



Sri Lanka Institute of Information Technology

FACTORY MANAGEMENT SYSTEM

Information Technology Project 2022

Project Report

Project ID: WD_B02_ITP_14

Submitted by:

	Name	IT number
1	Perera M.D.M. R	IT21003028
2	Samarasinghe D. S	IT21003332
3	Bishirhafi F.S.M. T	IT21006166
4	Serasinghe C.M	IT21004322
5	Weerawarna N. V	IT21110948
6	Vidanage D.S. D	IT21128868
7	Dissanayake D.M.J.C. B	IT21151392
8	Lekamge L.R.S. T	IT20159030

Submitted to:

Mr. S. M. B. Harshanath

2022/11/14

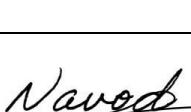
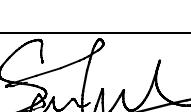
Declaration

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

Project Details

Project Title	Factory Management System (Lanka Mount Castle Pvt Ltd)
Project ID	WD_B02_ITP_14

Group Members

Reg. No	Name	Signature
IT21003028	Perera M.D.M. R	
IT21003332	Samarasinghe D. S	
IT21006166	Bishirhafi F.S.M. T	
IT21004322	Serasinghe C.M	
IT21110948	Weerawarna N. V	
IT21128868	Vidanage D.S. D	
IT21151392	Dissanayake D.M.J.C. B	
IT20159030	Lekamge L.R.S. T	

Abstract

Lanka Mount Castle Ltd is a company specializing in providing customized and non-customized garments embellishment. As a result of our research and interviews we have identified that, currently, the business faces many losses due to inefficiencies and the use of manual, unintegrated software. They have faced issues in retrieving data from other systems and due to this, they cannot track large volumes of data. The motivation behind developing this system is to minimize inefficiencies and satisfy clients' requirements by providing easy accessibility of data and simplifying the workload.

Therefore, we have proposed the development of an integrated and automated web application to solve the problems faced by the company and smooth the functioning of their business operations while increasing the survival rate of the company. Our project mainly focuses on the optimization of the production and maintenance aspects of the factory.

The newly developed system will be an alternative to their pen-and-paper approach to recording data. The operations department has been automated to allow easy communication and workflow management. Previously the sales, production, stock, and supplier management systems were unintegrated and difficult to manage, and communication was also slow, but now with the interconnectivity between these departments, communication can be made faster, and the operation department is made efficient. The new system will enable details of employees and machinery and maintenance to be managed efficiently while minimizing errors. The system will enable the company to meet demands on time and deliveries can be scheduled without delays. Moreover, the finance manager can benefit from the ease of access to data from other departments through this automation and integration, which was previously unavailable due to the traditional bookkeeping methods.

The technology we have used is the MERN stack as it is an open-source tool. We have used the mongo dB as the database which is a NoSQL database that follows a document-oriented DBMS. After analyzing the requirements to maintain large volumes of data we identified that a NoSQL database would be more suitable, as it has high data retrieval speed and is more efficient in handling large volumes of data. ReactJs and tailwind were used as the main development tool to create the user interfaces and design. ExpressJs and Node.js were used to develop the middleware and backend structure of the project and for the server-side development, which allows data binding and cross-platform deployment. "Postman" was used to evaluate the functionality of the system. We have also used other libraries such as syncfusion to develop analytical tools such as bar charts, pie charts, and line graphs. We have used jsPDF library to generate reports and download them in a PDF format. We have also used libraries such as sweetalerts to display popup alerts regarding the status of a transaction and provide feedback for users' inputs.

Overall, the newly integrated system will maximize sales and provide the optimum benefit to the user by tackling all the problems currently faced by the company. The new system will ensure they meet demands on time and tracking of details and communication is made more efficient. The problem of data accessibility due to the segregation of departments is solved by the interconnectivity between these departments. Moreover, the new system will allow them to retrieve data easily through the search function, and view insights and summarized data on the user's dashboard. They will also be able to retrieve reports for all the departments based on the date range and for all the records.

Acknowledgement

At the very outset of this report, we would like to take this opportunity to extend our heartfelt obligation to all the personages who have helped us in making this attempt a success. We would not have completed the project without their active direction, assistance, participation, and encouragement.

First, we would like to express our profound gratitude to Mr. S. M. B. Harshanath, Lecturer, Computer science and software engineering department for giving the full effort in guiding the team in achieving the goal as well as the encouragement to maintain our progress in track by giving us technical guidance and continuous supervision throughout the project.

A special thanks should go to our client, Mrs. Perera, Managing Director of Lanka Mount Castle Pvt. Ltd., for offering us the project to conduct in their factory and for always helping us to deal with the factory staff and the problem domain. Also, our sincere appreciation should go to the whole factory staff for helping us by providing accurate information.

We should thank the Information Technology Project (ITP) module lecturer in charge, the academic staff, and the degree coordinators of the Bachelor of Information Technology (Second Year) Degree program for giving us technical knowledge and guidance throughout the period.

Without the help and direction of parents and peers, no endeavor at any level can be completed. We would like to express our gratitude to our parents for all their assistance in providing us with the necessary data, knowledge, and guidance throughout the creation of this project. Despite their hectic schedules, they provided us with a variety of suggestions for making this project special and unique.

Finally, we are overwhelmed in all humility and gratitude to acknowledge everyone who has helped us transform these concepts into something greater than we could have ever imagined.

Table of Contents

Contents

Declaration.....	i
Abstract.....	ii
Acknowledgement.....	iii
Table of Contents	iv
List of Tables	v
List of Figures.....	v
List of Abbreviations	vi
1. Introduction	1
1.1 Background.....	1
1.2 Problem and Motivation	1
1.3 Literature Review	1
1.4 Aim and Objectives	2
1.4.1 Aim.....	2
1.4.2 Objectives.....	2
1.5 Solution Overview.....	2
1.6 Methodology.....	3
1.6.1 Feasibility Study.....	3
1.7 Structure of the Report	3
1.8 GitHub Repository Link	4
2. Requirements	5
2.1 Stakeholder Analysis	5
2.2 Requirements Analysis	6
2.2.1 Functional Requirements.....	6
2.2.2 Non-Functional Requirements	7
2.3 Requirements Modelling	7
2.3.1 Employee management system	8
2.3.2 Financial Management System	9
2.3.3 Sales Management System.....	10
2.3.4 Production Management System.....	11
2.3.5 Stock Management System	12
2.3.6 Supplier Management System.....	13
2.3.7 Machinery and Maintenance Management System.....	14
2.3.8 Transport Management System.....	15
3. Design and Development.....	16
3.1 Class Diagram	17
3.2 ER Diagram	18
3.3 Work Breakdown.....	18
3.4 Special Code Segments	19
4. Testing	21
4.1 Test Cases	21
4.1.1 Employee Management System.....	21
4.1.2 Finance Management System.....	22
4.1.3 Sales Management System.....	23
4.1.4 Production Management System.....	24
4.1.5 Stock Management System	25
4.1.6 Supplier Management System.....	26
4.1.7 Machinery and Maintenance Management System.....	27
4.1.8 Transport Management System.....	28
5. Evaluations and Conclusions.....	29
6. References	30

7. Appendix	I
7.1 Sequence Diagrams	I
7.2 Activity Diagrams	X
7.3 Screenshots	XVIII

List of Tables

Table 4.1 : Test cases - Employee Management System	21
Table 4.2 : Test cases - Finance Management System.....	22
Table 4.3 : Test cases - Sales Management System.....	23
Table 4.4 : Test cases - Production Management System.....	24
Table 4.5 : Test cases - Stock Management System	25
Table 4.6 : Test cases - Supplier Management System.....	26
Table 4.7 : Test cases - Machinery and Maintenance Management System.....	27
Table 4.8 : Test cases - Transport Management System.....	28

List of Figures

Figure 2.1 : Onion Diagram for Stakeholders.....	5
Figure 2.2 : Use Case Diagram – Employee Management System	8
Figure 2.3 : Use Case Diagram – Finance Management System	9
Figure 2.4 : Use Case Diagram – Sales Management System	10
Figure 2.5 : Use Case Diagram – Production Management System	11
Figure 2.6 : Use Case Diagram - Stock Management System	12
Figure 2.7 : Use Case Diagram - Supplier Management System.....	13
Figure 2.8 : Use Case Diagram – Machinery and Maintenance Management System	14
Figure 2.9 : Use Case Diagram - Transport Management System.....	15
Figure 3.1 : System Overview	16
Figure 3.2 : Class Diagram	17
Figure 3.3 : ER diagram.....	18
Figure 3.4 : Date Range Filter - Backend	19
Figure 3.5 : Date-Range-Filter - Backend.....	19
Figure 3.6 : Search function Implementation.....	20
Figure 3.7 : Search function Implementation.....	20
Figure 3.8 : Currency Formatting	20
Figure 7.1 : Sequence Diagram - Employee Management System.....	I
Figure 7.2 : Sequence Diagram - Finance Management System	II
Figure 7.3 : Sequence Diagram - Sales Management System	III
Figure 7.4 : Sequence Diagram - Production Management System	IV
Figure 7.5 : Sequence Diagram - Stock Management System	V
Figure 7.6 : Sequence Diagram - Supplier Management System	VI
Figure 7.7 : Sequence Diagram - Machinery Management System.....	VII
Figure 7.8 : Sequence Diagram - Maintenance Management System	VIII
Figure 7.9 : Sequence Diagram - Transport Management System	IX
Figure 7.10 : Activity Diagram - Employee Management System.....	X
Figure 7.11 : Activity Diagram - Finance Management System	XI

Figure 7.12 : Activity Diagram - Sales Management System.....	XII
Figure 7.13 : Activity Diagram - Production Management System.....	XIII
Figure 7.14 : Activity Diagram - Stock Management System	XIV
Figure 7.15 : Activity Diagram - Stock Management System	XV
Figure 7.16 : Activity Diagram - Supplier Management System.....	XV
Figure 7.17 : Activity Diagram - Machinery and Maintenance Management System.....	XVI
Figure 7.18 : Activity Diagram - Transport Management System.....	XVII
Figure 7.19 : Employee Dashboard.....	XVIII
Figure 7.20 : User Login	XVIII
Figure 7.21 : Attendance and Leaves Dashboard.....	XIX
Figure 7.22 : Finance Center.....	XIX
Figure 7.23 : Salary Dashboard.....	XX
Figure 7.24 : Orders Dashboard.....	XX
Figure 7.25 : Customer Dashboard	XXI
Figure 7.26 : Production Workflow	XXI
Figure 7.27 : Production Insights.....	XXII
Figure 7.28 : Stock Information.....	XXII
Figure 7.29 : Supplier Dashboard	XXIII
Figure 7.30 : Stock Utilization.....	XXIII
Figure 7.31 : Purchase Order Dashboard.....	XXIV
Figure 7.32 : Machinery Dashboard	XXIV
Figure 7.33 : Maintenance Dashboard	XXV
Figure 7.34 : Transport Dashboard	XXV
Figure 7.35 : Tables with search and filter.....	XXVI
Figure 7.36 : Driver Dashboard	XXVI
Figure 7.37 : Sales Invoices	XXVII
Figure 7.38 : Forms with validations	XXVII
Figure 7.39 : Theming Options.....	XXVIII
Figure 7.40 : Income statements	XXVIII
Figure 7.41 : Form with date validations	XXIX
Figure 7.42 : Alerts interface	XXIX

List of Abbreviations

1. API – Application Programming Interface
2. CRUD – Create, Read, Update, Delete
3. CSS – Cascading Style Sheets
4. ER diagram – Entity Relationship diagram
5. HTML – Hypertext Markup Language
6. Js – JavaScript
7. JSX – JavaScript XML
8. JWT – JSON Web Token
9. MERN – MongoDB, Express JS, React JS, Node JS
10. PDF – Portable Document Format
11. sd – sequence diagram
12. UML – Unified Modeling Language

1. Introduction

1.1 Background

Lanka Mountcastle (Pvt) Ltd. is a company that specializes in providing a wide range of embroidery and garments embellishment services to the Sri Lankan apparel industry. They provide both customized and non-customized garments to their customers. The client has requested a web application, which uses an internet platform that will provide an integrated and automated system with easy accessibility that aids in regulating the business operations with minimum complications.

1.2 Problem and Motivation

The company has encountered multiple issues due to inefficiencies within the factory. The main reason is that the existing system is manual and unintegrated. As a result, each business function works as isolated segments. They have also found that the maintenance of the system is complex, and it has resulted in the wastage of resources like paper because currently, the company keeps track of the details on paper before entering it into the computer. Moreover, they have also identified that their system has led to the inefficient handling of stocks, financial losses, production inefficiencies and difficulty in tracking the tasks. The system must keep track of large volumes of data and without a computerized system, the process seems to be tedious and faulty. They have also found that they are facing difficulties in forecasting future sales and production quantities as it is difficult to obtain and compare data from other departments.

The motivation to develop the system is due to the client wanting to reduce inefficiencies, and disruption of services and solve the major problems threatening the survival of their company. The client believes that with an integrated and automated system, there will be less chance of the occurrence of errors while recording data and it simplifies the workload. Easy accessibility of data can help in administering and monitoring tasks while minimizing losses to improve the financial status of the company.

1.3 Literature Review

Our system to be developed will be a complete system with all the functions required by the business in one application. All business functions will be accessible through one comprehensive system for the customer. The client investigated getting individuals apps to manage their business functions but was weary of having eight different applications with no integration to manage their business. To help them run their business, they sought us out to develop a fully integrated system. It is important to evaluate a project's feasibility before beginning it to decide whether it can be completed with the resources available. Afterward, since it is new to the software developer, it is difficult to understand the problem domain. To create the requirements specification, a literature review and data collection are needed once the framework and flow of the process have been established. As a result of interviews and questionnaires, we have decided that the client's business currently uses pen-and-paper solutions for tracking most functions. Some functions use desktop applications, such as finance management and employee management, but they are not interconnected or integrated. Currently, they are having trouble getting rid of their current system. This is because it is easy to use pen and paper systems that are easy to understand for anyone literate. This also reduces the requirements of the employees to use the system. Because everyone has been using the current system for years, implementing a revamped system will require a learning curve for anyone who isn't already accustomed to it. The newly designed system will be much more efficient in terms of time to use, effective in providing information, and will over time be easier to use. As the upgraded system is

implemented, there will be a backlog of paper documents to deal with. While the brand-new system may have some cons, it will prove to be much more profitable for the client's business over time since it was made specifically to maximize efficiency and profitability. The client will not be able to grow and expand their business in the future if the current system is maintained. This is because their pen-and-paper system prevents them from scaling the business. However, with our comprehensive system, it is merely a matter of adjusting a few variables to allow for a much larger business to be managed with the software. We started the project by researching several websites related to our project which really aid us to gain knowledge of the system. According to the research, Through the website of Sri Lanka Mountcastle PVT Limited, a business that provides the Sri Lankan apparel industry with the broadest selection of embroidery and garment embellishing services, we were able to get a clear idea about the system including its functional and non-functional requirements regarding our system.^[1] The TEAFMS (Tea Factory management system) project, which was a BIT final year project in 2011, explores the object-oriented approach to developing the software. The purpose of this project was also to address the factory management task of a tea factory, which served as a baseline for developing our project. TEAFMS is predominantly focused on the financial aspects of the factory, while our project, based on an embroidery manufacturing factory, focuses on the production and maintenance aspects of the factory. ^[2]

1.4 Aim and Objectives

1.4.1 Aim

- The aim of this project is to develop a web-based factory management system to increase the manageability and productivity of the client's factory.

1.4.2 Objectives

- Finalizing the product requirements before the start of the development process.
- Developing the system architecture for the factory management system.
- Developing the user interfaces for the system within a period of 1 month.
- Developing the individual business functions planned for the prototype version within a 1-month period.
- Launching a working prototype of the factory management system within a 2-month period.
- Developing the rest of the business functions planned for the last version within the next 2 months.
- Assessing the User experience of the factory management system using the client's feedback after the development of the system.
- Conducting system testing for the integrated system before the launch of the final product.
- Launching a fully functioning, integrated factory management by the end of a 4-month period.
- Presenting the final project report by the end of the 4-month period.

1.5 Solution Overview

A web application is proposed to be developed as a solution to the above problems. A web application will provide a reliable factory management system with the currently available resources without investing in additional resources. The web application has 8 different business functions to handle the business process of the factory. They are as follows.

- Employee management system
- Financial Management System
- Sales Management System
- Production Management System

- Stock Management System
- Supplier Management System
- Machinery and Maintenance Management System
- Transport Management System

1.6 Methodology

To identify the client's requirements, we scheduled interviews with our client. We use questionnaires for management-level employees and others who are going to use the new system to get ideas about how they would prefer the new system. We identify data that needs to be collected to generate reports. To implement the new integrated system, the current manual database system should be replaced by an automated database approach.

1.6.1 Feasibility Study

Before taking on the contract with Lanka Mountcastle, we had decided to conduct a feasibility study through third parties and relevant authorities. The project was evaluated in the following categories of feasibility.

