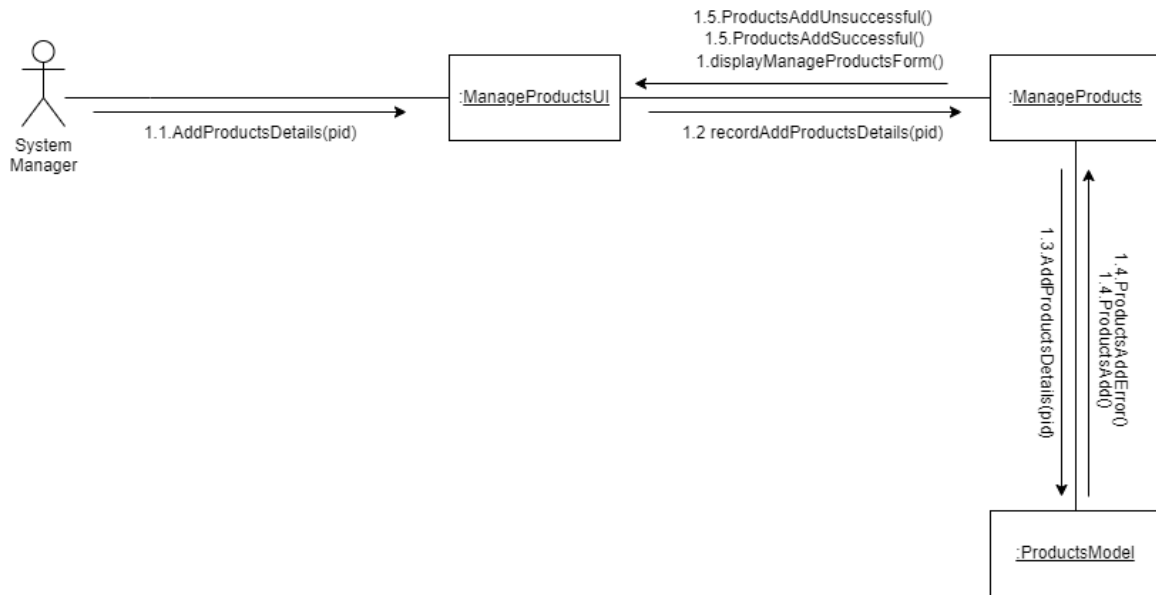


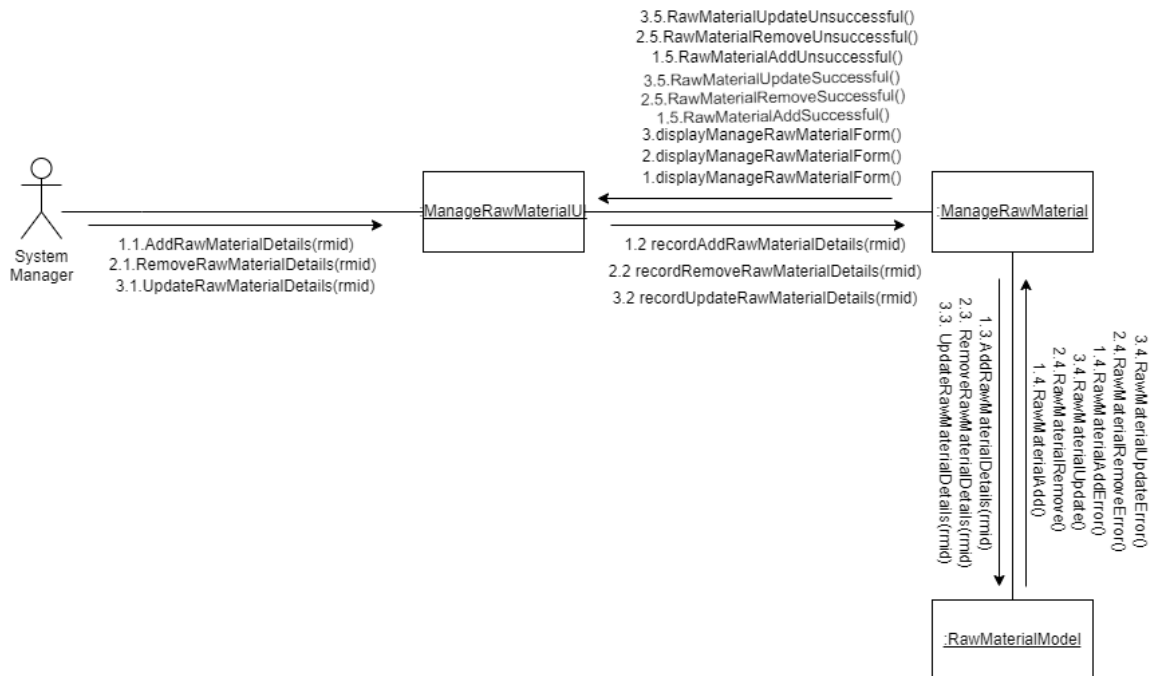
Figure 51(5.4): Manage Production Request



