ERP SYSTEM

An ERP System for a cloth Manufacturing Company

Created on June 2023

Presented by:

Omar Hossam Ahmed Mohamed	201900506
Mennatullah Mohamed Farouk Hussein	201900850
Fatema-elZahraa Mohammed	201900560
Hussein Khalid Hussein Abbas	201900271
Hossam Fawzi Ahmad Abdelfattah	201900263
Ahmed Reda Saeed Abdul Menem	201900035

Acknowledgment

We would like to express our deepest gratitude to our supervisor **Dr. Leila Abdelhamid** for her continuous support, motivation and immense knowledge. Her guidance helped us in all the time of research and writing of this project. We could not have imagined having a better advisor and mentor for our undergraduate study.

Besides our advisor, we would like to thank **the faculty of computer science and artificial intelligence at Helwan University** for providing all the facilities and necessary resources required for this project. Our sincere thanks go to the Information Systems department for providing us with the background knowledge and tools that made this project possible.

We would also like to thank **the department's teaching assistants** for their kind support and suggestions during the course of this project. Their assistance and insightful comments were invaluable in helping us overcome challenges that we encountered and keeping our progress on track.

Last but not the least, we wish to thank **all the people who have supported us to complete this project directly or indirectly**.

Abstract

This project presents the design and implementation of an enterprise resource planning (ERP) system for managing business processes and crucial information across various departments within an organization. The system contains modules for financial management, human resources, supply chain management, inventory management system and customer relationship management. The backend is developed using the .NET Core framework while the frontend user interface utilizes React.js. The system aims to integrate various departments and functions across an organization to improve information flow, responsiveness, and production. An entity-relationship model was developed to conceptualize system data requirements. The system utilizes a relational database to store and manage information. The design focuses on ease of use, flexibility and scalability to accommodate future growth. The ERP system has the potential to increase productivity, efficiency and reduce costs through automation and integration of key processes. It also provides key performance metrics and insights to aid management decision making across the organization.

Table Of Content

List of Tables

Chapter 1: Introduction

1.1 Overview

This ERP system aims to integrate and automate key business processes within the organization. It combines functions for financial management, human resources, supply chain, inventory management, and customer relationship management. The system works to improve efficiency, visibility and decision making across departments.

1.2 Objectives

The objectives of this ERP system are to:
- Integrate data across departments to facilitate information sharing
- Automate manual and repetitive workflows to save time and reduce errors
- Provide role-based access to data to allow employees insights for better performance
- Streamline operations through centralized processes
- Surface insights through reporting to improve planning and decision making

1.3 Purpose

The purpose of the ERP system is to improve the management of the organization's resources, operations and customer relationships through the integration and automation of business processes and data.

1.4 Scope

The scope of the ERP system includes:
- Designing the database schema and entity models
- Developing the backend system using .NET Core
- Building the frontend using React
- Integrating the various modules
- Testing and documenting the system

1.5 General Constraints

Constraints faced in developing the ERP system include:
- Time constraints during development and testing
- Resource constraints in accessing SMEs for requirements
- Difficulty simulating real-world data volumes for performance testing
- Limited user experience testing due to tester availability issues

Chapter 2: Project "Planning and analysis"

2.1 Project planning

2.1.1 Feasibility Study

A feasibility study was conducted to determine the viability of the project. The study considered technical, economic and schedule feasibility.

Technical Feasibility

The .NET Core and React technologies chosen for the project are technically feasible as they are robust, widely used and have a large community and support ecosystem. The database and other required technologies are also deemed to be technically feasible. The team members have the required technical skills and expertise to develop the proposed ERP system.

Economic Feasibility

The estimated costs of the project, including labor, hardware, software licenses and other expenses, were evaluated against the potential benefits. The cost of the current manual system in terms of errors, productivity losses and inefficiencies were also considered. The ERP system is expected to generate sufficient returns to justify the investment costs over the long term.

Schedule Feasibility

A project timeline of x months was proposed based on the project scope and team size. The scheduled was deemed feasible given the availability of resources and limited dependencies on external factors. Key milestones and deliverables were identified to track progress against the plan.

Based on the feasibility study across these factors, the ERP system project was determined to be viable and approved to proceed. The study will be revisited periodically to recalibrate cost estimates and ensure the project remains feasible.

2.1.2 Estimated Cost

The total estimated cost for developing the proposed ERP system depends on factors like the project scope, required functionality, hardware specifications, number of users, and more. Costs will be determined through market research, vendor quotes, and internal benchmarks where relevant.

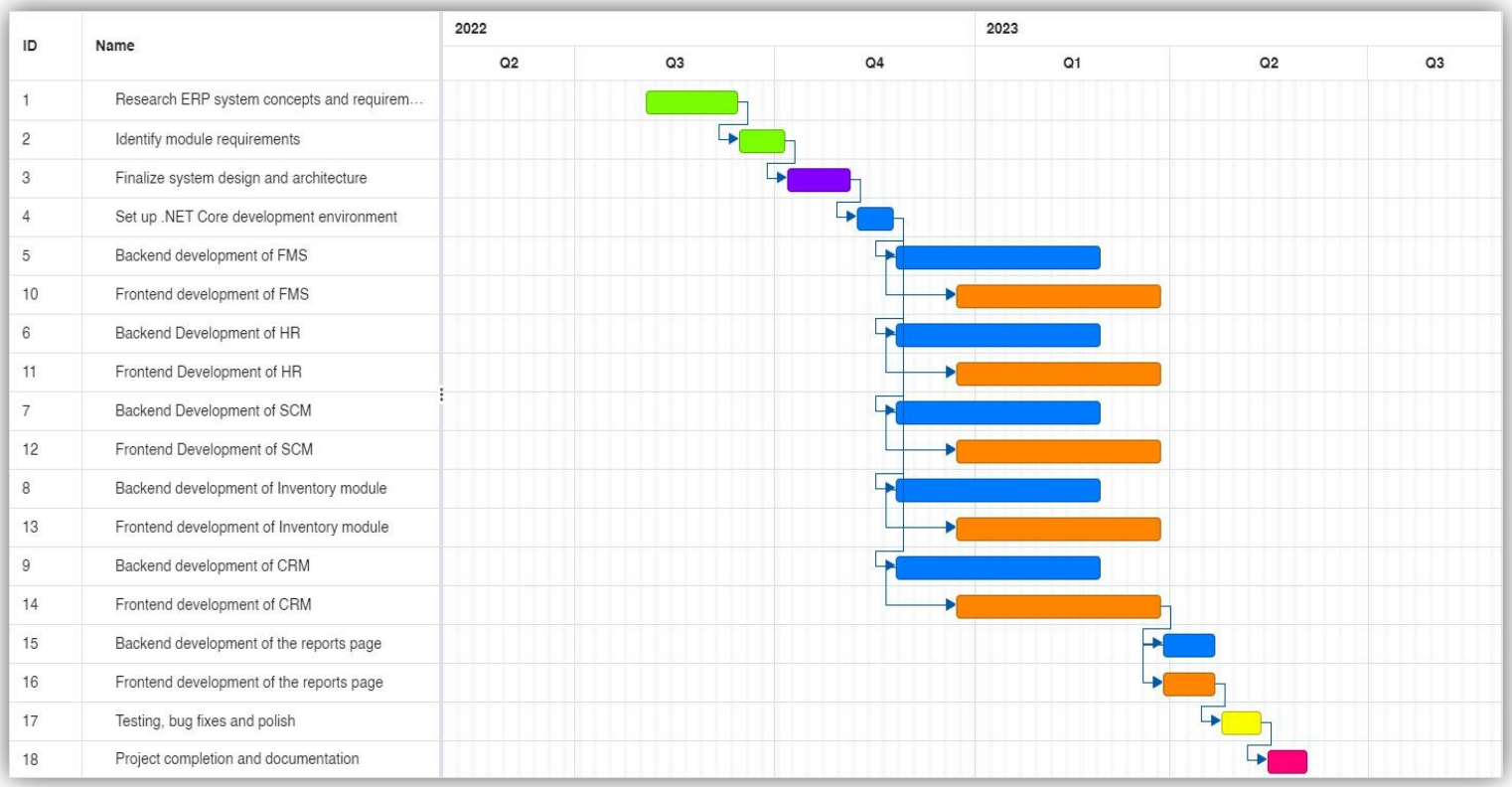
The cost estimate will cover major expense categories such as:

- Hardware
- Software
- Labor
- Other expenses including facilities, travel, training, and contingencies

The cost estimate will be treated as an initial projection to gauge feasibility. It will be revisited periodically and recalibrated based on better information and actual expenses incurred during implementation. Any significant variances from the estimate will be analyzed and corrective actions taken where needed.

The final cost of the project will depend on the approved scope, specifications, and resources allocated during implementation.

2.1.3 Gantt Chart



2.2 Analysis of existing system

The current manual system based on spreadsheets has several limitations:

• Inefficient and time-consuming manual processes lead to reduced productivity. On average, workers spend x hours per week on manual data entry and reconciliation.

• Data is siloed in different spreadsheets, making it difficult to track performance holistically or share information between departments in real time.

• Limited visibility into key metrics and performance indicators hinders management's ability to make data-driven decisions.

• The spreadsheet-based system is prone to errors from data entry mistakes, version control issues, and lack of input validation.

• The existing system lacks the flexibility and scalability to accommodate future business growth and expansion plans.

These limitations demonstrate the need for an automated ERP system to replace the current manual processes.

2.3 Need for new system

An ERP system is required to address the limitations of the existing spreadsheet-based system by:

• Automating workflows to improve efficiency and reduce manual work

• Integrating data into a centralized database for real-time information sharing and analysis

• Providing role-based dashboards and reports to make key metrics and KPIs visible

• Enforcing data controls, input validation and version management to minimize errors

• Utilizing a robust, scalable system architecture that can grow with business needs

The ERP system will help streamline operations, improve data-driven decision making, enhance compliance and reduce costs through automation. A centralized, integrated system will allow for greater visibility, control and performance management across the organization.

2.4 Analysis of the new system

2.4.1 User requirements

• Financial management users require the system to manage activities like accounts payable, receivable, payroll, and bookkeeping.

• Human resources users need the system to manage employee information, benefits, attendance, and performance.

• Supply chain users want the ability to place purchase orders, track inventory levels, and manage vendor relationships.

• Inventory management users require an inventory management system to keep track of inventory items, stock levels, reorders, item locations, and item attributes.

• Customer relationship management users need functionality to track customer accounts, cases, opportunities, and service requests.

Capturing the requirements of users across these key modules helped inform the functional and interface design of the ERP system.

2.4.2 System Requirements

- Windows 7 or higher, Mac OS X 10.10 or higher
- 4GB of RAM (8GB recommended)
- Quad-core processor
- Microsoft .NET Core 5runtime environment
- A modern web browser like Chrome, Firefox, Edge or Safari
- Stable internet connection
- SQL Server database or compatible DBMS to store data

The system is designed to run on modern desktop and laptop computers. The minimum hardware specifications listed above will allow the system to run properly but higher specifications are recommended for optimal performance. The .NET Core and Node environments must be installed to run the backend and frontend applications. A web browser is needed to access the user interface. An internet connection is needed for initializing the application and accessing SaaS solutions. A database is required to store and manage company data.

2.4.3 Functional Requirements

Financial module requirements

Requirement ID:	FR001
Requirement Title:	Create journal entries.
Requirement Rationale:	Journal entries are used to record financial transactions which is the core of accounting.
Requirement Description:	Journal entries record the date they were made on, the accounts the entry affects, the value of the debit and credit for each account, and a description for what this journal entry is for. The total debit must equal the total credit, in accordance with Double-entry bookkeeping. A unique identifying number is generated by the system for each entry. Journal entries are created by the user.

Requirement ID:	FR002
Requirement Title:	Create accounts.
Requirement Rationale:	An account is a record in an accounting system that tracks the financial activities of a specific asset, liability, equity, revenue, or expense.
Requirement Description:	Each account has a name/description, a category or more that it belongs to, and whether it increases with debits or credits. A unique identifying number is generated by the system for each account. Accounts are created by the user.

Requirement ID:	FR003
Requirement Title:	Create categories.
Requirement Rationale:	Used to organize accounts for further ease of filtering when generating financial statements or reviewing the chart of accounts.
Requirement Description:	The user specifies the category's name and description. The system generates a unique ID for the category.

Requirement ID:	FR004
Requirement Title:	Generate Journal.
Requirement Rationale:	View every journal entry in chronological order.
Requirement Description:	The system lists journal entries from every account by chronological order.

Requirement ID:	FR005
Requirement Title:	Generate a chart of accounts.
Requirement Rationale:	Keeping track of the accounts in the system.
Requirement Description:	An index of all the financial accounts in the general ledger, shows each account's ID, name/description, and categories. generated by the system.

Requirement ID:	FR006
Requirement Title:	Generate a general ledger.
Requirement Rationale:	The general ledger is the master set of accounts that summarize all transactions occurring within an entity.
Requirement Description:	The general ledger keeps track of and displays all financial transactions that occurred in the system organized into their appropriate accounts. generated by the system.

Requirement ID:	FR007
Requirement Title:	Automatic posting of journal entries .
Requirement Rationale:	Keeping all parts of the system in sync, automating laborious work.
Requirement Description:	The system must record journal entries into their appropriate accounts, and the effect must automatically show in the general ledger.

Requirement ID:	FR008
Requirement Title:	Generate a trial balance**.**
Requirement Rationale:	Trial balance is used to make sure the value of debits equals the value of credits.
Requirement Description:	Accounts from the general ledger are listed with either a debit balance or a credit balance, the total of the debit column and the credits column is then calculated. generated by the system.

Requirement ID:	FR009
Requirement Title:	Create a financial statement template.
Requirement Rationale:	The template is later used to generate statements.
Requirement Description:	The user specifies the title of the statement, the accounts included in the statement, and if needed, the accounts can be grouped into sections each with its own title and total.

Requirement ID:	FR010
Requirement Title:	Generate a financial statement from a template.
Requirement Rationale:	Financial statements showcase the financial position, performance and changes in the financial position of an enterprise.
Requirement Description:	The system uses the info from a financial statement template and generates the appropriate result.

Human Resource (HR) Module Requirements

Requirement ID:	FR001
Requirement Title:	Storing employee information in DBMS.
Requirement Rationale:	Know the workforce's data.
Requirement Description:	system should store the details about the employees (ID, Full Name, tax withholding, hours worked, salary) information in one location and can be accessed immediately by the authorized users(HR Manager).

Requirement ID:	FR002
Requirement Title:	Employees Time Management.
Requirement Rationale:	Show the time each employee's clocks in and out.
Requirement Description:	System should record the attendence and absence of employees, assign tasks time, and detect the deadline. In addition the system helps you to generate balanced shift schedules for all personnel, factoring in leave and holidays, sickness and other special occasions. All changes to shifts or work-related events can be logged to ensure fair compensation.

Requirement ID:	FR003
Requirement Title:	Employee training and development.
Requirement Rationale:	It is useful for new employees to train them if they need training.
Requirement Description:	System can provide training for new employees to learn how they work if they need it. If the system needs those employees, on the other hand the system will show the devolpment of its employees. By analyzing the performance of previous and existing employees, the system can point out areas that need the most attention with regard to training and development.

Requirement ID:	FR004
Requirement Title:	Bounces and Deduction calculations .
Requirement Rationale:	Automatically assign the bounces and deduction for employees.
Requirement Description:	System should calculate automatically the bounces or deduction profits of employees in some cases : When an employee works overtime, works on holidays, finishes his work before the deadline , he will gain bonus profit. On the other hand, when an employee doesn't finish his work ,absence, take vacation or give excuses , his salary will be dedcuted.

Requirement ID:	FR005
Requirement Title:	Automatic Payroll.
Requirement Rationale:	Automatically reports of employees payroll.
Requirement Description:	System should provide an HR manager with reports of employees payroll (Performance Incentives and Allowance Management, Reimbursement & Claims , and Income Tax calculation and Management).

Requirement ID:	FR006
Requirement Title:	Recruitment Management.
Requirement Rationale:	Help the HR manager to make his dceision for recruitment .
Requirement Description:	System will allow you to set up a screen to analyze your overall business performance, its objectives and help determine how many employees you need in each department, or specific to a project. The system gives you the place to store details of the skills needed for each job and build in workflows to guide HR recruitment processes. Additionally, the system can monitor your onboarding processes and employee performance. Such information is useful and can be used to inform future recruitment exercises.

Requirement ID:	FR007
Requirement Title:	Reports and analysis.
Requirement Rationale:	Help the HR manager to show all reports of the staff.
Requirement Description:	System should generate reports to make better decisions in HR.

Supply Chain Management (SCM) Module Requirements

Requirement ID:	FR001
Requirement Title:	Storing Raw materials and Finished products information.
Requirement Rationale:	getting quick and detailed information about items available in inventory.
Requirement Description:	System should store the required details about the raw materials and the finished products in the Inventory including ID, Name, Category, Description, Sales Price, Purchase Price, number of items in Inventory, Reordering Point.

Requirement ID:	FR002
Requirement Title:	Tracking Manufacturing Process.
Requirement Rationale:	Track raw materials undergoing in the manufacturing process.
Requirement Description:	System should track raw materials undergoing the manufacturing process, and notify users when the manufacturing process is done.

Requirement ID:	FR003
Requirement Title:	Storing Suppliers and Distributors Information.
Requirement Rationale:	Easily contact them when required.
Requirement Description:	System should store required information about the suppliers and distributors including their names, description about them, their emails, and their contact information.

Requirement ID:	FR004
Requirement Title:	Creating Product Manufacturing order.
Requirement Rationale:	prevent products-outage.
Requirement Description:	System should send raw materials to the manufacturing process when products reach reordering point in order to produce a bulk of finished products. Once the products are ready, inventory levels must be updated eventually.

Requirement ID:	FR005
Requirement Title:	Order Raw Materials from Supplier
Requirement Rationale:	prevent raw materials-outage
Requirement Description:	System should allow user to order raw material bulk from the suitable supplier when the materials in the inventory reaches the reordering point, once the materials are arrived. inventory must be updated eventually.

Requirement ID:	FR006
Requirement Title:	Create Distribution Order
Requirement Rationale:	meet demand
Requirement Description:	System should allow user to create Manufacturing orders and to be able to track these orders at their different states

Inventory Management Module Requirements

Requirement ID:	FR001
Requirement Title:	Manage inventory for raw materials
Requirement Rationale:	System allows user to view and track raw materials in inventory
Requirement Description:	System will be able to view, add, update raw materials in the inventory and to get notified is such raw material has reached the ROP

Requirement ID:	FR002
Requirement Title:	Manage inventory for finished Products
Requirement Rationale:	System allows user to view and track products in inventory
Requirement Description:	System will be able to view, add, update products in the inventory and to get notified is such a product has reached the ROP

Customer Relation Management (CRM) Module Requirements

Requirement ID:	001
Requirement Title:	**Contact management.**
Requirement Rationale	System shall be allowed to collect, store and act on data from prospects/leads/customers. System shall generate a profile of the customer with the collected data.
Requirement Description:	Any data recorded by the system (name/contact/type) about the (prospects/leads/customers) should be collected and stored in DBMS, tracked/viewed by the user, and used in any future operations.

Requirement ID:	002
Requirement Title:	**Customer Interaction Tracking.**
Requirement Rationale:	System shall provide tools (Dashboard) for handling and keeping track of your interactions with customers Users can track past interactions, attach relevant files, and view information about the contact.
Requirement Description:	Any transaction/interaction (such as invoices, purchase history and order status), is done on the System should be kept tracking and noticed by the Dashboards (act as a central hub of information) records of a customer's interactions.

Requirement ID:	003
Requirement Title:	**Marketing and Campaign**
Requirement Rationale:	System shall connect with the top standalone marketing automation solutions. System shall create, deliver, and track multi-channel marketing campaigns.

	a Campaign: a (marketing campaign) is a strategic sequence of steps and activities that promote our products/services, with a specific goal in mind.
Requirement Description:	The System shall start (social-media/email/event-based) marketing campaigns to attract new customers.

Requirement ID:	004
Requirement Title:	**Workflow Automation and Employee Management**
Requirement Rationale:	- System shall allow users to create to-do lists (helps users in their daily objectives). - System shall allow users to access a shared calendar list (helps users in noticing future appointments System shall allow users to connect with customers on social media.
Requirement Description:	The System should allow workflow automation to make it easier for the employee to finish daily work faster.

Requirement ID:	005
Requirement Title:	**Generate leads.**
Requirement Rationale:	- The leads: are the to-be customers that are acquired by the System. - System shall attract new potential customers (leads). - System shall provide pre-defined criteria to route them to the appropriate sales rep. - System shall generate leads and properly manage them. - System shall capture leads and store the data directly in the system.
Requirement Description:	The System shall come with embeddable web forms to capture leads and store it directly into the system and view their information in a list-view form.

Requirement ID:	006
Requirement Title:	**Sales**
Requirement Rationale:	- System shall manage sales opportunities from the first contact to sales closure.
Requirement Description:	The System shall manage/handle/view purchased products/services shall make a collaboration between sales and marketing teams.

2.4.4 Non-Functional Requirements

Usability
System must provide a convenient User Interface and user experience for the intended user, and full documentation must be supplied for the user too.

Reliability
System Failure rate must be 0% and do not exceed 5% for better SW experience and Business performance

Performance
Systems response time should not exceed 5 seconds AND System overall size should be max 2 GB.

Supportability
System must support all different platforms and all different Web Browsers.

All test cases must be maintained and covered before publishing the product in the market. Feature maintenance must me tracked after the application goes into production

Implementation
This web application must be implemented using

- React JS for front-end
- .Net Framework for back-end
- SQL Server For DB

Efficiency

4 GB ram, 1 gigahertz (GHz) or faster with 2 or more cores on a compatible 64-bit processor or System on a Chip (SoC) as minimum requirement.

Each module should not exceed 300 lines of code.

2.5 Advantages of the new system

• Integrated solution: By integrating financial management, HR, supply chain and CRM into a single system, data will be consistent across modules. This avoids data duplication and reconciliation issues.
• Automated workflows: Many routine tasks can be automated through the system, reducing manual efforts and human errors.

• Real-time information: Employees will have access to up-to-date information for better decision making.
• Increased productivity: Employees can manage tasks more efficiently through the system, leading to higher productivity.

• Improved controls: The new system will have stronger security controls and validation rules to reduce risks.

• Customization: The .NET Core and React technologies used allow for easy customization and extension of the system to meet evolving business needs.
• Scalability: The system is designed to scale efficiently to support potential business growth.

• Standardization: A standardized ERP system ensures consistent processes across departments.

• Visibility: Managers will have enhanced visibility into key performance metrics and operations.

• Reporting: The system's reporting features will provide data needed for analysis, planning and compliance.

2.6 Risk and Risk Management

Like any complex IT project, there are risks associated with implementing the new ERP system. However, proper risk management strategies can be put in place to minimize issues and maximize the chances of success.
Potential risks include:
• Budget overrun: There is a possibility of going over the initial budget estimates due to unforeseen issues or scope creep. Regular budget monitoring and change control procedures can help manage this risk.

• Schedule delays: Unanticipated challenges can lead to delays in the project schedule. Strong scheduling practices and buffer time in the schedule can mitigate this risk.

• Compatibility issues: There is a chance that the system may not integrate properly with existing legacy systems or third-party solutions. Thorough testing and prototyping can identify potential incompatibilities early.

• User resistance: Employees may be reluctant to adapt to the new system. Proper user training, feedback mechanisms and change management strategies can reduce user resistance.

• Vendor performance: There is a risk that the system vendors may not deliver as expected. Clear contracts and SLAs, along with continuous vendor evaluation, can manage vendor risk.

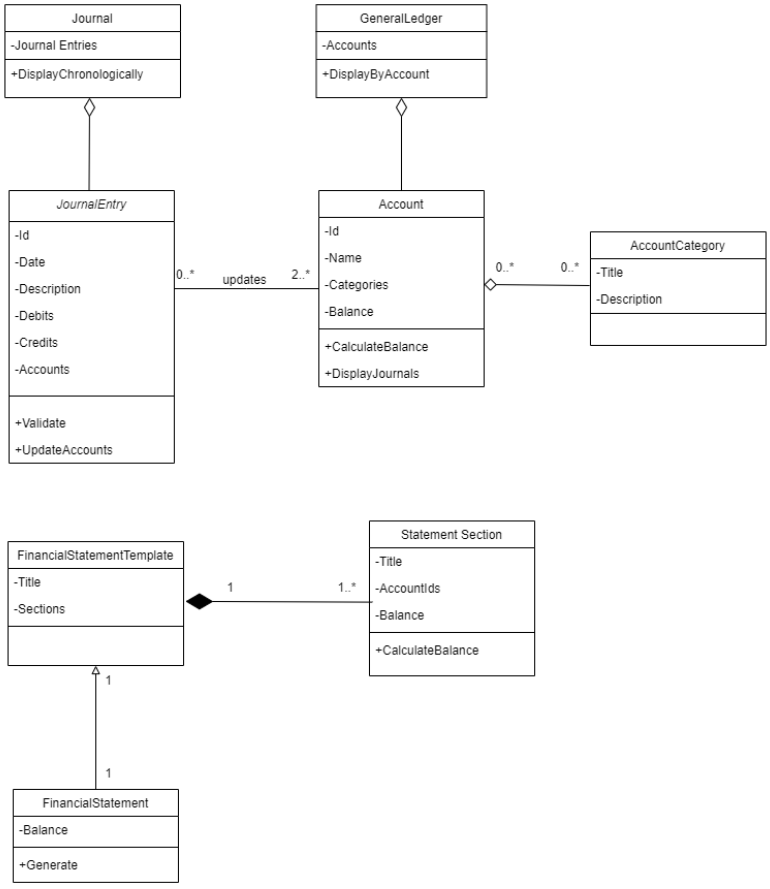
• Data migration issues: Migrating data from old systems to the new ERP can be complex and error-prone. Proper data validation, quality checks and testing of the migration process can minimize data issues.

To manage these risks, a comprehensive risk management plan will be developed that defines risk monitoring activities, contingency plans and fallbacks. Regular risk reviews and status reports will keep risks under control throughout the project lifecycle. With proper planning and execution, the risks associated with the new ERP system can be managed to an acceptable level.