- **Technical Feasibility** – We will be doing more Requirements Engineering to understand if the factory has the required technologies and resources for the system to be implemented, thus satisfying the technical feasibility conditions.
- **Operational Feasibility** – We will be consulting with a third-party Quality Surveying (QS) Organization that will ascertain if the project plan satisfies the client's requirements, for the client to proceed with the plan. The QS team will confirm if all the required resources are available to make the system operative.
- **Economic Feasibility** – We will be consulting with the client's Financial Team, to identify the economic and financial ability of the factory to ensure that the factory has the required capability to conduct the new system, including all the hardware and software costs. The new system will be beneficial in terms of the financial status of the factory.
- **Legal Feasibility** – We will be consulting with the client's Legal Team to identify any legal obstacles that we may have to face. Thus, ensuring that the project is legally doable and that there will be no unexpected future repercussions to the Development Team.
- **Schedule Feasibility** – Given the 14-week schedule within which the project needs to be completed, we have produced a Gantt chart which shows that the probability of completion is nearly 100% within the given time frame.

1.7 Structure of the Report

Our report has been divided into different sections, and each section and heading specify the topic discussed. We have also subdivided the sections to provide a clear understanding. The first chapter introduces our project. It describes our client, the problem and motivation behind developing this web application, the literature review explains the research process, the aims and objectives describe what our main motive is, the solution overview describes the overall web application we have developed and how it would help our client solve the current issues faced by the company, and the methodology describes the tools and methods used for data collection and requirements gathering.

The requirements section gives an outline of the stakeholders who would be directly and indirectly involved in using the web application. It further includes details about the requirements analysis that is the functional and non-functional requirements of the system. The requirements modelling describes all the UML diagrams and other diagrams that aided us in developing this application.

The design section includes the system architecture, class diagram and ER diagram. It also gives an outline of the tools and technology used to implement this system. The testing section describes the test cases conducted for each of the 8 management systems and along with that we have included an evaluation and conclusion. And for the concluding section we have included references, and other UML diagrams not included in the requirements section and screenshots of the user interfaces.

1.8 GitHub Repository Link

GitHub Repo Link: https://github.com/SLIITITP/y2_s2_wd_it_02-wd_b02_itp_14

2. Requirements

2.1 Stakeholder Analysis

Upon analyzing the requirements for this system, we were able to identify several different key stakeholders for the proposed system. They are as follows.

- Company owner
- Company employees
- Investors
- Customers
- Suppliers
- Government

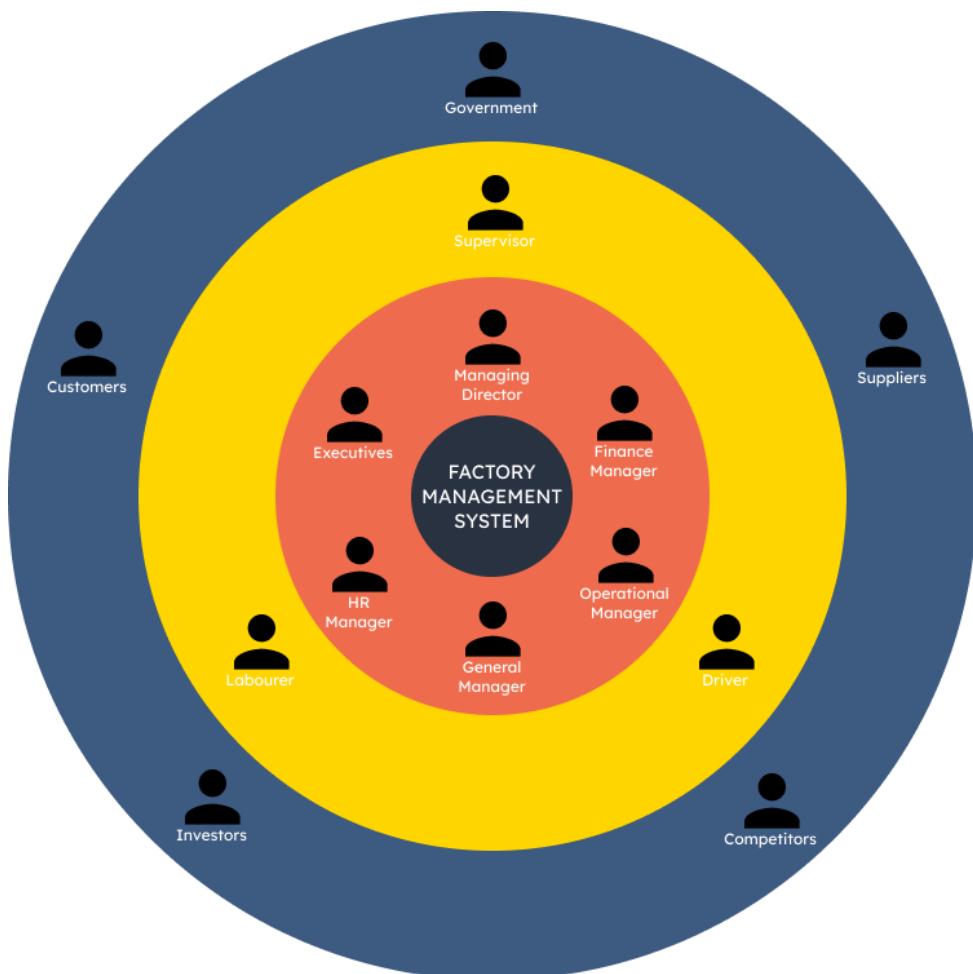


Figure 2.1 : Onion Diagram for Stakeholders

2.2 Requirements Analysis

Upon interviewing the stakeholders, and analyzing the current system, we were able to identify following functional and non-functional requirements

2.2.1 Functional Requirements

- Ability to login to the system with user credentials
- Ability to manage employee related details and be able to perform CRUD operations related to the employee details.
- Ability to record attendance of the employees.
- Ability to manage employee leaves and perform CRUD operations related to it.
- Ability to validate the user entered data properly.
- Ability to generate meaningful reports based on employee attendance and performance.
- Ability to manage daily cash transactions and all relevant CRUDs
- Ability to reconcile the finances through the finance function
- Ability to view financial statements such as the income statement, which accumulates information across the entire platform into a single view
- Having the capability of filtering the income statements date wise to view only relevant information
- Ability to generate these reports as required as PDFs by the user
- Ability to create, View and Manage sales orders
- Ability to generate and view sales reports.
- Ability to Download Invoice for billing purposes.
- Ability to register customers for invoices.
- Ability to filter information for a given time.
- Ability to retrieve pending stocks from inventory
- Ability to view, update and delete stock requisitions
- Ability to view, update and delete costed orders
- Ability to view orders according to the status
- Ability to track estimated and actual costs and view as summarized data as charts, tables and widgets
- Ability to generate costed reports
- Ability to communicating with the operational departments as an automated system with minimal inputs
- Ability to calculate the variances by comparing estimated and budgeted costs and calculation of labor costs, material costs, and total costs.
- Distribution of work to production employees
- Ability to insert details of new stock.
- Ability to make a stock purchase request.
- Ability to issue stocks.
- Ability to view, edit and delete stock records.
- Ability to view details of stocks requested, stocks received, additions and issues of stock, stock in hand and stock breakdown.
- Ability to view stocks by categories.
- Ability to track stock movements and view summarized data as charts, tables, and widgets
- Ability to generate reports.
- Ability to communicate with the production and supplier through an automated system with minimal inputs.
- Ability to filter data through date range and search function.
- Ability to provide meaningful alert messages about the status.
- Ability to calculate quantity needed to meet the reorder level and update damaged quantity.
- Ability to store and manage supplier details and perform CRUD operations of the supplier records
- Ability to select the best supplier for a specific order by search
- Ability to store and manage purchase order details and perform CRUD operations of the purchase order records

- Ability to filter the purchase orders placed according to a specific date range
- Generate reports based on supplier details and purchase order details
- Allows the Administrative to manage the details on machines on repairs and replacements, ownerships, renewal dates and schedules, depreciation of machines needed, and new purchases.
- Ability to find any detail related to the machines that are used in the factory in one place.
- Updating the system whenever a new machinery-related purchase happens.
- Ability to maintain a net book value which is calculated as the original costs of the machines when purchasing minus any accumulated depreciation of machines.
- Ability to generate Reports with the updated details and according to the dates.
- Ability to remove unnecessary records in the database when machines are unavailable, damaged, or no longer in need.
- Allow to feature a fault index with typical breakdowns and keep track of all maintenance.
- Ability in scheduling the examination of the machines and other resources based on data from the machinery management.
- Allows tracking past maintenance on the company resources.
- Allows scheduling preventative maintenance based on the date the equipment, machines, and vehicles were purchased, how long it has been in use, etc.
- Ability to update and remove all the details getting entered.
- Allows the maintenance and safety tasks to be assigned to the appropriate personnel.
- Ability to generate reports for the productive time of the company resources, fully compiled equipment, maintenance history, failures, maintenance expenses, depreciation charges, etc.
- Ability to manage transportation details and perform CRUD operations related to the transportation records.
- Ability to search for any transport records in the system.
- Ability to filter transportation records in the table with a given date range.
- Ability to manage driver details and perform CRUD operations on it.
- Ability to generate and download reports including every transportation detail and driver details.

2.2.2 Non-Functional Requirements

- Fast access speeds
- High information security
- High reliability
- Real time availability
- Easy accessibility
- User friendly Interface
- Information consistency
- Information accuracy
- Data integrity

2.3 Requirements Modelling

We have utilized several different UML diagrams to model the functional requirements of the proposed system.