**Figure 52(5.4): Manage Sales**

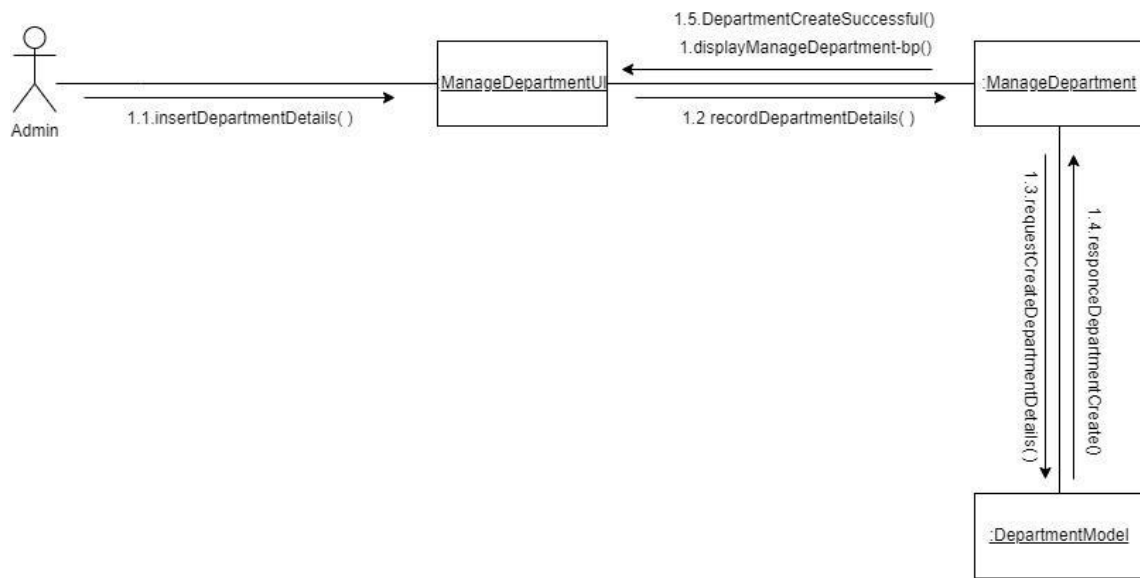


**Figure 53(5.4): Manage Products**



**Figure 54(5.4): Manage Raw Material**

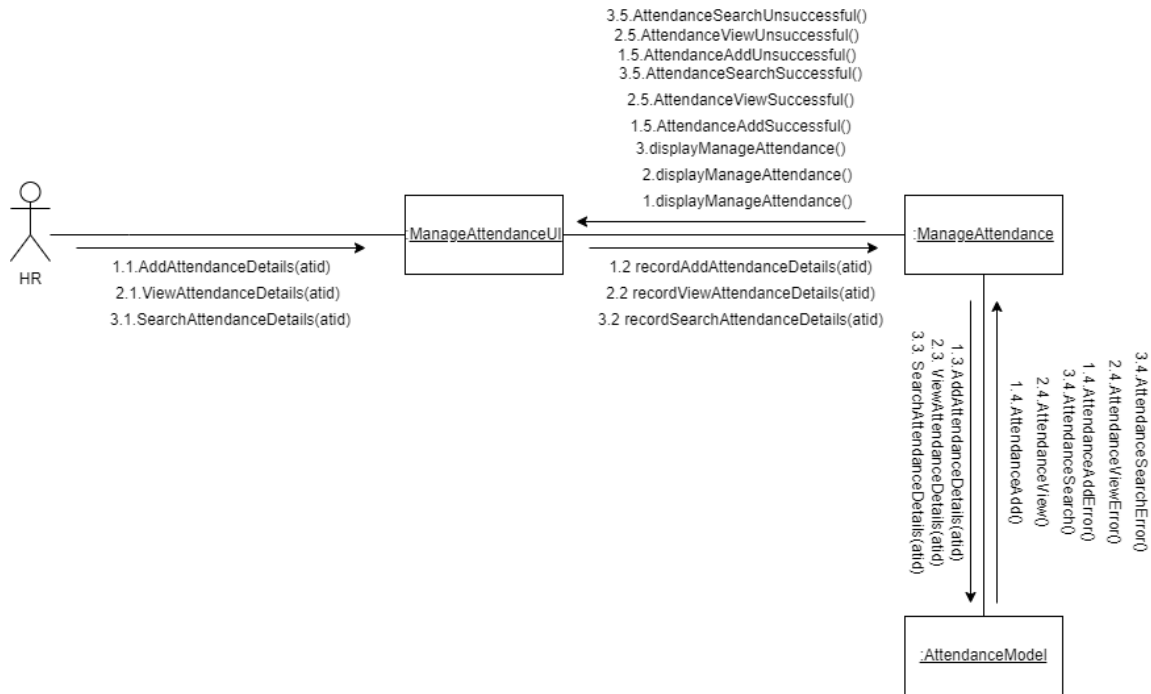




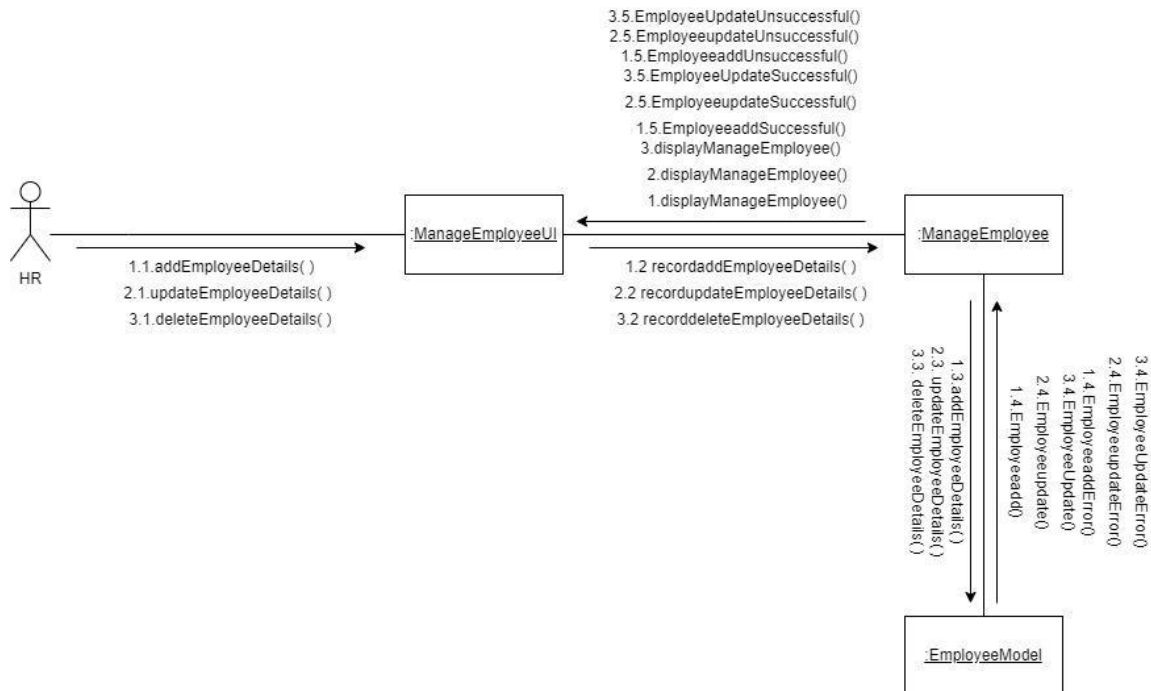
**Figure 55(5.4): Manage Department**



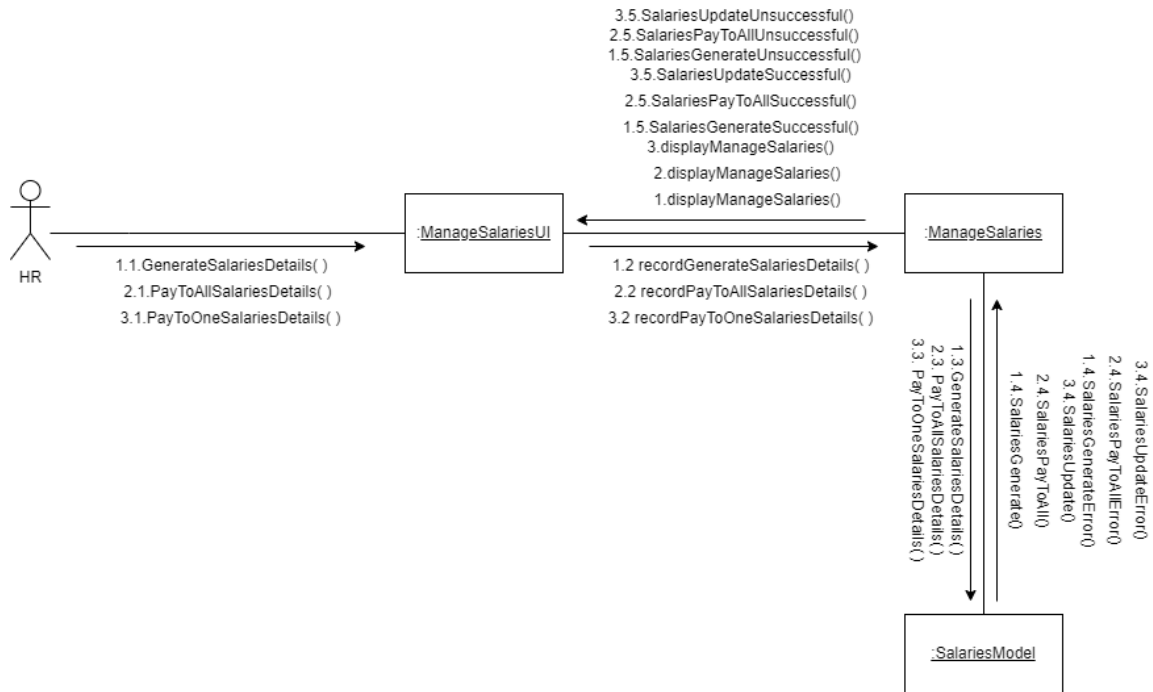
**Figure 56(5.4): Manage Backup**



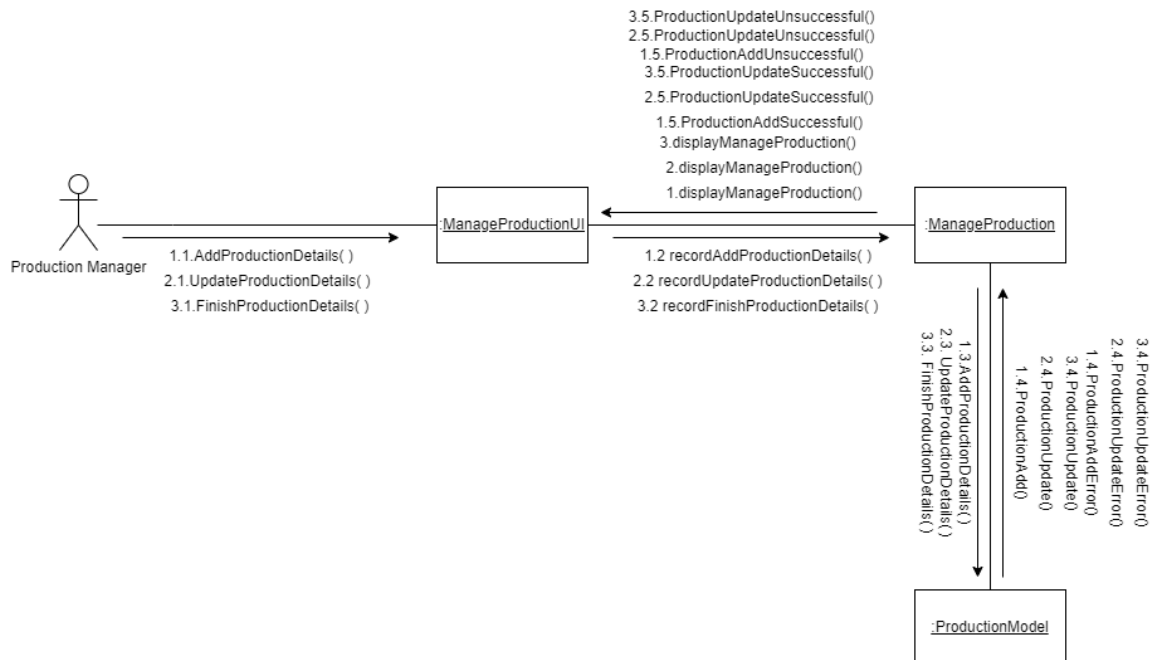
**Figure 57(5.4): Manage Attendance**



**Figure 58(5.4): Manage Employee**



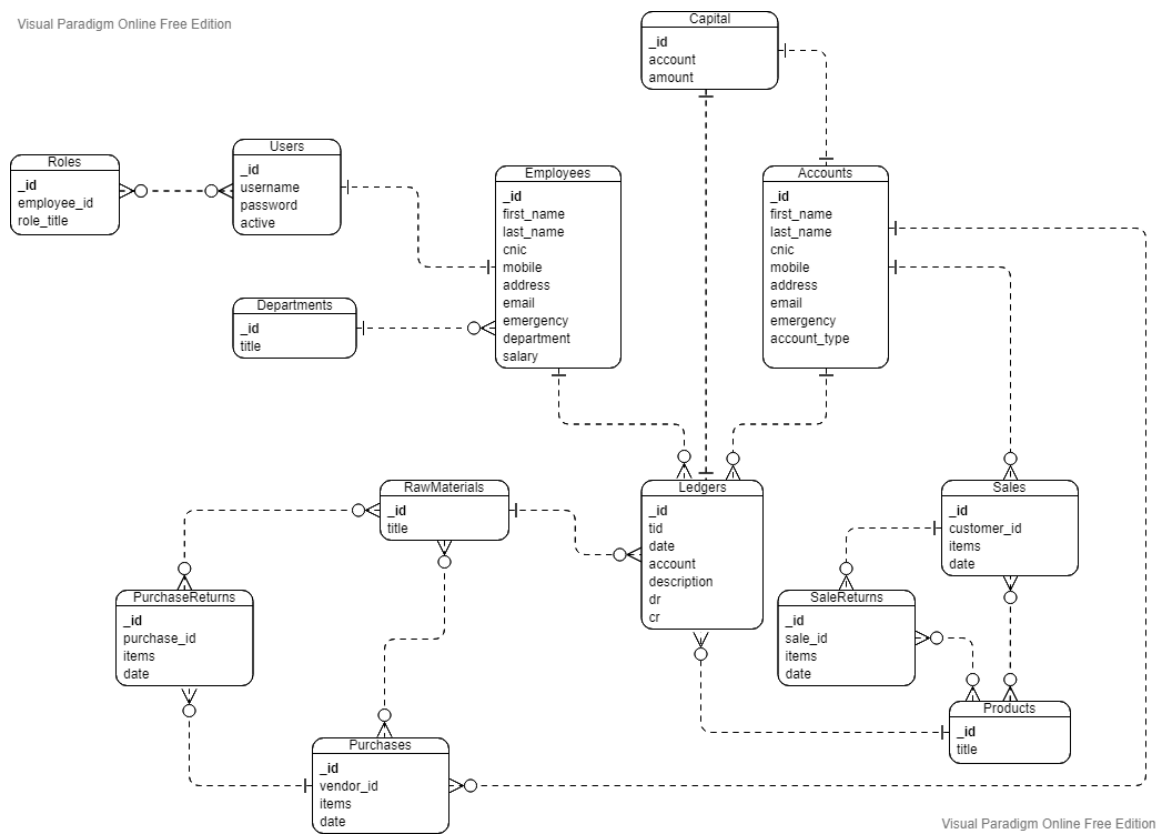
**Figure 59(5.4): Manage Salaries**



**Figure 60(5.4): Manage Production**

## 5.5 ERD

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

**Figure 61(5.5): ERD**

## 5.6 Data Dictionary

Table 24(5.6): Data Dictionary

Entity Name	Column Name	Data Type	Required	Accepts Null Values?
Roles	_id	Mongo.ObjectID	Yes	No
Roles	employee_id	Mongo.ObjectID	Yes	No
Roles	role_title	String	Yes	No
Users	_id	Mongo.ObjectID	Yes	No
Users	username	String	Yes	No
Users	password	String	Yes	No
Users	active	Boolean	Yes	No
Departments	_id	Mongo.ObjectID	Yes	No
Departments	title	String	Yes	No
Products	_id	Mongo.ObjectID	Yes	No
Products	title	String	Yes	No
RawMaterials	_id	Mongo.ObjectID	Yes	No
RawMaterials	title	String	Yes	No
Capitals	_id	Mongo.ObjectID	Yes	No
Capitals	account	Mongo.ObjectID	Yes	No
Capitals	amount	Number	Yes	No
Employees	_id	Mongo.ObjectID	Yes	No
Employees	first_name	String	Yes	No

Employees	last_name	String	Yes	No
Employees	cnic	Number	Yes	No
Employees	mobile	Number	Yes	No
Employees	address	String	Yes	No
Employees	email	String	No	No
Employees	emergency	Number	No	No
Employees	department	String	Yes	No
Employees	salary	Float	Yes	No
Accounts	_id	Mongo.ObjectID	Yes	No
Accounts	first_name	String	Yes	No
Accounts	last_name	String	Yes	No
Accounts	cnic	Number	Yes	No
Accounts	mobile	Number	Yes	No
Accounts	address	String	Yes	No
Accounts	email	String	No	No
Accounts	emergency	Number	No	No
Accounts	account_type	String	Yes	No
Ledgers	_id	Mongo.ObjectID	Yes	No
Ledgers	tid	Mongo.ObjectID	Yes	No
Ledgers	account	Mongo.ObjectID	Yes	No
Ledgers	date	String	Yes	No
Ledgers	description	String	Yes	No
Ledgers	dr	Number	Yes	No
Ledgers	cr	Number	Yes	No
Sales	_id	Mongo.ObjectID	Yes	No
Sales	customer_id	Mongo.ObjectID	Yes	No
Sales	items	String	Yes	No
Sales	date	String	Yes	No
SaleReturns	_id	Mongo.ObjectID	Yes	No
SaleReturns	sale_id	Mongo.ObjectID	Yes	No
SaleReturns	items	String	Yes	No
SaleReturns	date	String	Yes	No
Purchases	_id	Mongo.ObjectID	Yes	No

Purchases	vendor_id	Mongo.ObjectID	Yes	No
Purchases	items	String	Yes	No
Purchases	date	String	Yes	No
PurchaseReturns	_id	Mongo.ObjectID	Yes	No
PurchaseReturns	purchase_id	Mongo.ObjectID	Yes	No
PurchaseReturns	items	String	Yes	No
PurchaseReturns	date	String	Yes	No

## 6. IMPLEMENTATION DETAILS

---

### 6.1 Development Setup

Table 25(6.1): Development Setup

Tool/Technology	Description
VS Code	Used VS Code IDE for coding.
Postman	Postman is use to test API.
Google Chrome	To run web application
Mozilla Firefox	To run web application
MongoDB	NoSQL database
React JS	JS library for front end development
Express JS	JS frameforwork for RESTful API development
Node JS	JS enviroment
Redux Toolkit	JS library for state management
Json Web Token	For creating x-access token
Mongoose	JS library for MongoDB
Font Awesome	CDN for fonts and icons
React-Bootstrap	JS bootstrap library for frontend

### 6.2 Deployment setup

*How and where was your software deployed? Did you face any problems, how did you overcome these problems.*

### 6.3 Algorithms

**Add Account**



```

route.post("/", async (req, res) => {
  const data = req.body;