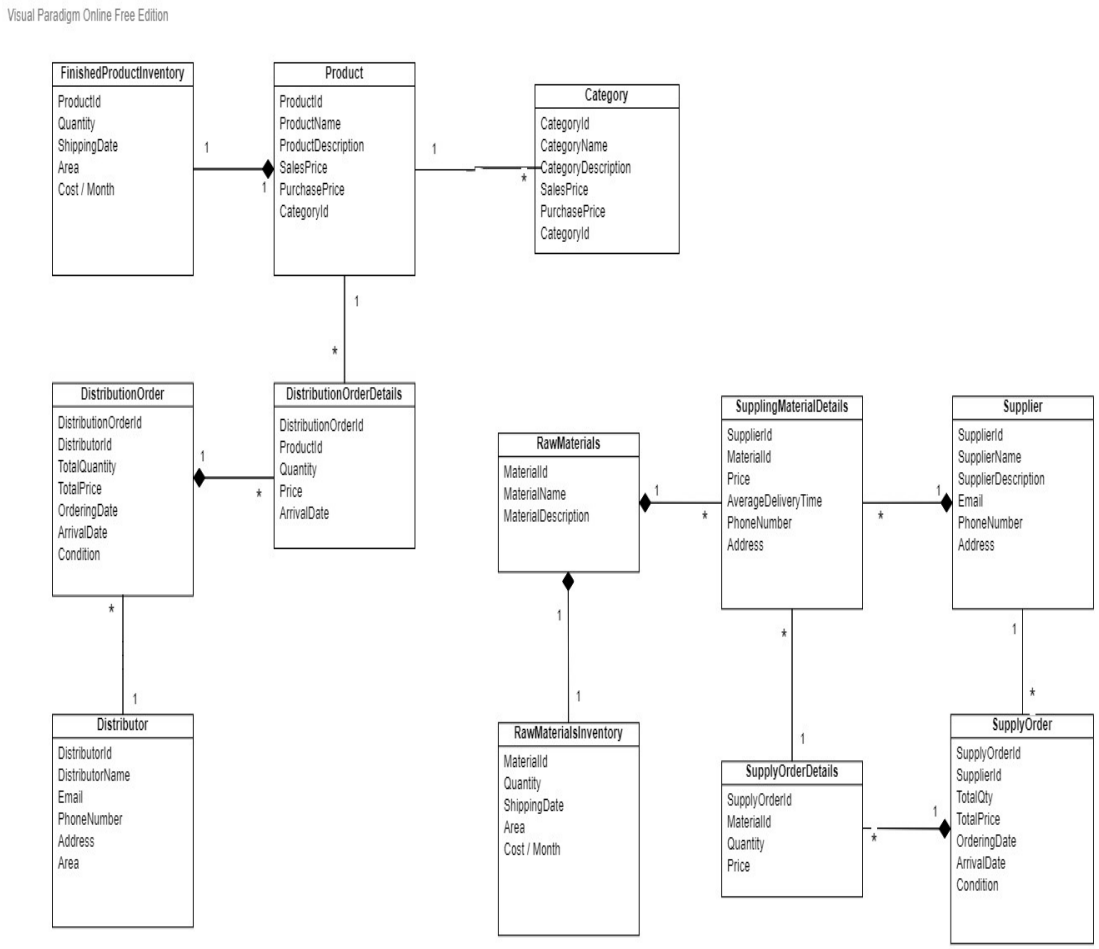
Chapter 3: Software Design

3.1 Design of database (ERD or Class) Diagram

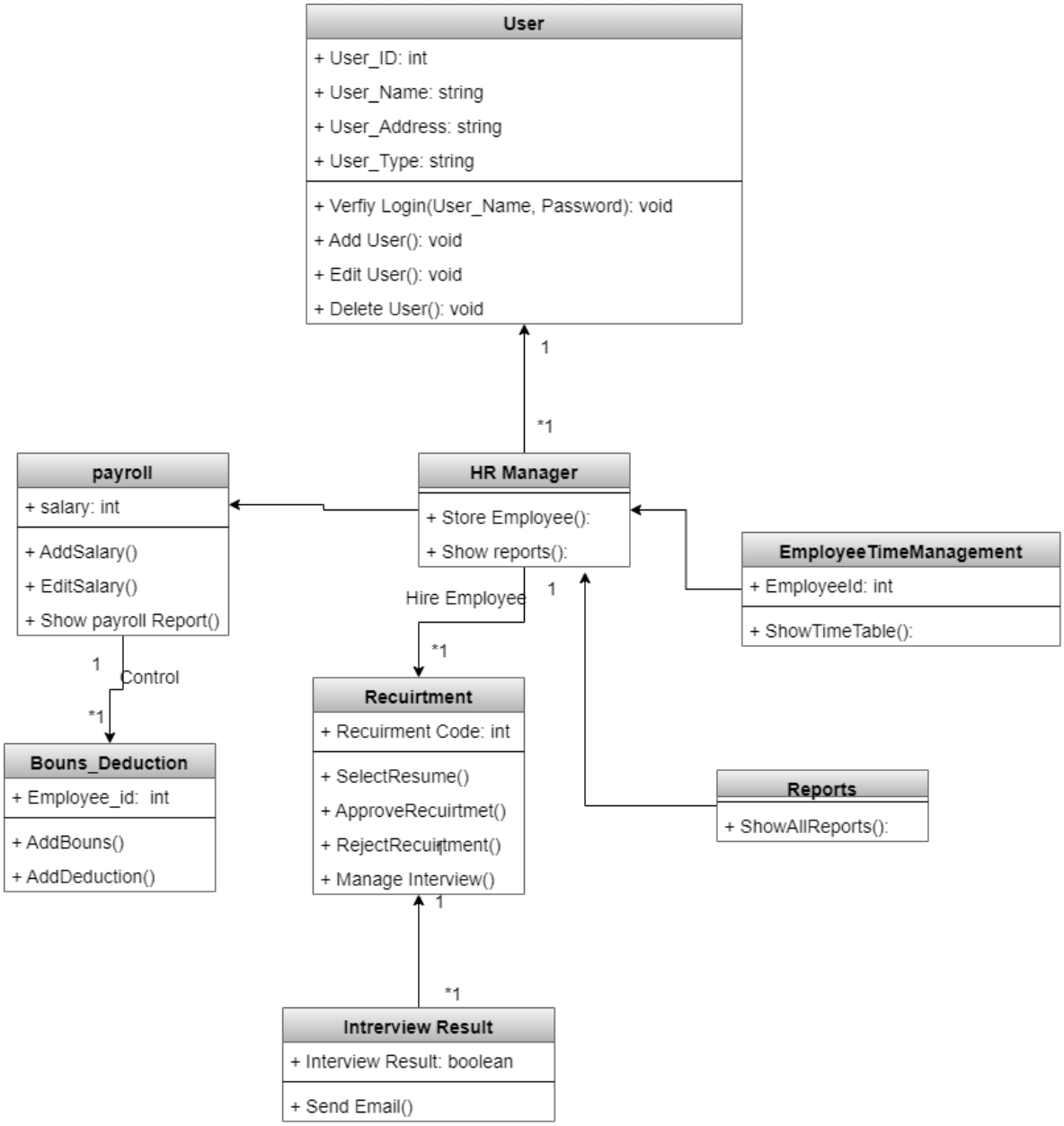
Financial management system Module UML Class Diagram



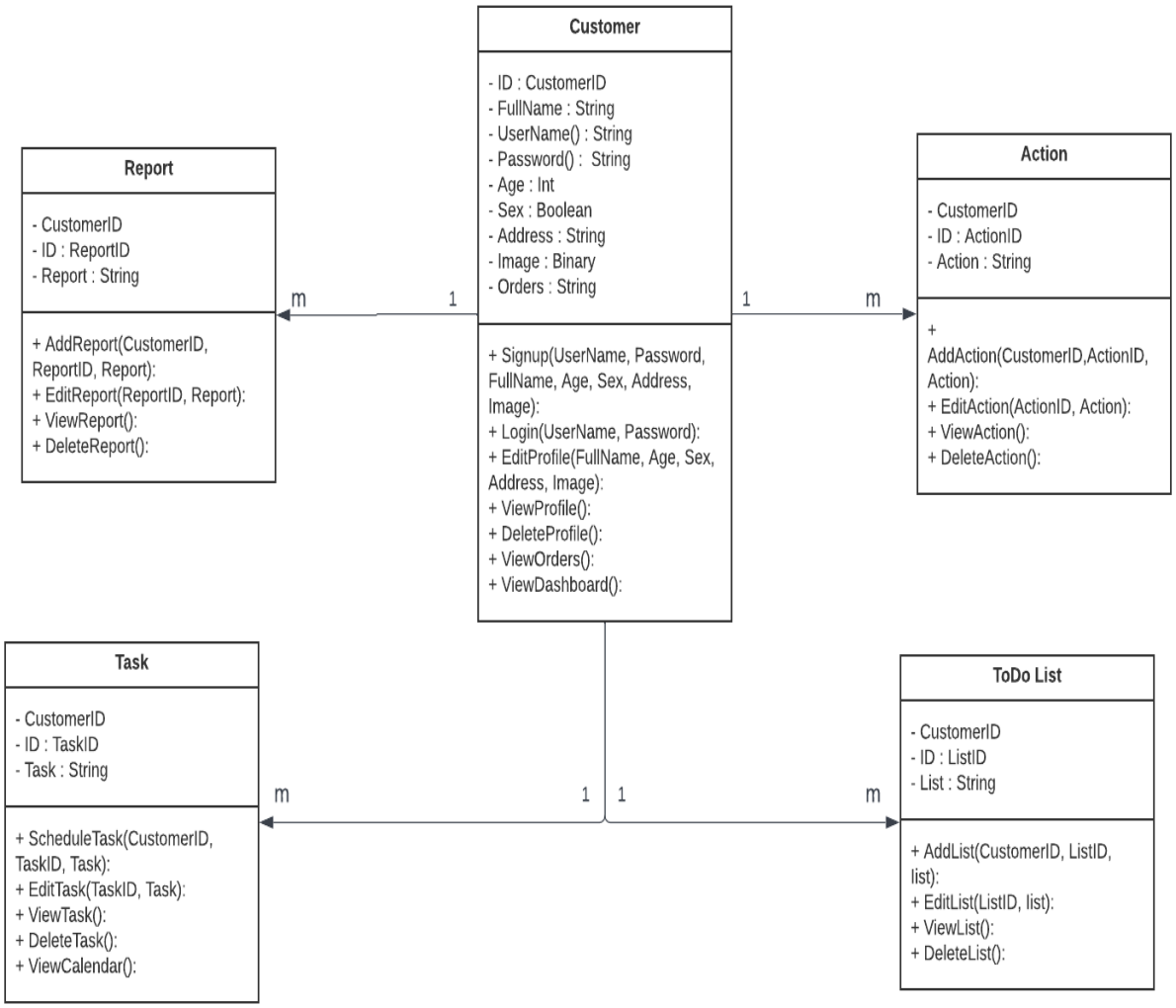
Supply Chain Management (SCM) Module UML Class Diagram



Human Resource Module UML Class Diagram

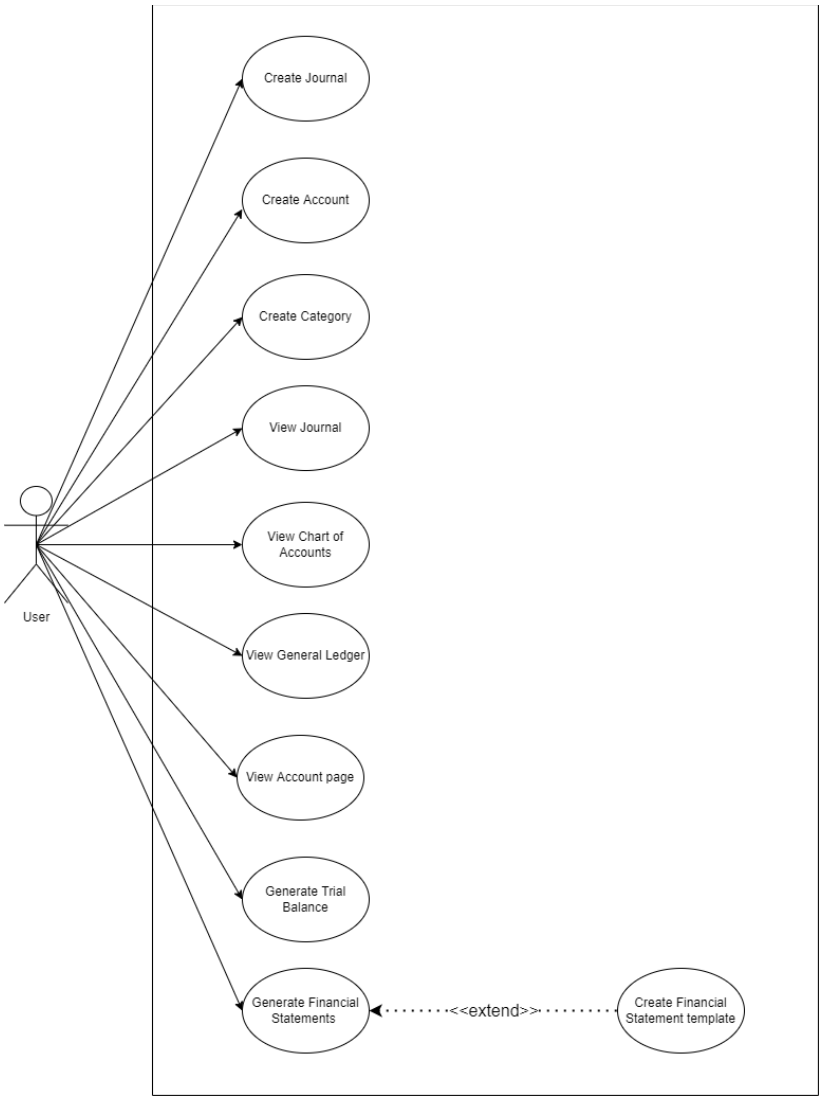


Customer Relation Management Module UML Class Diagram



3.2 Use case diagram

Financial management system Module Use Case Diagram



001

Use case scenario for “Create journal entries”

Actors: User (Accountant)

Precondition: At least two accounts that the debit and credit can go to exist

Main sequence:

1. User creates new journal entry
2. User fills out entry data
3. System validates entered data
4. Appropriate accounts are updated

Error flow sequence:

002

Use case scenario for “Create Accounts”

Actors: User (Accountant)

Precondition: None

Main sequence:

1. User creates a new account
2. User fills account data
3. System Validates data
4. Account is saved and shows up later in appropriate pages

Error flow sequence:

3. Display error and discard account

003

Use case scenario for “Create Account Category”

Actors: User (Accountant)

Precondition: None

Main sequence:

1. User creates a new category
2. User fills category data
3. System Validates data
4. Account is saved and shows up later in appropriate pages

Error flow sequence:

3. Display error and discard account

004

Use case scenario for “View Chart of accounts”

Actors: User (Accountant)

Precondition: None

Main sequence:

1. User clicks on the chart of accounts page
2. System fetches list of accounts from DB and displays them alongside their descriptive info (no journals for each account)
3. User can filter by account category

Error flow sequence:

2. Display no accounts found

005

Use case scenario for “View General Ledger”

Actors: User (Accountant)

Precondition: None

Main sequence:

1. User clicks on the general ledger page
2. System fetches account names, their pertaining journals, and balance from DB
3. Each account name is listed with its pertaining journals underneath and then its balance

Error flow sequence:

2. Display no accounts found

006

Use case scenario for “Generate Trial Balance”

Actors: User (Accountant)

Precondition: None

Main sequence:

1. User clicks on generate trial balance
2. System fetches account names and their balances from DB
3. System calculates total balances
4. A list of each account and its balance, with the total balance at the bottom is displayed to the user

Error flow sequence:

2. Display no accounts found

007

Use case scenario for “View Journal”

Actors: User (Accountant)

Precondition: None

Main sequence:

1. User clicks on Journal page
2. System fetches journals from DB
3. System sorts journals chronologically
4. Display list of every journal in the system chronologically sorted

Error flow sequence:

2. Display no journals found

008

Use case scenario for “Create Financial Statement Template”

Actors: User (Accountant)

Precondition: None

Main sequence:

1. User creates a template
2. User fills out required data
3. System validates data
4. Template is saved into DB

Error flow sequence:

3. Display error and discard template

009

Use case scenario for “Generate Financial Statement”

Actors: User (Accountant)

Precondition: A financial statement template exists

Main sequence:

1. User clicks generate financial statement
2. User chooses template
3. System fetches account names and their balances from DB
4. System calculates total balances for all accounts
5. Statement is shown with the appropriate sections and the total balance

Error flow sequence:

3. Display error accounts not found

010

Use case scenario for “View Journal Entry”

Actors: User (Accountant)

Precondition: A journal entry exists

Main sequence:

1. User clicks on journal anywhere
2. System fetches journal info from DB
3. Journal entry page is displayed with all its info

011

Use case scenario for “View Account page”

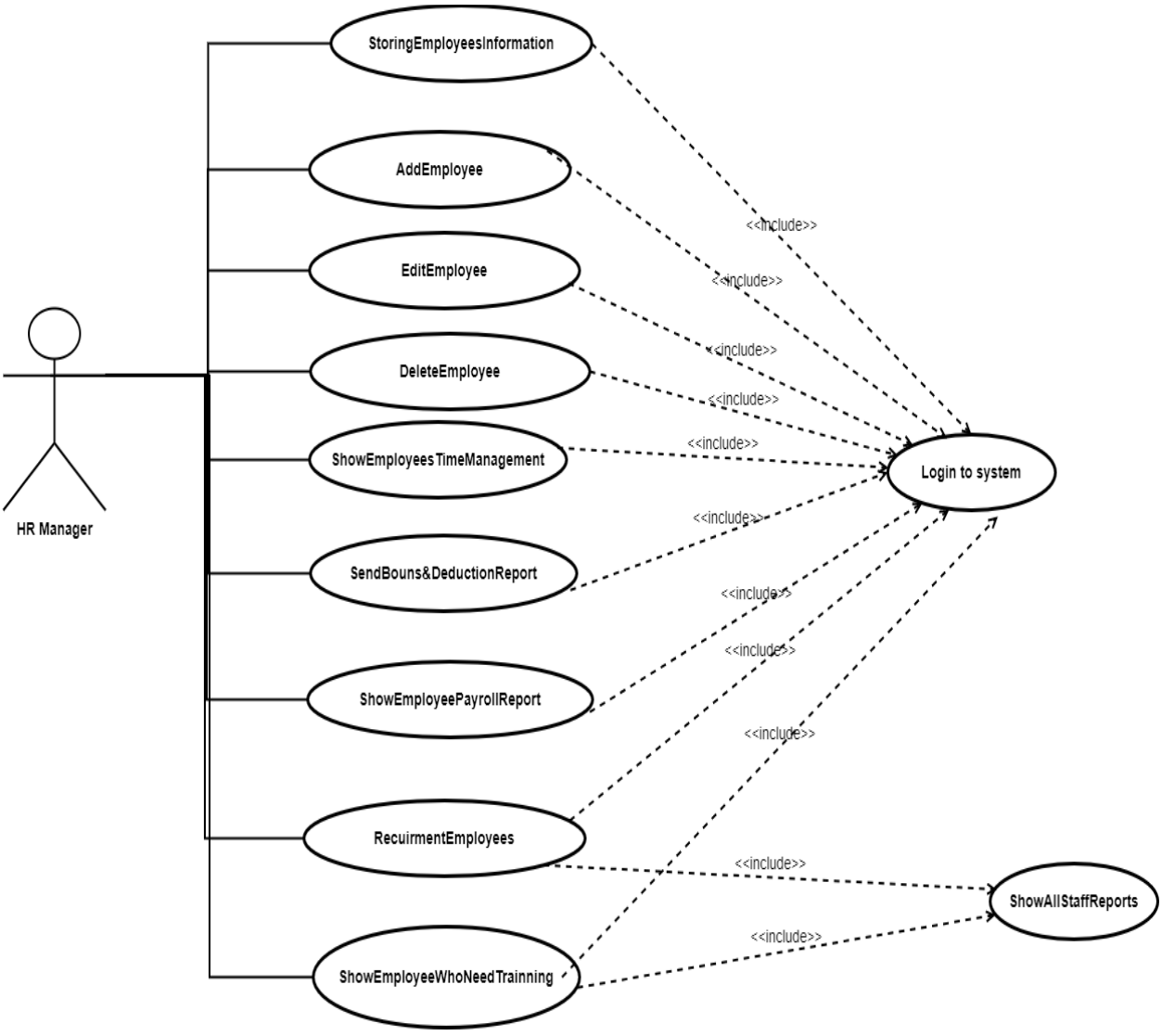
Actors: User (Accountant)

Precondition: An account exists

Main sequence:

1. User clicks on account anywhere
2. System fetches account info from DB
3. account page is displayed with all its info

Human Resource Module Use Case Diagram



Use case Name : Login To System
Summary: The System Validates HR Manager
Actors: HR Manager
Precondition: The ERP system must be available or on process or opened, and the login page must be opened.
Correct Flow Sequence(Main Sequence): 1. The HR manager clicks on its dashboard in the system. 2. The system displays the login form and the HR manager clicks on it. 3. The HR manager inserts his/her username and password. 4. The system checks the username and password. 5. If the username and password are correct then the system will display the HR module dashboard.
Error Flow Sequence(Alternative Sequence): Step 5: If the username or password is not correct, the System displays an Error message. And prompts for the correct username and password. Step 5: If the user tries to insert username and Password three times Without restarting the System will close. Step 1-4: If the user clicks on cancel, the system will go on the home page.
Post condition: username and password of manager is validated and go to HR module.

Use case Name: Storing Employees Information
Summary: The system store workforce's data
Actors: HR Manager
Dependency: Include login into system
Precondition: The ERP system must be available or on process or opened, and the DBMS is available.
Correct Flow Sequence(Main Sequence): 1. HR manager login to the system. 2. HR manager clicks on Employees Info. 3. The system displays The Employees Information (ID, Full Name, tax withholding, hours worked, salary information).
Error Flow Sequence(Alternative Sequence): Step 3: If there is an error in DBMS the system will send an error message notification.
Post condition: HR manager sees the Employees Information that can be accessed.

Use case Name :	Add Employee
Summary:	Successfully add new Employee
Actors:	HR Manager
Dependency:	Include login into system
Precondition:	The ERP system must be available or on process or opened, and the DBMS is available.
Correct Flow Sequence(Main Sequence): 1. HR manager login to the system. 2. HR manager goes to Employees Info. 3. The system displays The Employees Information (ID, Full Name, tax withholding, hours worked, salary information). 4. HR manager clicks on add employee. 5. The system displays the form of add employee. 6. The HR manager fills in the form of Employee's information. 7. HR manager clicks on save. 8. The system sends notification of successfully saved and saved new employee information in DBMS.	
Error Flow Sequence(Alternative Sequence): Step 7: If there is an error in DBMS the system will send an error message notification.	
Post condition:	New employee is added successfully.

Use case Name :	Edit Employee
Summary:	Successfully Edit Employee
Actors:	HR Manager
Dependency:	Include login into system
Precondition:	The ERP system must be available or on process or opened, and the DBMS is available.
Correct Flow Sequence(Main Sequence): 1. HR manager login to the system. 2. HR manager clicks on Employees Info. 3. The system displays The Employees Information (ID, Full Name, tax withholding, hours worked, salary information). 4. HR manager clicks on Edit employee. 5. The system displays the form of an Edit employee. 6. HR manager fills the form of Employee's information to edit information. 7. HR manager clicks on save. 8. The system sends notification of successfully saved and save new changes in employee's information in DBMS.	
Error Flow Sequence (Alternative Sequence): Step 3: If there is an error in DBMS the system will send an error message notification.	
Post condition:	Employee's Information is edited successfully.

Use case Name:	Delete Employee
Summary:	Successfully Delete Employee
Actors:	HR Manager
Dependency:	Include login into system
Precondition:	The ERP system must be available or on process or opened, and the DBMS is available.
Correct Flow Sequence(Main Sequence): 1. HR manager login to the system. 2. HR manager clicks on Employees Info. 3. The system displays The Employees Information (ID, Full Name, tax withholding, hours worked, salary information). 4. HR manager clicks on Delete employee. 5. The HR manager selects the employee that will be deleted. 6. The system sends notification of successfully Deleted and saves new changes in employee's information in DBMS.	
Error Flow Sequence (Alternative Sequence): Step 3: If there is an error in DBMS the system will send an error message notification.	
Post condition:	Employee is deleted successfully.

Use case Name :	Show Employee Time Management
Summary:	Show report of employee time management
Actors:	HR Manager
Dependency:	Include login into system
Precondition:	The ERP system must be available or on process or opened, and the DBMS is available.
Correct Flow Sequence(Main Sequence): 1. HR manager login to the system. 2. HR manager clicks on Employees Time Management. 3. The system displays The Employees Time Management Report.	
Error Flow Sequence(Alternative Sequence): Step 3: If there is an error in the page of the report the system will send an error message.	
Post condition:	HR manager see the Employees Time management Report.

Use case Name :	Send Bonus & Deduction Report
Summary:	Successfully send bonus & deduction report to finance module
Actors:	HR Manager
Dependency:	Include login into system
Precondition:	The ERP system must be available or on process or opened, and the DBMS is available.
Correct Flow Sequence(Main Sequence): 1. HR manager login to the system. 2. HR manager clicks on Bonus & Deduction Report. 3. The system displays the bonus & deduction report. 4. HR manager clicks on Send. 5. If the report is sent to the finance module then the system shows a message successfully sent.	
Error Flow Sequence(Alternative Sequence): Step 3: If there is an error in the report page then the system will send an error message notification.	
Post condition:	The Bonus & Deduction Report is sent successfully to Finance Module

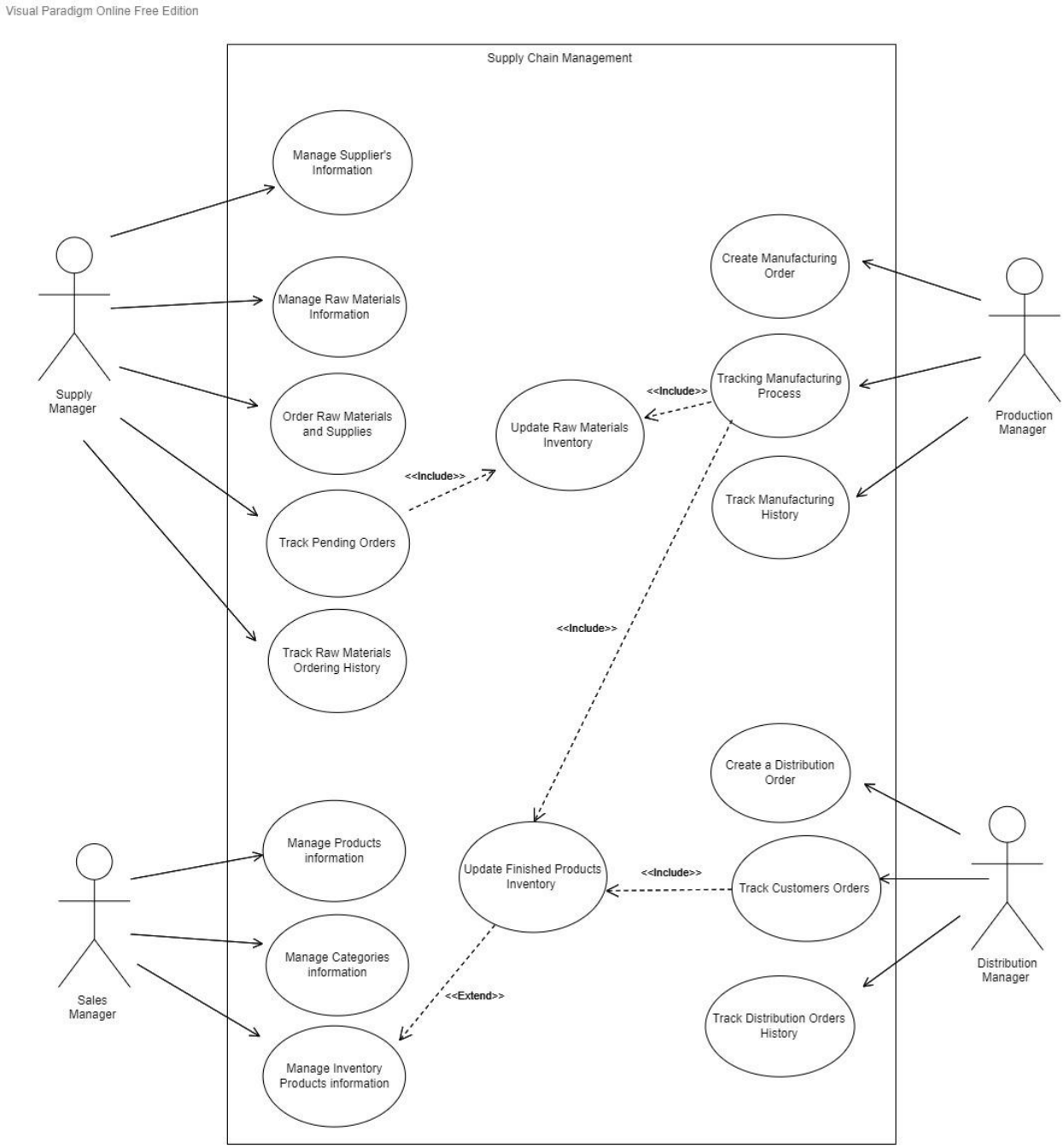
Use case Name :	Show Employee Payroll Report
Summary:	show the payroll report
Actors:	HR Manager
Dependency:	Include login into system
Precondition:	The ERP system must be available or on process or opened, and the DBMS is available.
Correct Flow Sequence(Main Sequence): 1. HR manager login to the system. 2. HR manager clicks on Payroll Report. 3. The system displays Payroll Report	
Error Flow Sequence(Alternative Sequence): Step 3: If there is an error in the Payroll Report page then the system will send an error message notification.	
Post condition:	HR manager sees the Payroll Report.