2.3.1 Employee management system

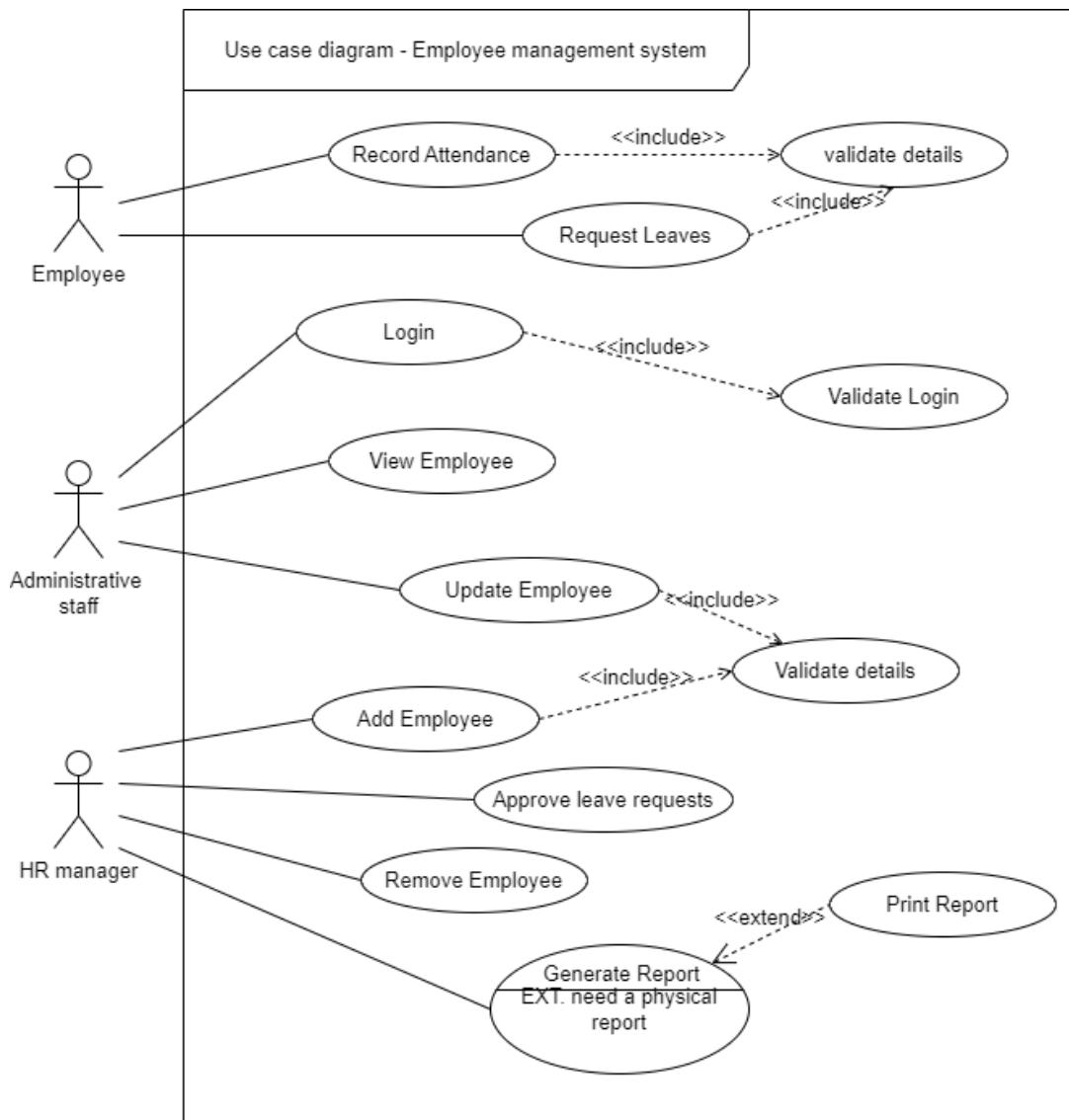


Figure 2.2 : Use Case Diagram – Employee Management System

2.3.2 Financial Management System

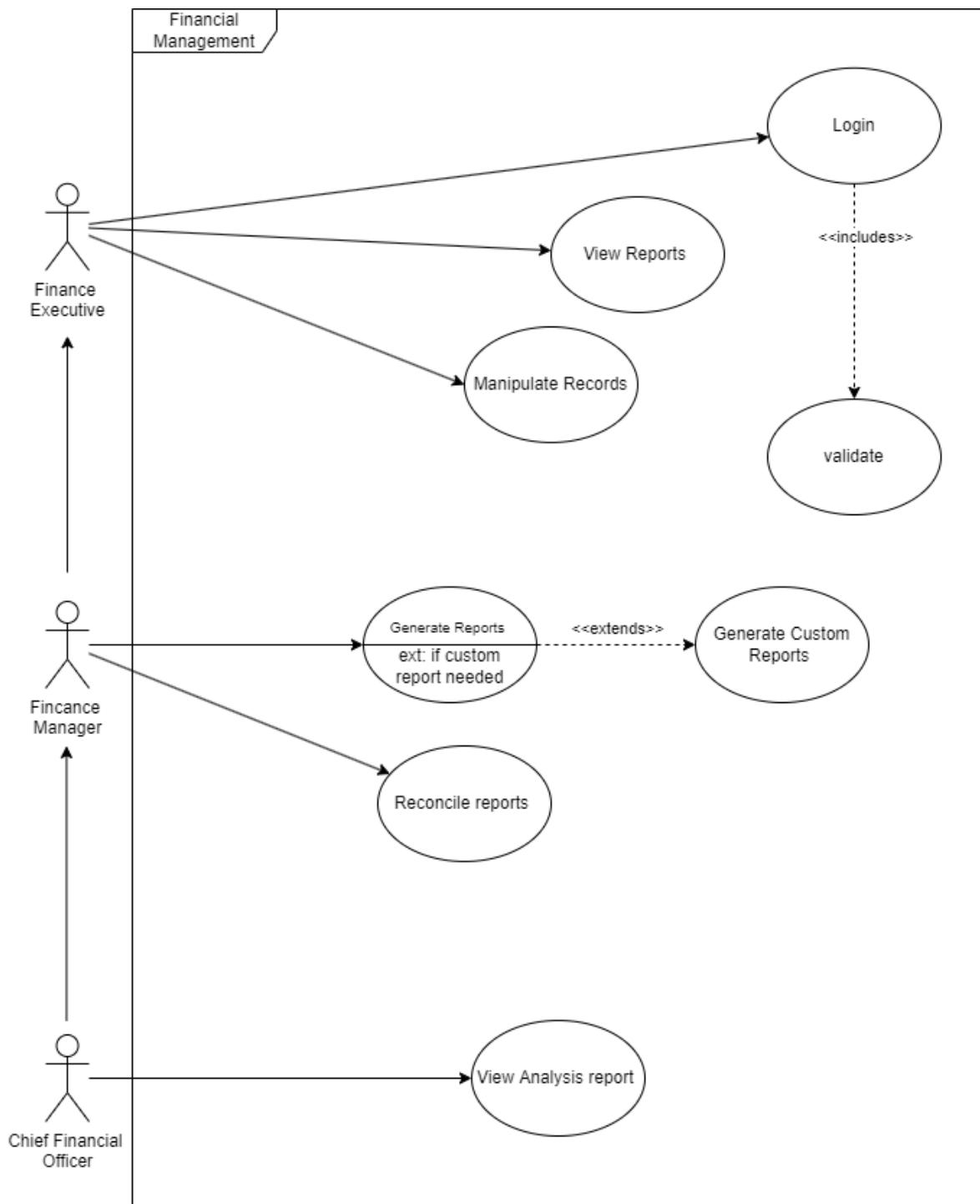


Figure 2.3 : Use Case Diagram – Finance Management System

2.3.3 Sales Management System

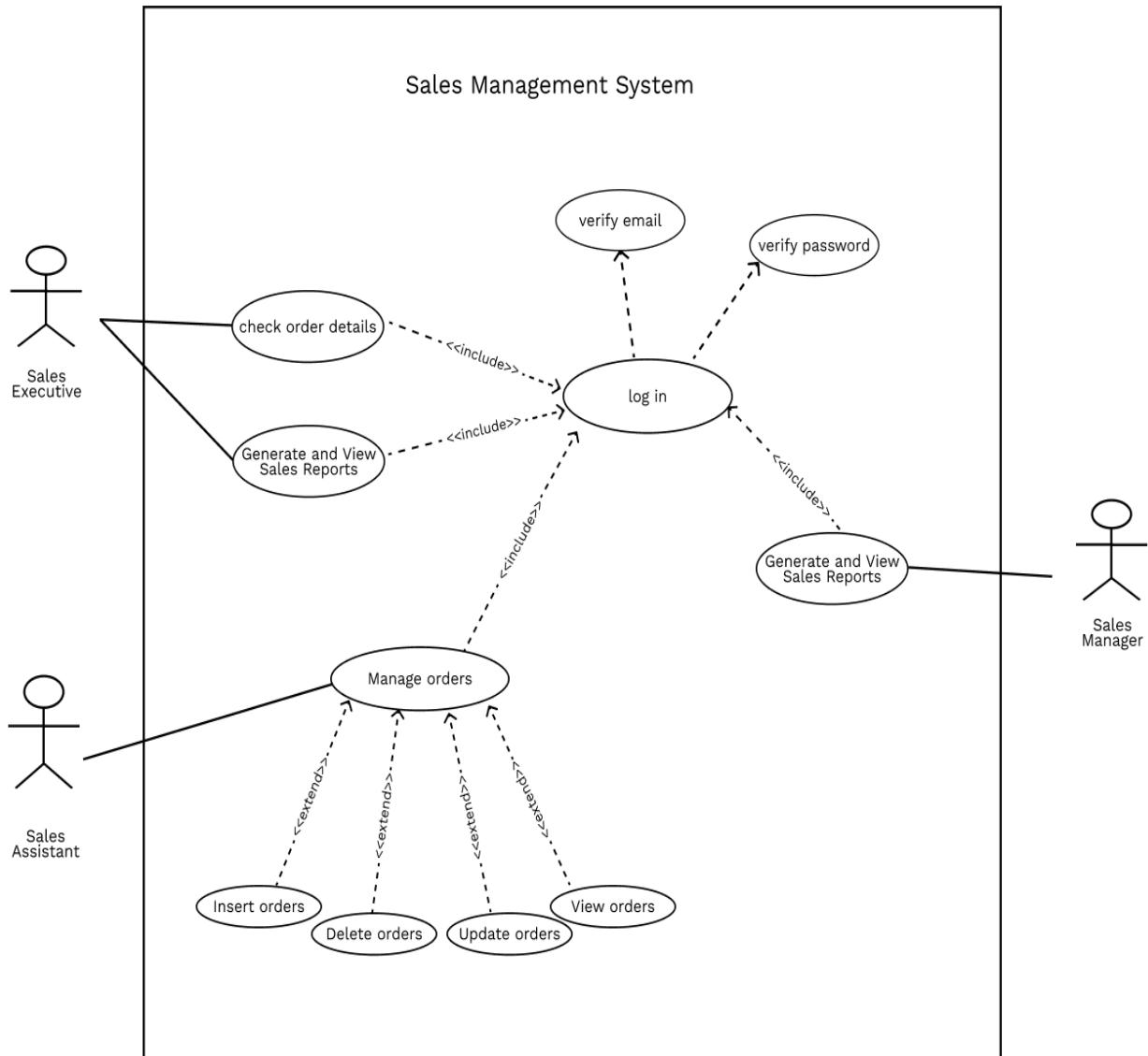


Figure 2.4 : Use Case Diagram – Sales Management System

2.3.4 Production Management System

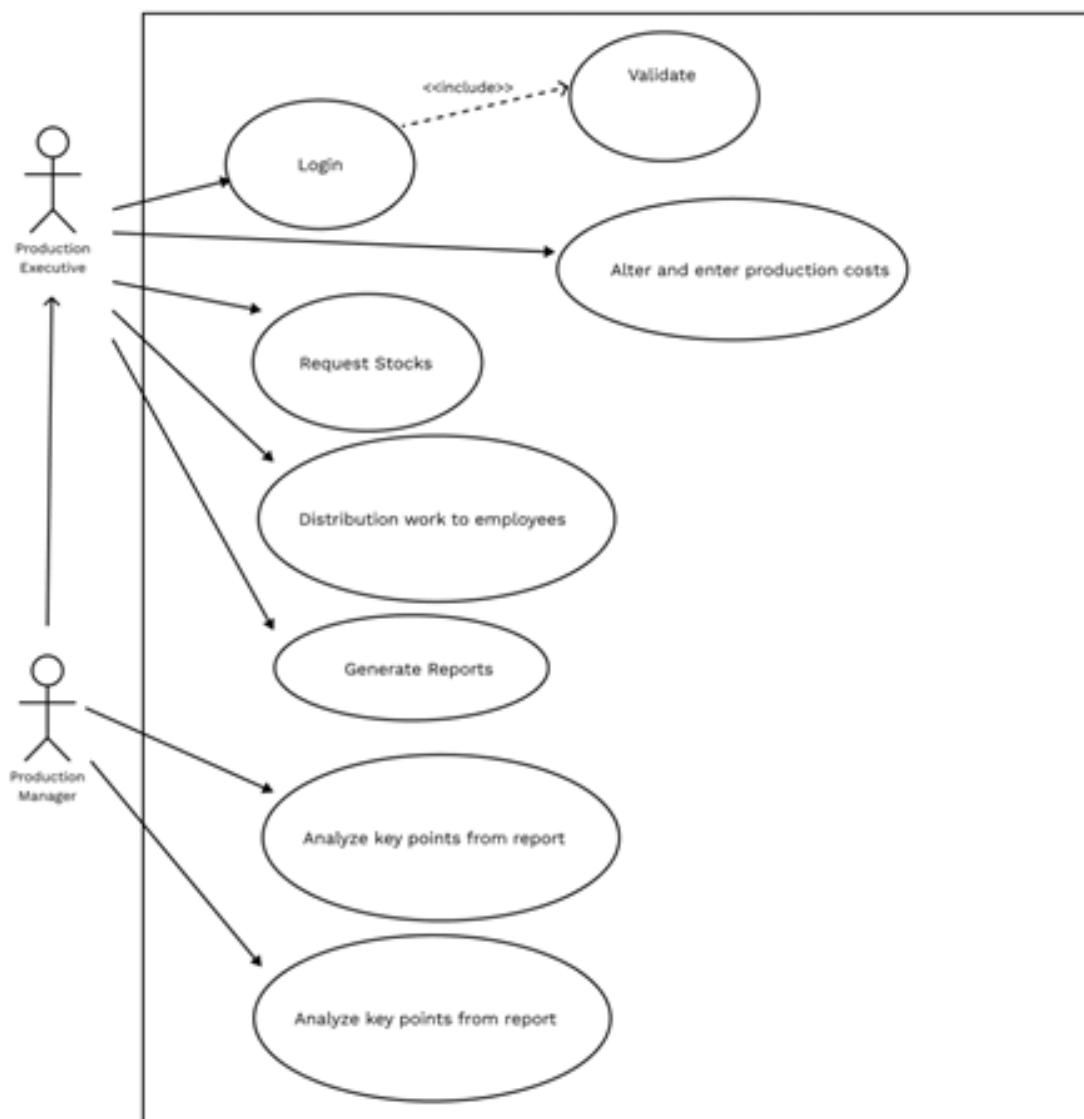


Figure 2.5 : Use Case Diagram – Production Management System

2.3.5 Stock Management System

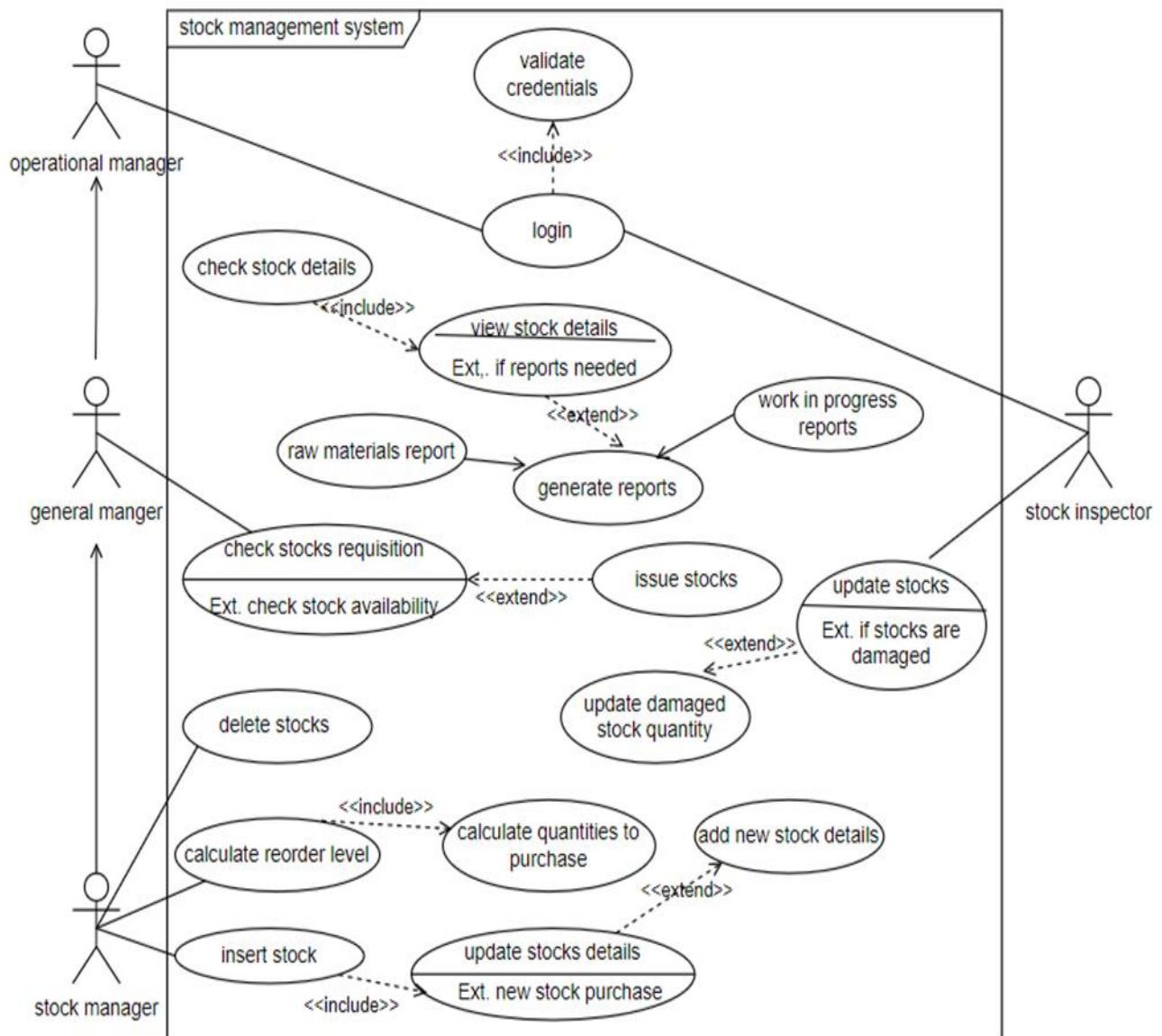


Figure 2.6 : Use Case Diagram - Stock Management System

2.3.6 Supplier Management System

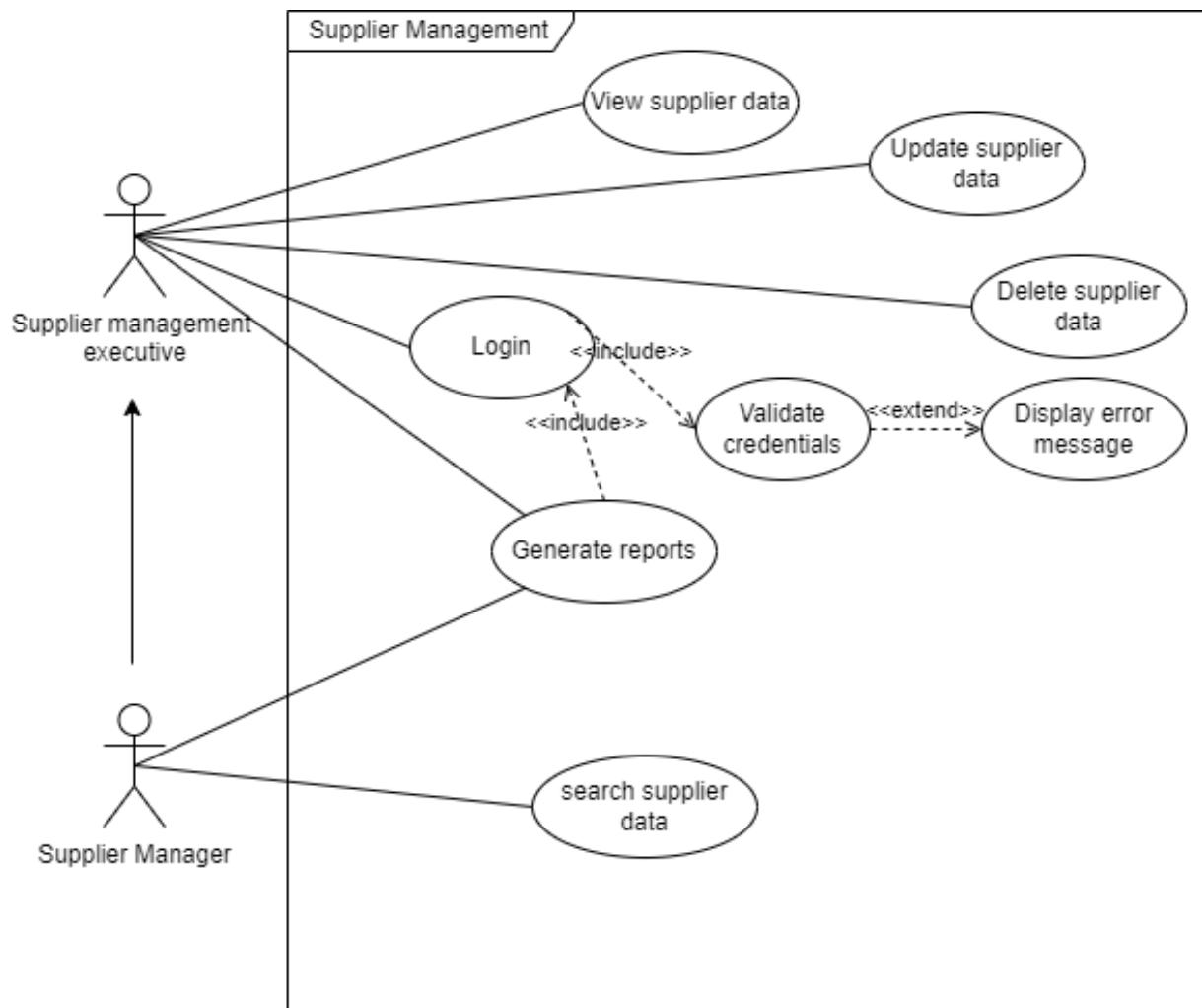


Figure 2.7 : Use Case Diagram - Supplier Management System

2.3.7 Machinery and Maintenance Management System

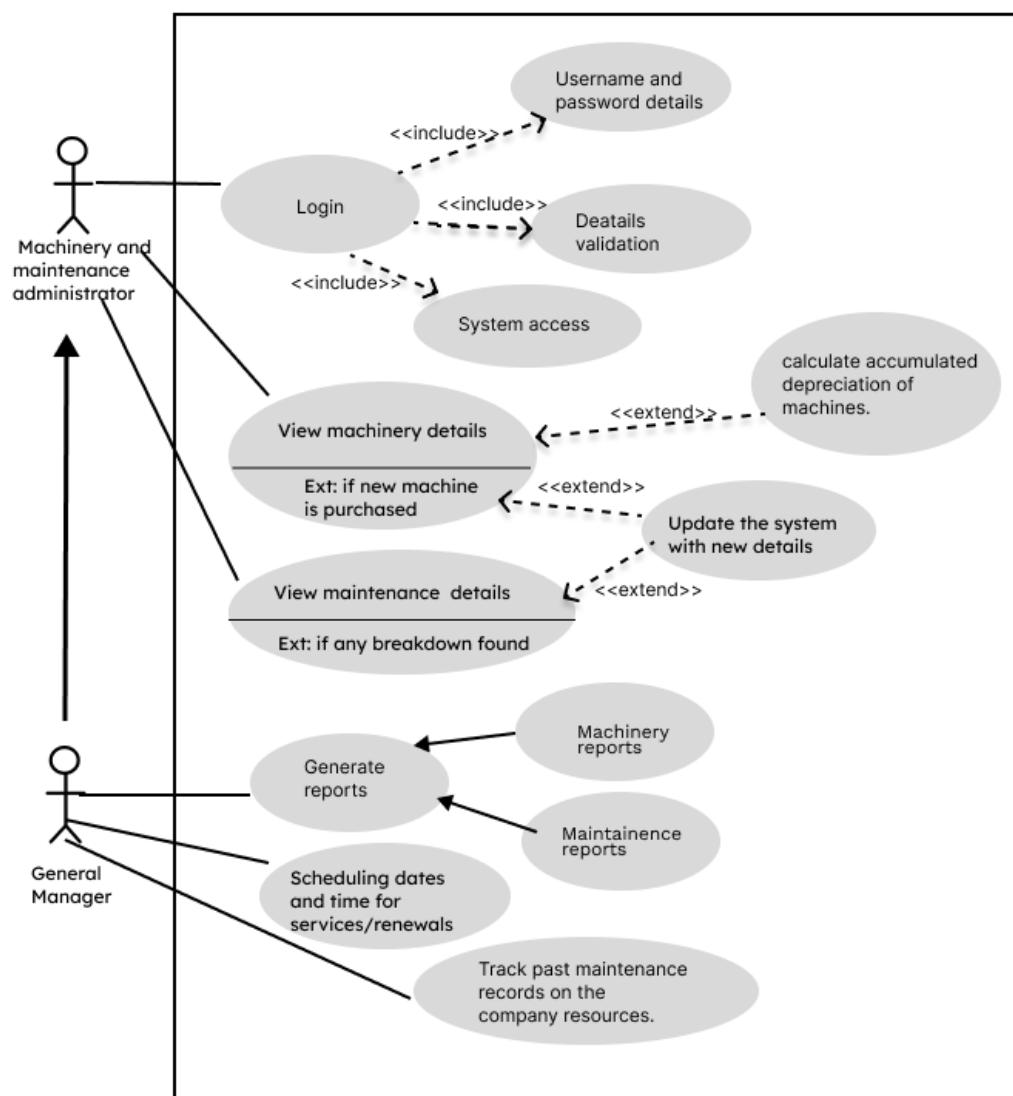


Figure 2.8 : Use Case Diagram – Machinery and Maintenance Management System

2.3.8 Transport Management System

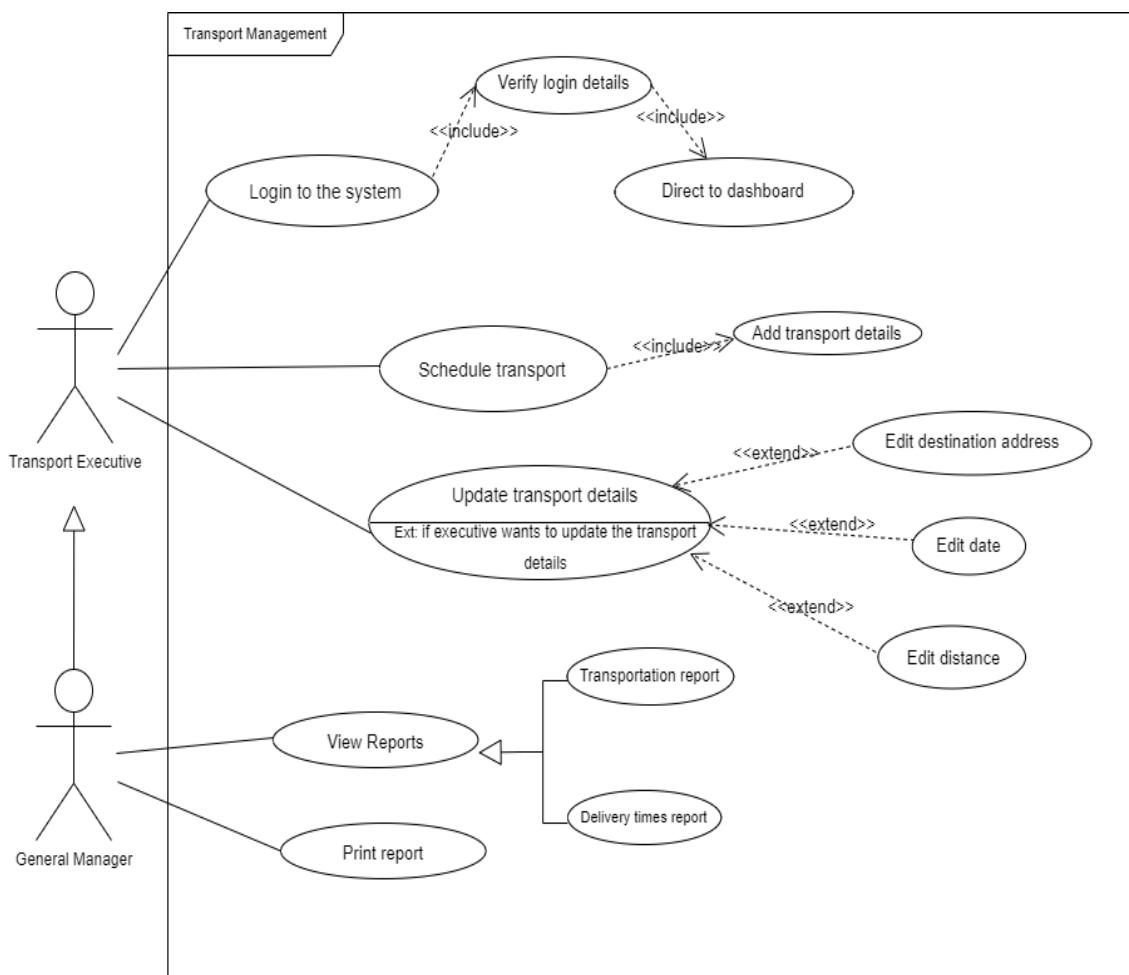


Figure 2.9 : Use Case Diagram - Transport Management System

3. Design and Development

In the designing phase of our project, we have made the use of several designing tools to draw UML diagrams, Wireframes, and create prototypes. Draw.io was used to build UML diagrams. When designing our wireframes, we used Mock Flow as our wireframing tool. After wireframing, we used Figma for prototyping. Agile development was used as the main method for developing our web application.

Our factory management system is 2 tier web-based application which provides the necessary functionality to the client.

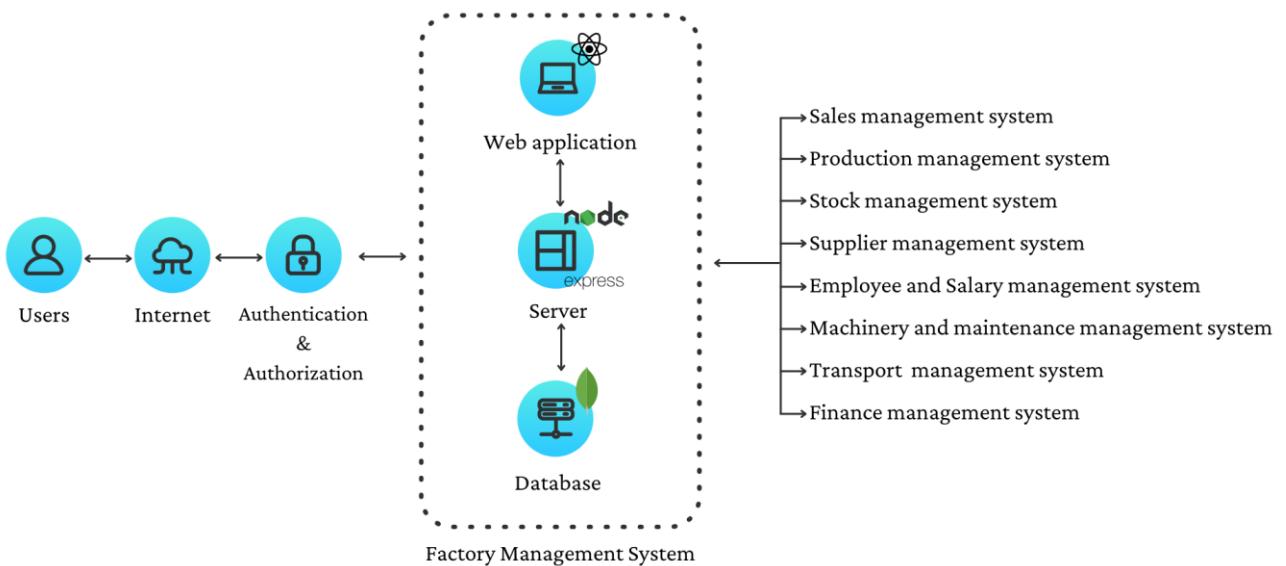


Figure 3.1 : System Overview

In our project, we utilize MERN stack technologies to implement the factory management system. The front end was developed using React Js and the middleware was implemented using Express Js. The backend database was implemented using MongoDB. In addition, we used React bootstrap, HTML, JSX, and yarn packages. We used JSX for our project because it enables us to write HTML in React. Alternatively, we considered using AngularJS, Java Spring Boot, and