  if (!data.fname || data.fname == "") {
    return setResponse(405, "First name is required", null, res);
  }
  if (iterateWithFunction(data.fname, isNumeric)) {
    return setResponse(405, "First name can not contain number", null, res);
  }
  if (iterateWithFunction(data.fname, isSymbol)) {
    return setResponse(405, "First name can not contain symbol", null, res);
  }
  if (!data.lname || data.lname == "") {
    return setResponse(405, "Last name is required", null, res);
  }
  if (iterateWithFunction(data.lname, isNumeric)) {
    return setResponse(405, "Last name can not contain number", null, res);
  }
  if (iterateWithFunction(data.lname, isSymbol)) {
    return setResponse(405, "Last name can not contain symbol", null, res);
  }
  if (!data.cnic || data.cnic == "") {
    return setResponse(405, "CNIC is required", null, res);
  }
  if (iterateWithFunction(data.cnic, isAlphabat)) {
    return setResponse(
      405,
      "CNIC can not contain alphabat or symbol",
      null,
      res
    );
  }
  if (iterateWithFunction(data.cnic, isSymbol)) {
    return setResponse(405, "CNIC can not contain symbol", null, res);
  }
  if (data.cnic.split("").length < 13 || data.cnic.split("").length > 13) {

```

```

    return setResponse(405, "CNIC length should be 13", null, res);
}
const cnicExist = await Account.findOne({ cnic: data.cnic });
if (cnicExist) {
    return setResponse(405, "Account already exist with this cnic", null, res);
}
if (!data.mobile || data.mobile === "") {
    return setResponse(405, "Mobile number is required", null, res);
}
if (iterateWithFunction(data.mobile, isAlphabat)) {
    return setResponse(
        405,
        "Mobile number can not contain alphabat or symbol",
        null,
        res
    );
}
if (iterateWithFunction(data.mobile, isSymbol)) {
    return setResponse(405, "Mobile number can not contain symbol", null, res);
}
if (data.mobile.split("").length < 11 || data.mobile.split("").length > 11) {
    return setResponse(405, "Mobile number length should be 11", null, res);
}
if (!data.address || data.address === "") {
    return setResponse(405, "Address is required", null, res);
}
if (data.email) {
    const validEmail =
        /^[a-zA-Z0-9.!#$%&'*/=?^_`{}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/;
    if (!data.email.match(validEmail)) {
        return setResponse(405, "Email address is not valid", null, res);
    }
    const emailExist = await Account.findOne({ email: data.email });
    if (emailExist) {
        return setResponse(

```

```

    405,
    "Account already exist with this email",
    null,
    res
);
}
}
if (data.emergency) {
    if (iterateWithFunction(data.emergency, isAlphabat)) {
        return setResponse(
            405,
            "Emergency number can not contain alphabat or symbol",
            null,
            res
        );
    }
    if (iterateWithFunction(data.emergency, isSymbol)) {
        return setResponse(
            405,
            "Emergency number can not contain symbol",
            null,
            res
        );
    }
    if (
        data.emergency.split("").length < 11 ||
        data.emergency.split("").length > 11
    ) {
        return setResponse(
            405,
            "Emergency number length should be 11",
            null,
            res
        );
    }
}

```

```

    }
    if (!data.accountType || data.accountType === "") {
        return setResponse(405, "Account Type is required", null, res);
    }
    data.status = true;
    const newAccount = new Account(data);
    newAccount.save();
    return setResponse(201, "Account Created", newAccount, res);
});