Use case Name :	Show All Staff Reports
Summary:	show report
Actors:	HR Manager
Dependency:	Include login into system
Precondition:	The ERP system must be available or on process or opened, and the DBMS is available.
Correct Flow Sequence(Main Sequence): 1. HR manager login to the system. 2. The HR manager clicks on the show report. 3. The system displays all reports of employees.	
Error Flow Sequence(Alternative Sequence): Step 3: If there is an error in DBMS the system will send an error message notification.	
Post condition:	HR manager sees the staff reports.

Use case Name :	Recruitment Employee
Summary:	Successfully Recruitment Employee
Actors:	HR Manager
Dependency:	Include login into system
Precondition:	The ERP system must be available or on process or opened, and the DBMS is available, show all staff reports.
Correct Flow Sequence(Main Sequence): 1. HR manager login to the system. 2. HR manager clicks on Recruitment employee. 3. The system displays the employee position that is needed. 4. HR manager clicks on an advertisement for a job. 5. The HR manager clicks on the schedule for the interview. 6. HR manager clicks on send accepted mail if an employee accepts.	
Error Flow Sequence(Alternative Sequence): Step 6: If the employee failed on interview then HR manager send rejected mail.	
Post condition:	Successfully Recruitment employee.

Use case Name : Show Employee Who Need Training
Summary: Successfully add training to employee
Actors: HR Manager
Dependency: Include login into system
Precondition: The ERP system must be available or on process or opened, and the DBMS is available, show all staff reports.
Correct Flow Sequence(Main Sequence): 1. HR manager login to the system. 2. HR manager clicks to show all staff reports. 3. The system displays the staff report. 4. HR managers select who needs training. 5. HR manager clicks on submit.
Error Flow Sequence(Alternative Sequence): Step 3: If there is an error in DBMS the system will send an error message notification.
Post condition: Successfully assign training to employees who need training.

Supply Chain Management (SCM) and IM Module Use case Diagram:



ID:	UC_SCM_01
Title:	Manage Supplier's Information
Description:	System allows the user/s to Manage CRUD operations on the supplier's information that the organization deals with.
Primary Actor:	Supply Manager
Preconditions:	User Logs in In to the System.
Postconditions:	Organizations Supplier's Data are Updated or Modified
Main Scenario:	1. User Logs in to the System 2. User views a table containing all Suppliers Information 2.1. information contains Id, suppliers Name, suppliers Description, supplier's raw material/s being supplied and at what cost and time, suppliers contact info including phone number and email as well as his address. 3. User can then either add a new Supplier, Update or Delete an existing Supplier 4. System acknowledge user about modification
Extensions:	1. System Fails to authenticate User • System prevents user from accessing supplier's table

ID:	UC_SCM_02
Title:	Manage Raw Material's Information
Description:	System allows the user/s to Manage CRUD operations on the Raw materials as well as raw materials included in the Warehouse.
Primary Actor:	Supply Manager
Preconditions:	User Logs in In to the System.
Postconditions:	Organizations raw Materials Data are Updated or Modified

Main Scenario:	1. User Logs in to the System 2. User views a table containing all raw materials Information 2.1. information includes ID, Name, Description, Quantity, Area in Inventory, cost in inventory and arrival date. 3. User can then either add a new raw material, Update or Delete an existing raw material data. 4. System acknowledge user about modification
Extensions:	1. System Fails to authenticate User • System prevents user from accessing raw material's data.

ID:	**UC_SCM_03**
Title:	Order Raw Materials from Supplier
Description:	User can order bulk of raw materials from one or more suppliers when needed required for the production of finished products
Primary Actor:	Supply Manager
Preconditions:	Raw Materials reached the ROP or new Materials are required
Postconditions:	Update raw Materials Inventory about new bulk arrival
Main Scenario:	1. User Logs in to the System 2. User creates a supply order from the intent supplier containing the type of material required as well as the amount of quantity needed. 3. System responds with the total cost and the expected arrival time range invoice 4. Inventory updated its data accordingly
Extensions:	1. Raw materials inventory reaches re-ordering point. • System Automatically notifies user/s.

ID:	UC_SCM_04
Title:	Track Orders from Supplier
Description:	User can check the status of an order
Primary Actor:	Supply Manager
Preconditions:	Order is created
Postconditions:	User gets updated about the orders current status, either pending, shipped, fulfilled
Main Scenario:	1. User Logs in to the system 2. User access the list of orders 3. Condition of each order must be associated with it. either Pending, Shipped, Out for Delivery, Fulfilled or Cancelled 4. User can change the status accordingly
Extensions:	None

ID:	UC_SCM_05
Title:	Manage Product's Information
Description:	User Manage CRUD operations on the organization's products information being produced.
Primary Actor:	Sales Manager
Preconditions:	User Logs in In to the System.
Postconditions:	Final Products Data are Updated or Modified
Main Scenario:	1. User Logs in to the System 2. User views a table containing all Products Information 2.1. information contains Id, Name, Description, sales price, purchasing price, category as well as products in inventory information. 3. User can then either add a new Product, Update or Delete an existing Product 4. System acknowledge user about modification
Extensions:	1. System Fails to authenticate User • System prevents user from accessing product's table

ID:	UC_SCM_06
Title:	Manage Categories Information
Description:	User Manage CRUD operations on the organization's Category's information
Primary Actor:	Sales Manager
Preconditions:	User Logs in In to the System.
Postconditions:	Categories Data are Updated or Modified
Main Scenario:	1. User Logs in to the System 2. User views a table containing all Categories Information 2.1. information contains Id, Name, Description and the number of products belonging to that category. 5. User can then either add a new Category, Update or Delete an existing Category. 6. System acknowledge user about modification
Extensions:	1. System Fails to authenticate User • System prevents user from accessing categories table

ID:	UC_SCM_07
Title:	Manage Products in Inventory Information
Description:	User Manage CRUD operations on the organization's products information in the warehouse.
Primary Actor:	Sales Manager or System.
Preconditions:	Product is registered in the Warehouse Repository.
Postconditions:	Products Inventory Data are Updated or Modified
Main Scenario:	1. User Logs in to the System 2. User views a table containing all Products in the Inventory and can manage their information. 2.1. information contains productid, quantity, cost in inventory per month, shipping date 3. a new Product can be added to the Inventory, or an existing product in the inventory can be either Updated or Deleted 4. System acknowledge user about modification
Extensions:	1. System Fails to authenticate User • System prevents user from accessing inventory table

ID:	**UC_SCM_08**
Title:	Update Raw Materials Inventory
Description:	Prevents Raw Materials Stock Outages
Primary Actor:	Automated System
Preconditions:	New Bulk of raw Materials arrives OR Materials are consumed for production
Postconditions:	Raw materials inventory Materials quantity are either incremented or decremented.
Main Scenario:	1. If materials are taken from the inventory for production then materials data are decremented. 2. If a supply order is delivered, then materials data are incremented
Extensions:	1. Raw materials reach the ROP • System creates automatically a supply order to prevent stock outage

ID:	**UC_SCM_09**
Title:	Update Finished Products Inventory
Description:	Prevents Finished Products Stock Outages
Primary Actor:	Automated System
Preconditions:	New Bulk of Products are Produced OR Products order are being created for distribution.
Postconditions:	Products inventory quantity are either incremented or decremented.
Main Scenario:	1. If products are taken from the inventory for distribution purpose then products data are decremented. 2. If a production order is completed, products are taken to the warehouse and products qty in the inventory are incremented
Extensions:	1. Products reach the ROP • System creates automatically a manufacturing/production order to prevent stock outage

ID:	UC_SCM_10
Title:	Create Manufacturing Order
Description:	Create a Manufacturing order to produce finished products form raw materials
Primary Actor:	Production Manager or System
Preconditions:	Finished products have reached the ROP
Postconditions:	Order for Manufacturing is created
Main Scenario:	1. Products reaches ROP 2. System notifies user 3. System automatically creates a manufacturing order providing a list of the required materials associated with the quantity of each. 4. Requested amount of finished products must be provided too
Extensions:	1. Order already exists • Avoid duplication of the same order

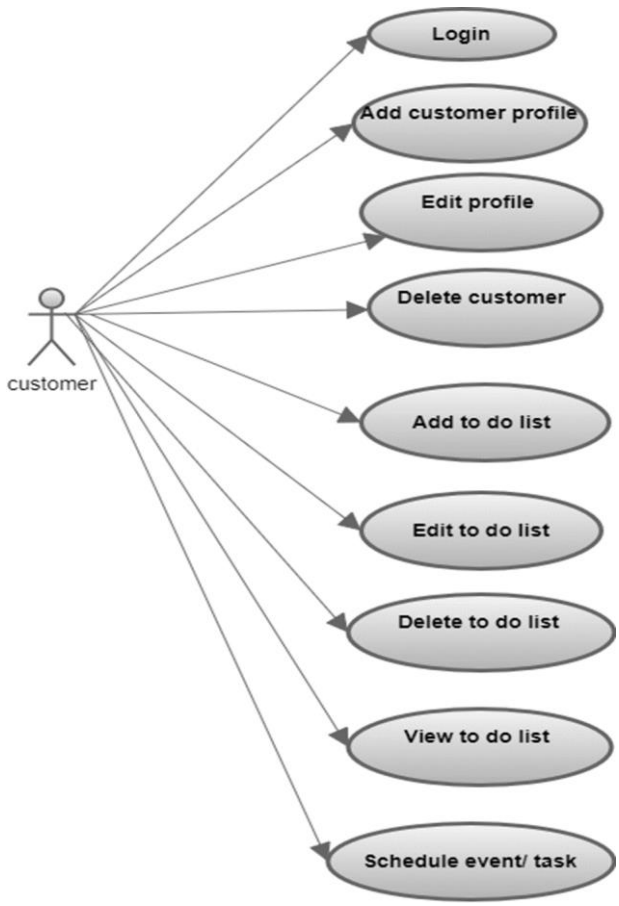
ID:	UC_SCM_11
Title:	Track Manufacturing Orders
Description:	User gets feedback about order's status
Primary Actor:	Production Manager
Preconditions:	Order is created
Postconditions:	User get to know the order's condition
Main Scenario:	1. User logs in 2. User checks recent orders placed 3. Order status can be either Pending OR Manufacturing OR Shipped
Extensions:	None

ID:	UC_SCM_12
Title:	Create a Distribution Order
Description:	Create a Distribution order to deliver finished products to the customers
Primary Actor:	Distribution Manager
Preconditions:	Order is Requested from a Customer
Postconditions:	Distribution Order is created
Main Scenario:	1. Distributor requests a bulk of products. 2. System Creates a Distribution order including a list of the required products associated with the intent quantity. 3. Total Price is calculated accordingly
Extensions:	1. Products Reached ROP • System Creates Manufacturing Order

ID:	UC_SCM_13
Title:	Track Distribution Orders
Description:	User can check the status of an order
Primary Actor:	Distribution Manager
Preconditions:	Order is created by user or system
Postconditions:	User gets updated about the orders current condition
Main Scenario:	1. User Logs in to the system 2. User access a list of current distribution orders 3. Condition of each order must be associated with it. either Pending, Shipped, Out for Delivery, Fulfilled, Received or Cancelled
Extensions:	None

ID:	UC_SCM_14
Title:	Manage Distributors Information
Description:	User Can manage crud operations on the distributors
Primary Actor:	Distribution Manager
Preconditions:	none
Postconditions:	Distributor info is either added, updated or removed from/to the system.
Main Scenario:	1.User Logs in to the system 2.User access the list of all Distributors. 3.CRUD operation is done
Extensions:	None

Customer Relation Management (CRM) Module Use case Diagram



Use Case Name	Login To System
Summary	The System Validates a customer
Actors	Customer
Precondition	The ERP system must be available or on process or opened, and the login page must be opened.
Correct Flow Sequence (Main Sequence): 1. Customer click on its dashboard in the system. 2. The system display the login form and Customer click on it. 3. The Customer insert his/her username and password. 4. The system check the username and password. 5. If the username and password are correct then the system will display the CRM module dashboard.	
Error Flow Sequence (Alternative Sequence): Step 5: If the username or password is not correct, the System displays an Error message. and prompts for the correct username and password. Step 5: If the user tries to insert username and Password three times Without restarting the System it will close. Step 1-4: If the user clicks on cancel, the system will go on the home page.	
Post condition:	username and password of Customer is validated and go to CRM module.

Use case Name	Adding Customer Profile
Summary	The system adds customer data
Actors	Customer
Dependency	Include login into system
Precondition	The ERP system must be available or on process or opened, and the DBMS is available.
Correct Flow Sequence(Main Sequence): 1. Customer login to the system. 2. Customer click on profile Info. 3. The system creates a new profile and store it into the database(ID, Full Name, Sex, Age, Address, Image >>"optional").	
Error Flow Sequence(Alternative Sequence): Step 3: If there is error in DBMS the system will notify the user with data is mistaken.	
Post condition:	Profile is created successfully for the customer.

Use case Name:	Edit Profile
Summary:	change his/her profile info.
Actors:	Customer
Dependency:	Include login into system
Precondition	The ERP system must be available or on process or opened, and the DBMS is available.
Correct Flow Sequence(Main Sequence): 1. Customer login to the system. 2. Customer goes to Profile Info. 3. The system displays his profile(ID, Full Name, Sex, Age, Address, Image). 4. Customer click on edit profile. 5. The system display form profile info. 6. Customer fill the form changing profile info. 7. Customer click on save. 8. The system send notification of successfully saved and save new profile information in DBMS.	
Error Flow Sequence(Alternative Sequence): Step 7: If there is error in DBMS the system will send error message notification.	
Post condition	Profile is edited successfully.

Use case Name	Delete Customer
Summary:	Successfully Delete Profile
Actors:	Customer
Dependency:	Include login into system
Precondition:	The ERP system must be available or on process or opened, and the DBMS is available.
Correct Flow Sequence(Main Sequence): 1. Customer login to the system. 2. Customer click on Profile Info. 3. The system displays The Employees Information (ID, Full Name, Sex, Age, Address, Image). 4. Customer click on delete Profile. 5. The system display form of Delete Profile. 6. Customer fill the form of Delete Profile information. 7. Customer click on save. 8. The system send notification of successfully saved and save new changes in employee's information in DBMS.	
Post condition:	Customer's Profile is deleted successfully.

Use case Name :	Add To-Do List
Summary:	Adding To-Do List
Actors:	Customer
Dependency	Include login into system
Precondition	The ERP system must be available or on process or opened, and the DBMS is available.
Correct Flow Sequence(Main Sequence):	1. Customer login to the system. 2. Customer click on Dashboard Info. 3. The system displays The Dashboard. 4. Customer click on Add To-Do List. 5. Customer fill a form of Adding To-Do List. 6. Customer clicks on save. The system send notification of successfully Deleted and save new changes in employee's information in DBMS.
Error Flow Sequence(Alternative Sequence):	Step 3: If there is error in DBMS the system will send error message notification.
Post condition:	Customer Added To-Do List Successfully

Use case Name	**Edit To-Do List**
Summary:	**Editing Todo List**
Actors:	**Customer**
Dependency:	**Include login into system**
Precondition	**The ERP system must be available or on process or opened, and the DBMS is available, show all staff reports.**
Correct Flow Sequence(Main Sequence):	**1. Customer login to the system.** **2. Customer click on Dashboard Info.** **3. The system displays The Dashboard.** **4. Customer click on Edit To-Do List.** **5. Customer fill a form of Editing To-Do List.** **6. Customer clicks on save.** **7. The system send notification of successfully Deleted and save new changes in employee's information in DBMS.**
Error Flow Sequence(Alternative Sequence):	**Step 6: If the Customer failed. Notify him/her data is mistaken.**
Post condition:	**Customer Successfully Edited To-Do List.**

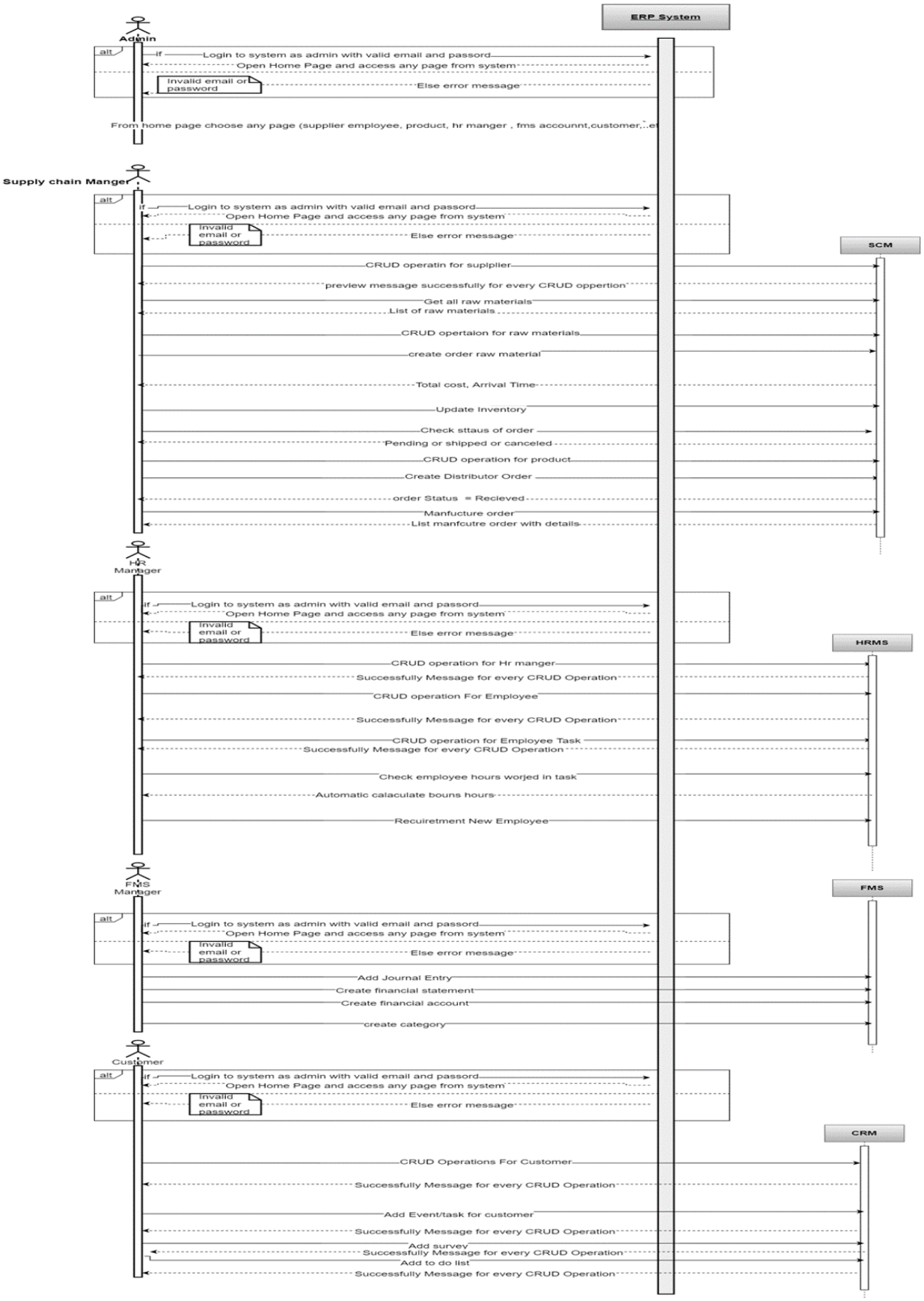
Use case Name	**View To-Do List**
Summary:	**Successfully view ToDo List**
Actors:	**Customer**
Dependency:	**Include login into system**
Precondition:	**The ERP system must be available or on process or opened, and the DBMS is available, show all staff reports.**
Correct Flow Sequence(Main Sequence):	**1. Customer login to the system.** **2. Customer click on Dashboard Info.** **3. The system displays The Dashboard.** **4. Customer click on View To-Do List.**
Error Flow Sequence(Alternative Sequence):	**Step 3: If there is error in DBMS the system will send error message notification.**

Use case Name :	**Delete To-Do List**
Summary:	**Deleting Todo List**
Actors:	**Customer**
Dependency:	**Include login into system**
Precondition:	**The ERP system must be available or on process or opened, and the DBMS is available, show all staff reports.**
Correct Flow Sequence(Main Sequence):	**8. Customer login to the system.** **9. Customer click on Dashboard Info.** **10. The system displays The Dashboard.** **11. Customer click on Delete To-Do List.** **12. Customer fill a form of Delete To-Do List.** **13. Customer clicks on save.** **14. The system send notification of successfully Deleted and save new changes in employee's information in DBMS.**
Error Flow Sequence (Alternative Sequence):	**Step 6: If the Customer failed. Notify him/her data is mistaken.**
Post condition:	**Customer Successfully Deleted To-Do List.**

Use case Name :	**Schedule Event/Task**
Summary:	**Scheduling Event/Task**
Actors:	**Customer**
Dependency:	**Include login into system**
Precondition:	**The ERP system must be available or on process or opened, and the DBMS is available, show all staff reports.**
Correct Flow Sequence (Main Sequence):	**48. Customer login to the system.** **49. Customer click on Dashboard Info.** **50. The system displays The Dashboard.** **51. Customer click on View Calendar.** **52. Customer click on schedule Event / Task** **53. Customer fill the form of events.**
Customer clicks on save.	**1. The system send notification of successfully Deleted and save new changes in employee's information in DBMS.**
Error Flow Sequence (Alternative Sequence):	**Step 3: If there is error in DBMS the system will send error message notification.**
Post condition:	**Customer Schedule Event/Task Successfully**