MySQL for our project, but in the end, we chose the MERN stack. due to its high performance, easy integration with 3rd party technologies, ability to use JavaScript throughout the frontend and backend, and its high availability of developers to provide future development and maintenance to the system. Themes were an essential part of our application to maintain the consistency and the cleanliness of the final product. We added designs to our app with colors, styles, icons, charts and more. Tailwind CSS theme is one of the trending CSS frameworks used in developing modern web applications to develop an app with less CSS use of predefined CSS helper classes. “jsPDF” library was used to generate reports and download them in a PDF format. We have also used libraries such as “sweetalerts” to display popup alerts regarding the status of the functionalities and we have used “bcryptjs” library to encrypt the user login password. This module permits the hashing of passwords rather than storing them in plaintext. Also, we have used JWT to share security information between a client and a server. We used the visual studio code as our primary integrated development environment. The API was tested

using the “Postman” tool and it easily sends an API request to the web server and receives the response, whatever it is. For workflow visualization, we took the benefits of using the Kanban board to manage the workload in a successful manner. GitHub was used for the version control to track and manage different versions of our project code, which helps us collaborate with all the group members and to manage the main repository and integrate our project.

3.1 Class Diagram

Link to the Full Class Diagram:

https://drive.google.com/file/d/1hyCi_G0p4AUD5hrwQV9MygS1jnwlEbX/view

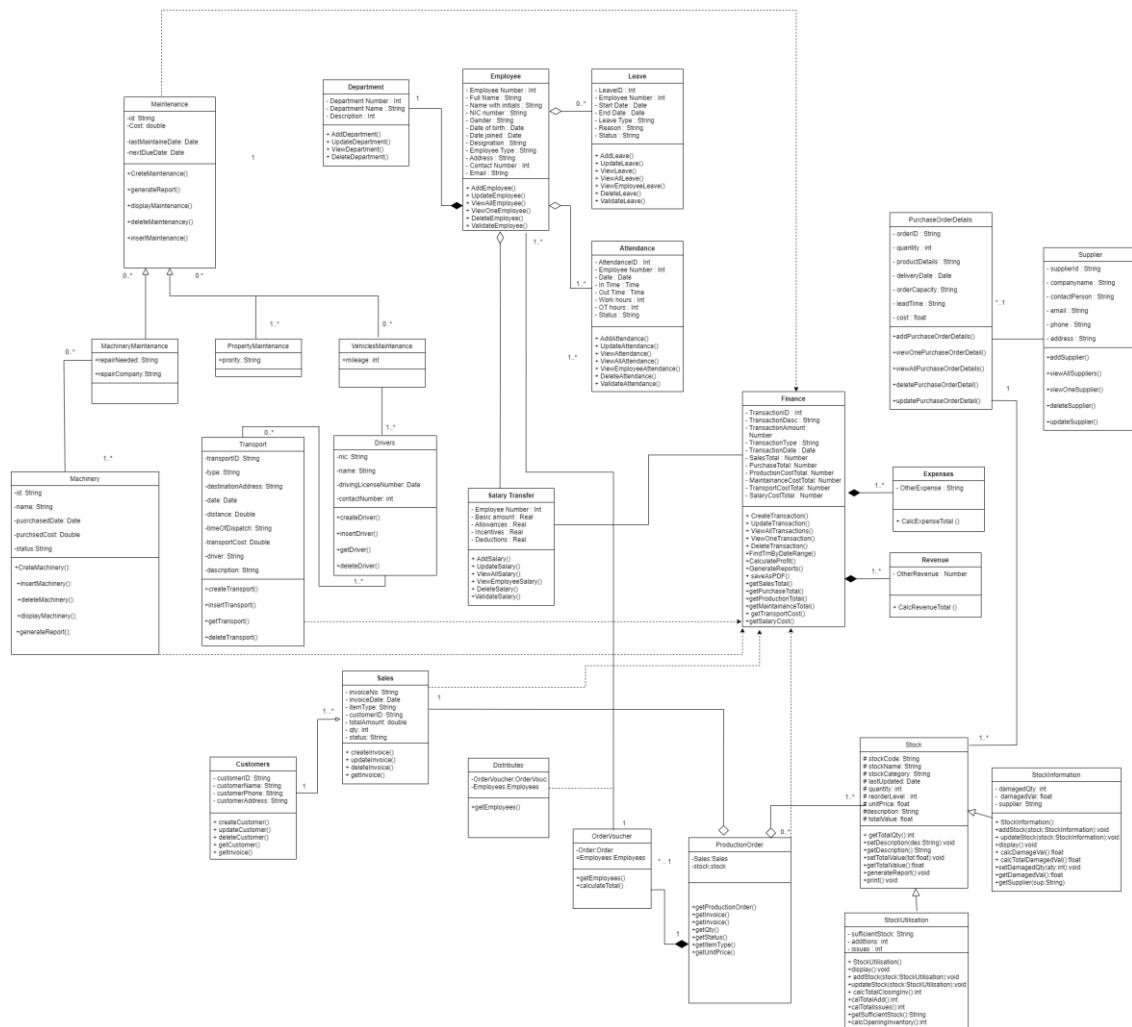


Figure 3.2 : Class Diagram

3.2 ER Diagram

Link to the Full ER Diagram:

<https://drive.google.com/file/d/16c1pW0-sYc861H447hlJjaWkVzCPgI4r/view>

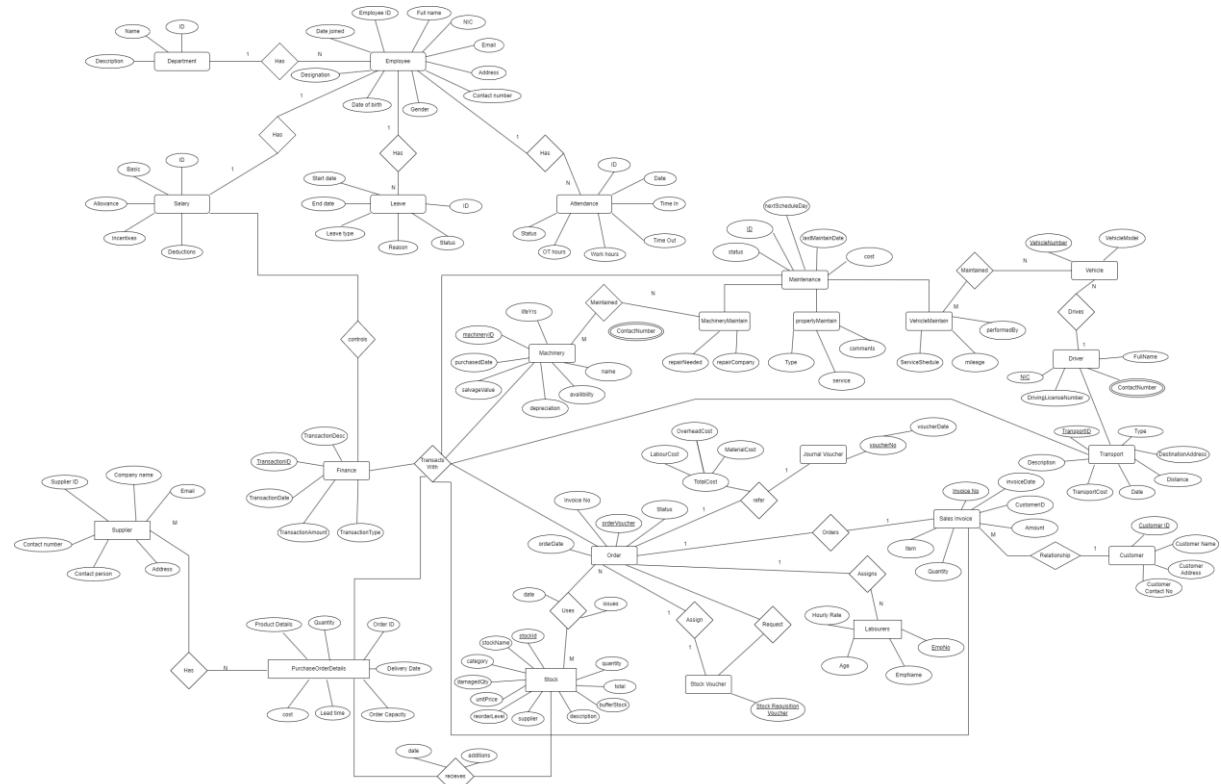


Figure 3.3 : ER diagram

Further designing diagrams ([Sequence diagrams](#), [Activity Diagrams](#) and [System Screenshots](#)) will be attached to the appendix.