```

### **Generate Salaries**

```

route.post("/", async (req, res) => {
    const date = new Date();
    const year = date.getFullYear();
    const month = date.getMonth() + 1;
    const day = date.getDate();
    const myDate = month + "/" + year;
    const salaryExist = await Salary.findOne({ date: myDate });
    if (salaryExist) {
        return setResponse(
            405,
            "Salaries already gennerated for this month",
            null,
            res
        );
    }
    const employees = await Employee.find({});
    const attendance = await Attendance.find({});
    const newAttendance = [];
    attendance.map((i) => {
        if (i.date.search(myDate) > -1) {
            i.checkIn =
                parseInt(i.checkIn[0] + "" + i.checkIn[1]) * 60 +
                parseInt(i.checkIn[3] + "" + i.checkIn[4]);
            i.checkOut =
                parseInt(i.checkOut[0] + "" + i.checkOut[1]) * 60 +

```

```

        parseInt(i.checkOut[3] + "" + i.checkOut[4]));
    i.checkOut = i.checkOut - i.checkIn;
    newAttendance.push(i);
  }
});
const final = [];
employees.map((i) => {
  const cnic = i.cnic;
  const temp = {
    account: i._id,
    amount: 0,
    date: myDate,
  };
  newAttendance.map((j) => {
    if (j.cnic === cnic) {
      temp.amount = temp.amount + parseInt(j.checkOut);
    }
  });
  temp.amount = parseInt(temp.amount) * i.salaryPerMinute;
  final.push(temp);
});
await Salary.insertMany(final);
return setResponse(200, "Salaries generated", null, res);
});

```

### **Add Raw Material**

```

route.post("/", async (req, res) => {
  const data = req.body;
  const response = {
    status: null,
    message: null,
    data: null,
  };
  if (!data.title || data.title === "") {
    response.status = 405;
    response.message = "Title field is required";
  }

```

```

    return res.status(response.status).json(response);
  }
  const isExist = await RawMaterial.findOne( { title: data.title } );
  if (isExist) {
    response.status = 405;
    response.message = "Raw Material already exist";
    return res.status(response.status).json(response);
  }
  const rawMaterial = new RawMaterial(data);
  await rawMaterial.save();
  response.status = 201;
  response.message = "Raw Material added";
  response.data = rawMaterial;
  return res.status(response.status).json(response);
});

```

#### **Add Purchase**

```

route.post("/", async (req, res) => {
  const data = req.body;
  if (!data.vendor) {
    return setResponse(405, "Vendor is required", null, res);
  }
  const vendorExist = await Account.findOne( {
    _id: ObjectId(data.vendor),
    accountType: "vendor",
  });
  if (!vendorExist) {
    return setResponse(405, "Account type not allowed", null, res);
  }
  if (!data.paidAmount) {
    return setResponse(405, "Paid Amount is required", null, res);
  }
  if (iterateWithFunction(data.paidAmount, isAlphabat)) {
    return setResponse(
      405,
      "Amount can not contain alphabat or symbol",
    );
  }

```

```

    null,
    res
  );
}
if (!data.rawMaterial) {
  return setResponse(405, "Raw Material is required", null, res);
}
if (data.rawMaterial.length === 0) {
  return setResponse(405, "Raw Material is required", null, res);
}

if (
  data.rawMaterial.some(async (i) => {
    if (!i.material) {
      return true;
    }
    const myRawMaterial = await RawMaterial.findOne({
      _id: ObjectId(i.material),
    });
    if (!myRawMaterial) {
      return true;
    }
    if (!i.quantity) {
      return true;
    }
    if (parseInt(i.quantity) < 1) {
      return true;
    }
    if (!i.unitPrice) {
      return true;
    }
    if (parseFloat(i.unitPrice) < 0.1) {
      return true;
    }
  })

```

```

    ) {
        return setResponse(405, "Invalid Material", null, res);
    }
    const date = new Date();
    const day = date.getDate();
    const month = date.getMonth() + 1;
    const year = date.getFullYear();
    const now = `${month}/${day}/${year}`;
    data.date = now;
    const count = await Purchase.find({});
    data.invoice = count.length + 1;
    const newPurchase = new Purchase(data);
    newPurchase.save();
    return setResponse(201, "Purchase Added", newPurchase, res);
});

```

### **Add Employee**

```

route.post("/", async (req, res) => {
    const data = req.body;
    if (!data.fname || data.fname === "") {
        return setResponse(405, "First name is required", null, res);
    }
    if (iterateWithFunction(data.fname, isNumeric)) {
        return setResponse(405, "First name can not contain number", null, res);
    }
    if (iterateWithFunction(data.fname, isSymbol)) {
        return setResponse(405, "First name can not contain symbol", null, res);
    }
    if (!data.lname || data.lname === "") {
        return setResponse(405, "Last name is required", null, res);
    }
    if (iterateWithFunction(data.lname, isNumeric)) {
        return setResponse(405, "Last name can not contain number", null, res);
    }
    if (iterateWithFunction(data.lname, isSymbol)) {
        return setResponse(405, "Last name can not contain symbol", null, res);
    }
}

```



```

}
if (!data.cnic || data.cnic == "") {
    return setResponse(405, "CNIC is required", null, res);
}
if (iterateWithFunction(data.cnic, isAlphabat)) {
    return setResponse(
        405,
        "CNIC can not contain alphabat or symbol",
        null,
        res
    );
}
if (iterateWithFunction(data.cnic, isSymbol)) {
    return setResponse(405, "CNIC can not contain symbol", null, res);
}
if (data.cnic.split("").length < 13 || data.cnic.split("").length > 13) {
    return setResponse(405, "CNIC length should be 13", null, res);
}
const cnicExist = await Employee.findOne({ cnic: data.cnic });
if (cnicExist) {
    return setResponse(405, "Employee already exist with this cnic", null, res);
}
if (!data.mobile || data.mobile == "") {
    return setResponse(405, "Mobile number is required", null, res);
}
if (iterateWithFunction(data.mobile, isAlphabat)) {
    return setResponse(
        405,
        "Mobile number can not contain alphabat or symbol",
        null,
        res
    );
}
if (iterateWithFunction(data.mobile, isSymbol)) {
    return setResponse(405, "Mobile number can not contain symbol", null, res);
}

```

```

    }
    if (data.mobile.split("").length < 11 || data.mobile.split("").length > 11) {
        return setResponse(405, "Mobile number length should be 11", null, res);
    }
    if (!data.address || data.address == "") {
        return setResponse(405, "Address is required", null, res);
    }
    if (data.email) {
        const validEmail =
            /^[a-zA-Z0-9.!#$%&'+=?^_`{}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)$/;
        if (!data.email.match(validEmail)) {
            return setResponse(405, "Email address is not valid", null, res);
        }
        const emailExist = await Employee.findOne({ email: data.email });
        if (emailExist) {
            return setResponse(
                405,
                "Employee already exist with this email",
                null,
                res
            );
        }
    }
    if (data.emergency) {
        if (iterateWithFunction(data.emergency, isAlphabat)) {
            return setResponse(
                405,
                "Emergency number can not contain alphabat or symbol",
                null,
                res
            );
        }
        if (iterateWithFunction(data.emergency, isSymbol)) {
            return setResponse(
                405,

```

```

        "Emergency number can not contain symbol",
        null,
        res
    );
}
if (
    data.emergency.split("").length < 11 ||
    data.emergency.split("").length > 11
) {
    return setResponse(
        405,
        "Emergency number length should be 11",
        null,
        res
    );
}
}
if (!data.department || data.department == "") {
    return setResponse(405, "Department is required", null, res);
}
if (!ObjectId.isValid(data.department)) {
    return setResponse(405, "Department type is invalid", null, res);
}
if (!data.salary || data.salary == "") {
    return setResponse(405, "Salary is required", null, res);
}
if (iterateWithFunction(data.salary, isAlphabat)) {
    return setResponse(
        405,
        "Salary can not contain alphabat or symbol",
        null,
        res
    );
}
if (iterateWithFunction(data.salary, isSymbol)) {

```

```

    return setResponse(405, "Salary can not contain symbol", null, res);
}
data.status = true;
const eSalary = parseFloat(data.salary);
const minuteSalary = (eSalary / 40) / 60;
data.salaryPerMinute = Math.round(minuteSalary);
const newEmployee = new Employee(data);
newEmployee.save();
return setResponse(201, "Employee Created", newEmployee, res);
});

```

### **Capital Management**

```

const route = require("express").Router();
const { Capital, Account } = require("../models/model");
const ObjectId = require("mongoose").Types.ObjectId;

function setResponse(status = null, message = null, data = null, resp) {
    const response = {};
    response.status = status;
    response.message = message;
    response.data = data;
    return resp.status(response.status).json(response);
}

const isAlphabat = (val) => {
    return isNaN(val);
};

const isSymbol = (val) => {
    const ascii = val.charCodeAt(0);
    if (ascii < 48 || ascii > 122) {
        return true;
    }
    if (ascii > 57 && ascii < 65) {
        return true;
    }
}

```

```

    if (ascii > 90 && ascii < 97) {
        return true;
    }
    return false;
};

```

```

const iterateWithFunction = (data, func) => {
    return data.split("").some((i) => {
        return func(i);
    });
};

```

```

// Get all capitals
route.get("/", async (req, res) => {
    const capitals = await Capital.find();
    return setResponse(200, null, capitals, res);
});

```

```

// Get single capital
route.get("/:id", async (req, res) => {
    if (!ObjectId.isValid(req.params.id)) {
        return setResponse(405, "Invalid ID", null, res);
    }
    const capital = await Capital.findOne({ _id: ObjectId(req.params.id) });
    if (!capital) {
        return setResponse(404, "Capital not found", capital, res);
    }
    return setResponse(200, null, capital, res);
});