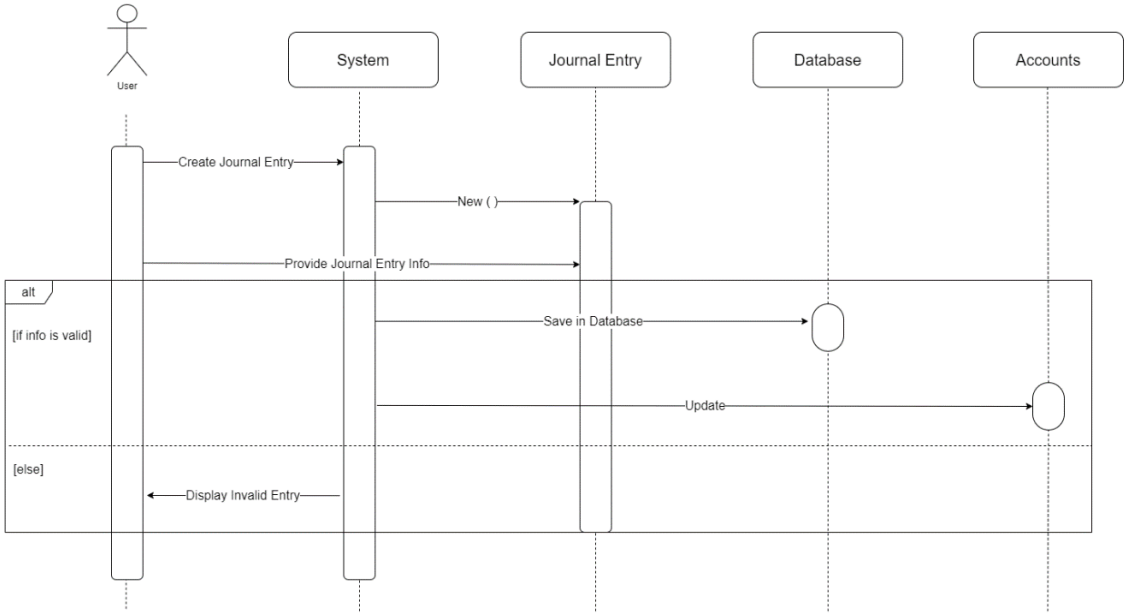
3.3 sequence diagrams

System Sequence Diagram

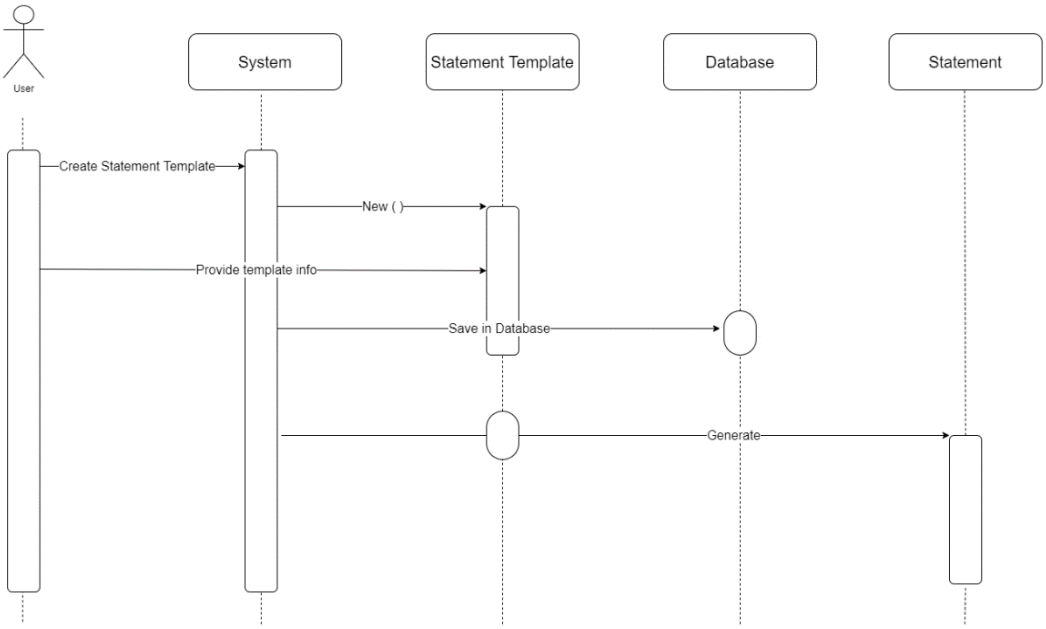


FMS

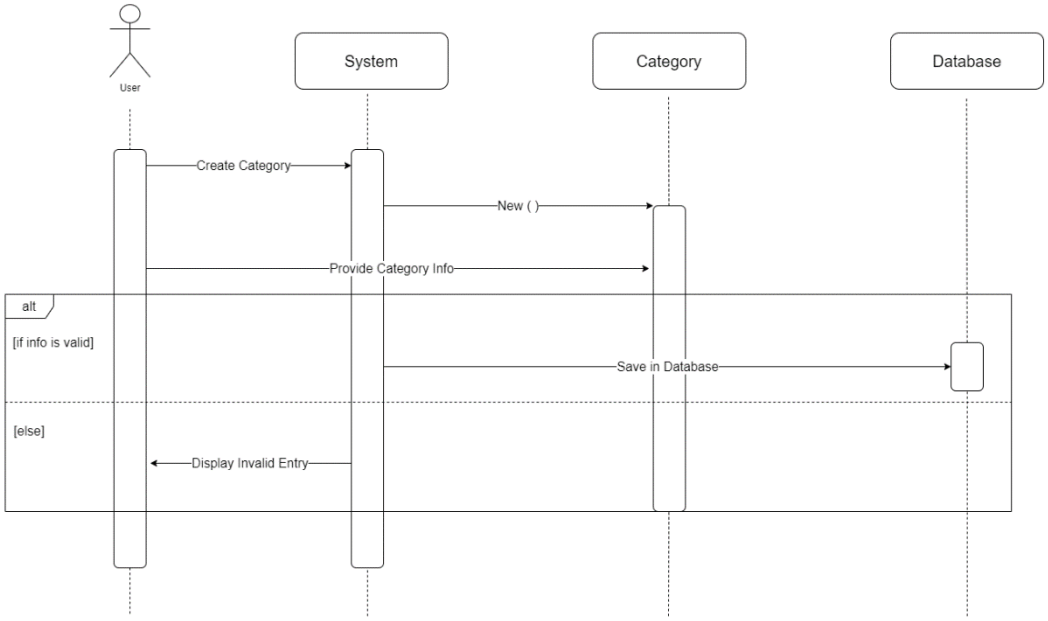
Add journal entry



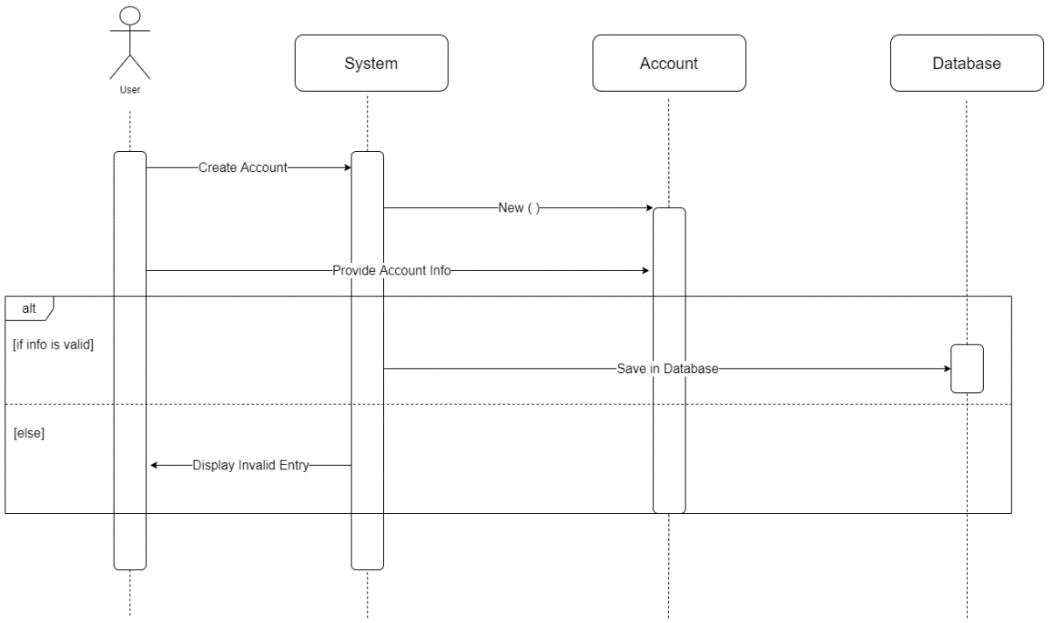
Generate Financial Statement



Create Category

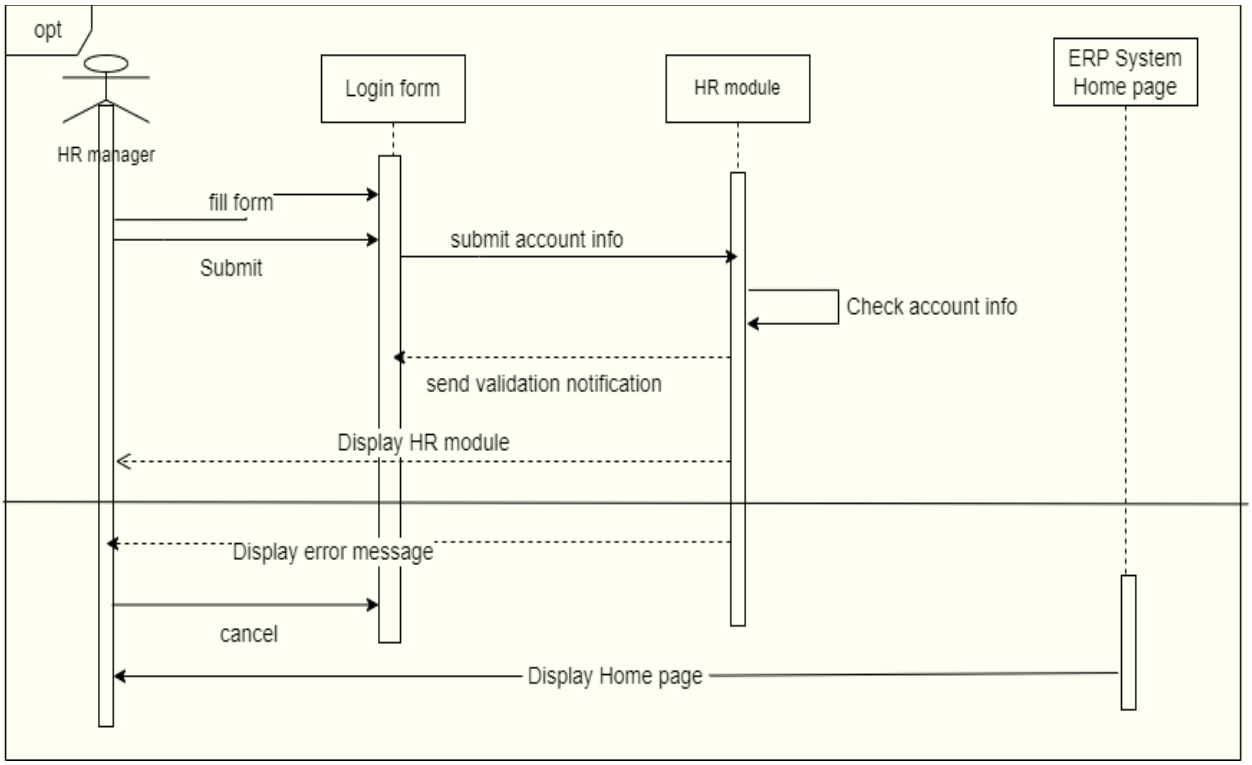


Create Account

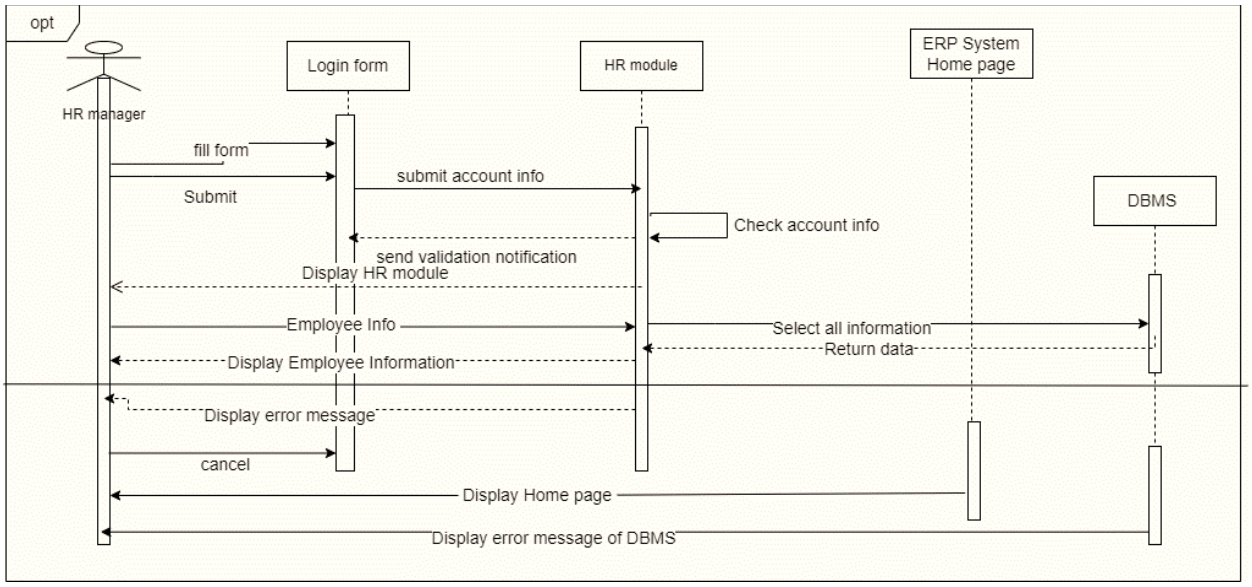


HR

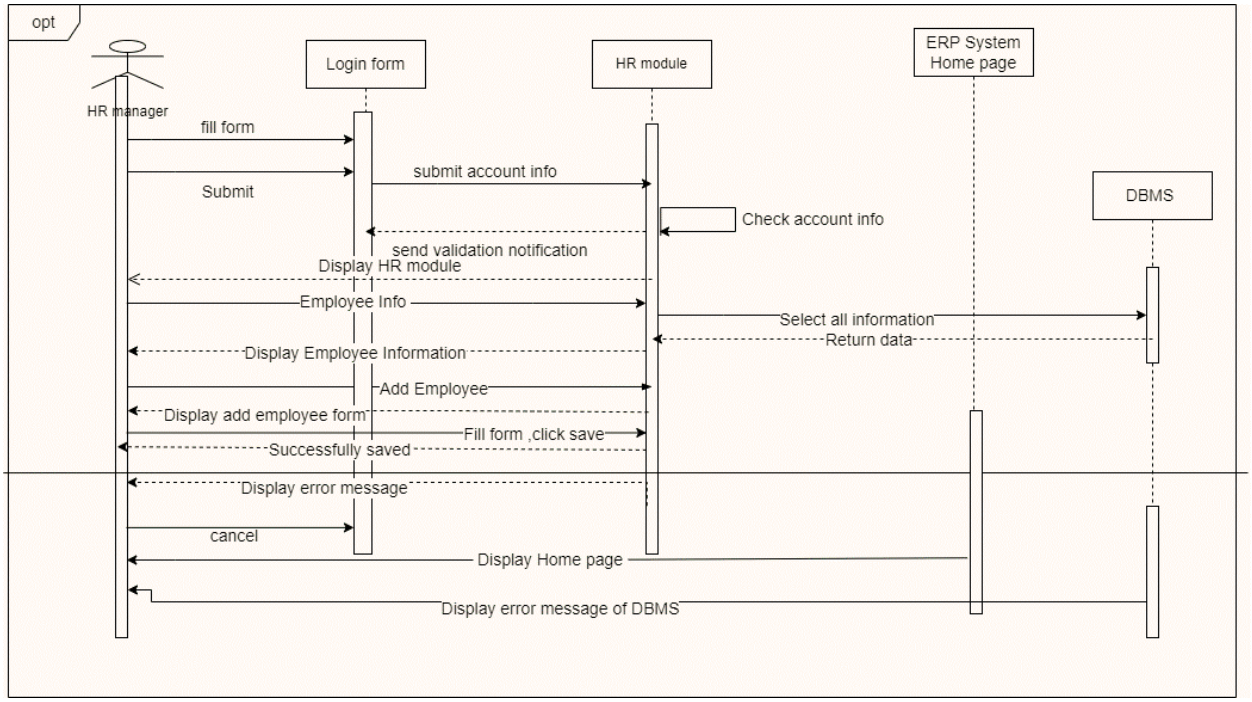
1. Sequence Diagram for login to system:



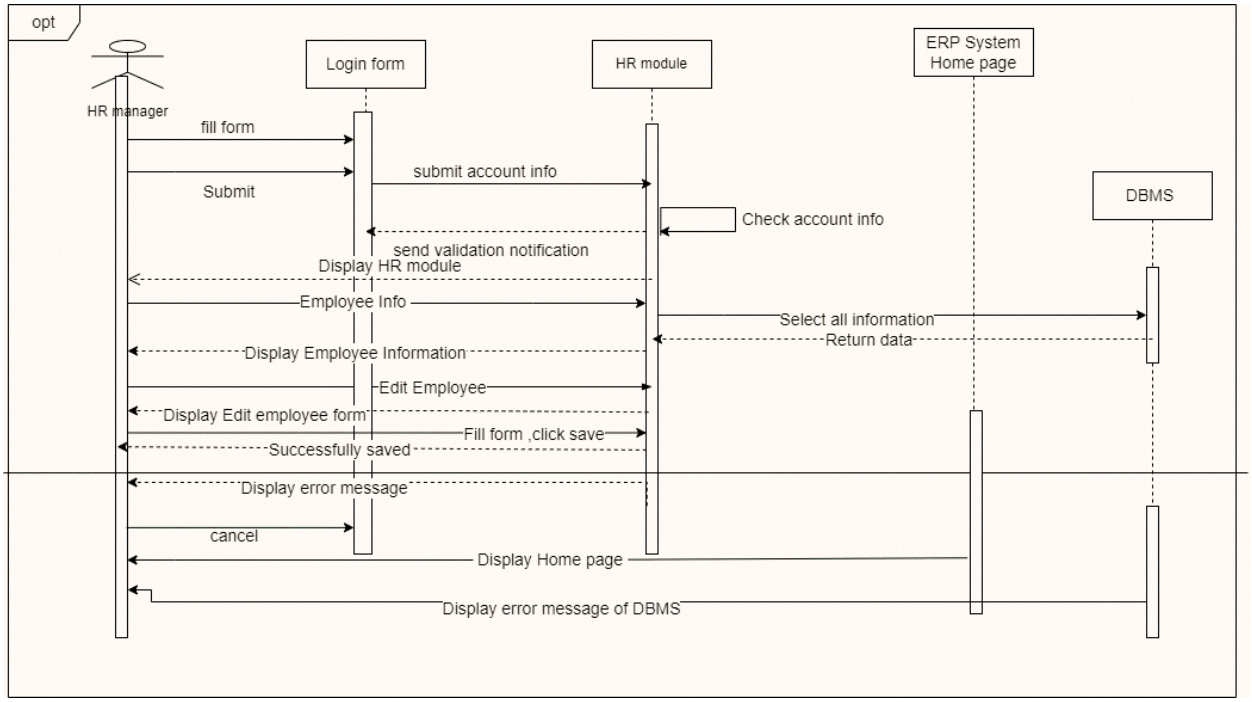
2. Sequence Diagram for Storing Employees Information:



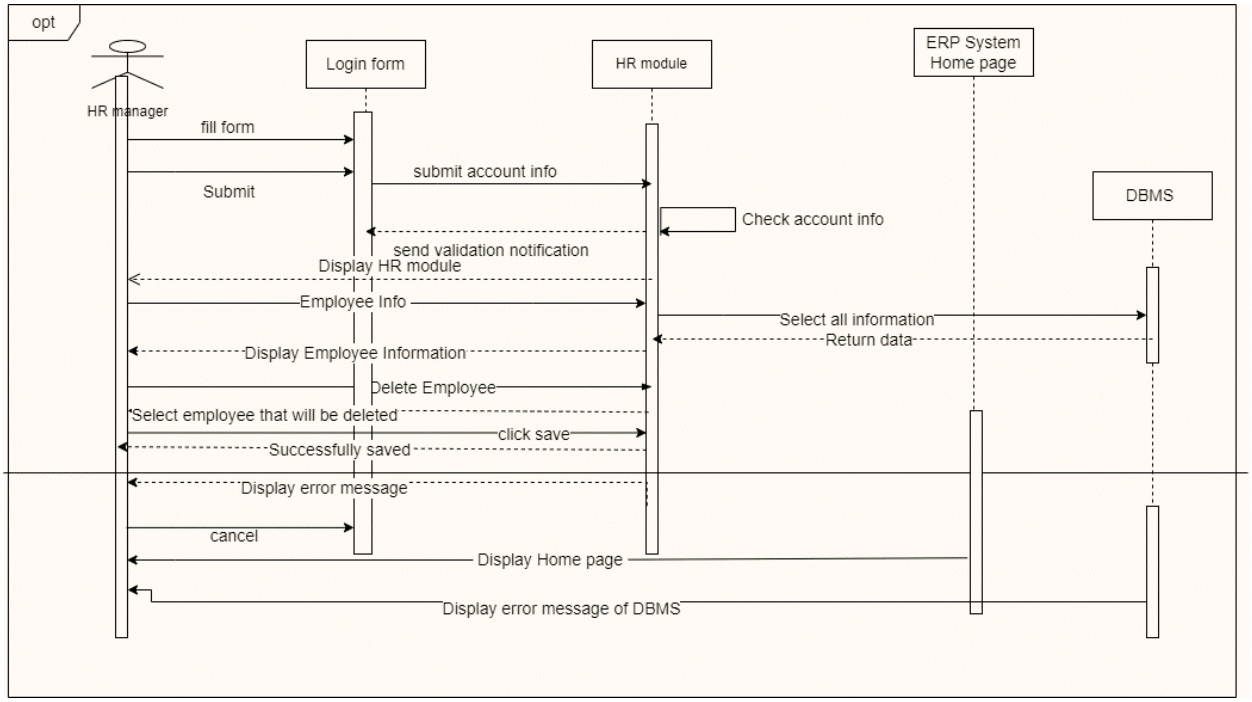
3. Sequence Diagram for Add Employee:



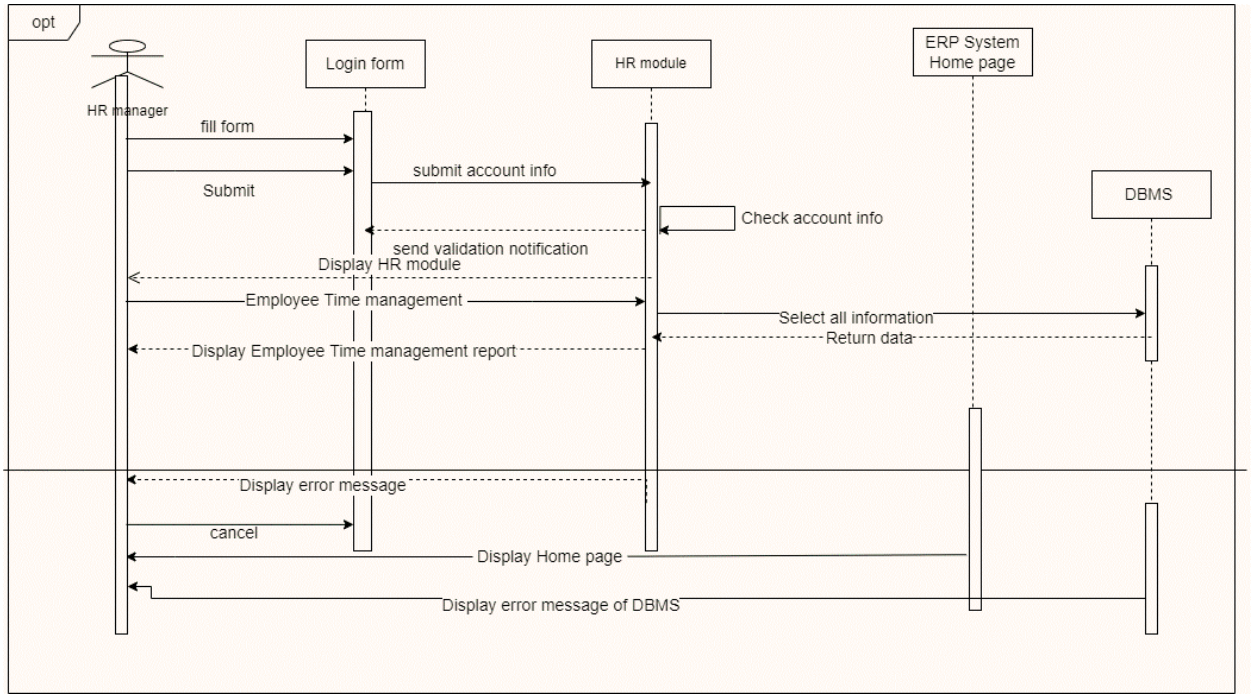
4. Sequence Diagram for Edit Employee:



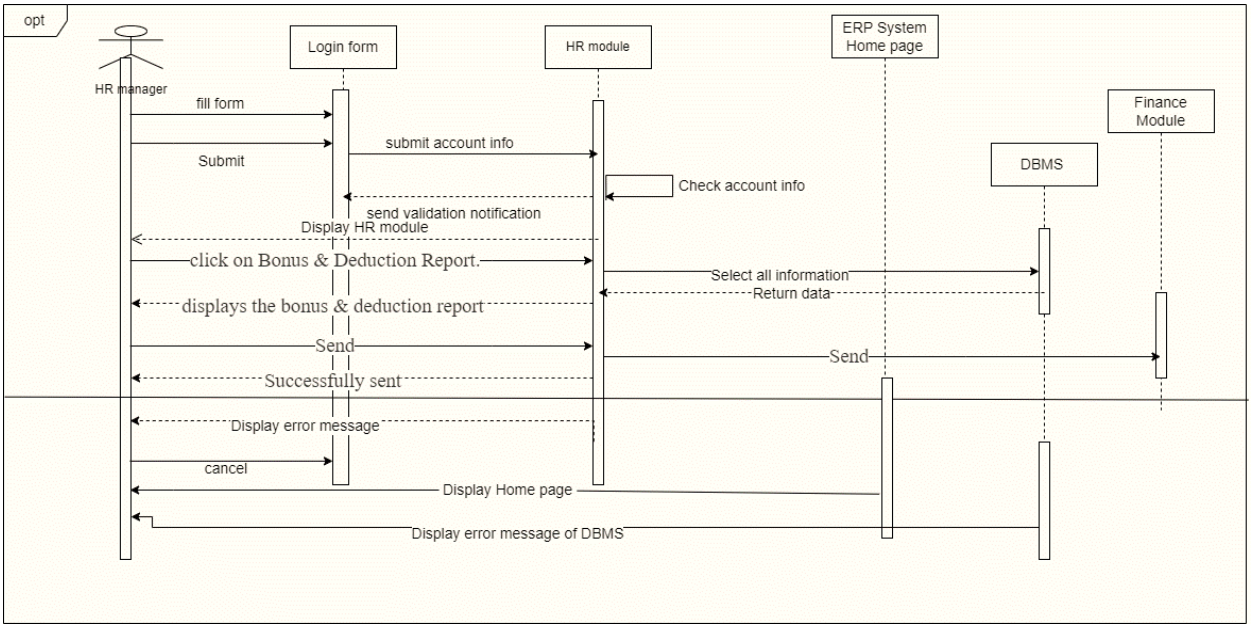
5. Sequence Diagram for Delete Employee:



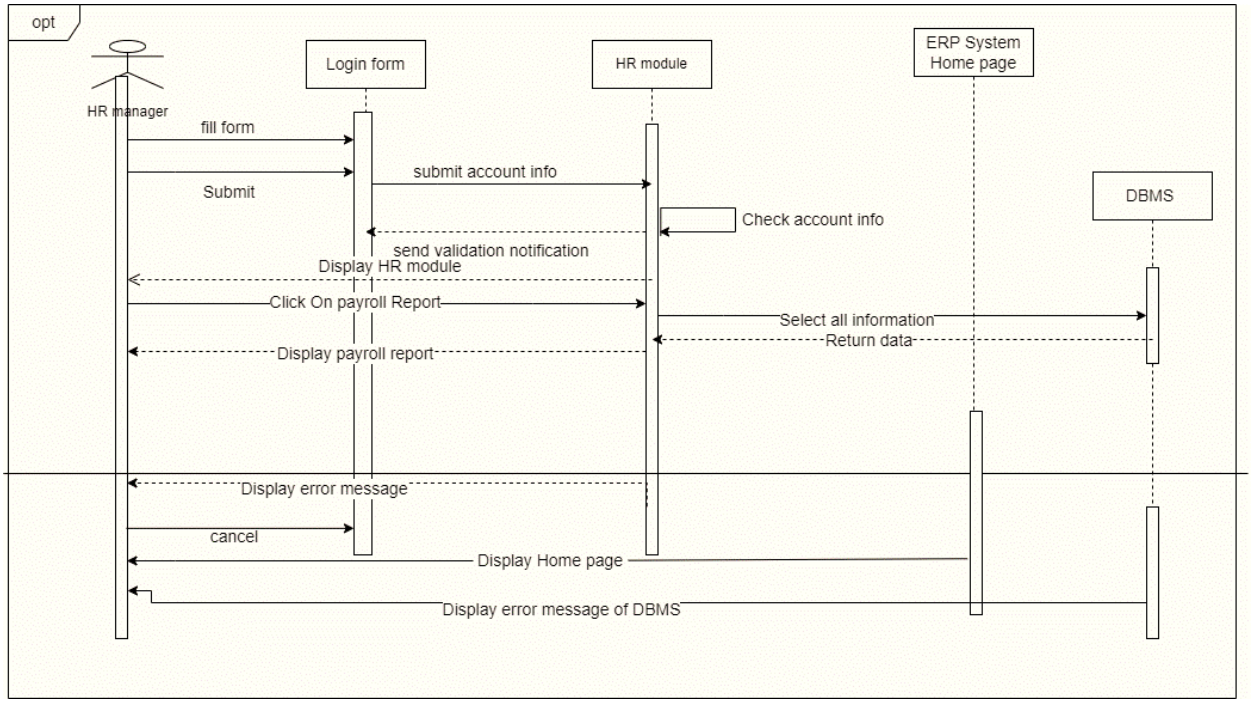
6. Sequence Diagram for Show Employee Time Management:



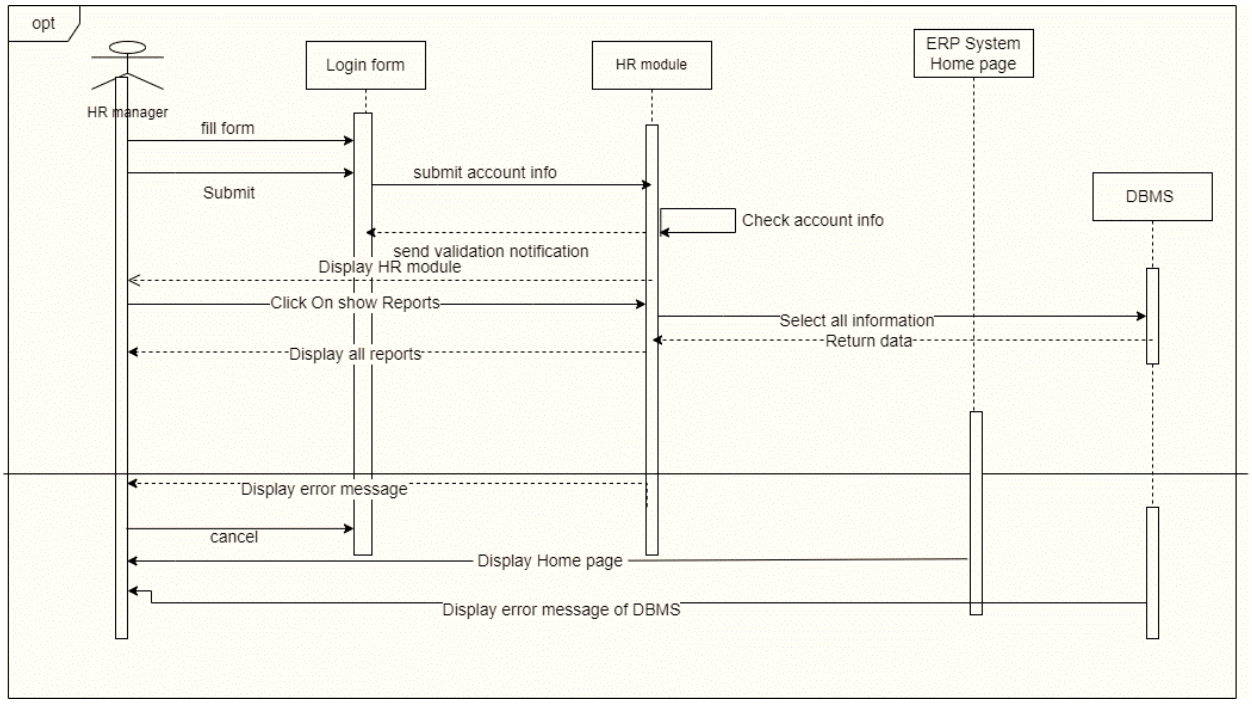
7. Sequence Diagram for Send Bonus & Deduction Report:



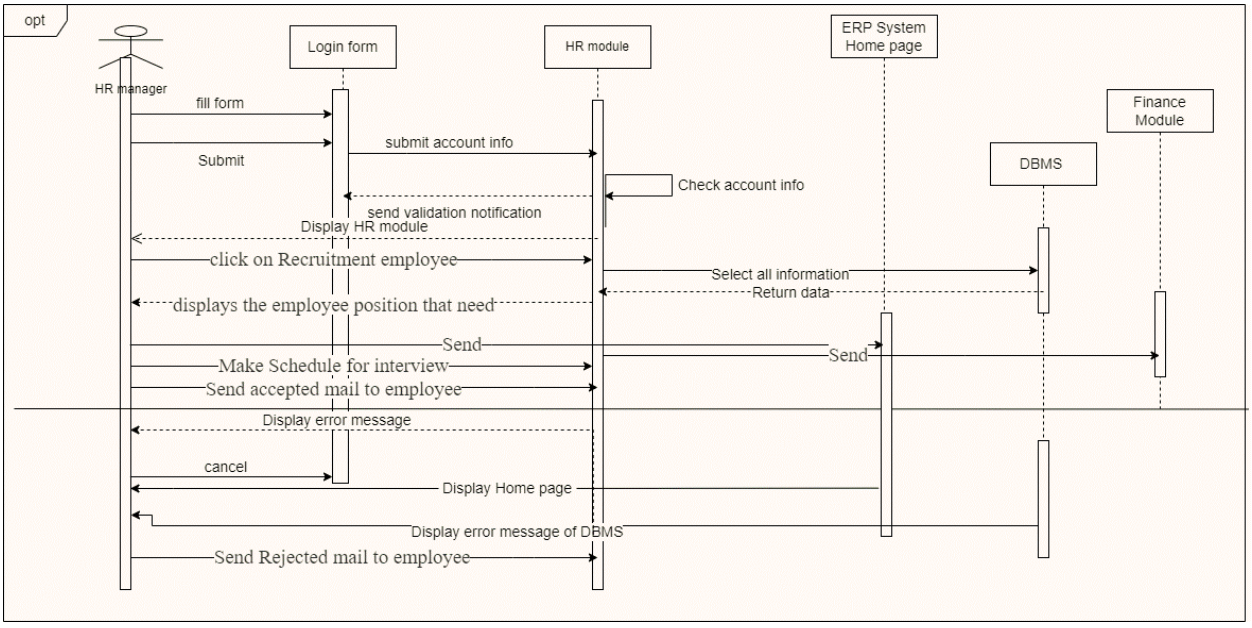
8. Sequence Diagram for Show Employee Payroll Report:



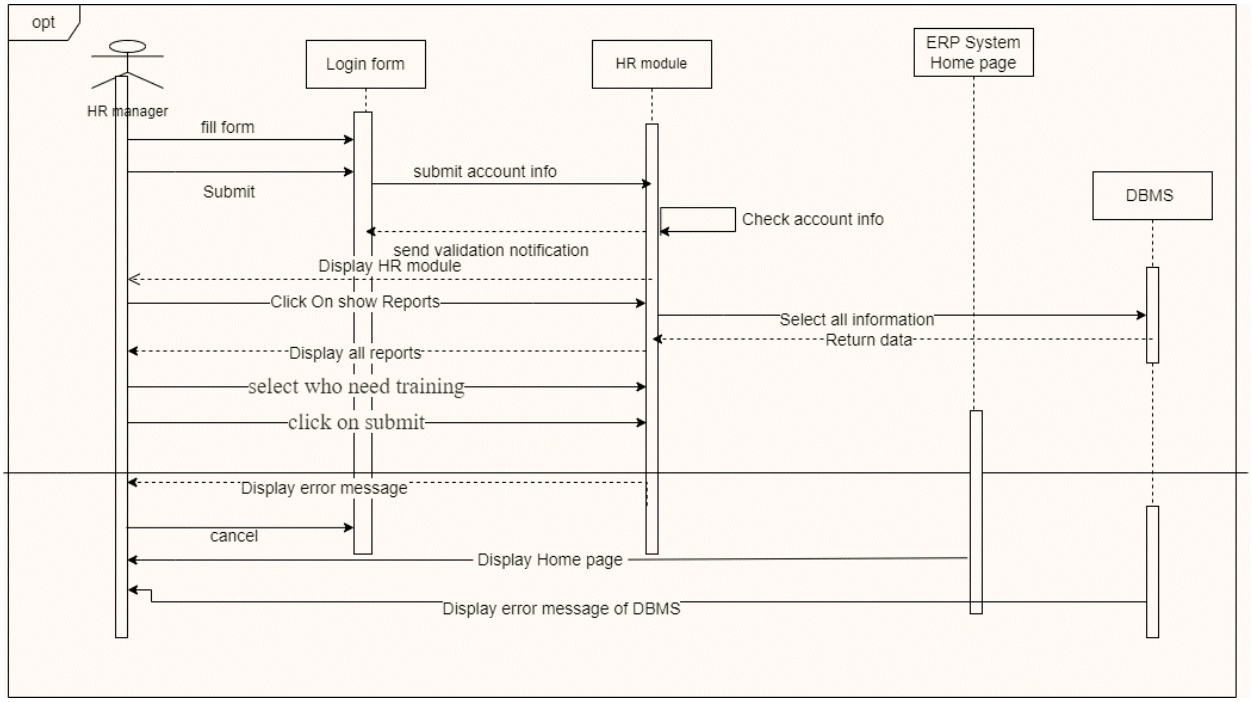
9. Sequence Diagram for Show All Staff Reports:



10. Sequence Diagram for Recruitment Employee:

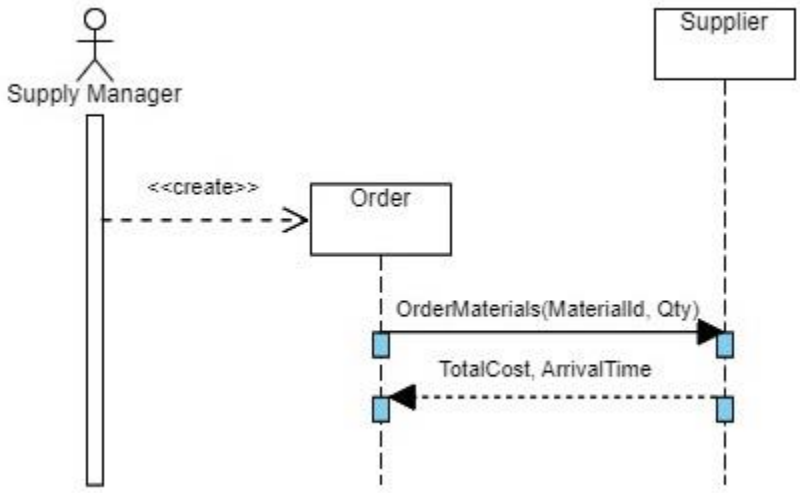


11. Sequence Diagram for Show Employees Who Need Training:

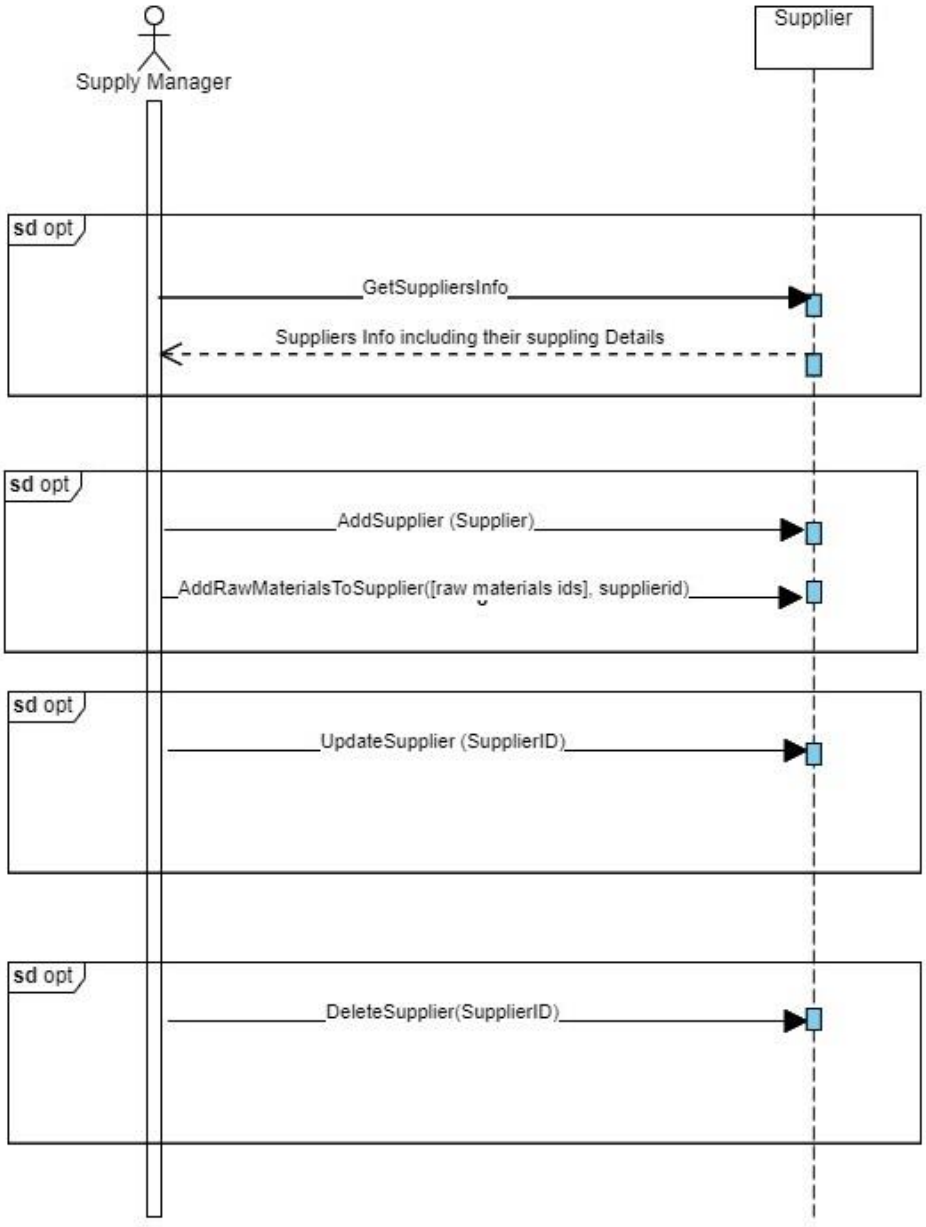


Supply Chain Management (SCM)an IM (Inventory Management):