3.3 Work Breakdown

Reg. No	Name	Developed section
IT21003028	Perera M. D. M. R	Finance management system
IT21003332	Samarasinghe D. S	Production management system
IT21006166	Bishirhafi F. S. M. T	Stock management system
IT21004322	Serasinghe C. M	Machinery and maintenance management system
IT21110948	Weerawarna N. V	Transport management system
IT21128868	Vidanage D. S. D	Supplier management system
IT21151392	Dissanayake D. M. J. C. B	Sales management system
IT20159030	Lekamge L.R.S. T	Employee management system

3.4 Special Code Segments

```
export const getDateRangeFinance = async (req, res) => {
  try {
    const DS = req.params.DS;
    const DE = req.params.DE;
    const finacedata = await FD.aggregate([
      {
        $match: { trnRecordedDate: { $gte: new Date(DS), $lte: new Date(DE) } }
      }
    ]);
    res.status(200).json(finacedata);
  } catch (error) {
    res.status(404).json({message : error});
  }
}
```

Figure 3.4 : Date Range Filter - Backend

This function is used to pass a date range to the database and retrieve only the relevant results as well as information within that given date range, this implements features of MongoDB as well as NodeJS, and ExpressJs.

```
const filterDate = () => {
  if (dateRangeRef.value && dateRangeRef.value.length > 0) {
    const start = (dateRangeRef.value[0]);
    const end = (dateRangeRef.value[1]);

    setDateStart(start);
    setDateEnd(end);
    navigate('/FinanceDateRange',{state:{DS:start,DE:end}});
  } else {
    dateConf();
    setDateStart('');
    setDateEnd('');
  }
}
```

Figure 3.5 : Date-Range-Filter - Backend

This function is used to supplement the prior function wherein this will be used in ReactJS to get user input on date values.

<div>
 <input type="text" className="block w-400 rounded-md
 bg-gray-100 focus:bg-white dark:text-black"
 placeholder="Search Here"
 onChange={(e) => {
 setSearchTerm(e.target.value);
 }} />
 </div>

Figure 3.7 : Search function Implementation

Figure 3.6 : Search function Implementation

These functions are used for the search function wherein it will take input from the user on the frontend of the page and pass it along to the filter function where it will return results to the user based on the given inputs.

```
const formatter = new Intl.NumberFormat('en-US', {  
  style: 'currency',  
  currency: 'LKR',  
  minimumFractionDigits: 2,  
  currencyDisplay: 'symbol',  
});
```

Figure 3.8 : Currency Formatting

This function was used to display any and all monetary values in the required format as per the report it was held in.

4. Testing

We started testing the system from the beginning with unit testing. The API calls were tested using “Postman.” We evaluated both the backend and frontend validations using different test cases. Then each of the unit functions were then integrated and tested for errors. The sub systems related to production workflow were tested thoroughly during the integration testing. Finally, once everything was developed, we moved on to test the complete system.

4.1 Test Cases

4.1.1 Employee Management System

Table 4.1 : Test cases - Employee Management System

Testing Function 1: Updating Employee Records	
Test case ID: 0001	Test designed by: Reg. No - IT2015930 Name - Lekamge L R S T
Test priority (High/Medium/Low):	Medium
Test Description: When updating employee details, it should only save the updated employee details if it passes all the validations	

Preconditions (if there are any): The employee should already exist in the system
Dependencies (if there are any): None

Test steps:
Log into site -> Employee Dashboard -> Click All Employees-> Select the relevant employee -> Click Edit button -> Enter Details -> Click Submit button

Pass-condition:
Entered details are validated and updated details are added to the system and displayed in the employee profile.

Test ID	Test Inputs	Expected Output	Actual Output	Result	Comments
1.1	Enter an invalid email address	An error saying invalid email format	An error saying invalid email format	Pass	

1.2	Enter a valid email address	Alert saying Data successfully gets added	Alert saying Data successfully gets added	Pass	
-----	-----------------------------	---	---	------	--

4.1.2 Finance Management System

Table 4.2 : Test cases - Finance Management System

Testing Function 2: Entering Date Range for Incomes Statement					
Test case ID: 0002	Test designed by: Reg. No - IT21003028 Name - M. D. M. R. Perera				
Test priority (High/Medium/Low):	Medium				
Test Description: When entering a date range, an income statement should be generated with the information of all sectors of the business relevant only between those given date ranges					
Preconditions (if there are any): other functions must have information to populate the table					
Dependencies (if there are any): jsPDF (to generate the report as a PDF)					
Test steps:					
Enter a date range ->filter the view					
Pass-condition:					
If only the relevant information is used to populate the table using the given date range					
Test ID	Test Inputs	Expected Output	Actual Output	Result	Comments
2.1	Date range where other functions have values	Table to be populated with information	Table is populated with relevant information	Pass	
2.2	No date range is entered	Error given asking user to enter a relevant date range	Error is shown asking user to enter a relevant date range	Pass	

4.1.3 Sales Management System

Table 4.3 : Test cases - Sales Management System

Testing Function 3: Create New Sales Invoice	
Test case ID: 0003	Test designed by: Reg. No - IT21151392 Name - Dissanayake D.M.J.C. B
Test priority (High/Medium/Low):	High
Test Description: New Sales Invoice should have a unique Invoice Number	

Preconditions (if there are any): User should be logged into the system with valid login.

Dependencies (if there are any): None

Test steps:

Log into site > Sales Dashboard > Click New Invoice > Enter Invoice Details > Click Save

Pass-condition:

Entered invoice details are validated and a new invoice is created and added to the system and can be seen on the invoice table

Test ID	Test Inputs	Expected Output	Actual Output	Result	Comments
3.1	Invoice No – INV00001 Date – 10/11/2022 Customer ID – 00001 Item Name – Shirts Quantity – 100 Total Amount – 100000 Order status - Pending	Add invoice to the system and alert a success message. And the new invoice can be seen in the Invoice table	Invoice was added to the system and the user was alerted. New invoices can be seen on the invoice page.	Pass	
3.2	Invoice No – INV00001 (Duplicate Invoice Number)	Check for duplicate invoice number and alert user with a message and navigate back to the page.	Identified as a duplicate number, user was alerted and navigated back to the page.	Pass	

4.1.4 Production Management System

Table 4.4 : Test cases - Production Management System

Testing Function 4: Passing a Stock Requisition from Production Department										
Test case ID: 0004	Test designed by: Reg. No - IT21003332 Name - Devindu Samarasinghe									
Test priority (High/Medium/Low):	High									
Test Description: When creating a stock requisition form from the given pending invoices, the order status should be changed and passed down to stock requested table and to be shown as in production in sales table in sales management system.										
Preconditions (if there are any): Invoice must be in the pending list table first.										
Dependencies (if there are any): None										
Test steps: View production workflow dashboard => press pending stocks widget => Retrieve pending stocks => Click one pending stock order => Update the form => Click submit and receive the message.										
Pass-condition: <ul style="list-style-type: none"> The status of the invoice in sales management should change Once the status has changed in the stock requisition order to pending it should be shown in the stock requisition table Stock requisition order should automatically pass down to stock management system 										
Test ID	Test Inputs	Expected Output	Actual Output	Result	Comments					
4.1	Entering a negative overhead cost	Display an error alert saying the value mentioned is less than zero	Display an error alert saying the estimated overhead value cannot be less than zero	Pass						
4.2	Creating a stock requisition without a date	Alerting the user to pass a date within the current date and the date the invoice has been created.	Alerting the user to pass a date within the current date and the date the invoice has been created.	Pass						

4.1.5 Stock Management System

Table 4.5 : Test cases - Stock Management System

Testing Function 5: Issuing Stocks									
Test case ID: 0005		Test designed by: Reg. No - IT21006166 Name - Bishirhafi. F. S. M. T							
Test priority (High/Medium/Low):		High							
Test Description: When issuing stocks, the stocks should be issued only if the required quantity of stock is available.									
Preconditions (if there are any): Stock needs to be requested by production									
Dependencies (if there are any): None									
Test steps:									
View Stock Utilization dashboard -> view stocks requisitions -> click issue stock -> view the details -> click submit									
Pass-condition:									
<ul style="list-style-type: none"> The production status needs to change. The quantity available needs to decrease after the issue of stocks. A new record is added to the stock utilization table. 									
Test ID	Test Inputs	Expected Output	Actual Output	Result	Comments				
5.1	The input for the quantity of stock requested for production is lower than the available stock.	Display issued stock alert, change the production status, and reduce the stocks quantity available and display the record in the stock's utilization table.	Display issued stock alert, change the production status, and reduce the stocks quantity available and display the record in the stock's utilization table	Pass					
5.2	The input for the quantity of stock requested for production is greater than the available stock	Alert the user about the quantity of stocks required for production and navigate to stock purchase request form.	Alert the user about the quantity of stocks required for production and navigate to stock purchase request form.	Pass					

4.1.6 Supplier Management System

Table 4.6 : Test cases - Supplier Management System

Testing Function 6: Adding a new supplier record					
Test case ID: 0006	Test designed by: Reg. No - IT21128868 Name -Vidanage D.S.D				
Test priority (High/Medium/Low):	Medium				
Test Description: When entering supplier details, for the details to get successfully added to the system it should be having a unique supplier ID					
Preconditions (if there are any): Administrator should be logged into the system with valid credentials					
Dependencies (if there are any): None					
Test steps: Login to the site -> Supplier Details Dashboard -> Click create new supplier -> Enter the supplier details -> Click submit					
Pass-condition:					
Entered details are validated and a new supplier record is created and added to the system					
Test ID	Test Inputs	Expected Output	Actual Output	Result	Comments
6.1	Enter an invalid supplier id	Display an error message saying “This Supplier ID already exists”	Display an error message saying “This Supplier ID already exists”	Pass	
6.2	Enter a valid supplier id	Display “Supplier details added successfully” message	Display “Supplier details added successfully” message	Pass	

4.1.7 Machinery and Maintenance Management System

Table 4.7 : Test cases - Machinery and Maintenance Management System

Testing Function 7: Adding Machinery Maintenance	
Test case ID: 0007	Test designed by: Reg. No - IT21004322 Name - Serasinghe C M
Test priority (High/Medium/Low):	Medium
Test Description: When adding a machinery maintenance, it should be successfully added only if the machine is available in the machinery details	

Preconditions (if there are any): A maintenance record needs to be added
Dependencies (if there are any): None

Test steps:
Log into site -> Maintenance Dashboard -> Click new machinery maintenance -> Enter Details -> Click Save

Pass-condition:
Entered details are validated and a new maintenance is created and added to the system

Test ID	Test Inputs	Expected Output	Actual Output	Result	Comments
7.1	Enter an invalid machinery id number which is not recorded under machinery details	An error saying machine is not in use	An error saying machine is not in use	Pass	
7.2	Enter a valid machinery id number	Data successfully gets added	Data successfully gets added	Pass	

4.1.8 Transport Management System

Table 4.8 : Test cases - Transport Management System

Testing Function 8: Adding a new transport record	
Test case ID: 0008	Test designed by: Reg. No - IT21110948 Name - Weerawarna N. V
Test priority (High/Medium/Low):	Medium
Test Description: When entering transport details, it should be in valid format and added successfully to the system	
Preconditions (if there are any): Administrator should be logged into the system with valid credentials	
Dependencies (if there are any): None	
Test steps:	
Login in as administrator > Click the transport management dashboard > Click New Transport > Enter the transport details > Click add transport button > View successful message	
Pass-condition:	
Entered details are validated and a new transportation record is created and added to the system	

Test ID	Test Inputs	Expected Output	Actual Output	Result	Comments
8.1	Enter an invalid transportation id	Display a validation error message	Display a validation error message	Pass	
8.2	Enter all transport details in correct format	Display transport details successfully added message	Display transport details successfully added message	Pass	

5. Evaluations and Conclusions

Evaluation is where the system is tested. Different methods can be used to test systems. When it comes to system quality, evaluation is crucial since the system must complete the required work without any errors. This is a two-step method consisting of validation and verification.^[3] Verification is the technique used to determine whether the system satisfies the client's needs. The validation process verifies that each system function works as the client as planned. To drastically reduce errors, testing is done at various phases of the system. Software testing is most of the time integrated with system development in today's context to ease debugging and reduce errors. Throughout the development stage, every page of our system was tested. This may be regarded as unit testing, and tests were conducted using both white box and black box techniques. In the white box technique, every code segment is inspected to ensure that the logic is correct and in the black box testing, the page or module is utilized to conduct the necessary job and then the database is queried to determine whether the task was completed successfully.^[4] The system's integration, functionality, and other aspects were also evaluated. Regression testing, interface testing, defect testing, and system testing were conducted to address bugs and errors in the system integration.

This project was initiated to create a fully functional web application for Lanka MountCastle (Pvt) Ltd. Before commencing the project, a feasibility analysis was conducted after discussions with the client to identify all the needs. The system's primary functions were split among the group's eight members, and user-friendly interfaces were created by considering the client's requirements. The system was created as a web-based application with React Js functioning as the frontend and Node Js serving as the backend framework. Since the client intended to use the application online, web-based applications were chosen over standalone systems. Implementation of this system had a massive impact on their prevailing system. This system allows keeping track of every task conducted for each job separately, which was the biggest achievement of the system. Also, this system can easily accessibility of data and can help in administering and monitoring tasks while minimizing losses to improve the financial status of the company. According to the company, the system is used by administrators and corporate representatives. The administrators can manage tasks like employees, finance, sales, production, stock, supplier, machinery, and maintenance, and transport management with a user-friendly interface. As a result, this web application will transform the company's workflow and have a positive impact on its future growth.

6. References

- [1] Z. M. Nawhal, "Web Based Factory Management System for Deenside Tea Factory," <https://dl.ucsc.cmb.ac.lk/jspui/handle/123456789/1268>, 2013.
- [2] W3School, "React_jsx," [Online]. Available: https://www.w3schools.com/react/react_jsx.asp.
- [3] softwaretestinghelp, "Verification vs Validation," [Online]. Available: <https://www.softwaretestinghelp.com/what-is-verification-and-validation>.
- [4] IBM, "Software Testing," [Online]. Available: <https://www.ibm.com/topics/software-testing/>.
- [5] MondoDB, "MongoDB documentation," MondoDB, [Online]. Available: <https://www.mongodb.com/docs/>.
- [6] N. JS, "Node JS Documentation," Node JS, [Online]. Available: <https://nodejs.org/en/docs/>.
- [7] E. JS, "Express JS Documentation," Express JS, [Online]. Available: <https://expressjs.com/>.
- [8] React, "React Documentaion," React, [Online]. Available: <https://reactjs.org/docs/getting-started.html>.
- [9] Syncfusion, "Syncfusion React UI Components," Syncfusion , [Online]. Available: <https://ej2.syncfusion.com/home/react.html>.
- [10] Tailwind CSS, "Tailwind Documentation," Tailwind Labs Inc, [Online]. Available: <https://tailwindcss.com/docs/installation>.
- [11] MrRio, "jsPDF documentation," jsPDF, [Online]. Available: <https://raw.githack.com/MrRio/jsPDF/master/docs/index.html>.