```

```

// Add new capital
route.post("/", async (req, res) => {
    const data = req.body;
    if (!data.account || data.account === "") {
        return setResponse(405, "Account is required", null, res);
    }
}

```

```

    }
    if (!ObjectId.isValid(data.account)) {
        return setResponse(405, "Invalid ID", null, res);
    }
    const accountExist = await Account.findOne({
        _id: ObjectId(data.account),
        accountType: "investor",
    });
    if (!accountExist) {
        return setResponse(405, "Invalid account id", null, res);
    }
    if (!data.amount || data.amount === "") {
        return setResponse(405, "Amount is required", null, res);
    }
    if (iterateWithFunction(data.amount, isAlphabat)) {
        return setResponse(
            405,
            "AMOUNT can not contain alphabat or symbol",
            null,
            res
        );
    }
    if (iterateWithFunction(data.amount, isSymbol)) {
        return setResponse(405, "Amount can not contain symbol", null, res);
    }
    const newCapital = new Capital(data);
    newCapital.save();
    return setResponse(201, "Capital Added", newCapital, res);
});

// Delete capital
route.delete("/:id", async (req, res) => {
    if (!ObjectId.isValid(req.params.id)) {
        return setResponse(405, "Invalid ID", null, res);
    }

```

```

const capitalExist = await Capital.findOne( {
  _id: ObjectId(req.params.id),
});
if (!capitalExist) {
  return setResponse(404, "Capital not found", null, res);
}
const deleted = await Capital.deleteOne( { _id: ObjectId(req.params.id) });
return setResponse(200, "Capital deleted", deleted, res);
});

```

```

module.exports = route;

```

### **Sales Management**

```

const route = require("express").Router();
const { Sale, Product, Account } = require("../models/model");
const ObjectId = require("mongoose").Types.ObjectId;

function setResponse(status = null, message = null, data = null, resp) {
  const response = {};
  response.status = status;
  response.message = message;
  response.data = data;
  return resp.status(response.status).json(response);
}

const isAlphabat = (val) => {
  return isNaN(val);
};

const iterateWithFunction = (data, func) => {
  return data.split("").some((i) => {
    return func(i);
  });
};

```

```

function getQuantity(data, id) {

```

```

    if (data.length === 0) return 0;
    let qty = 0;
    for (let i = 0; i < data.length; i++) {
        if (data[i].item === id) {
            qty = qty + parseInt(data[i].quantity);
        }
    }
    return qty;
}

// Get all sales
route.get("/", async (req, res) => {
    const sales = await Sale.find({});
    return setResponse(200, null, sales, res);
});

// Get single sale
route.get("/:id", async (req, res) => {
    if (!ObjectId.isValid(req.params.id)) {
        return setResponse(405, "Invalid ID", null, res);
    }
    const sale = await Sale.findOne({ _id: ObjectId(req.params.id) });
    if (!sale) {
        return setResponse(404, "Sale record not found", null, res);
    }
    return setResponse(200, null, sale, res);
});

// Add new sale
route.post("/", async (req, res) => {
    const data = req.body;
    if (!data.customer) {
        return setResponse(405, "Customer is required", null, res);
    }
    const customerExist = await Account.findOne({

```



```

    _id: ObjectId(data.customer),
    accountType: "customer",
  });
  if (!customerExist) {
    return setResponse(405, "Account type not allowed", null, res);
  }
  if (!data.receivedAmount) {
    return setResponse(405, "Received Amount is required", null, res);
  }
  if (iterateWithFunction(data.receivedAmount, isAlphabat)) {
    return setResponse(
      405,
      "Amount can not contain alphabat or symbol",
      null,
      res
    );
  }
  if (!data.product) {
    return setResponse(405, "Product is required", null, res);
  }
  if (data.product.length === 0) {
    return setResponse(405, "Product is required", null, res);
  }

  if (
    data.product.some(async (i) => {
      if (!i.item) {
        return true;
      }
      const myProduct = await Product.findOne({
        _id: ObjectId(i.item),
      });
      if (!myProduct) {
        return true;
      }
    })
  ) {
    return setResponse(405, "Product is required", null, res);
  }

```

```

    if (!i.quantity) {
      return true;
    }
    if (parseInt(i.quantity) < 1) {
      return true;
    }
    if (!i.unitPrice) {
      return true;
    }
    if (parseFloat(i.unitPrice) < 0.1) {
      return true;
    }
  })
) {
  return setResponse(405, "Invalid iTem", null, res);
}

const date = new Date();
const day = date.getDate();
const month = date.getMonth() + 1;
const year = date.getFullYear();
const now = `${month}/${day}/${year}`;
data.date = now;
const count = await Sale.find({});
data.invoice = count.length + 1;
const newSale = new Sale(data);
newSale.save();
return setResponse(201, "Sale Added", newSale, res);
});

// Return item
route.put("/:id", async (req, res) => {
  const data = req.body;
  if (!ObjectId.isValid(req.params.id)) {
    return setResponse(405, "Invalid ID", null, res);
  }
}

```

```

if (!data.product || data.product.length === 0) {
  return setResponse(405, "Nothing to return", null, res);
}
const mySale = await Sale.findOne({ _id: ObjectId(req.params.id) });
if (
  data.product.some((i) => {
    if (!i.quantity) {
      return true;
    }
    if (!i.item) {
      return true;
    }
    if (parseInt(i.quantity) < 1) {
      return true;
    }
    if (
      parseInt(getQuantity(mySale.returns, i.item)) +
      parseInt(i.quantity) >
      parseInt(getQuantity(mySale.product, i.item))
    ) {
      return true;
    }
  })
) {
  return setResponse(405, "Invalid Return", null, res);
}
const updatedData = [...mySale.returns, ...data.product];
const update = await Sale.updateOne(
  { _id: ObjectId(req.params.id) },
  { $set: { returns: updatedData } }
);
return setResponse(200, "Return Successful", update, res);
});

module.exports = route;

```

## 6.4 Constraints

### 6.4.1 Assumptions

*Things we assume will be true.*

*e.g.:*

- *We will receive all necessary technical support from the engineers at cMeRun, Select and Mellon Bank to help design the interfaces between their systems and enGyro.*
- *All database maintenance will be handled by the client.*
- *There will be no real-time interfacing with any accounting systems.*

### 6.4.2 System constraints

*A constraint specifies how the system must operate or how it must be built*

### 6.4.3 Restrictions

*Constraints applied on the system by the client*

### 6.4.4 Limitations

*Services your software is unable to provide*

## 7. TESTING

### 7.1 Extended Test Cases

Table 26(7.1): Test Case 1 Login

Test Case ID: 1			Test Design By: M Haris Siddiqui			
Test Module Name: login			Test Design date:12/7/22			
Test Priority: high			Test Executed By: M Haris Siddiqui			
Test title: Valid Employee Information			Test Executed Date: 12/7/22			
Description: Test the login module with valid and invalid login details						
Pre-Condition:						
Dependencies:						
Step	Test Step	Test Date	Expected Result	Actual Result	Status	Notes
1	Navigate to login page		Login form is displayed	Login form is displayed	Pass	
2	User submit form without employee id and password	Id: Password:	System displays “Employee id and password is required”	System displays “Employee id and password is required”		
3	User submit form with valid employee id and empty password field	Id:12 Password:	System displays” Password is required”	System displays” Password is required”		
4	User submit form with empty employee id and valid password field	Id: Password:1234	System displays” Employee id is required”	System displays” Employee id is required”		

5	User submit form with valid employee id and invalid password field	Id:123 Password:1255	System displays” invalid password”	System displays” invalid password”		
6	User submit form with invalid employee id and valid password field	Id:12>@ Password:12345678	System displays” invalid employee id”	System displays” invalid employee id”		
7	User submit form with valid employee id and valid password field	Id:12 Password:12345678	System displays” form submitted successfully”	System displays” form submitted successfully”		
<b>Post Condition:</b> User login						

**Table 27(7.1): Test Case 2 Forget Password**

<b>Test Case ID:</b> 1.1				<b>Test Design By:</b> M Haris Siddiqui		
<b>Test Module Name:</b> Forget Password				<b>Test Design date:</b> 12/7/22		
<b>Test Priority:</b> high				<b>Test Executed By:</b> M Haris Siddiqui		
<b>Test title:</b> Forget Password				<b>Test Executed Date:</b> 12/7/22		
<b>Description:</b> Employee forget the password						
<b>Pre-Condition:</b>						
<b>Dependencies:</b>						
Step	Test Step	Test Date	Expected Result	Actual Result	Status	Notes
1	Navigate to login page		Login form is displayed	Login form is displayed	Pass	

2	Navigate to forget password page		Forget form is displayed	Forget form is displayed		
3	User submit form without employee id	Id:	System display" employee id is required"	System display" employee id is required"		
4	User submit form containing alphabet in employee id	Id:12345gfifu	System display" invalid employee id"	System display" invalid employee id"		
5	User submit form containing employee id length not equal to 13	Id: 1234567	System display" invalid employee id"	System display" invalid employee id"		
6	User submit form containing employee id length equal to 13	Id:1234567892345	System display" Password reset link sent to your email address"	System display" Password reset link sent to your email address"		
<b>Post Condition:</b> New password send successfully.						