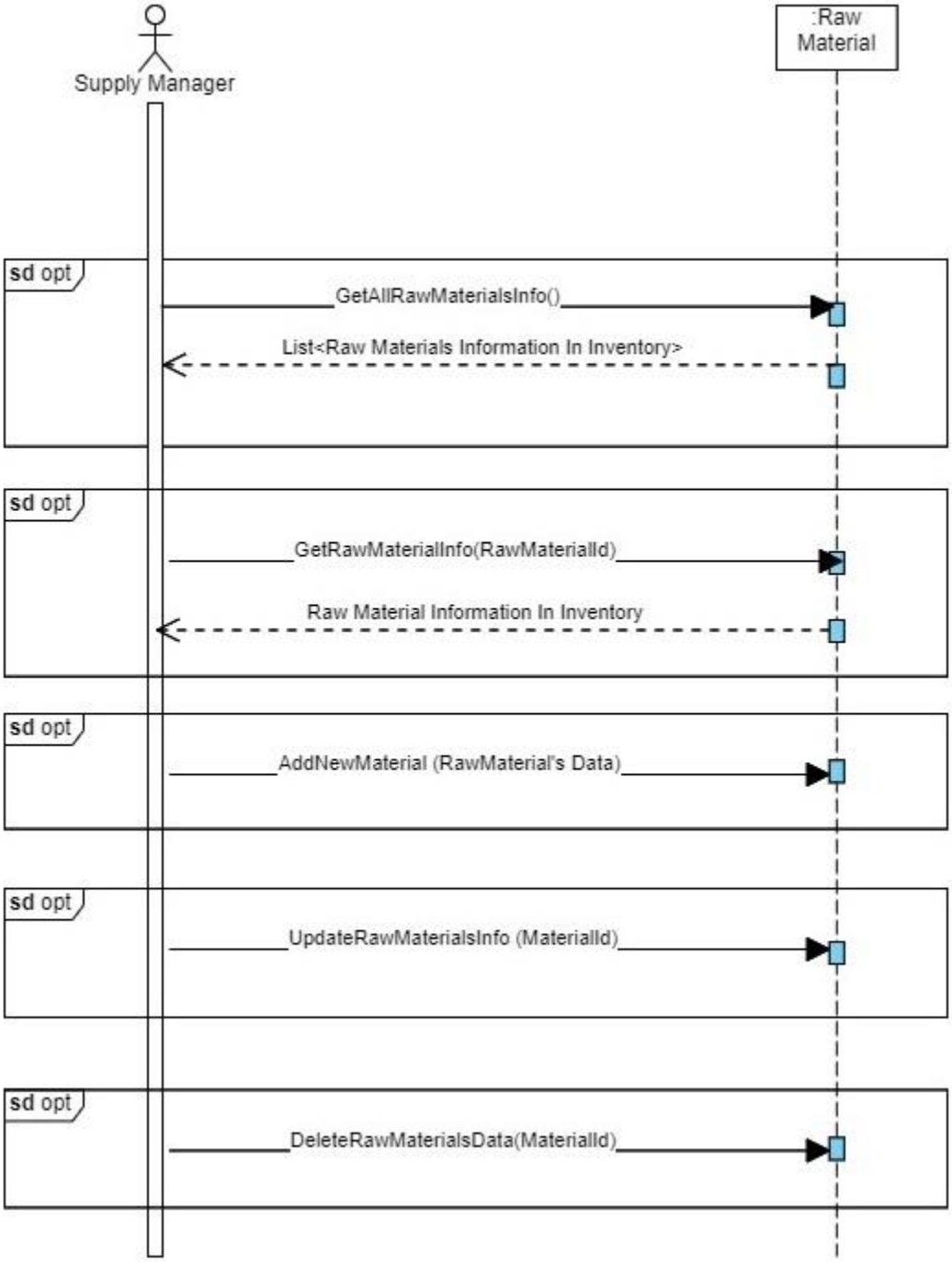
1- Create Order from Supplier



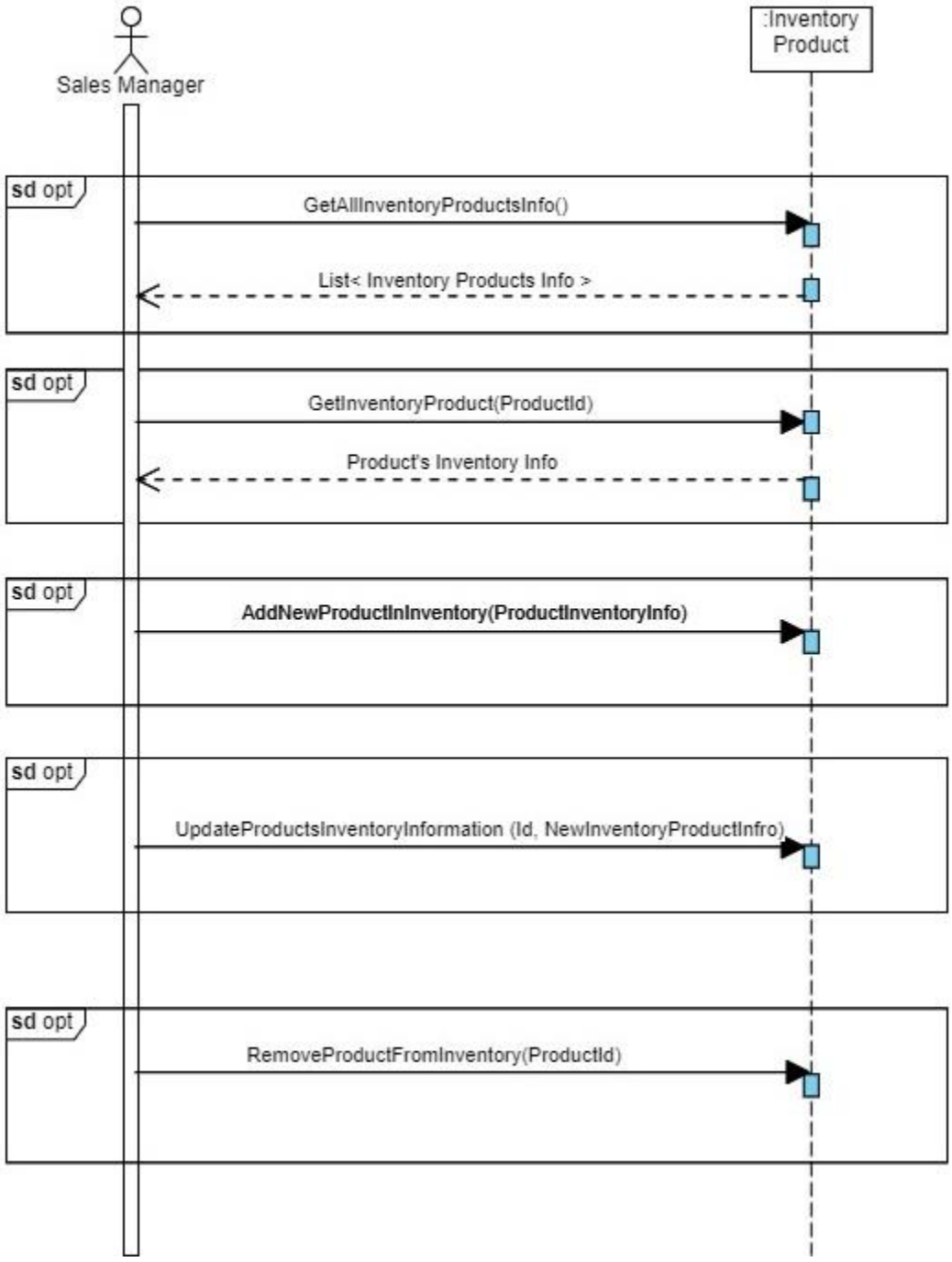
2- Manage Suppliers Info



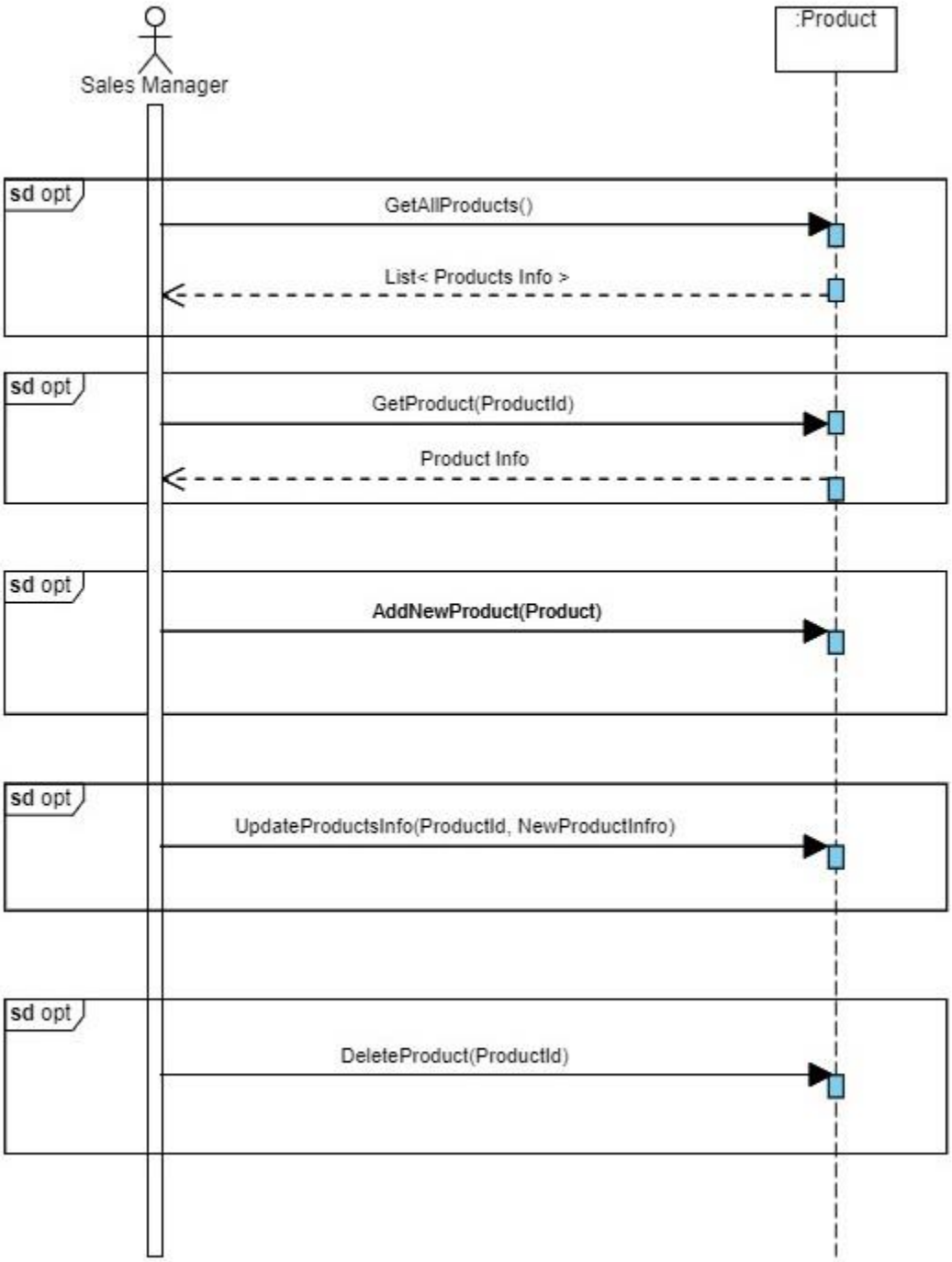
3- Manage Raw Materials Info



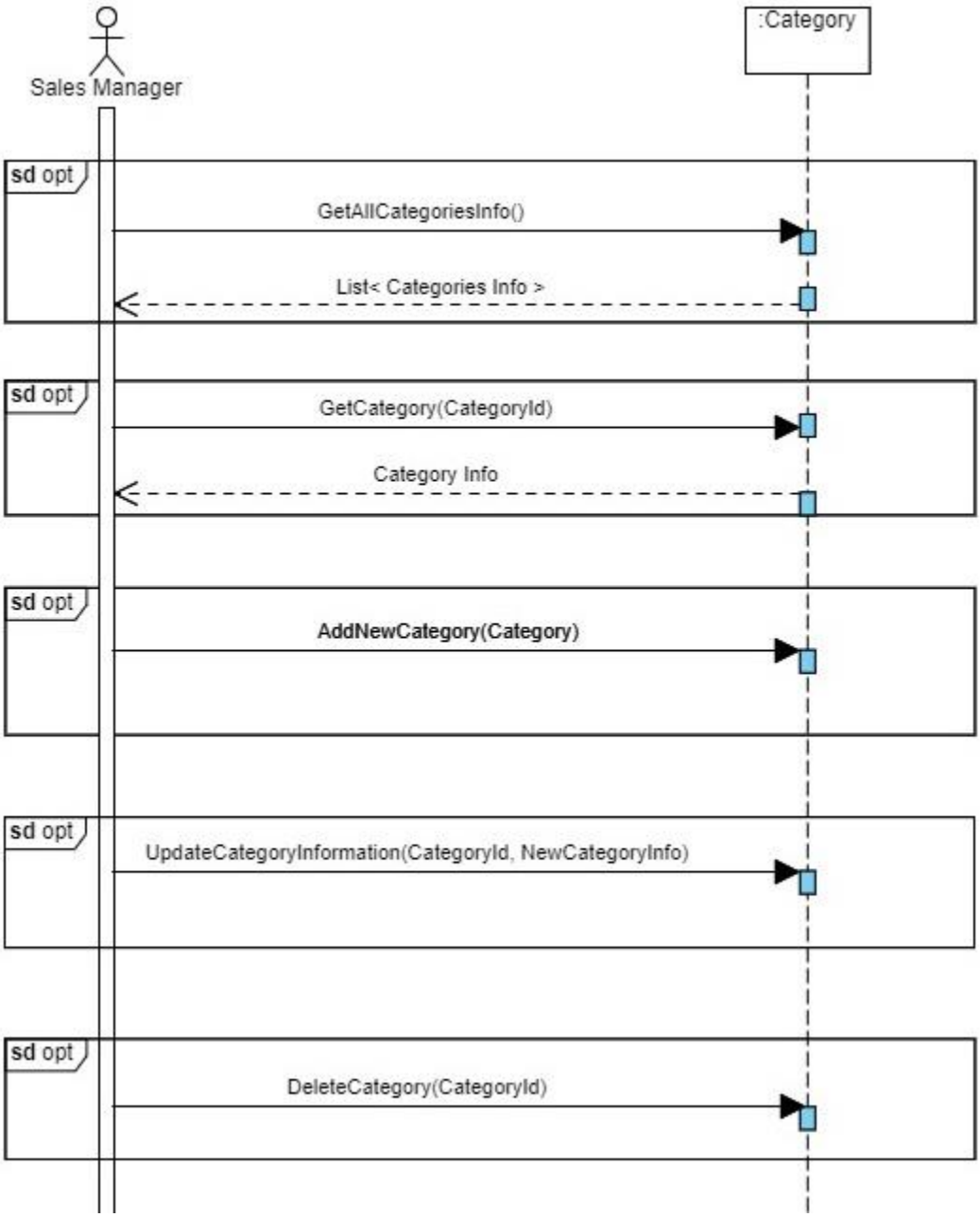
4- Manage Products in Inventory



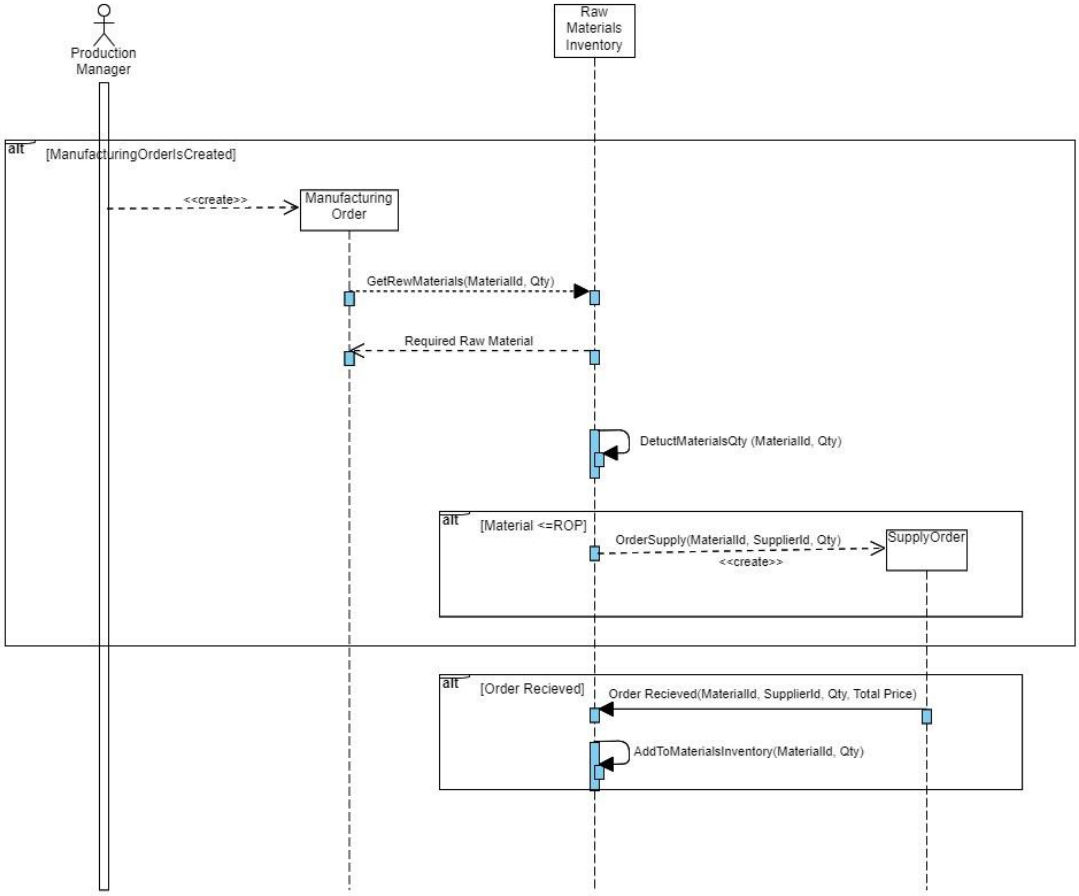
5- Manage Products Info



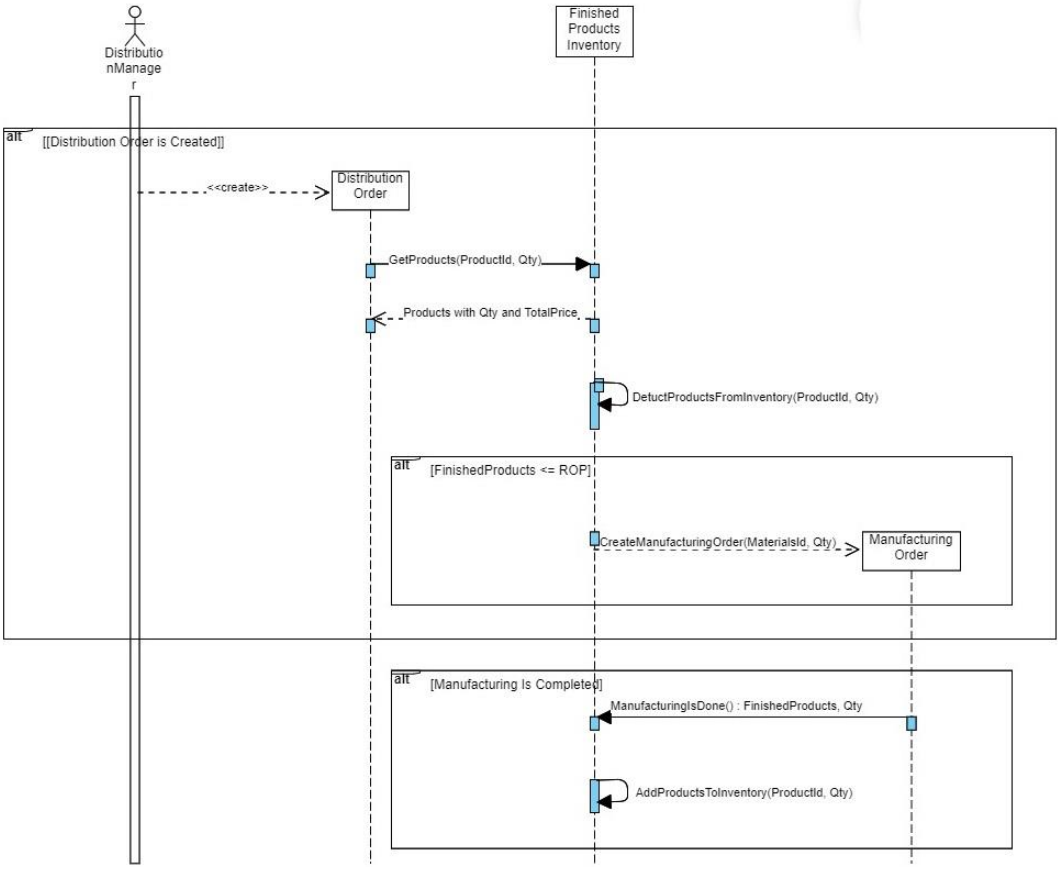
6- Manage Categories Info



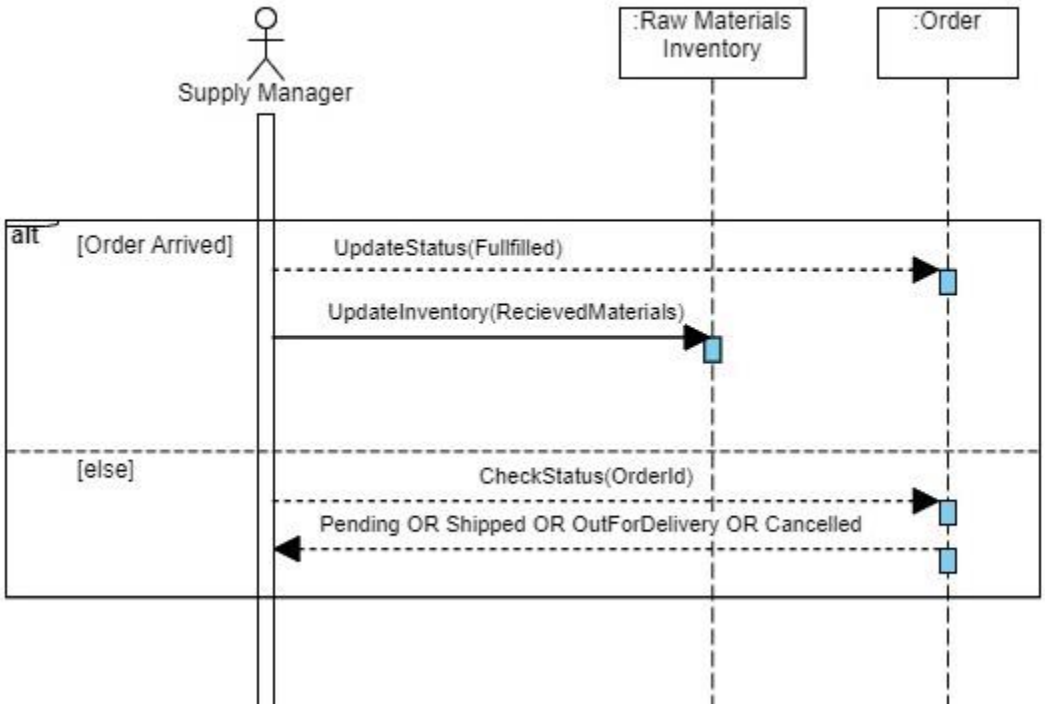
7- Manage manufacturing



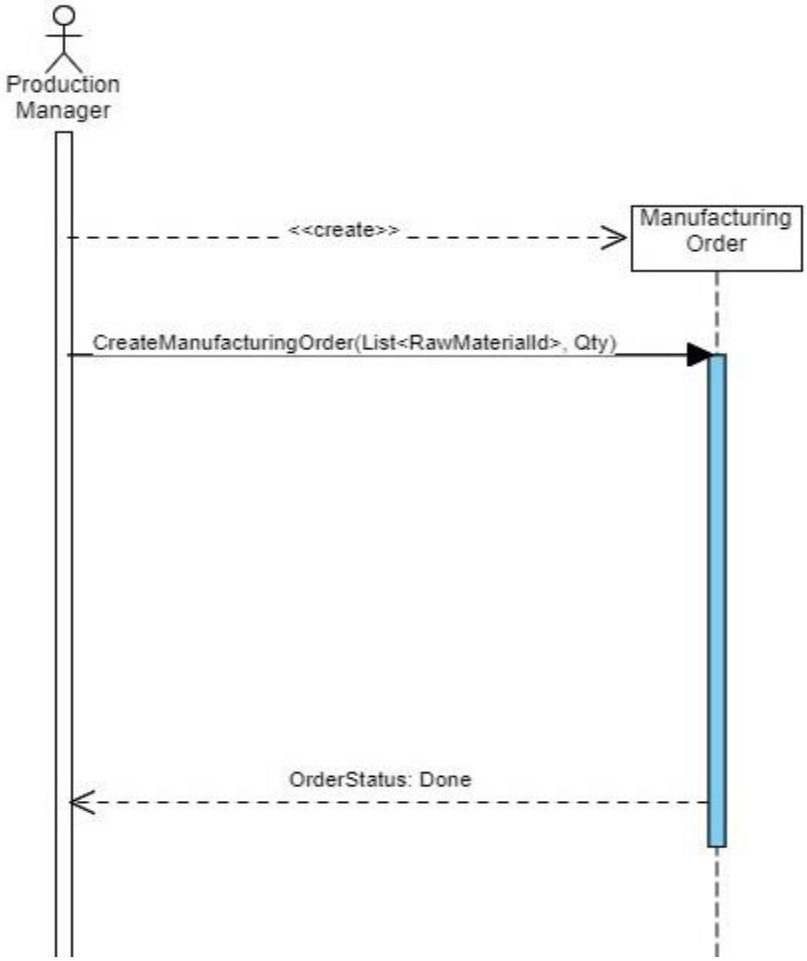
8- Create Distribution Order



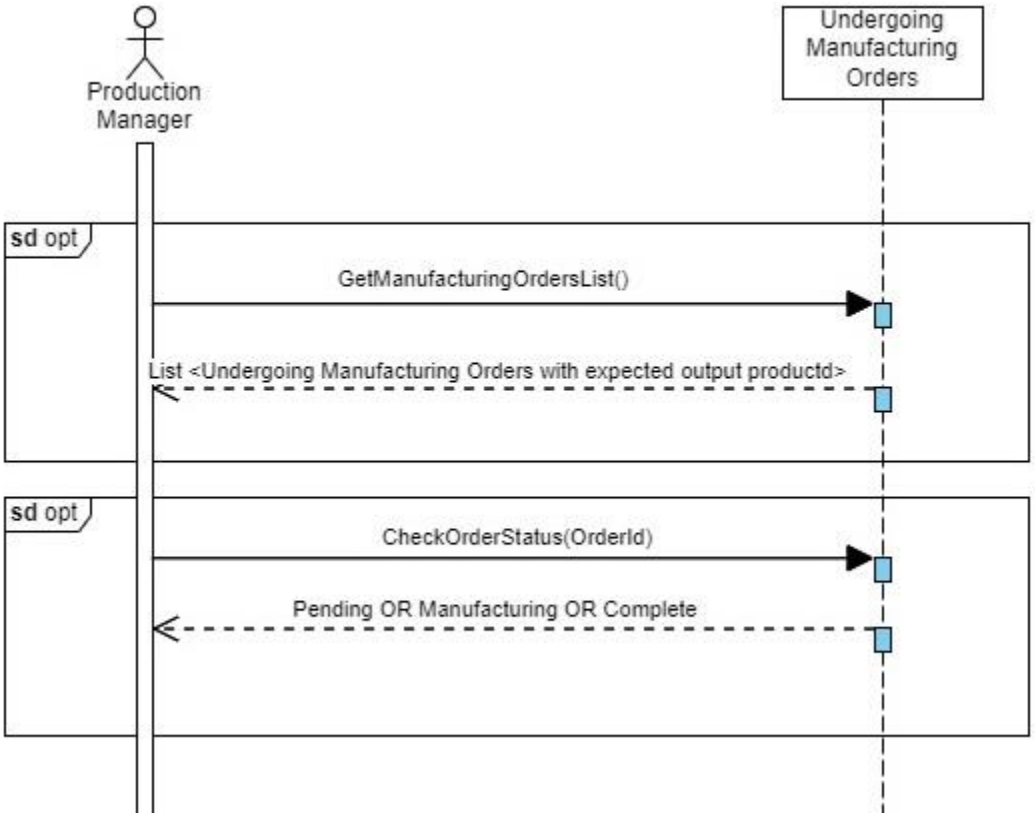
9- Track Orders from Supplier



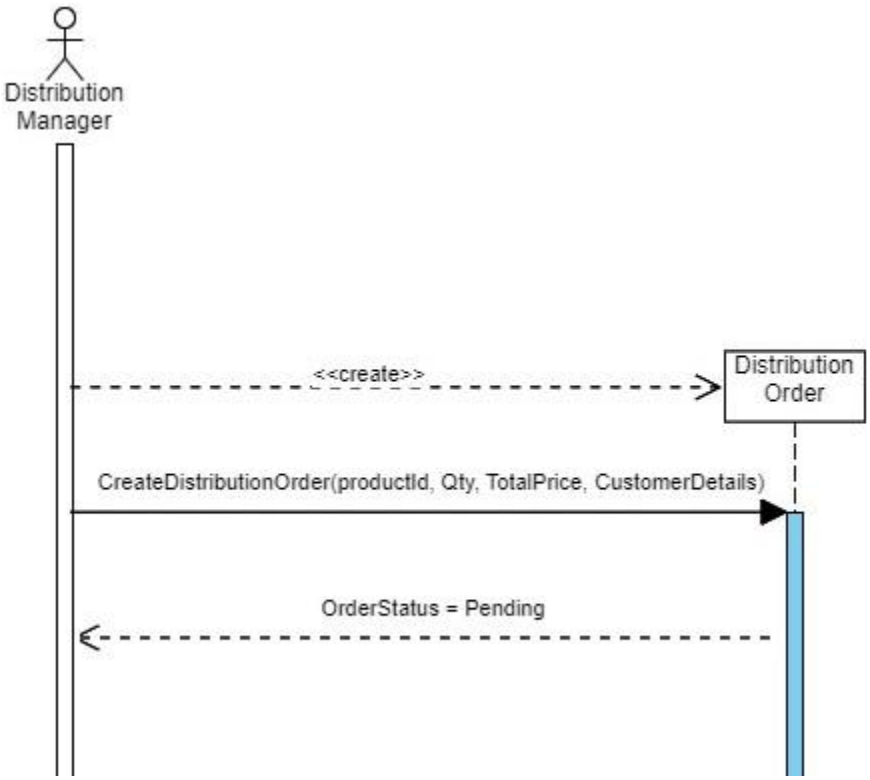
10- Create Manufacturing Order



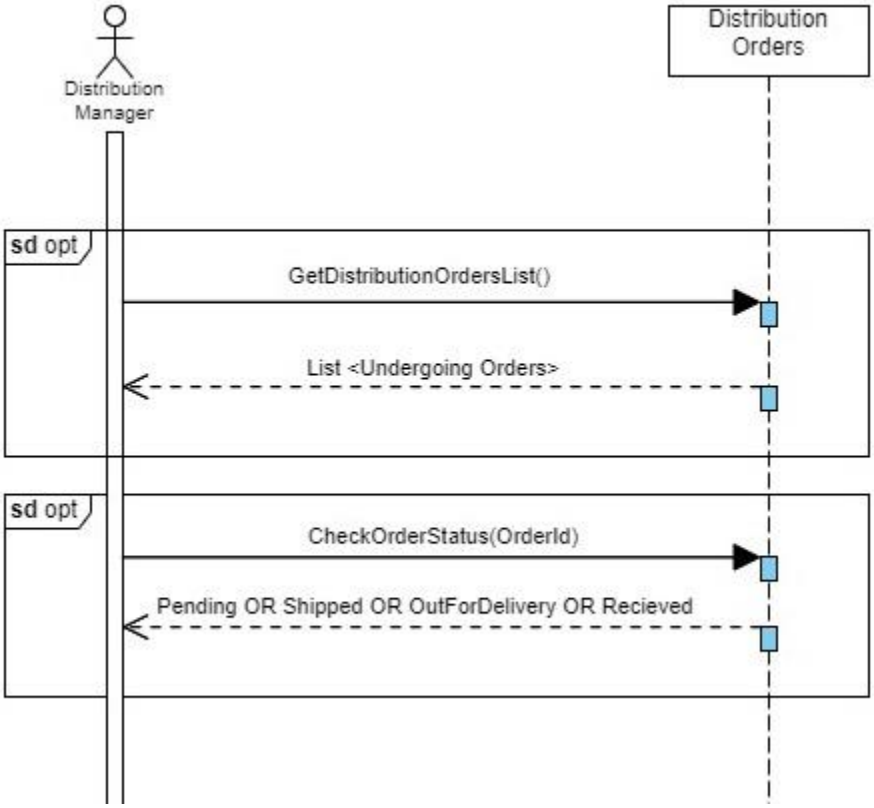
11- Track Manufacturing Order



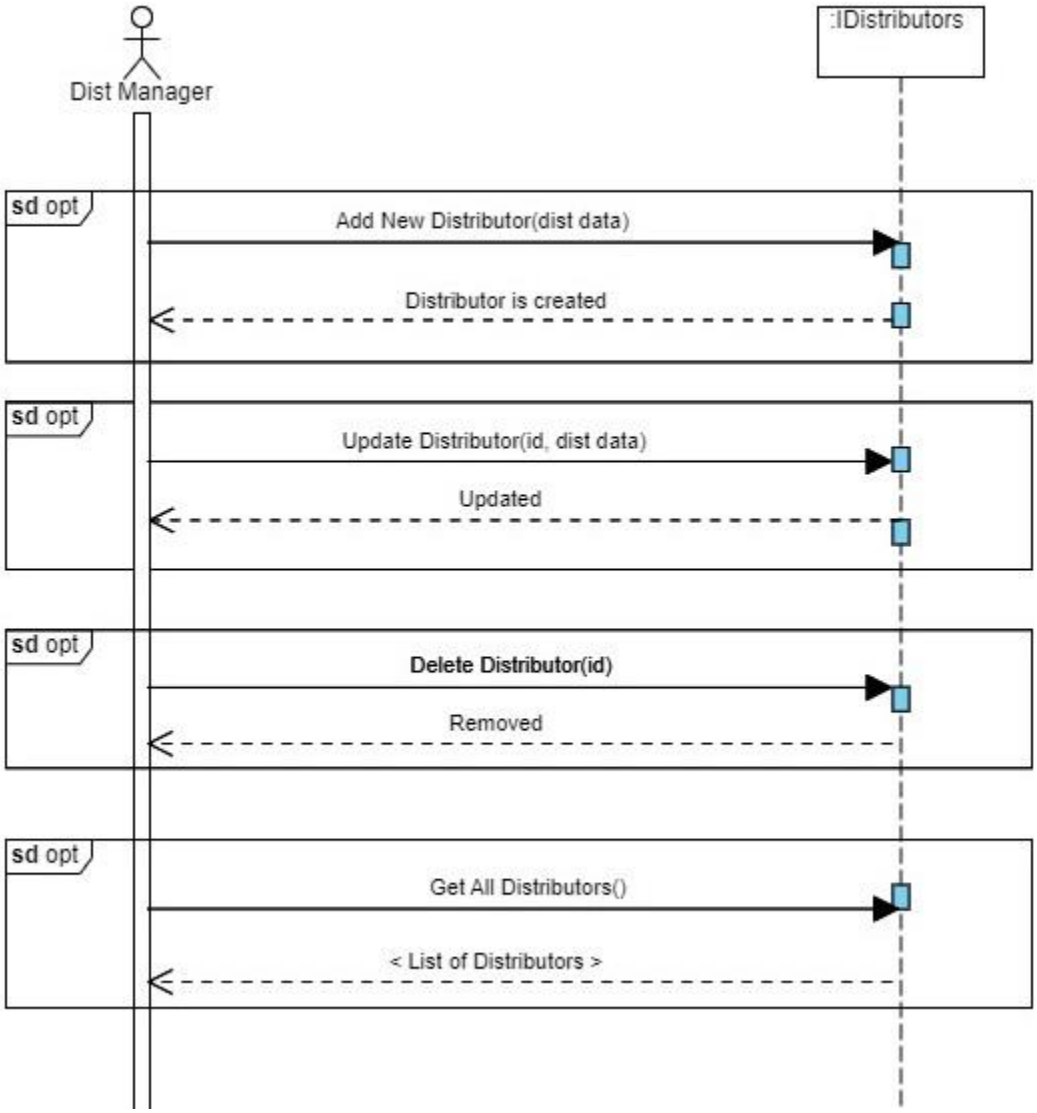
12-Create Distribution Order



13-Track Distribution Order

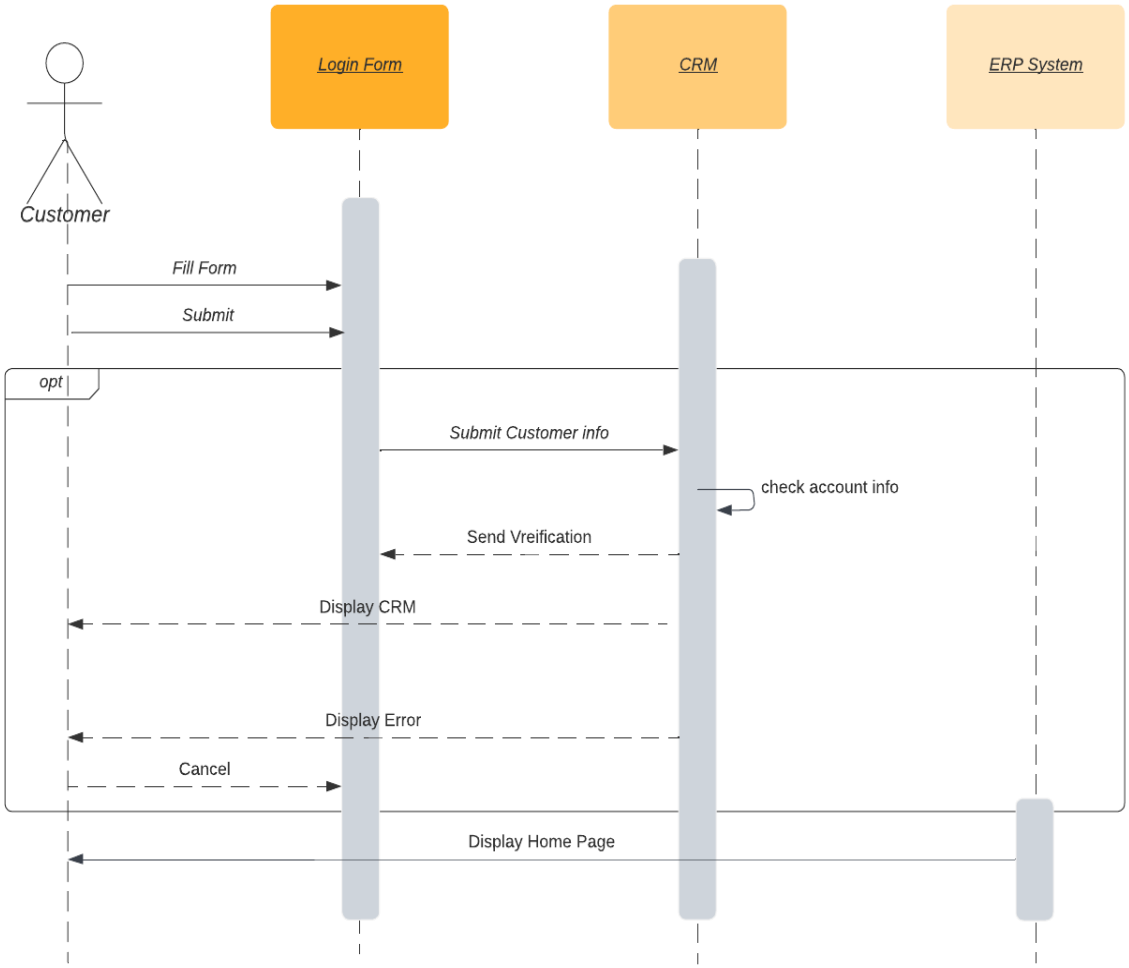


14-Manage Distributors Info

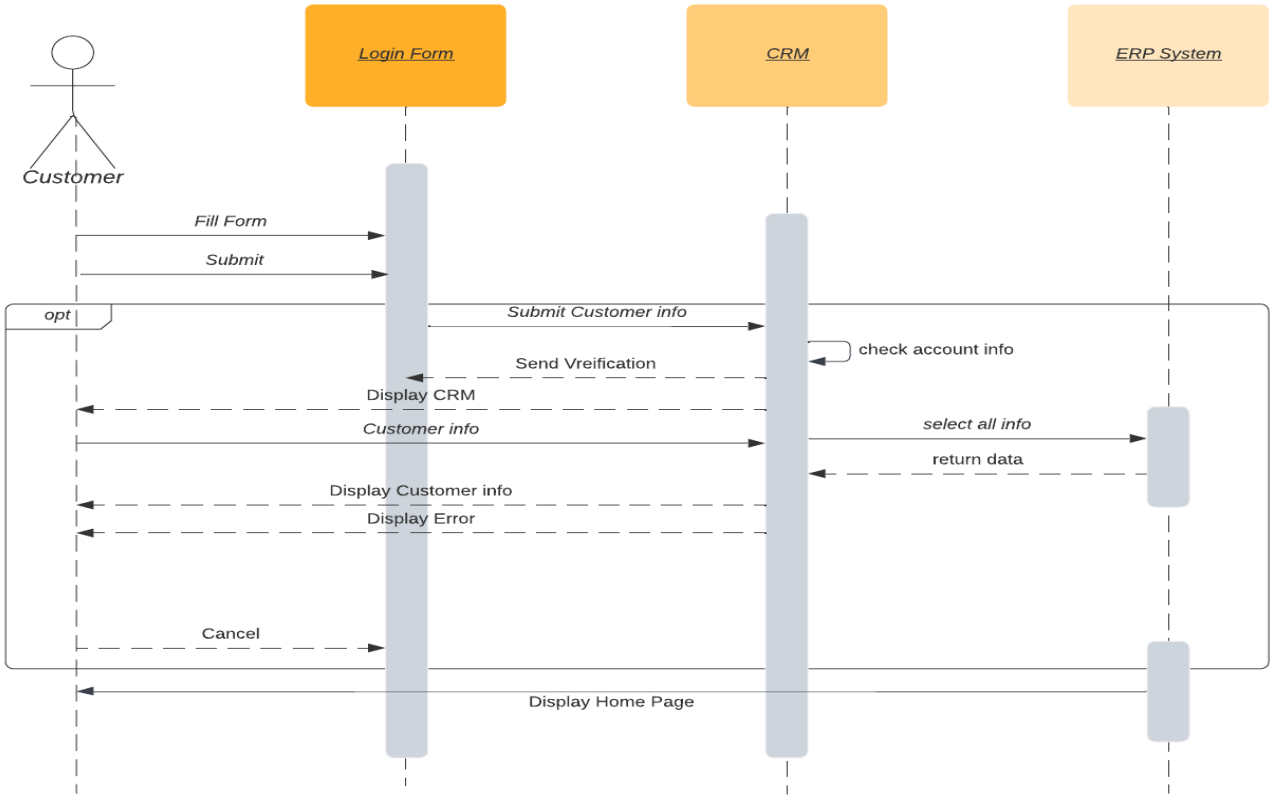


CRM

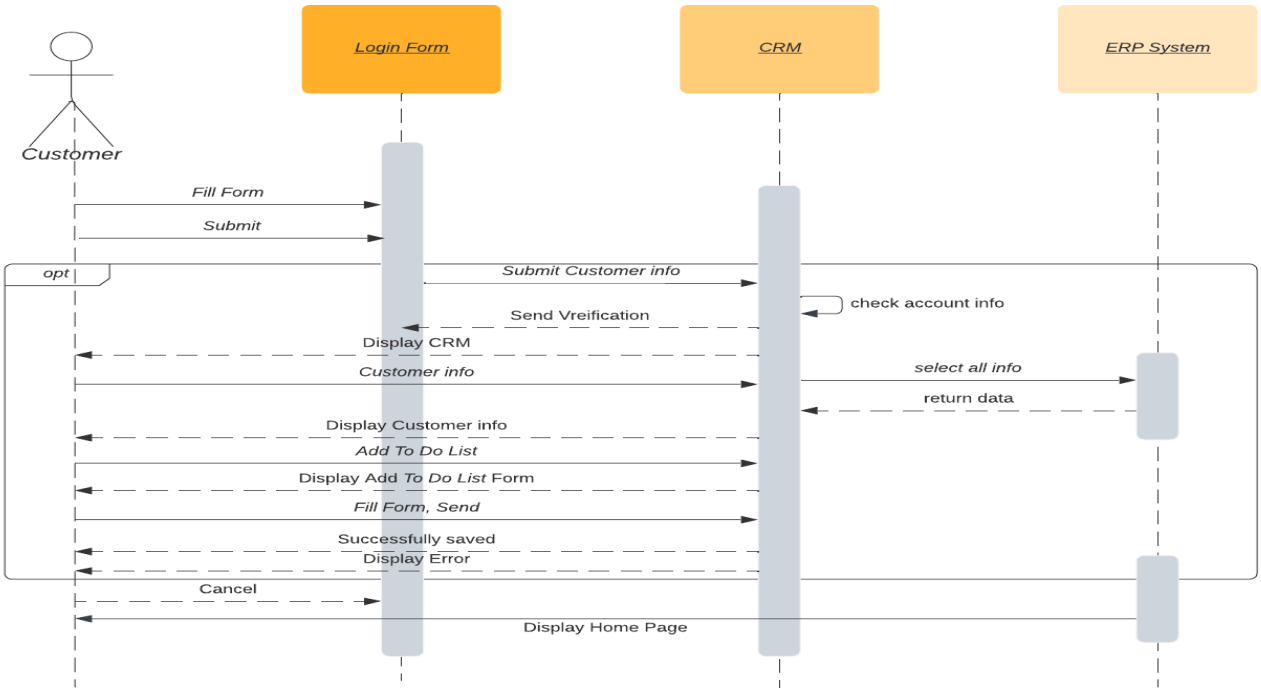
1 - Customer Login



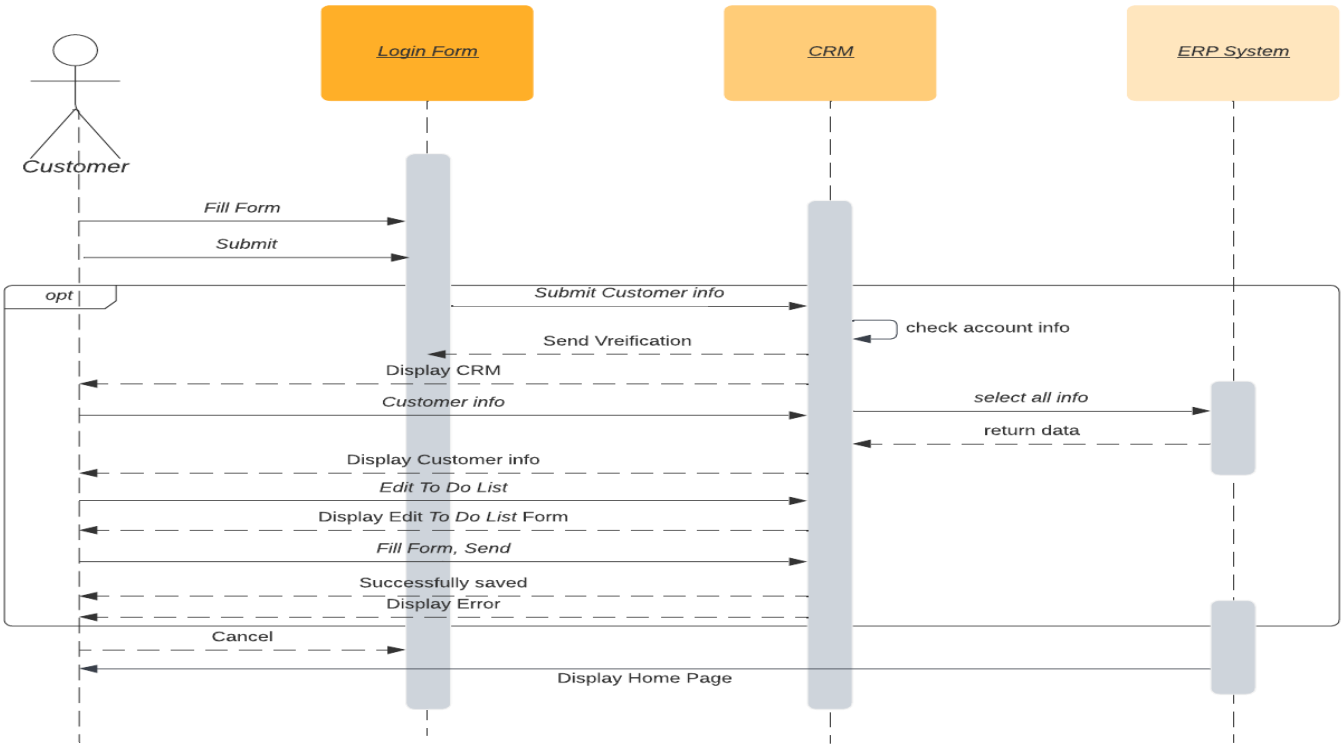
2 – Add Profile



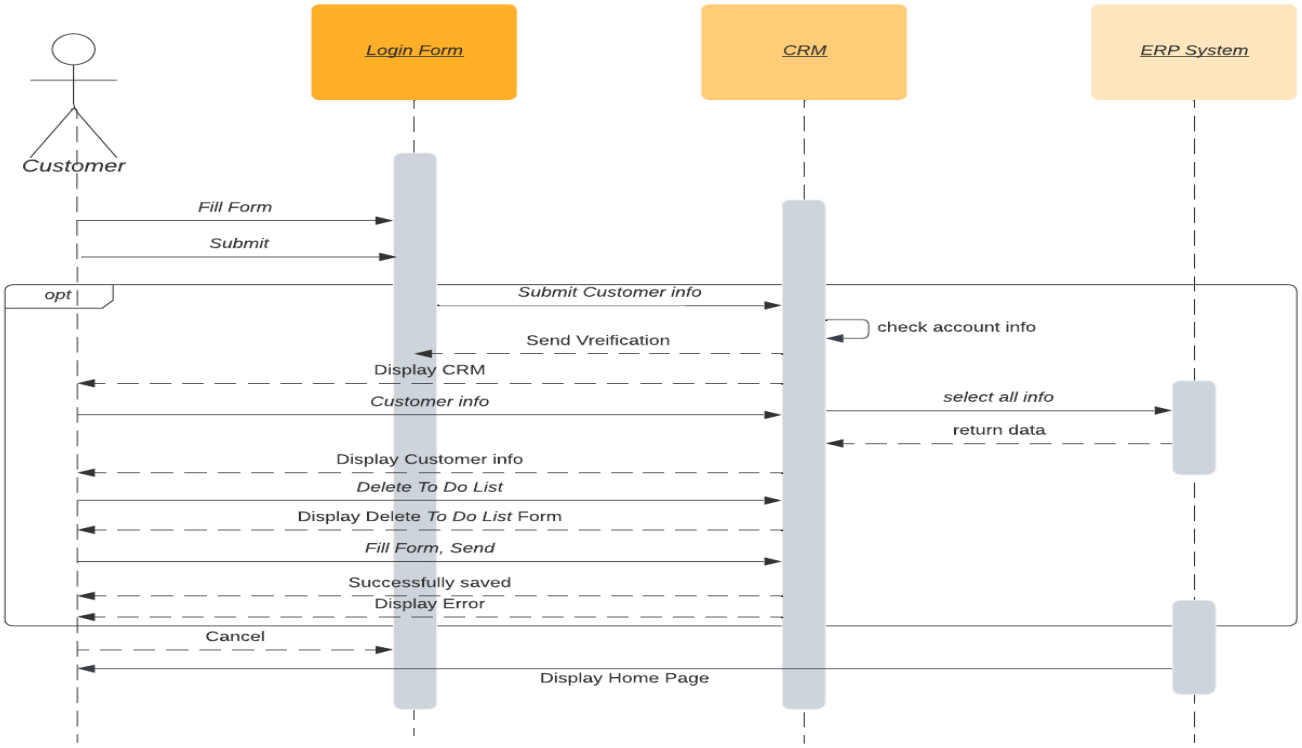
4 – Add To-Do List



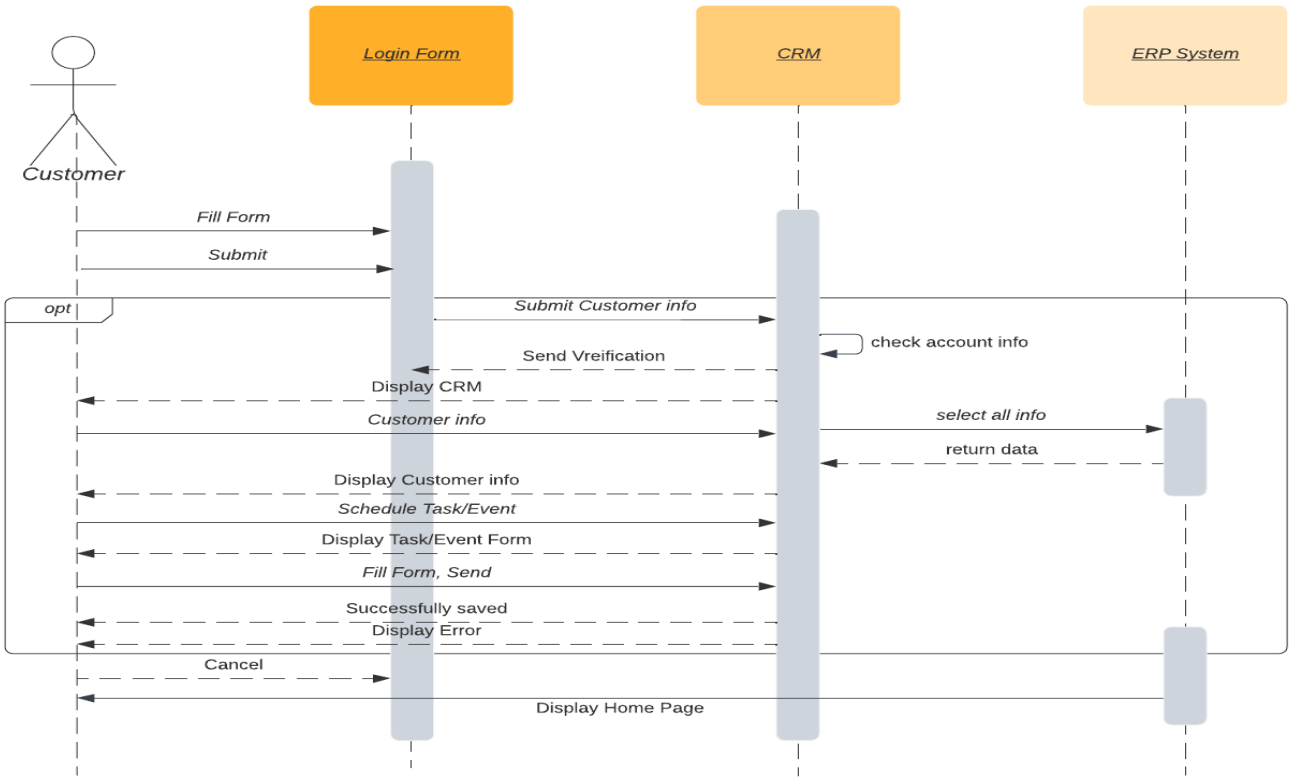
8 – Edit To-Do List



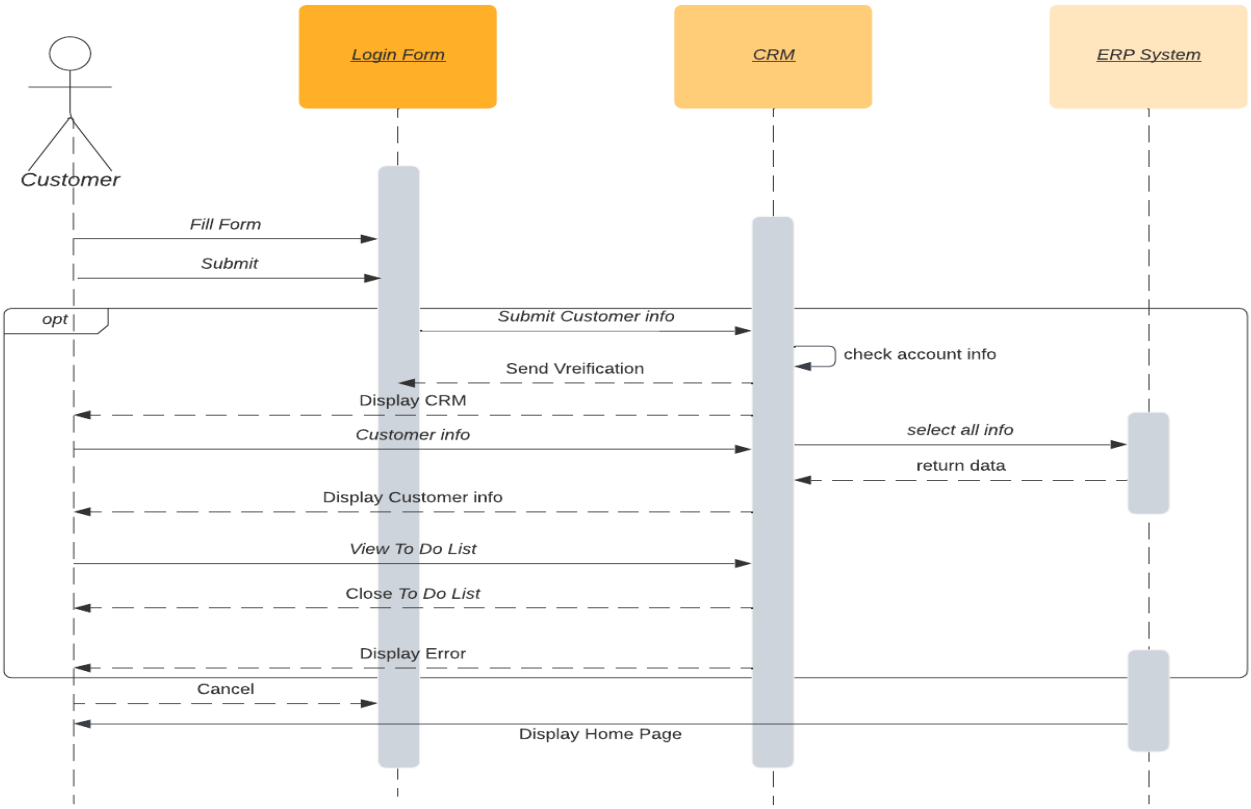
9 – Delete To-Do List



12 – Schedule Task / Event