7. Appendix

7.1 Sequence Diagrams

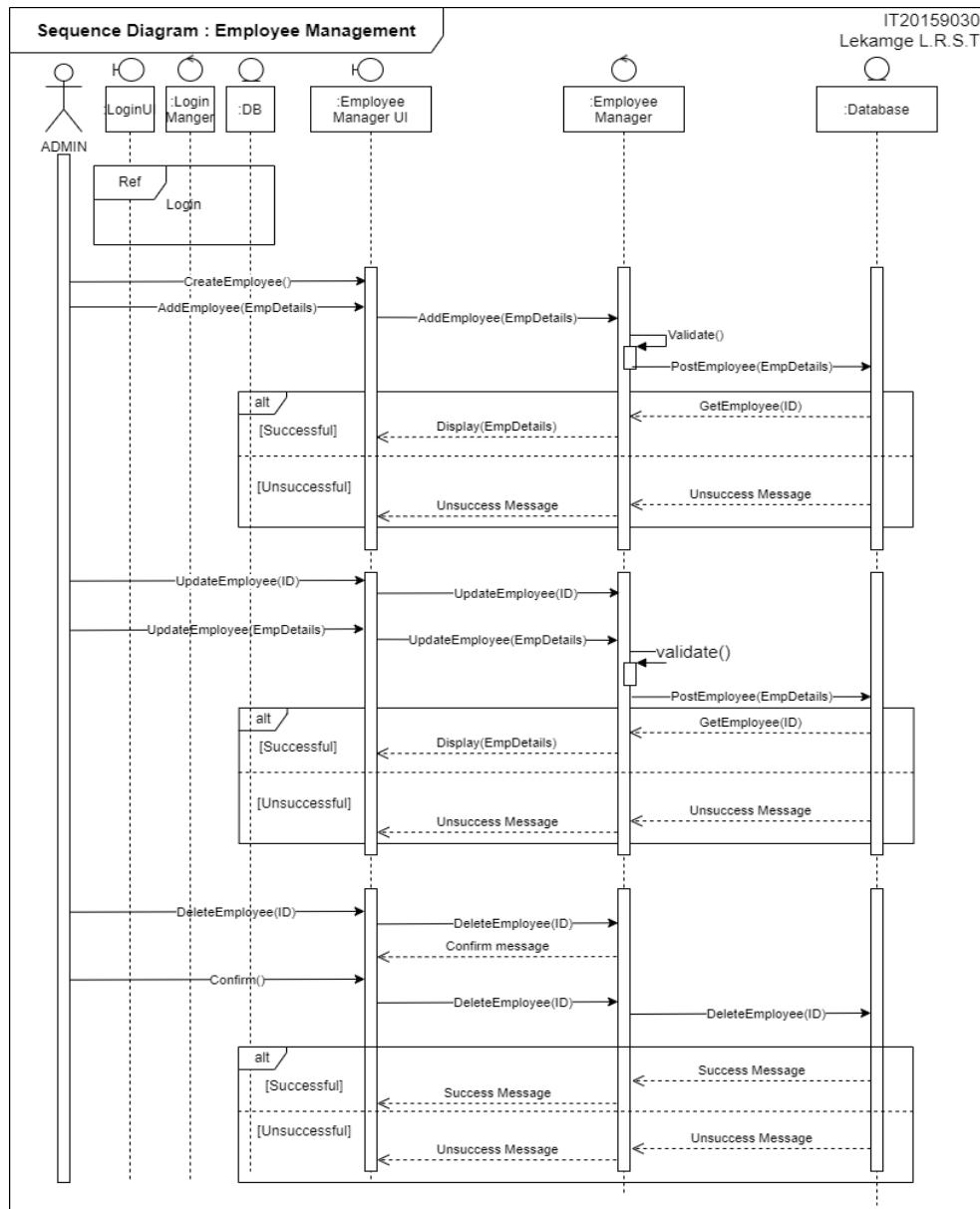


Figure 7.1 : Sequence Diagram - Employee Management System

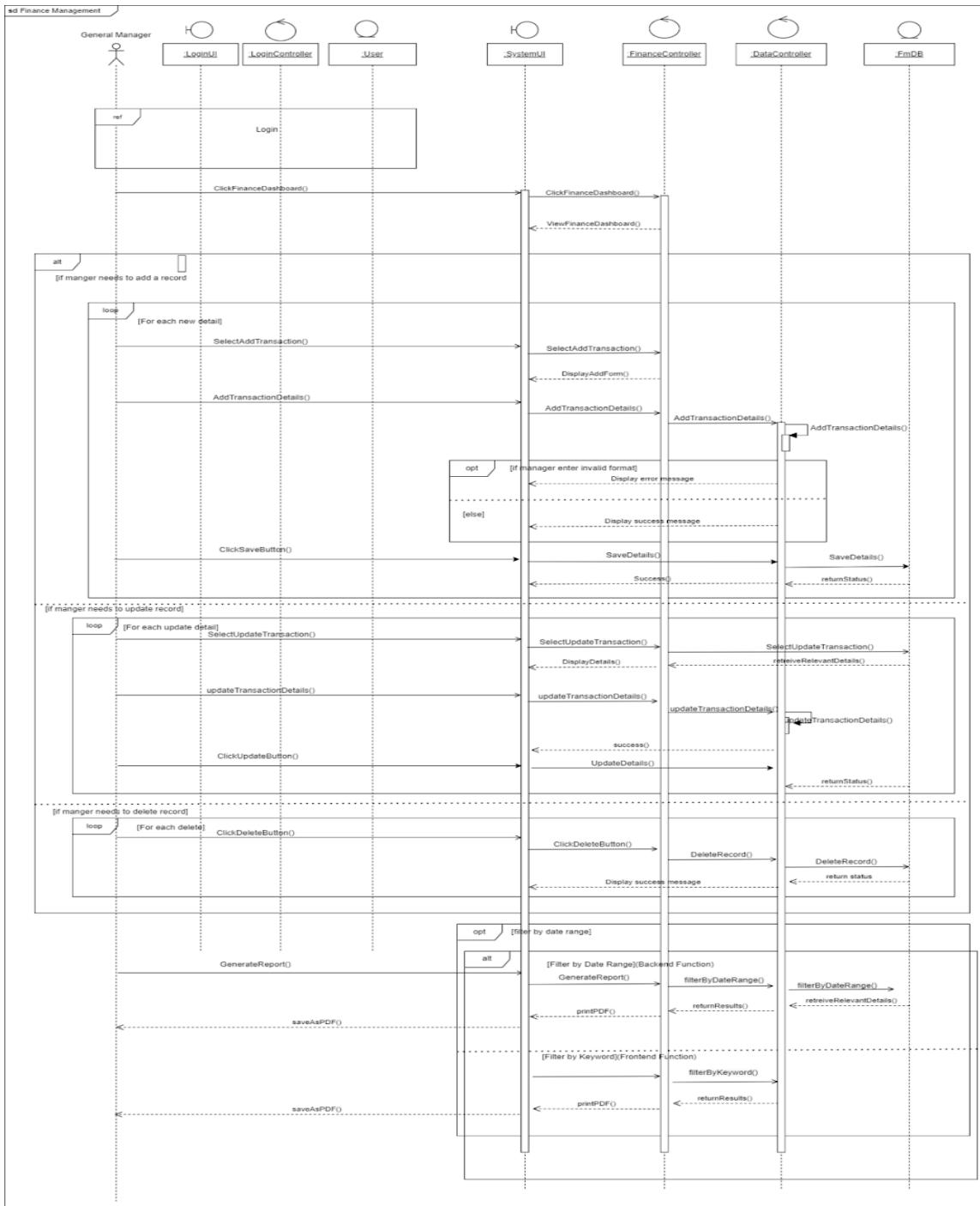


Figure 7.2 : Sequence Diagram - Finance Management System

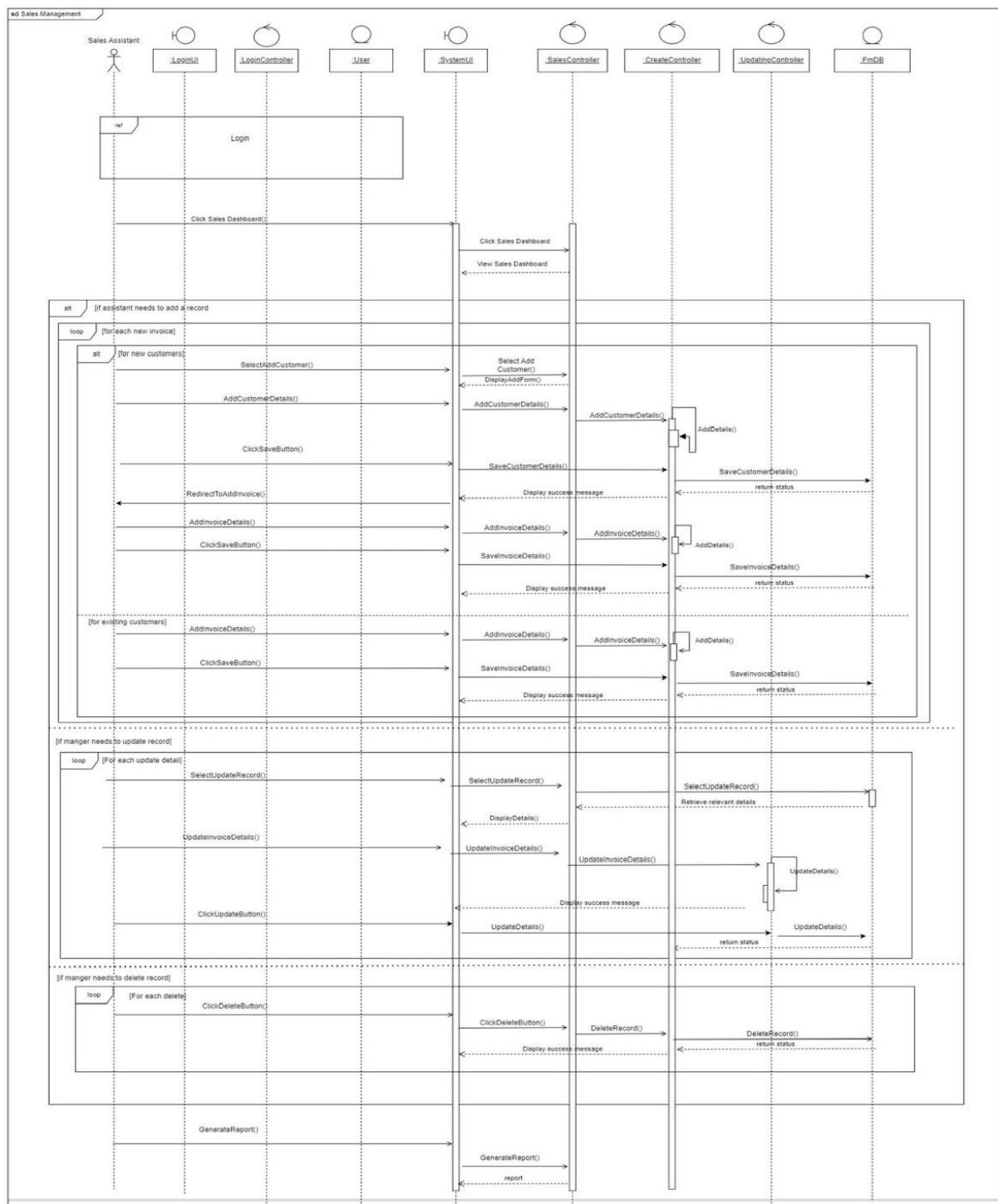


Figure 7.3 : Sequence Diagram - Sales Management System

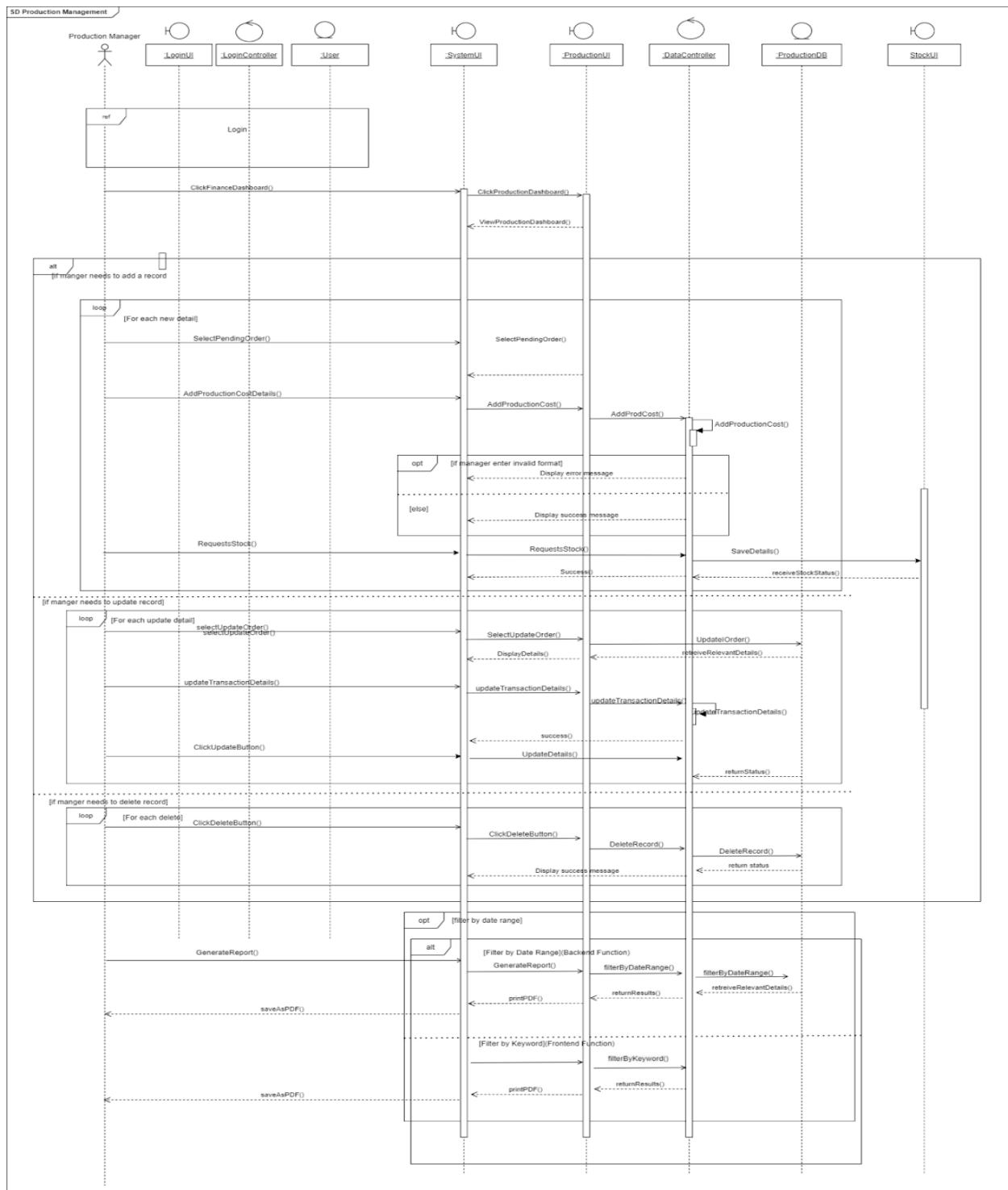


Figure 7.4 : Sequence Diagram - Production Management System

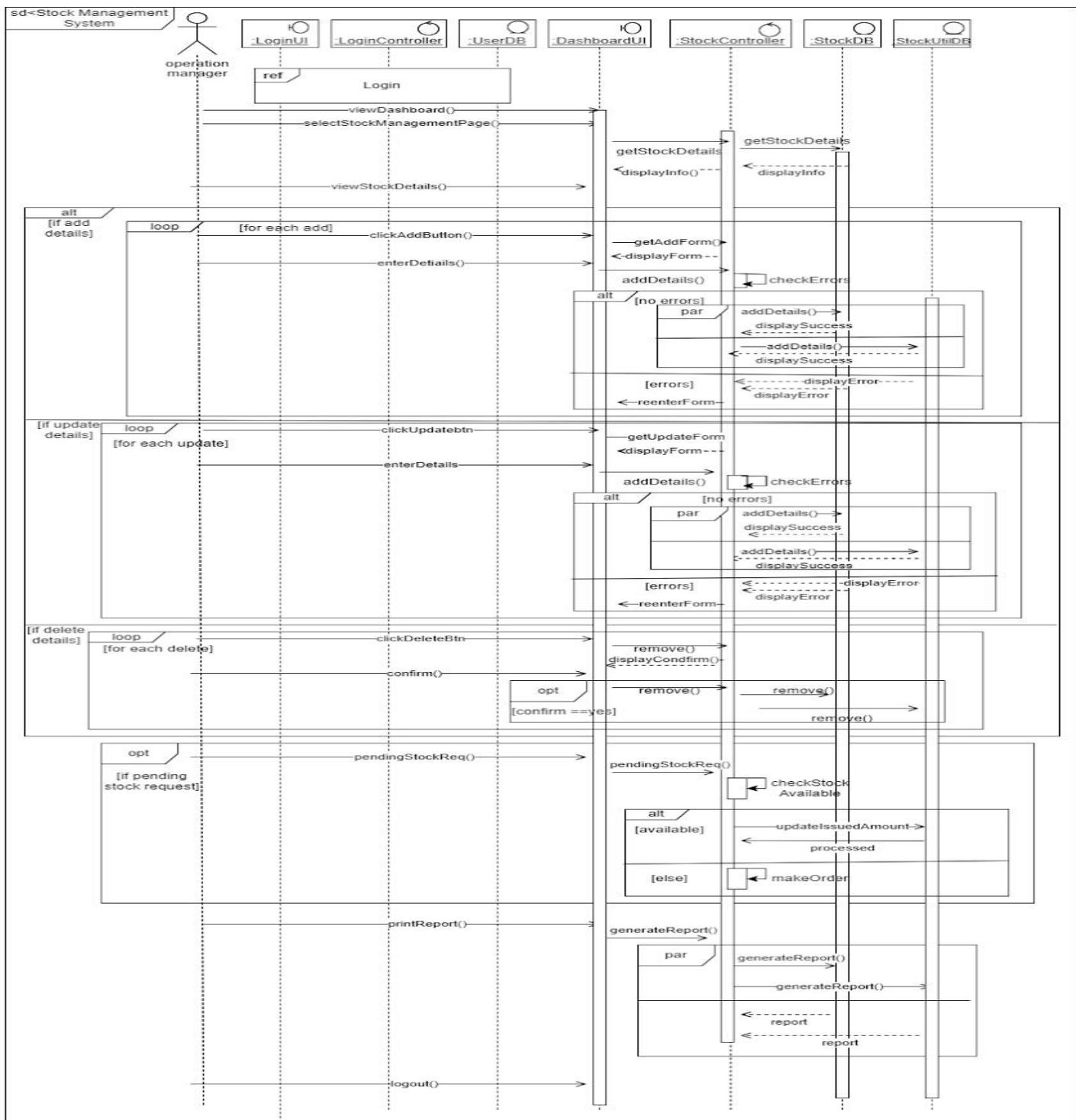