**Table 28(7.1): Test Case Manage Account**

Test Case ID: 3				Test Design By: M Haris Siddiqui		
Test Module Name: Manage Account				Test Design date: 12/7/22		
Test Priority: high				Test Executed By: M Haris Siddiqui		
Test title: Add account / Update Account				Test Executed Date: 12/7/22		
Description: Actor would be able to add and update accounts.						
Pre-Condition:						
Dependencies:						
Step	Test Step	Test Date	Expected Result	Actual Result	Status	Notes
1	Navigate to add / Update account page		Add account form is displayed	Add account form is displayed	Pass	
2	User Submit form with empty first name field	First name:	System displays" First name is required"	System displays" First name is required"		
3	User submit form in which first name containing number or special character	First name:Ha@1	System displays" Invalid first name"	System displays" Invalid first name"		
4	User Submit form with empty last name field	Last name:	System displays" last name is required"	System displays" last name is required"		
5	User submit form in which last name containing number or	Last name:sid@12	System displays" Invalid last name"	System displays" Invalid last name"		



	special character					
6	User submit form with empty CNIC	CNIC:	System displays "CNIC is required".	System displays "CNIC is required".		
7	User submit form in which CNIC containing alphabet or special character	CNIC:234sa@	System displays "CNIC is invalid".	System displays "CNIC is invalid".		
8	User submit form in which CNIC length is not equal to 13	CNIC:123456	System displays "CNIC is invalid".	System displays "CNIC is invalid".		
9	User submit form with empty mobile number field	Mobile number:	System displays "Mobile number is required "	System displays "Mobile number is required "		
10	User submit form containing alphabet and special character in mobile number field	Mobile No:12kdd@	System displays "Mobile number is invalid "	System displays "Mobile number is invalid "		
11	User submit form in which mobile number length is	Mobile No:123456				

	not equal to 11					
12	User submit form with empty address field	Address:	System display” Address is required ”	System display” Address is required ”		
13	User submit form with invalid email address format	Email address:harissid@gmail.com	System displays” Email address is invalid”	System displays” Email address is invalid”		
14	User submit form containing alphabet and special char in emergency number field	Emergency number:123h@	System display” Emergency number is invalid”	System display” Emergency number is invalid”		
15	User submit form in which emergency number length is not equal to 11	Emergency number:123456	System display” Emergency number is invalid”	System display” Emergency number is invalid”		
14	User submit form in which account type is not selected		System display “Account type is required”	System display “Account type is required”		

15	User submit form in which opening balance field is empty		System Display” Opening balance required” is	System Display” Opening balance is required”		
16	User submit form containing alphabet and special char in opening balance field	Opening balance:12g@	System display” Opening balance is invalid”	System display” Opening balance is invalid”		
17	User submit form in which opening balance is not selected		System display” Balance type is required”	System display” Balance type is required”		
18	User submit form with correct first name, Last name, CNIC, Mobile number, Address, Email address, emergency number, Account type, opening balance, opening balance type,		System display” Account added successfully”	System display” Account added successfully”		
<b>Post condition:</b> User account added successfully.						

**Table 29(7.1): Test Case 4 Manage Capital**

Test Case ID: 4				Test Design By: M Haris Siddiqui		
Test Module Name: Manage Capital				Test Design date: 12/8/22		
Test Priority: high				Test Executed By: M Haris Siddiqui		
Test title: Add / Update Capital				Test Executed Date: 12/8/22		
Description: Test update and add capital with valid and invalid details						
Pre-Condition:						
Dependencies:						
Step	Test Step	Test Date	Expected Result	Actual Result	Status	Notes
1	Navigate to add capital page		Add capital form is displayed	Add capital form is displayed	Pass	
2	User submit form in which account field is not selected		System displays “Amount is required”.	System displays “Amount is required”.		
3	User submit form in which amount field is empty	Amount :	System displays “Amount is required”.	System displays “Amount is required”.		

4	User submit form containing special char and alphabet in amount field	Amount:123d@	System displays "Amount is invalid".	System displays "Amount is invalid".		
5	User submit form with valid information	Amount:30000	System display" Form submitted successfully"	System display" Form submitted successfully"		
<b>Post Condition:</b> User added or updated capital successfully.						

**Table 30(7.1): Test Case 6 Manage Expense**

Test Case ID: 6				Test Design By: M Haris Siddiqui		
Test Module Name: Manage Expense				Test Design date: 12/8/22		
Test Priority: high				Test Executed By: M Haris Siddiqui		
Test title: Add expense				Test Executed Date: 12/8/22		
Description: Actor will able to add expense.						
Pre-Condition:						
Dependencies:						
Step	Test Step	Test Date	Expected Result	Actual Result	Status	Notes
1	User navigate to add expense page		Expense page is displayed	Expense page is displayed	Pass	
2	User submit form in which expense title is empty	Expense title:	System displays “Expense title is required”.	System displays “Expense title is required”.		

3	User submit form in which expense amount is negative number	Expense amount:-12	System display “Expense amount should be greater than 0”.	System display “Expense amount should be greater than 0”.		
4	User submit form in which expense amount field is empty	Expense amount:	System displays “Expense amount is required”.	System displays “Expense amount is required”		
5	User submit form in which expense amount is not a number	Expense amount: dhdh	System displays “Expense amount should be a number”.	System displays “Expense amount should be a number”.		
<b>Post Condition:</b> User added expense successfully.						

**Table 31(7.1): Test Case 7 Manage Payments (Payments In)**

<b>Test Case ID:</b> 7				<b>Test Design By:</b> M Haris Siddiqui		
<b>Test Module Name:</b> Manage Payment				<b>Test Design date:</b> 12/8/22		
<b>Test Priority:</b> Middle				<b>Test Executed By:</b> M Haris Siddiqui		
<b>Test title:</b> Payment In				<b>Test Executed Date:</b> 12/8/22		
<b>Description:</b> Test whether payment In details are valid						
<b>Pre-Condition:</b>						
<b>Dependencies:</b>						
Step	Test Step	Test Date	Expected Result	Actual Result	Status	Notes
1	User navigate to Payment In page		Payment In form is displayed	Payment In form is displayed	Pass	

2	User submitted form in which customer account is not selected		System display "Customer account is required".	System display "Customer account is required".		
3	User submitted form in which sale invoice is not selected		System display "Sale invoice is required".	System display "Sale invoice is required".		
4	User submitted form in which amount field is empty		System display "Amount is required".	System display "Amount is required".		
5	User submitted form containing alphabet in amount field		System display "Please enter amount in numbers only".	System display "Please enter amount in numbers only".		
6	User submitted form in which amount is greater than receivable amount		System display "Amount can't be greater than receivable amount".	System display "Amount can't be greater than receivable amount".		
<b>Post Condition:</b> User manage payment In successfully						

**Table 32(): Test Case 8 Manage Payments (Payments Out)**

<b>Test Case ID:</b> 7.1	<b>Test Design By:</b> M Haris Siddiqui
<b>Test Module Name:</b> Manage Payment	<b>Test Design date:</b> 12/8/22
<b>Test Priority:</b> Middle	<b>Test Executed By:</b> M Haris Siddiqui

Test title: Payment out				Test Executed Date: 12/8/22		
Description: Test the payment out details are valid or not						
Pre-Condition:						
Dependencies:						
Step	Test Step	Test Date	Expected Result	Actual Result	Status	Notes
1	User navigate to Payment out page		Payment out form is displayed	Payment out form is displayed	Pass	
2	User submitted form in which vendor account is not selected		System display “Vendor account is required”.	System display “Vendor account is required”.		
3	User submitted form in which purchase invoice is not selected		System display “Purchase invoice is required”.	System display “Purchase invoice is required”.		
4	User submitted form in which amount field is empty		System display “Amount is required”.	System display “Amount is required”.		
5	User submitted form containing alphabet in amount field		System display “Please enter amount in numbers only”.	System display “Please enter amount in numbers only”.		
6	User submitted form in which amount is		System display“Amount can’t be greater	System display“Amount can’t be greater than payable amount		



	greater than payable amount		than payable amount			
<b>Post Condition:</b> User manage payment out successfully						

**Table 33(7.1): Test Case 9 Manage Purchases**

Test Case ID: 9				Test Design By: M Haris Siddiqui		
Test Module Name: Manage Purchases				Test Design date: 12/8/22		
Test Priority: Middle				Test Executed By: M Haris Siddiqui		
Test title: Add Purchases				Test Executed Date: 12/8/22		
Description: Test the add purchase field.						
Pre-Condition:						
Dependencies:						
Step	Test Step	Test Date	Expected Result	Actual Result	Status	Notes
1	Navigate to add purchase page		Add purchase form is displayed	Add purchase form is displayed	Pass	
2	User submit form in which vendor field is not selected		System display “Vendor field is required”	System display “Vendor field is required”		
3	User submit form in which paid amount field is empty	Paid amount:	System display “Paid amount field is required”	System display “Paid amount field is required”		
4	User submit form in which paid amount field contain alphabet	Paid amount:12s	System display “Paid amount is invalid”	System display “Paid amount is invalid”		