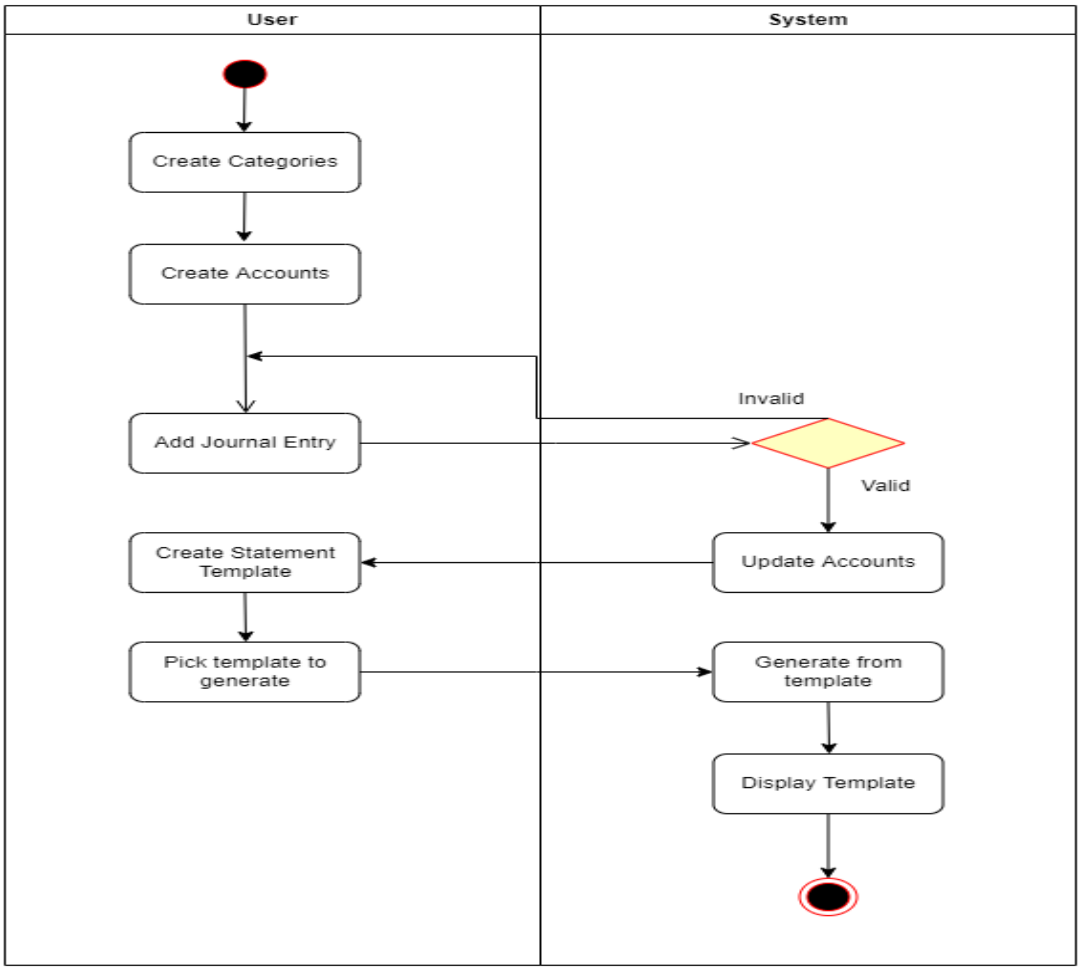


14 - View To-Do List

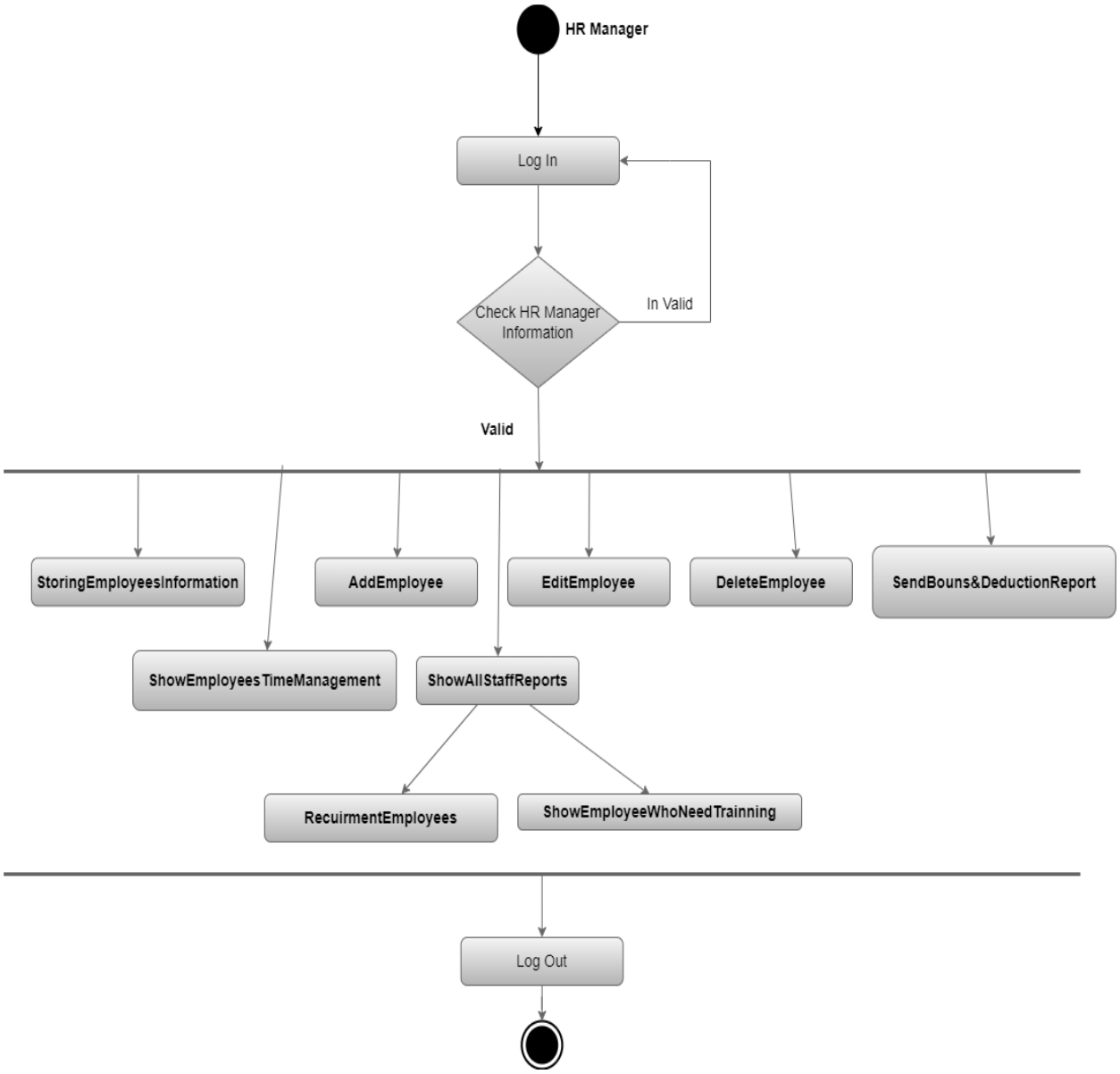


3.4 activity diagram

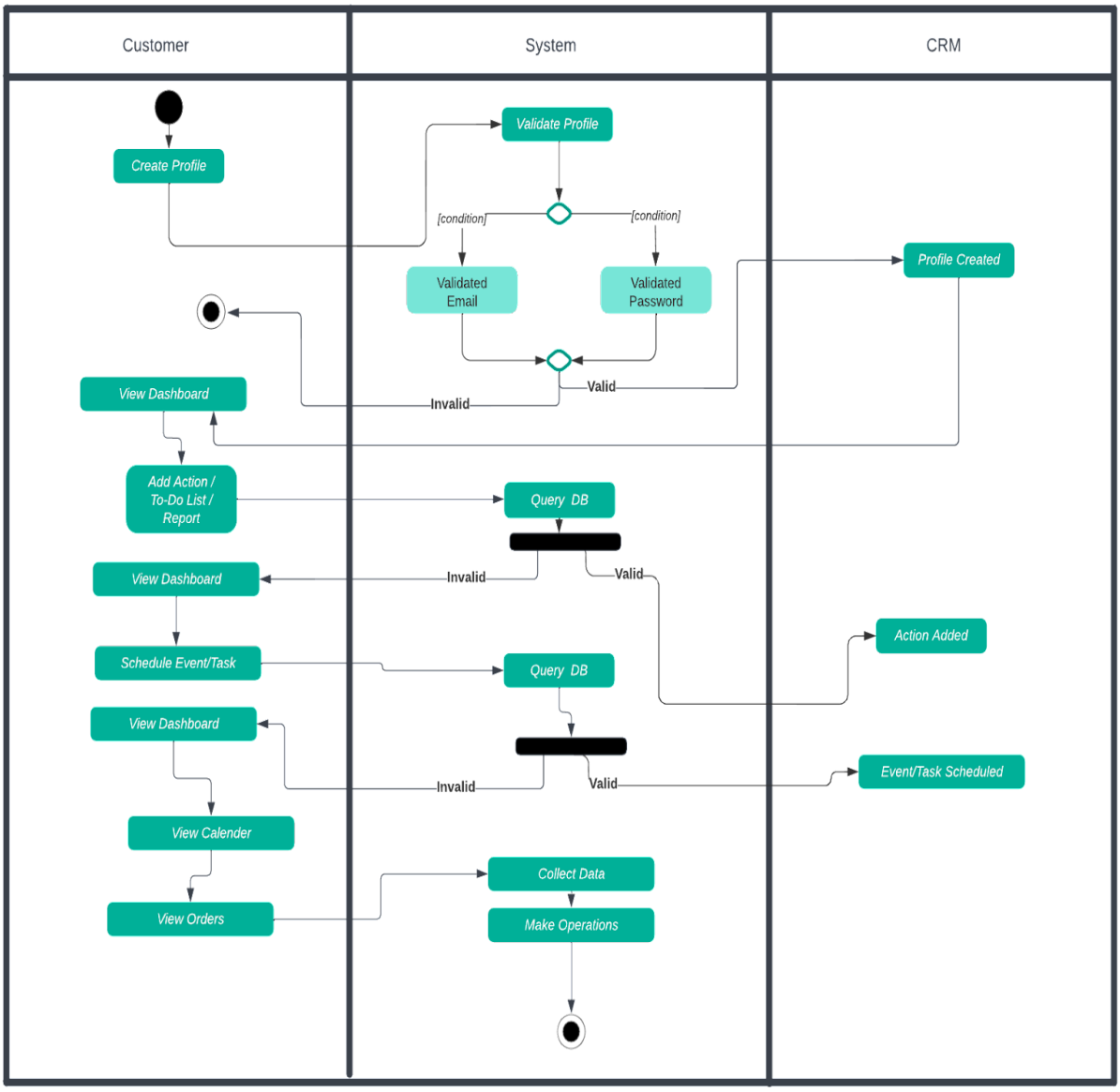
FMS



HR


HR Manager
Log In
Check HR Manager Information
In Valid
Valid
StoringEmployeesInformation
AddEmployee
EditEmployee
DeleteEmployee
SendBouns&DeductionReport
ShowEmployeesTimeManagement
ShowAllStaffReports
RecuirmentEmployees
ShowEmployeeWhoNeedTrainning
Log Out

CRM


Customer
System
CRM
Create Profile
Validate Profile
[condition]
[condition]
Validated Email
Validated Password
Profile Created
Invalid
Valid
View Dashboard
Add Action / To-Do List / Report
Query DB
Invalid
Valid
View Dashboard
Action Added
Schedule Event/Task
Query DB
View Dashboard
Invalid
Valid
Event/Task Scheduled
View Calender
Collect Data
View Orders
Make Operations

Chapter 4: Implementation

4.1 Software Architecture

- Software architecture refers to the high-level design of a software system that determines how the various components of the system interact with each other and how they are organized to achieve the desired functionality and performance.

- The infrastructure of this system depends on the **client-server architecture.**

- The **client side** is a React software *(a JavaScript library for building user interfaces)* which is responsible for presenting information to the user and receiving input from them.