Figure 7.5 : Sequence Diagram - Stock Management System

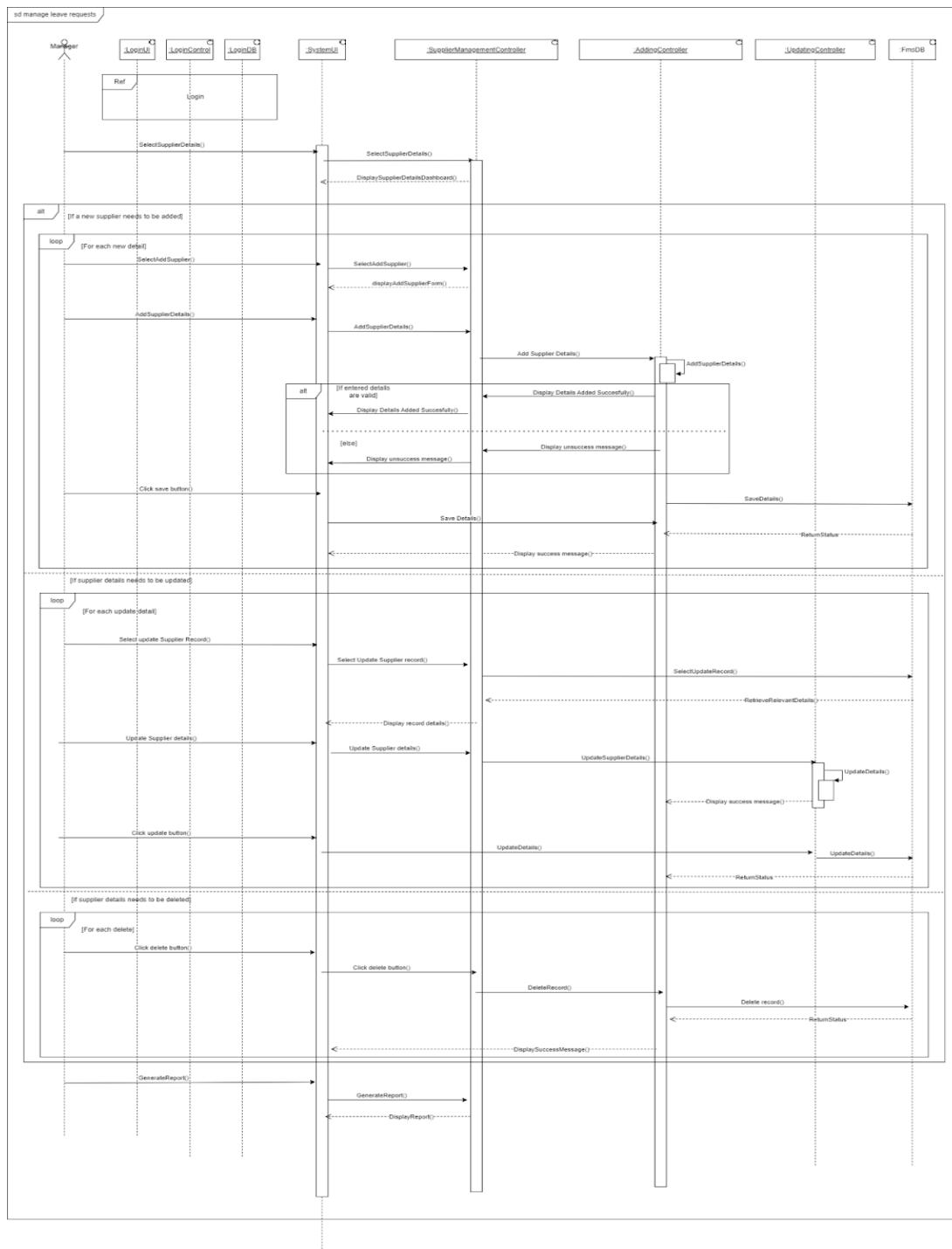


Figure 7.6 : Sequence Diagram - Supplier Management System

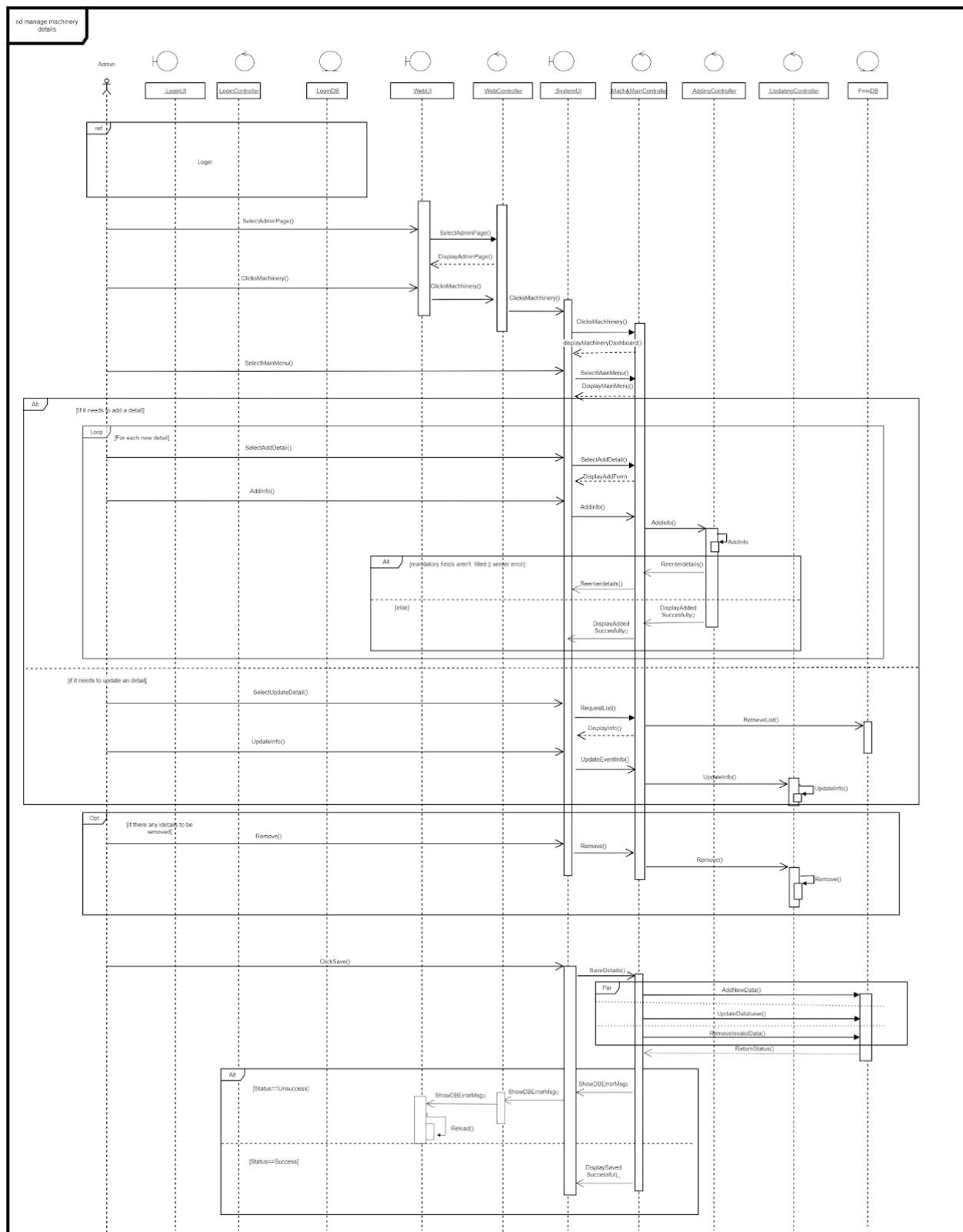


Figure 7.7 : Sequence Diagram - Machinery Management System

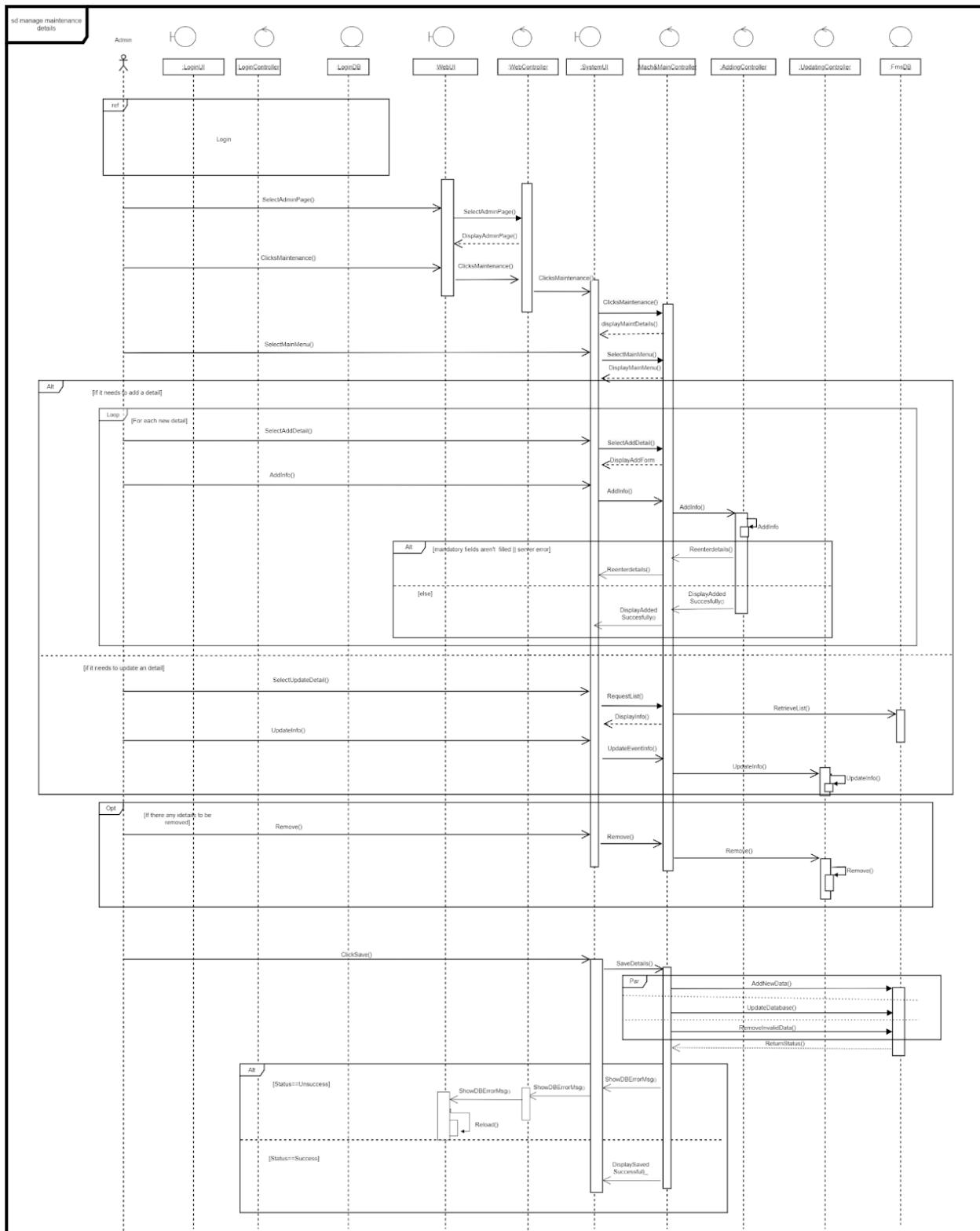


Figure 7.8 : Sequence Diagram - Maintenance Management System

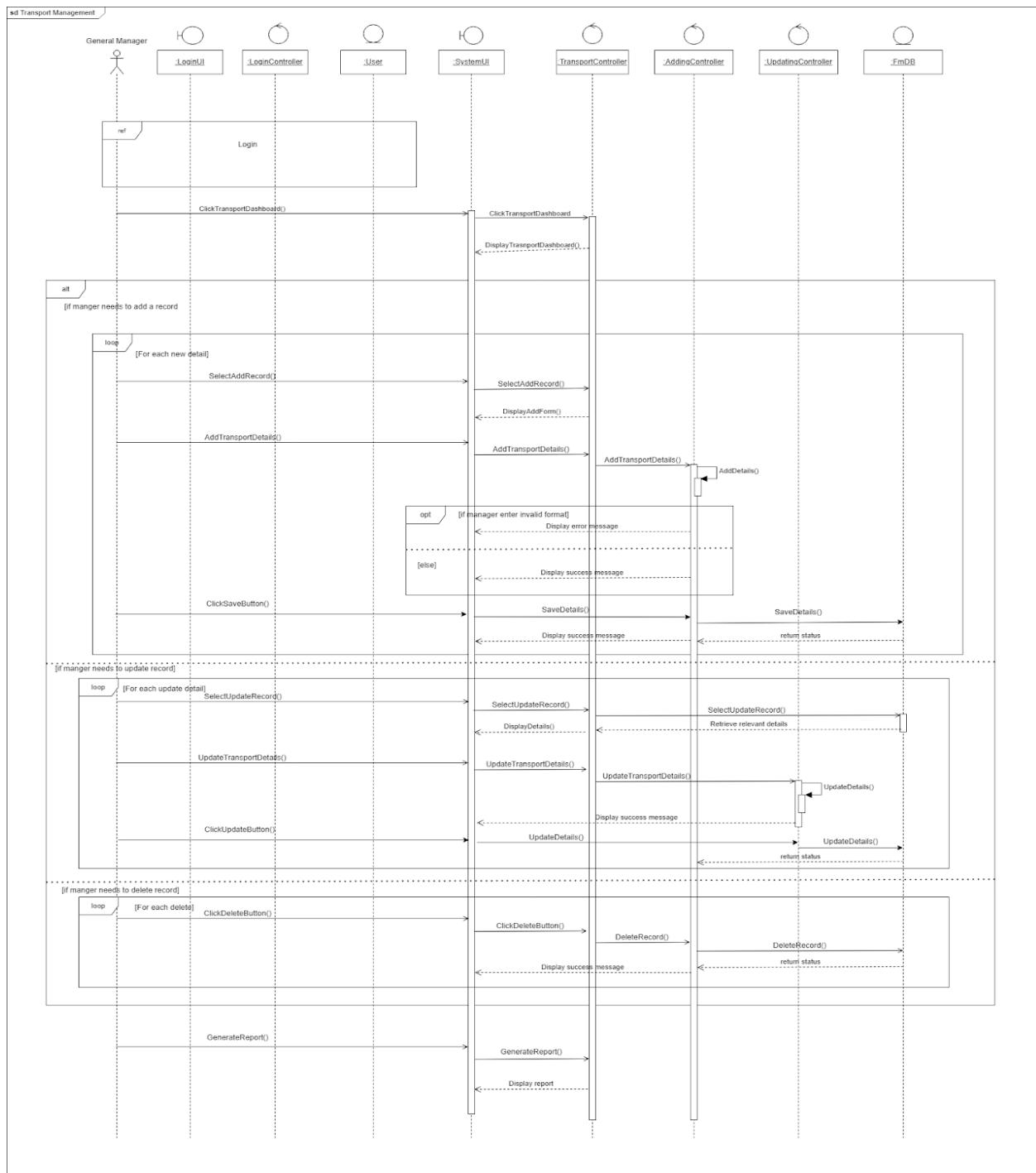


Figure 7.9 : Sequence Diagram - Transport Management System

7.2 Activity Diagrams