5	User submit form in which paid amount field contain negative number	Paid amount:-12	System display "Paid amount is invalid"	System display "Paid amount is invalid"		
6	User submit form in which raw material field is not selected		System display "Raw material field is required"	System display "Raw material field is required"		
7	User submit form in which raw material quantity field is empty	Raw material quantity:	System display "Raw material quantity field is required"	System display "Raw material quantity field is required"		
8	User submit form in which raw material quantity field contain alphabet	Raw material quantity:12s	System display "Raw material quantity is invalid"	System display "Raw material quantity is invalid"		
9	User submit form in which raw material quantity field contain negative number	Raw material quantity:-12	System display "Raw material quantity is invalid"	System display "Raw material quantity is invalid"		
10	User submit form in which per unit price field is empty	Per unit price:	System display "Per unit price field is required"	System display "Per unit price field is required"		
11	User submit form in which per unit price	Per unit price:123g	System display "Per	System display "Per unit price is invalid"		

	field contain alphabet		unit price is invalid"			
12	User submit form in which per unit price field contain negative number	Per unit price:-123	System display "Per unit price is invalid"	System display "Per unit price is invalid"		
13	User submit form in which expense title field is empty	Expense title:	System display "Expense title field is required"	System display "Expense title field is required"		
14	User submit form in which expense cost field is empty	Expense cost:	System display "Expense cost field is required"	System display "Expense cost field is required"		
<b>Post Condition:</b> User added purchases successfully						

**Table 34(7.1): Test Case 10 Add sale**

Test Case ID: 11			Test Design By: M Zeeshan Fiaz			
Test Module Name: Add sale			Test Design date: 12/7/22			
Test Priority: high			Test Executed By: M Zeeshan Fiaz			
Test title: Valid Add sale Information			Test Executed Date: 12/7/22			
Description: Test the add sale module with valid and invalid data						
Pre-Condition:						
Dependencies:						
Step	Test Step	Test Date	Expected Result	Actual Result	Status	Notes

1	Navigate to Sale management page		Sale management is displayed	Sale management is displayed	Pass	
2	User submit form without selecting customer id	Customer Id:	System displays "Customer id is required"	System displays "Customer id is required"		
3	User submit form with empty paid amount	Paid amount:	System displays "Paid amount is required"	System displays "Paid amount is required"		
4	User submit form that contains alphabets in Paid Amount	Paid Amount: 1200@78	System displays "Paid amount is invalid"	System displays "Paid amount is invalid"		
5	User submit form that contains Negative number in Paid Amount	Paid Amount: -12500	System displays "Paid amount is invalid"	System displays "Paid amount is invalid"		
6	User submit form without selecting Product	Product:	System displays "Product field is required"	System displays "Product field is required"		

7	User submit form without selecting Product quantity	Product quantity:	System displays "Product quantity field is required"	System displays "Product quantity field is required"		
---	---	-------------------	--	--	--	--

8	User submit form that contains	Product quantity: 675@67	System displays "Product quantity is invalid"	System displays "Product quantity is invalid"		
---	--------------------------------	--------------------------	---	---	--	--

	Alphabets in Product quantity					
9	User submit form that contains Negative number in Product quantity	Product quantity: -656756	System displays "Product quantity is invalid"	System displays "Product quantity is invalid"		
10	User submit form that contains alphabets in Per Unit Price	Per Unit price:	System displays "Per unit price is invalid"	System displays "Per unit price is invalid"		
11	User submit form that contains Negative number in Per unit price	Per unit price: -236154	System displays "Per unit price is invalid"	System displays "Per unit price is invalid"		
12	User submit form with empty Expense title	Expense title:	System displays "Expense title field is required"	System displays "Expense title field is required"		
13	User submit form with empty Expense cost	Expense Cost:	System displays "Expense cost field is required"	System displays "Expense cost field is required"		

14	User submit form that contains alphabets in Expense cost	Expense cost: 145@65	System displays "Expense cost is invalid"	System displays "Expense cost is invalid"		
----	--	----------------------	---	---	--	--

15	User submit form that contains negative number in Expense cost	Expense cost: -18768	System displays" Expense cost is invalid"	System displays" Expense cost is invalid"		
16	User submit form with valid Customer, Paid Amount, Product field, Product quantity, Per unit price, Expense title and Expense cost fields.	Customer: Paid Amount: Product: Product quantity: Per Unit price: Expense title: Expense Cost:	System displays" Sale added successfully"	System displays" Sale added successfully"		
<b>Post Condition:</b> Sale added successfully in system.						

**Table 35(7.1): Test Case 11 Return Sale**

Test Case ID: 11.1			Test Design By: M Zeeshan Fiaz			
Test Module Name: Return sale			Test Design date: 12/7/22			
Test Priority: high			Test Executed By: M Zeeshan Fiaz			
Test title: Valid Return sale Information			Test Executed Date: 12/7/22			
Description: Test the return sale module with valid and invalid data						
Pre-Condition:						
Dependencies:						
Step	Test Step	Test Date	Expected Result	Actual Result	Status	Notes

1	Navigate to Sale management page		Sale management is displayed	Sale management is displayed	Pass	
2	User submit form without selecting Item field	Item Id:	System displays "Item field is required"	System displays "Item field is required"		
3	User submit form with empty return quantity field	Return quantity:	System displays" Return quantity field is required"	System displays" Return quantity field is required"		
4	User submit form that contains alphabets in Return quantity field	Return quantity: 128@962	System displays" Return quantity is invalid"	System displays" Return quantity is invalid"		
5	User submit form that contains Negative number in Return quantity field	Return quantity: -128648	System displays" Return quantity is invalid"	System displays" Return quantity is invalid"		

6	User submit form with valid Item id and Return quantity fields.	Item id: Return quantity:	System displays" Sale return added successfully"	System displays" Sale return added successfully"		
<b>Post Condition:</b> Sale return added successfully in system.						

**Table 36(7.1): Test Case 12 Manage Products**

Test Case ID: 12			Test Design By: M Zeeshan Fiaz			
Test Module Name: Manage products			Test Design date: 12/7/22			
Test Priority: high			Test Executed By: M Zeeshan Fiaz			
Test title: Valid product information			Test Executed Date: 12/7/22			
Description: Test the manage product module with valid and invalid data						
Pre-Condition:						
Dependencies:						
Step	Test Step	Test Date	Expected Result	Actual Result	Status	Notes
1	Navigate to Product management page		Product management is displayed	Product management is displayed	Pass	
2	User submit form without selecting product title field	Product title:	System displays “Product title is required”	System displays “Product title is required”		
3	User submit form without selecting product code field	Product code:	System displays “Product code is required”	System displays “Product code is required”		
4	User submit form that contains valid Product title and product code fields	Product title: Product code:	System displays” Product added successfully”	System displays” Product added successfully”		



**Post Condition:** Product added successfully in system.

**Table 37(7.1): Test Case 13 Manage Raw Material**

Test Case ID: 13			Test Design By: M Zeeshan Fiaz			
Test Module Name: Manage Raw Material			Test Design date: 12/7/22			
Test Priority: high			Test Executed By: M Zeeshan Fiaz			
Test title: Valid raw material information			Test Executed Date: 12/7/22			
Description: Test the manage raw material module with valid and invalid data						
Pre-Condition:						
Dependencies:						
Step	Test Step	Test Date	Expected Result	Actual Result	Status	Notes
1	Navigate to Raw Material Management page		Raw Material Management is displayed	Raw Material Management is displayed	Pass	
2	User submit form without selecting Department title field	Department title:	System displays “Department title is required”	System displays “Department title is required”		
3	User submit form with valid Raw Material title field	Raw Material title:	System displays “Raw Material added successfully”	System displays “Raw Material added successfully”		
Post Condition: Raw Material added successfully						

**Table 38(7.1): Test Case 14 Manage Departments**

Test Case ID: 14			Test Design By: M Zeeshan Fiaz			
Test Module Name: Manage Departments			Test Design date: 12/7/22			
Test Priority: high			Test Executed By: M Zeeshan Fiaz			
Test title: Valid Department information			Test Executed Date: 12/7/22			
Description: Test the manage department module with valid and invalid data						
Pre-Condition:						
Dependencies:						
Step	Test Step	Test Date	Expected Result	Actual Result	Status	Notes
1	Navigate to Department Management page		Department Management is displayed	Department Management is displayed	Pass	
2	User submit form without selecting Raw Material title field	Raw Material title:	System displays “Raw Material title is required”	System displays “Raw Material title is required”		
3	User submit form with valid Raw Material title field	Raw Material title:	System displays “Department added successfully”	System displays “Department added successfully”		
Post Condition: Department added successfully						

**Table 39(7.1): Test Case 15 Search Attendance**

<b>Test Case ID:</b> 16		<b>Test Design By:</b> M Zeeshan Fiaz	
<b>Test Module Name:</b> Search Attendance		<b>Test Design date:</b> 12/7/22	
<b>Test Priority:</b> high		<b>Test Executed By:</b> M Zeeshan Fiaz	

<b>Test title:</b> Valid Search attendance information			<b>Test Executed Date:</b> 12/7/22			
<b>Description:</b> Test the search attendance module with valid and invalid data						
<b>Pre-Condition:</b>						
<b>Dependencies:</b>						
Step	Test Step	Test Date	Expected Result	Actual Result	Status	Notes
1	Navigate to Attendance Management page		Attendance Management is displayed	Attendance Management is displayed	Pass	
2	User submit form without selecting Date field	Date:	System displays “Date is required”	System displays “Date is required”		
3	User submit form with valid Date field	Date:	System displays “Attendance searched successfully”	System displays “Attendance searched successfully”		
<b>Post Condition:</b> Attendance searched successfully.						