- On the other hand, the **server-side** of a system consists of the back-end infrastructure that processes the requests from the clients and provides responses for them.
  (.NET core framework is used to build the server-side of the system by providing a number of features including support for network communication protocols such as TCP/IP and HTTP, as well as libraries for handling data serialization, encryption, and authentication)

- The server-side infrastructure often includes a **web server, application server, database server** and **ORM.**

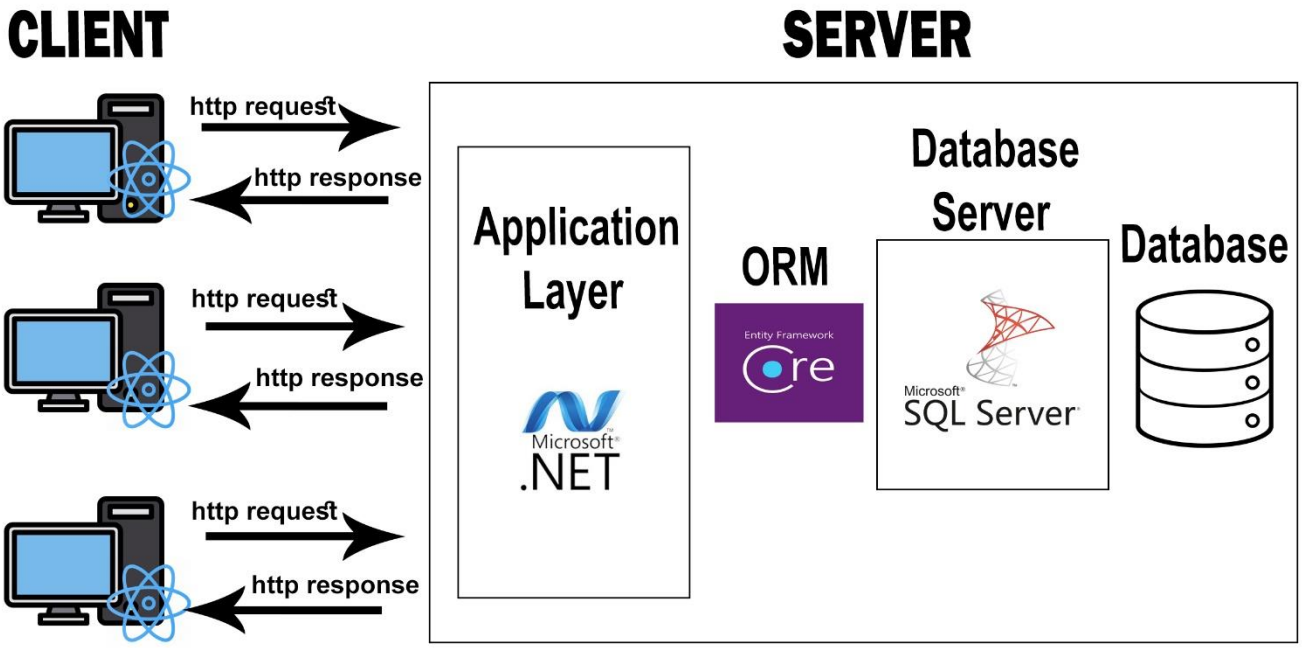
  1. **Web server:** A web server is responsible for serving web pages to clients that request them. It receives HTTP requests from clients (such as web browsers or mobile devices), processes them, and sends back HTTP responses that contain the requested web content.

  2. **Application server**: An application server is responsible for executing the business logic of a web application. The application server receives requests from the web server, processes them, and sends back responses that contains the results of the business logic. The application server may also interact with other components of the system, such as a database server, to retrieve or update data. Currently, The Application Server is a dot net core web app that runs on the machines localhost, which will further be published on a remote application server (Microsoft IIS) as mentioned in the future work.
  3. **Database server:** A database server is responsible for storing and retrieving data that is used by the web application. The database server receives requests from the application server to read or write data, processes them, and sends back responses that contain the requested data. The database server in this application is MS SQL Server

  4. **Object-relational mapping (ORM):** is a technique that allows software developers to map data between an object-oriented programming language and a relational database management system. It is a way for developers to interact with databases using object-oriented programming concepts rather than raw SQL providing a layer of abstraction that makes it easier to manipulate and query data. In our Case, Our System is Using Entity Frame Work Core 6 for providing ORM capability.

Together, these components form the backbone of a client-server architecture. and Communication between these components is typically carried out using standard **HTTP** and **TCP/IP protocols.** These protocols define the format and structure of the messages exchanged between the client and server, as well as the rules for establishing and maintaining a connection.

Diagrammatically, the communication between client and server of the System can be represented as follows:



- **The Process starts when the client machine consumes the Web APIs provided by the Application layer through sending an Http request to the server through a http method (either get, post, put or delete) and any associated data are sent through the request's header or the body.**
- **the server then processes the request, interact with the ORM to get the Data from the Database and returns the requested data to the client through an API response (data is either stored in the response header or in the response body)**

4.2 Pseudocode, Flowchart or workflow

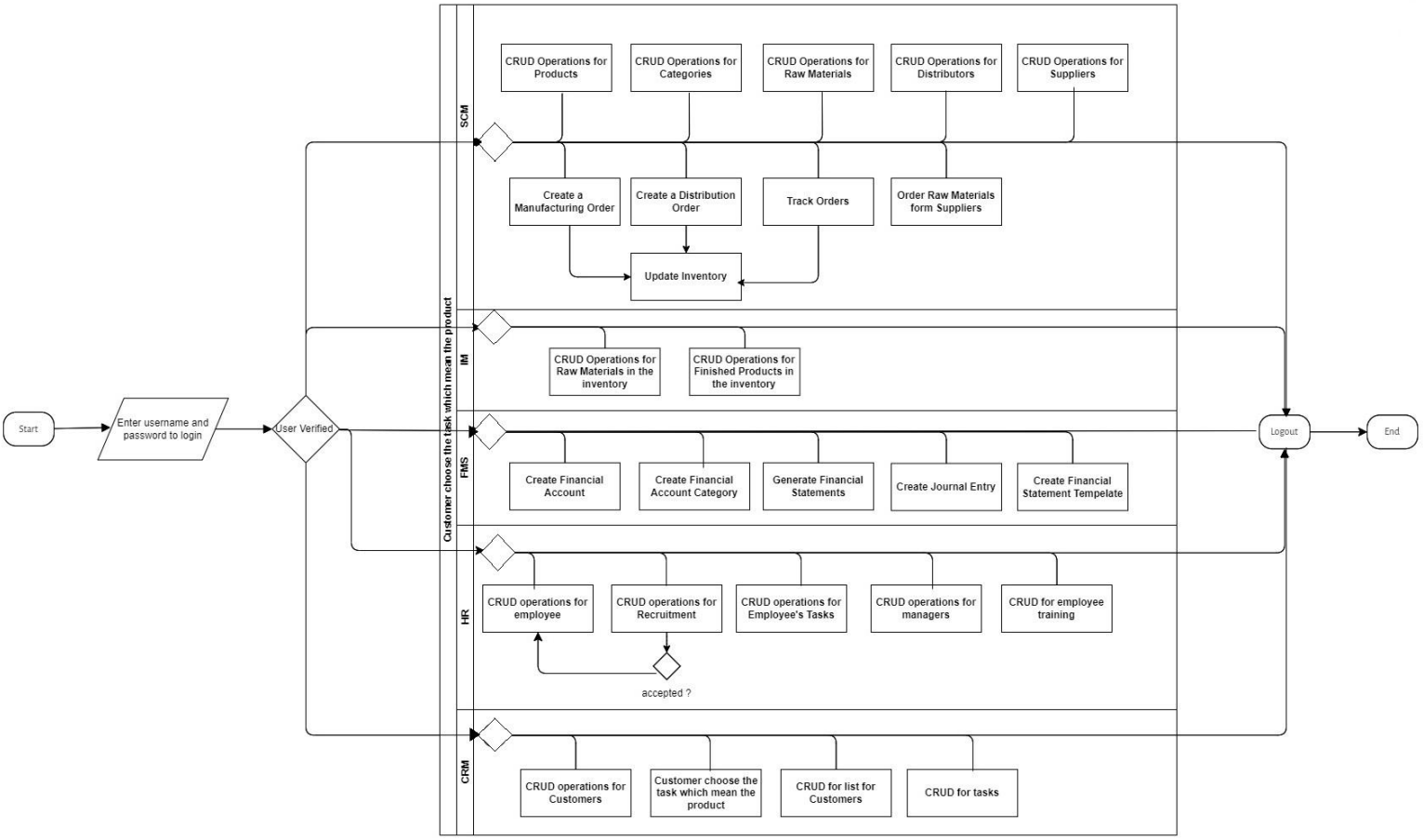
1. Pseudocode

Pseudocode for the system including communication between client and server are as follows:

1. User logs in in to the system
2. Send requests to the server to verify user
3. Server Receives request from client
4. If user is authenticated, he will be logged in successfully with a successful token otherwise not
5. Display dashboard and Home Page
6. Allow user to create, read, update, or delete data according to his roles in the system
7. User can send requests to server to perform CRUD operations
8. Server Receives the request/s from client
9. Server Processes the request/s
10. Retrieve or modify data in the database
11. Send response to client
12. client Receive responses from server
13. Display data on user interface
14. Log out and close connection

2. FLOW CHART

The following diagram represents a Hight-level flow chart diagram of the system



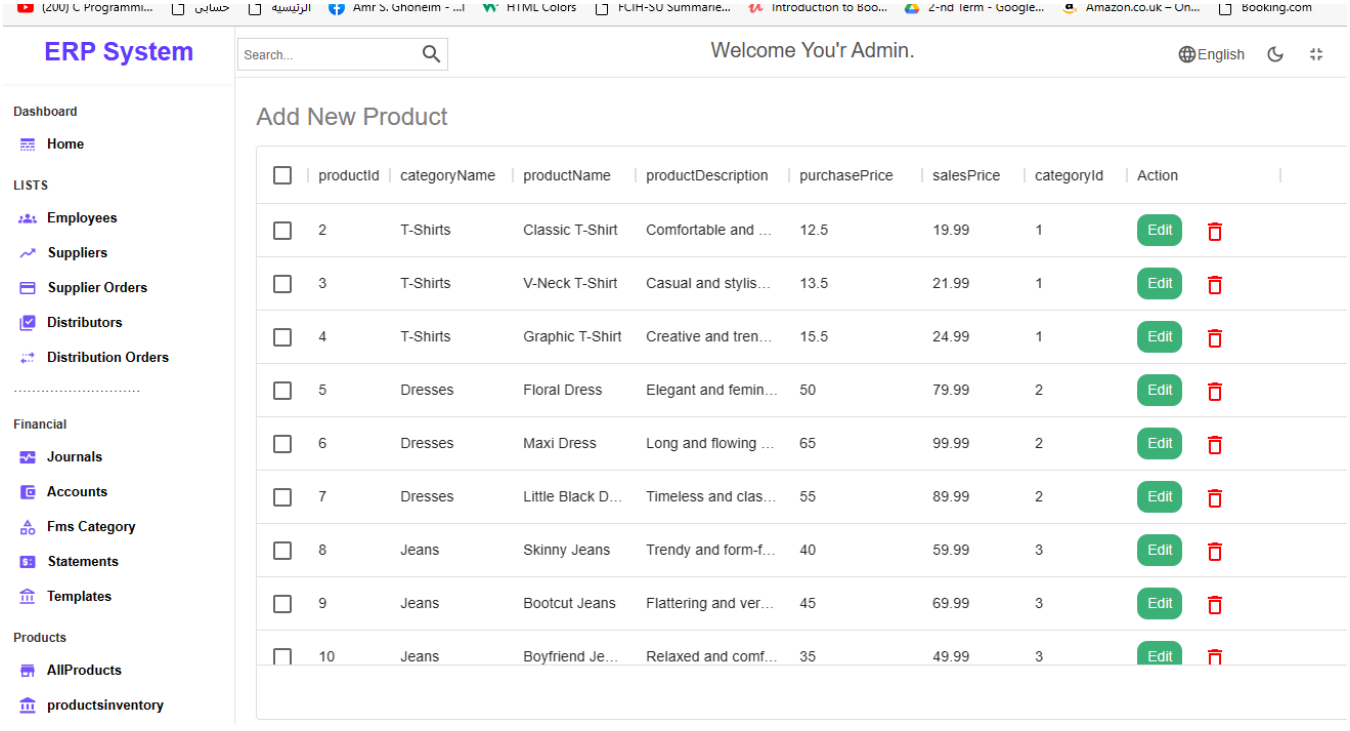
4.3 Code Snippets

Get all Products

```
[HttpGet]
0 references
public async Task<IActionResult> GetAllProducts()
{
    try
    {
        var products = await _unitOfWork.Product.GetAllAsync(new List<string>(){ "Category" });
        return Ok(_mapper.Map<List<ProductDTO>>(products));
    }
    catch (Exception ex)
    {
        return StatusCode(500, new ErrorExceptionResponse(500,null,ex.Message));
    }
}
```

```
public
src
  api
  Components
  context
    AccountContext.js
    AllproductContext.jsx
    CategoryContext.jsx
    darkModeContext.js
    darkModeReducer.js
    DistributionOrdersContext.js
    DistributorContext.js
    employee.jsx
    FmsCategoryContext.js
```

```
export function ProductInventoryContextProvider({children})
{
    const [data, setData] = useState([]);
    async function getProductInventory()
    {
     const ProductInventory = await axios.get(`https://localhost:44393/api/GetAllProductsInInventory`);
     setData(ProductInventory.data);
    }
    async function getProductInventorydel()
    {
     const ProductInventory = await axios.get(`https://localhost:44393/api/GetAllProductsInInventory`);
     return ProductInventory.data;
    }
    async function getProductInventoryById(id)
    {
        const productinventoryObject = await axios.get(`https://localhost:44393/api/GetProductInInventoryById/${i
        return productinventoryObject ;
```



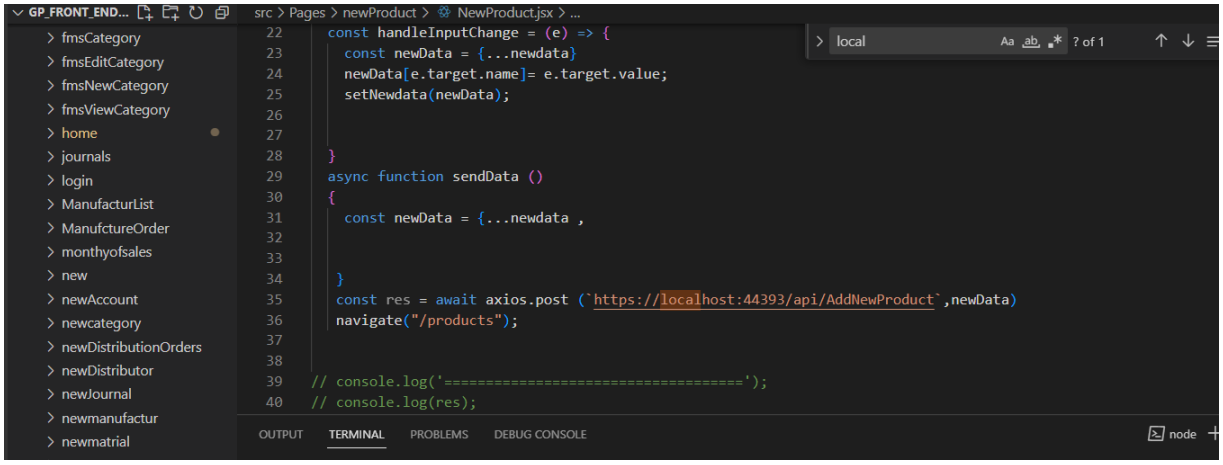
Add a New Product

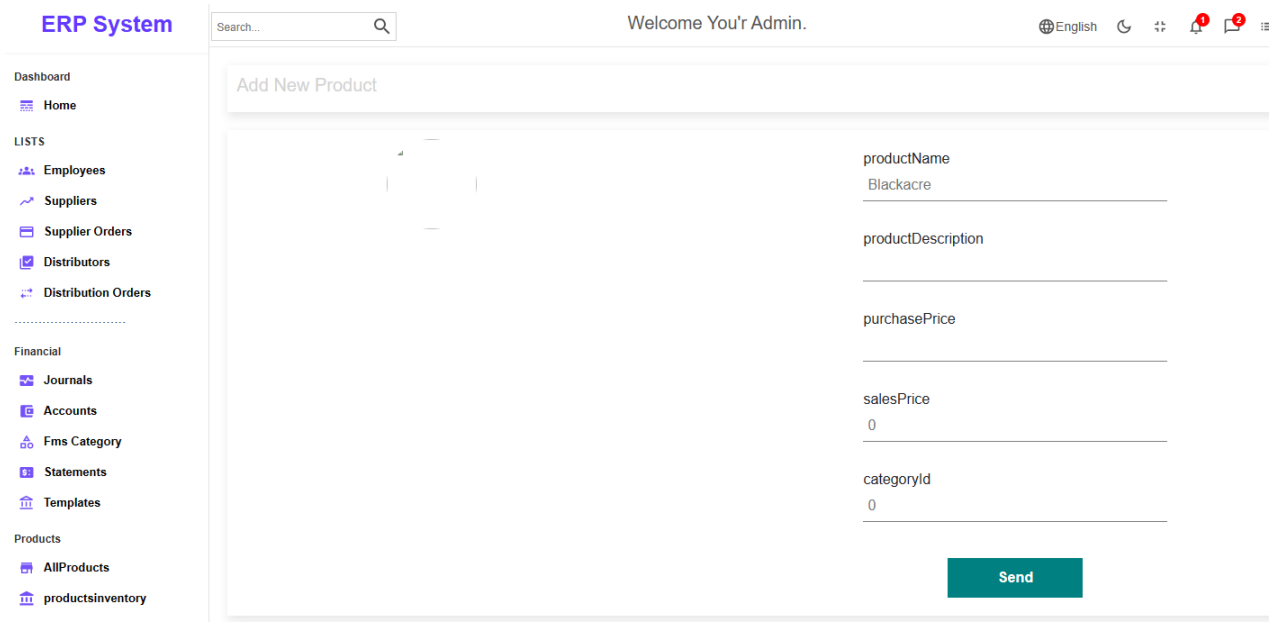
```
[HttpPost]
0 references
public async Task<IActionResult> AddNewProduct([FromBody] AddProductDTO product)
{
    if (!ModelState.IsValid)
    {
        return BadRequest(ModelState);
    }

    if (!this.ValidateCategoryId(product.CategoryId))
          return BadRequest(new ErrorApiResponse(400,"Invalid Category Id is Sent.") );

    try
    {
        _unitOfWork.Product.InsertAsync(_mapper.Map<TbProduct>(product));
        await _unitOfWork.Save();

        return NoContent();
    }
    catch (Exception ex)
    {
        return StatusCode(500, new ErrorExceptionResponse(500, null, ex.Message));
    }
}
```





Chapter 5: Testing

In this Chapter We are going to discuss our Enterprise Resource Planning (ERP) System testing, present some of type of testing, some bug reports and test cases we examined in our system.

5.1 Unit Testing

Unit testing is an essential component of the software development process. It involves testing individual units or components of the ERP system to ensure they function correctly. Unit tests should be automated and run frequently during the development process to catch issues early and provide confidence in the code's functionality.

5.2 Integrated Testing

Integrated testing involves testing the interactions and integration between different modules or components of the ERP system. The goal is to ensure that these components work together correctly and exchange data seamlessly.

5.3 Additional Testing

5.3.1 Performance Testing

This involves testing the system's performance under different loads and stress conditions to identify any bottlenecks or performance issues.

5.3.1 Security Testing

This focuses on identifying vulnerabilities and weaknesses in the system's security mechanisms, ensuring that sensitive data is protected.

5.5.3 User Acceptance Testing (UAT)

UAT involves testing the system with real users or stakeholders to ensure it meets their requirements and expectations.

5.3.4 Regression Testing

Regression testing ensures that previously implemented features and functionalities continue to work correctly after new changes or up- dates are introduced.

5.5.5 Accessibility Testing

This type of testing ensures that the ERP system is accessible to users with disabilities, conforming to accessibility standards and guidelines

5.4 Bug Reports

5.4.1 First Bug Report:

First Bug Report	
Bug Summary	**Application cannot add new employee training on clicking add new**
Bug ID	**It will be automatically created by the BUG Tracking tool once you save this bug**
Build Number	**Version Number 5.0**
Severity	**MEDIUM**
Priority	**HIGH**
Assigned to	**Mennatuallah Mohamed**
Reported By	**our back-end Team**
Reported On	**5/4/2023**
Status	**Active**
Environment	**Windows 10 /SQL Server 2019/ VS 2019 / React**
Description	**when Click on Add New Button in Employee Training Page System can't add it successfully**
Steps To Repro-duce	**1) Login into the application 2) Navigate to the Employee Training Add New 3) Filled To add New Employee Training 4) Clicked on 'Add New' button**
Expected result	**On clicking Add New button, should be prompted to a success message "Added Successfully".**
Actual result	**Application can't add employee training successfully on clicking the Add New button while creating a new Employee Training.**

5.4.2 Second Bug Report

Second Bug Report	
Bug Summary	**Application cannot update employee training on clicking update**
Bug ID	**It will be automatically created by the BUG Tracking tool once you save this bug**
Build Number	**Version Number 5.0**
Severity	**MEDIUM**
Priority	**HIGH**
Assigned to	**Mennatuallah Mohamed**
Reported By	**our back-end Team**
Reported On	**25/4/2023**
Status	**Active**
Environment	**Windows 10 /SQL Server 2019/ VS 2019 / React**
Description	**when click on update Button in Employee Training Page System can't update it successfully.**
Steps To Reproduce	**1) Login into the application 2) Navigate to the Employee Training update. 3) Filled To update Employee Training 4) Clicked on 'Update' button**
Expected result	**On clicking Update button, should be prompted to a success message "Updated Successfully".**
Actual result	**Application can't Update employee training successfully. on clicking the Update button while creating a new Employee Training.**

5.5 What is test case?

A Test Case is a set of actions executed to verify a particular feature or functionality of your software application.

5.5.1 **Test Case for login**

- Description: In this Test Case we are going to test login page.
- Prerequisite: system must be opened.
- Test Data: Email and password.
- Test Steps:
    1. open system.
    2. Insert email.
    3. Insert Password.
    4. click on enter button.
- Expected Results: The application should log in to home page.
- Result: pass.

- Test Scenario objective: Check Login Functionality there many possible.

Test case scenario				
Step NO	Description	Excepted Result	Actual Result	**Error Type**
1	Check results on enter valid Email and Password.	User navigates to home page	User goes to home success-fully.	
2	Check results on entering invalid Email and password.	User cannot navigate to home Page and see error message.	users see error message in-valid email or password.	
3	Check response when Email is empty, and Login Button is pressed.	User cannot navigate to home page and see error message.	Users see error message please enter email or password.	

5.5.2 Test Case for Add Employee

- Description: In this Test Case we are going to test Add Employee function.
- Prerequisite: system must be opened, and the user (Admin or HR Manger) must be logged in.
- Test Data: employee Full Name, tax Withholding, hours Worked, date Of Joining, attendance Time, holidays, employee Salary and HR Id.
- Test Steps:
    1. open system.
    2. Insert email.
    3. Insert Password.
    4. click on enter button.
    5. login to home page.
    6. choose Employee page.
    7. Insert Data for Employe.
    8. Click on Add New.
- Expected Results: The application should Add New Employee successfully.
- Result: pass.

Test case scenario				
Step NO	**Description**	**Excepted Result**	**Actual Result**	**Error Type**
1	Navigate to Employee Page.	User navigate to employee page.		
2	Enter employee fullname.	User can enter employee full name.		
3	Enter tax with holding	User can enter tax withholding.		
4	Enter hour worked	User can enter hours worked		
5	Enter date Of Joining	User can enter date of Joining		
6	Enter attendance Time	User can enter attendance time.		
7	Enter holidays	User can enter holidays		
8	Enter employee Salary	User can enter employee salary.		
9	Enter HR Id.	User can enter HRId.		
10	Press Add New	User Successfully Add New Employee.	Passed	

Chapter 6: Results and Discussion

6.1 Results

6.1.1 Expected result

Our goal was to develop an ERP system tailored for small to medium-sized businesses, to automate and integrate key functions within realistic constraints. We planned modules for financial management, supply chain, inventory, HR, customer management and a business intelligence module to provide insights. While striving to incorporate useful features, the system was not intended to match commercial ERPs aimed at large enterprises.

6.1.2 Actual results

While we successfully developed the core ERP modules to support basic functionalities, some advanced features were beyond the scope given resource and time constraints. The financial management module lacks aspects like financial years. However, we were able to implement templates and generate basic financial statements and reports. The business intelligence module had to be simplified considerably.

6.2 Discussion

Though not a complete commercial-grade ERP, our system prototype demonstrates the potential for:

- Streamlining routine tasks through automation
- Gaining data-driven insights from integrated reports
- Expanding functionality over time by adding useful features

With further development focused on delivering value, the system could evolve into a comprehensive in-house solution tailored to the needs of small to medium-sized businesses.

Chapter 7: Conclusion

In conclusion, this report has presented an overview of the project and its progress. The initial objective of the project was to develop a web-based ERP platform for managing an Enterprise Core business operation and provide managers with a summarized report. The project team successfully achieved this objective by designing and developing a user-friendly system that incorporates all the required features. The system can be accessed through a web browser, and it is compatible with various operating systems and devices. Additionally, the platform has been tested, and it has been found to be stable and reliable.

In order to enhance the project, the project team would require additional resources. First, the platform could benefit from the integration of artificial intelligence (AI) and machine learning (ML) technologies. This would enable the platform to learn from historical data and make accurate predictions on future demand. Additionally, the platform could be further enhanced by incorporating a mobile application to enable users to access it on-the-go. Secondly, the project team could improve the platform's security features by implementing multi-factor authentication, encryption, and regular security audits. Finally, the project team could improve the platform's scalability by implementing cloud-based technologies, which would enable the platform to handle higher volumes of traffic and data. With these enhancements, the system would be more efficient, secure, and user-friendly, thereby providing a better experience for the Enterprise.

Chapter 8: Future work

Moving forward, there are several areas that could be improved to further enhance the project. Firstly, is to deploy the system on a remote cloud server to assist the enormous growth of data.

Secondly, is to continue to update and add new features to the platform in response to changing customer needs and technological advancements. This could include incorporating new AI and ML algorithms, developing additional integrations with external systems, and improving analytics and reporting capabilities to have a better way of viewing insights.

Another area for future work is to expand the platform to new markets and geographies. This could involve adapting the system to support multiple languages and currencies, as well as complying with local regulations and standards.

Finally, the project team could also focus on optimizing the platform's performance and scalability. This could involve implementing new technologies such as serverless computing, microservices, and containerization. The team could also conduct regular load testing to identify and address any performance bottlenecks.

Overall, the future work involved in improving the project is extensive, and the project team should prioritize their efforts based on the objectives. By continuing to innovate and improve the system, the project team can ensure that it remains a valuable tool for managing Business Operations for years to come.

Bibliography

- Odoo | https://www.odoo.com/
- SAP ERP | https://www.sap.com/products/erp.html
- Oracle (2021). Oracle Cloud Infrastructure | https://www.oracle.com/erp/
- Lucid Charts | https://www.lucidchart.com/pages/
- Visual Paradigm | https://online.visual-paradigm.com/
- simplilearn.com what is client server? Feb 14, 2023 | www.simplilearn.com/what-is-client-server-architecture-article
- Kimball, R., & Ross, M. (2013). The data warehouse toolkit: The definitive guide to dimensional modeling (3rd ed.). Wiley.
- O'Leary, D. E. (2000). Enterprise resource planning systems: Systems, life cycle, electronic commerce, and risk. Cambridge University Press.