**Table 40(7.1): Test Case 16 Marked Attendance**

<b>Test Case ID:</b> 16.1	<b>Test Design By:</b> M Zeeshan Fiaz
<b>Test Module Name:</b> Marked Attendance	<b>Test Design date:</b> 12/7/22
<b>Test Priority:</b> high	<b>Test Executed By:</b> M Zeeshan Fiaz
<b>Test title:</b> Valid Marked attendance information	<b>Test Executed Date:</b> 12/7/22
<b>Description:</b> Test the marked attendance module with valid and invalid data	
<b>Pre-Condition:</b>	
<b>Dependencies:</b>	

Step	Test Step	Test Date	Expected Result	Actual Result	Status	Notes
1	Navigate to Attendance Management page		Attendance Management is displayed	Attendance Management is displayed	Pass	
2	User submit form without selecting Employee Id field	Employee ID:	System displays "Employee ID is required"	System displays "Employee ID is required"		
3	User submit form that contains length of 13 digits in Employee ID field	Employee ID: 156 987 123 3214	System displays "Invalid Employee ID"	System displays "Invalid Employee ID"		
4	User submit form that contains alphabets in Employee ID field	Employee ID: 123@786 78681	System displays "Invalid Employee ID"	System displays "Invalid Employee ID"		
5	User submit with valid Employee Id field	Employee ID:	System displays "Attendance marked successfully"	System displays "Attendance marked successfully"		
<b>Post Condition:</b> Attendance marked successfully.						

**Table 41(7.1): Test Case 17 Add/Update Employees**

<b>Test Case ID:</b> 17	<b>Test Design By:</b> M Zeeshan Fiaz
<b>Test Module Name:</b> Add/Update Employees	<b>Test Design date:</b> 12/7/22
<b>Test Priority:</b> high	<b>Test Executed By:</b> M Zeeshan Fiaz
<b>Test title:</b> Valid ADD/UPDATE Employees information	<b>Test Executed Date:</b> 12/7/22

Description: Test the Add/Update module with valid and invalid data						
Pre-Condition:						
Dependencies:						
Step	Test Step	Test Date	Expected Result	Actual Result	Status	Notes
1	Navigate to Employee Management page		Employee Management is displayed	Employee Management is displayed	Pass	
2	User submit form with empty First name field	First name:	System displays “First name is required”	System displays “First name is required”		
3	User submit form that contains number or special characters in First name field	First Name: Abdul@Habib786	System displays “Invalid First name”	System displays “Invalid First name”		
4	User submit form with empty Last name field	Last name:	System displays” Last name is required”	System displays” Last name is required”		
5	User submit form that contains number or special characters in Last name field	Last name:	System displays ”Invalid Last name”	System displays ”Invalid Last name”		
6	User submit form with empty CNIC field	CNIC:	System displays” CNIC field is required”	System displays” CNIC field is required”		

7	User submit form that contains the special characters or alphabets in CNIC field	CNIC: 35201-78@7864-0	System displays” Invalid CNIC”	System displays” Invalid CNIC”		
8	User submit form whose CNIC length is not 13 digits	CNIC: 35301-55568798-9	System displays“ Invalid CNIC”	System displays“ Invalid CNIC”		
9	User submit form with empty Mobile Number field	Mobile Number:	System displays” Mobile Number required” is	System displays” Mobile Number required” is		
10	User submit form that contains alphabets or special characters in Mobile Number field	Mobile Number: 0316-400@863	System displays” Mobile Number required” us	System displays” Mobile Number required” us		
11	User submit the form where mobile number length is not 11	Mobile Number: 0321-58461295	System displays” Mobile Number invalid” is	System displays” Mobile Number invalid” is		
12	User submit form with empty address field	Address:	System displays” Address required” is	System displays” Address required” is		

13	User submit form with invalid Email Address field	Address: 123 765#gmail.com	System displays" Email Address is required"	System displays" Email Address is required"		
14	User submit form that contains special characters or alphabets in Emergency number field	Emergency Number:	System displays" Invalid Emergency Number"	System displays" Invalid Emergency Number"		
15	User submit form where Emergency number length is not 11	Emergency Number: 0321-23541125	System displays" Invalid Emergency Number"	System displays" Invalid Emergency Number"		
16	User submit form with empty department field	Department:	System displays" Department is required"	System displays" Department is required"		
17	User submit form with empty Salary field	Salary:	System displays" Salary is required"	System displays" Salary is required"		
18	User submit form that contains alphabets or special characters in Salary field	Salary: 174\$4521 Pkr	System displays" Salary is Invalid"	System displays" Salary is Invalid"		

19	User submit form with valid First name, Last name, CNIC, Mobile Number, Address, Emergency Number, Department and Salary	First Name: Last Name: CNIC: Mobile Number: Address: Emergency Number: Department: Salary:	System displays” Employee Added/Updated successfully”	System displays” Employee Added/Updated successfully”		
<b>Post Condition:</b> Employees Added/Updated successfully.						

**Table 42(7.1): Test Case 18 Add/Update Production**

Test Case ID: 19			Test Design By: M Zeeshan Fiaz			
Test Module Name: Add/Update Production			Test Design date: 12/7/22			
Test Priority: high			Test Executed By: M Zeeshan Fiaz			
Test title: Valid Add/Update Production information			Test Executed Date: 12/7/22			
Description: Test the add/update production module with valid and invalid data						
Pre-Condition:						
Dependencies:						
Step	Test Step	Test Date	Expected Result	Actual Result	Status	Notes
1	Navigate to Production Management page		Production Management is displayed	Production Management is displayed	Pass	
2	User submit form with empty Product code field	Product Code:	System displays “Product code is required”	System displays “Product code is required”		



3	User submit form with empty Product quantity field	Product quantity:	System displays “Product quantity required” is	System displays “Product quantity required” is		
4	User submit form that contains alphabets in Product quantity field	Product quantity: 1234sa45	System displays” Invalid Product quantity”	System displays” Invalid Product quantity”		
5	User submit with negative number product quantity field	Product quantity: -12563	System displays” Invalid Product quantity”	System displays” Invalid Product quantity”		
6	User submit form with empty Raw Material field	Raw Material:	System displays” Raw Material is required”	System displays” Raw Material is required”		
7	User submit form with empty Raw Material quantity field	Raw Material quantity:	System displays” Raw Material quantity is required”	System displays” Raw Material quantity is required”		
8	User submit form that contains alphabets in Raw Material quantity field	Raw Material quantity: 231k212	System displays” Invalid Raw Material quantity”	System displays” Invalid Raw Material quantity”		
9	User submit form with negative number in Raw Material quantity field	Raw Material quantity: -1346232	System displays” Invalid Raw Material quantity”	System displays” Invalid Raw Material quantity”		

10	User submit form with empty Expense title field	Expense title:	System displays” Expense title is required”	System displays” Expense title is required”		
11	User submit form with empty Expense cost field	Expense cost:	System displays” Expense cost is required”	System displays” Expense cost is required”		
12	User submit form that contains alphabets in Expense cost field	Expense cost:	System displays” Expense cost is invalid”	System displays” Expense cost is invalid”		
13	User submit form that contains the negative number in expense cost field	Expense cost: -12162323	System displays” Invalid Expense cost”	System displays” Invalid Expense cost”		
14	User submit form with valid Product code, Product quantity, Raw Material, Raw Material quantity, Expense title and Expense cost fields	Product code: Product quantity: Raw Material: Raw Material quantity: Expense title: Expense cost:	System displays” Production added/updated successfully”	System displays” Production added/updated successfully”		
<b>Post Condition:</b> Production added/updated successfully.						

7.2 Decision Table

7.2.1 Code snippet

7.2.2 Decision coverage table

Table 43(7.2): Decision Coverage Table

Conditions	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7	Rule 8
New customer (15%)	T	T	T	T	F	F	F	F
Loyalty card (10%)	T	T	F	F	T	T	F	F
Coupon (20%)	T	F	T	F	T	F	T	F
Actions								
Discount (%)	X	X	20	15	30	10	20	0

7.3 Traceability Matrix

7.3.1 RID vs UCID (requirements vs use cases)

Table 44(7.3): RID vs UCID

U C I D/ R I D	R 0 1	R 0 2	R 0 3	R 0 4	R 0 5	R 0 6	R 0 7	R 0 8	R 0 9	R 1 0	R 1 1	R 1 2	R 1 3	R 1 4	R 1 5	R 1 6	R 1 7	R 1 8	R 1 9	R 2 0	R 2 1	R 2 2	R 2 3	R 2 4	R 2 5	R 2 6
----------------------------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------

U C- 1	✓																			✓	✓	✓	✓	✓	✓	✓
U C- 2		✓																		✓	✓		✓	✓	✓	✓
U C- 3			✓																	✓	✓		✓	✓	✓	✓
U C- 4				✓																✓	✓		✓	✓	✓	✓
U C- 5					✓															✓	✓		✓	✓	✓	✓
U C- 6						✓														✓	✓		✓	✓	✓	✓
U C- 7							✓													✓	✓		✓	✓	✓	✓
U C- 8								✓												✓	✓		✓	✓	✓	✓
U C- 9									✓											✓	✓		✓	✓	✓	✓
U C- 10										✓										✓	✓		✓	✓	✓	✓
U C- 11											✓									✓	✓		✓	✓	✓	✓

U C- 12													✓							✓	✓			✓	✓	✓	✓
U C- 13														✓						✓	✓			✓	✓	✓	✓
U C- 14															✓					✓	✓			✓	✓	✓	✓
U C- 15																✓				✓	✓			✓	✓	✓	✓
U C- 16																	✓			✓	✓			✓	✓	✓	✓
U C- 17																		✓		✓	✓			✓	✓	✓	✓
U C- 18																			✓	✓	✓			✓	✓	✓	✓
U C- 19																				✓	✓	✓			✓	✓	✓