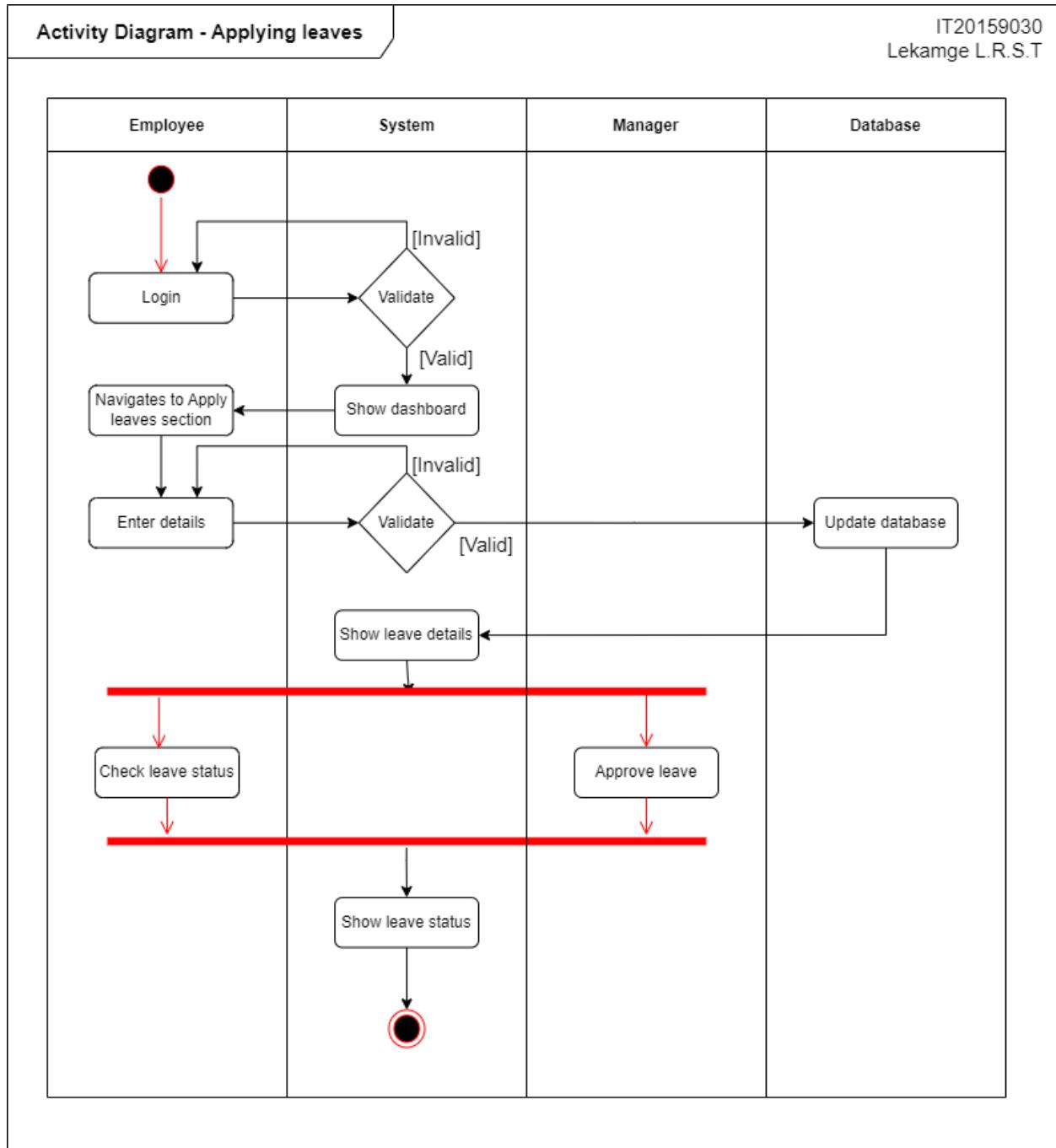


Figure 7.10 : Activity Diagram - Employee Management System

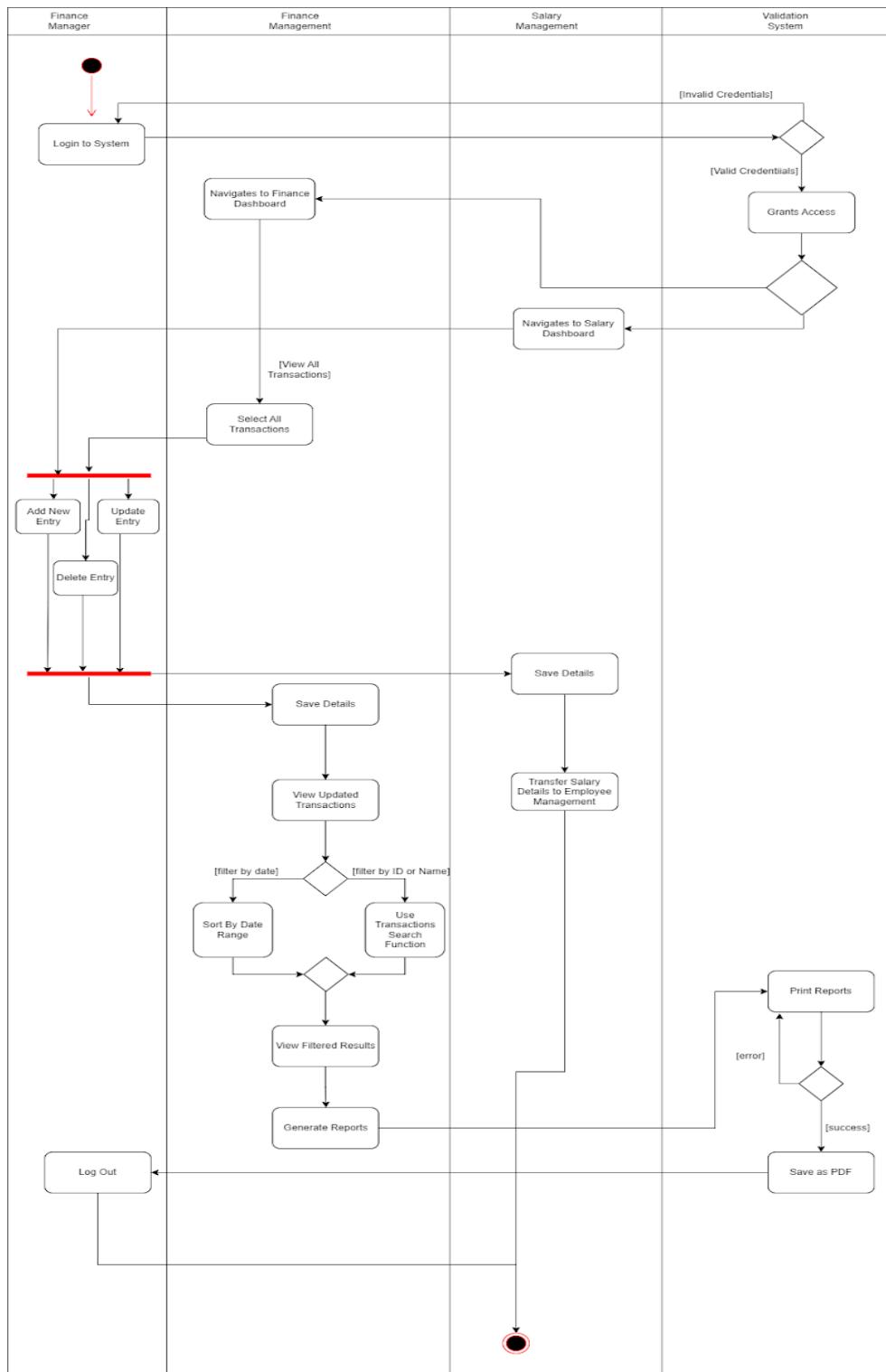


Figure 7.11 : Activity Diagram - Finance Management System

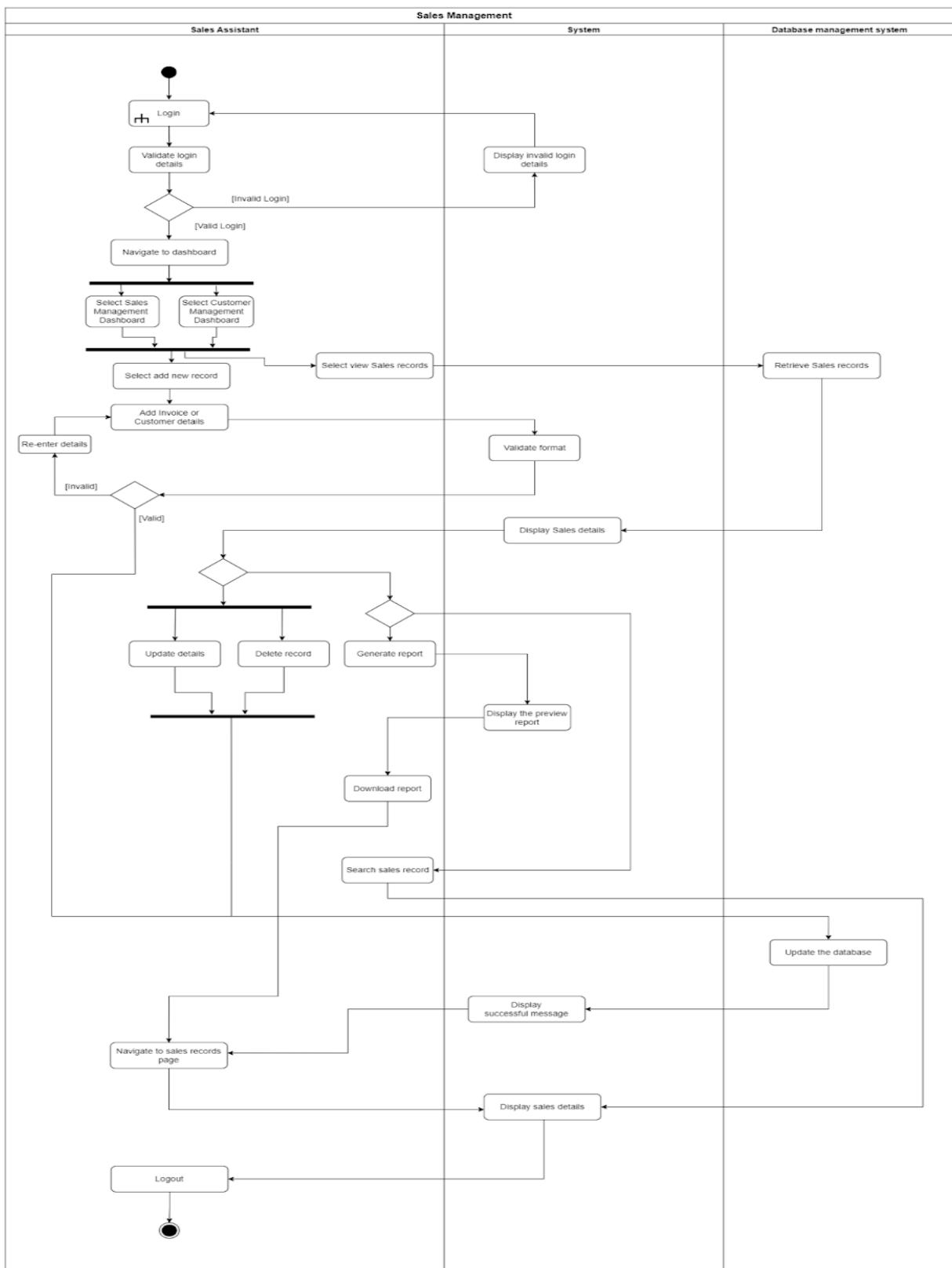


Figure 7.12 : Activity Diagram - Sales Management System

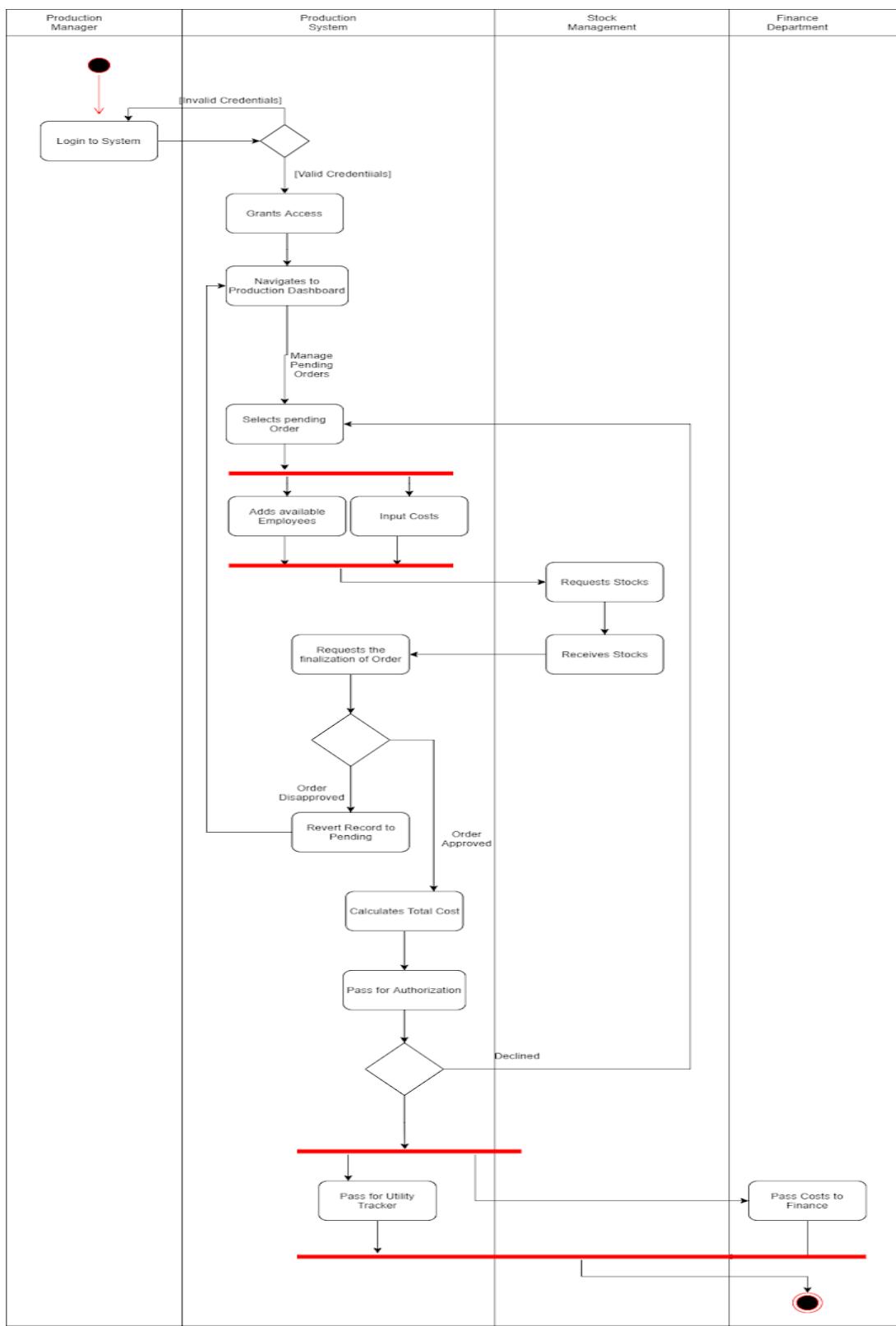


Figure 7.13 : Activity Diagram - Production Management System

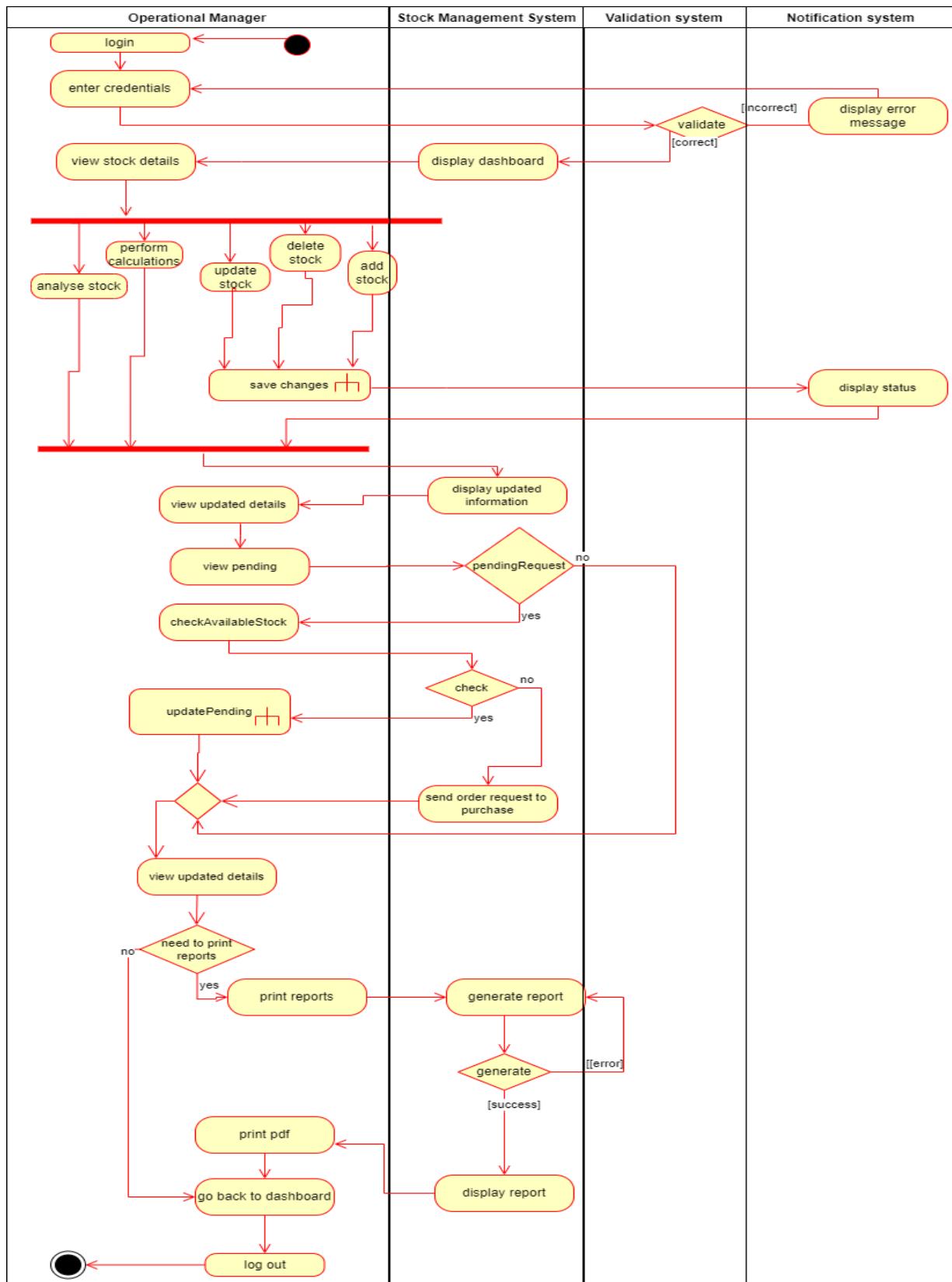


Figure 7.14 : Activity Diagram - Stock Management System

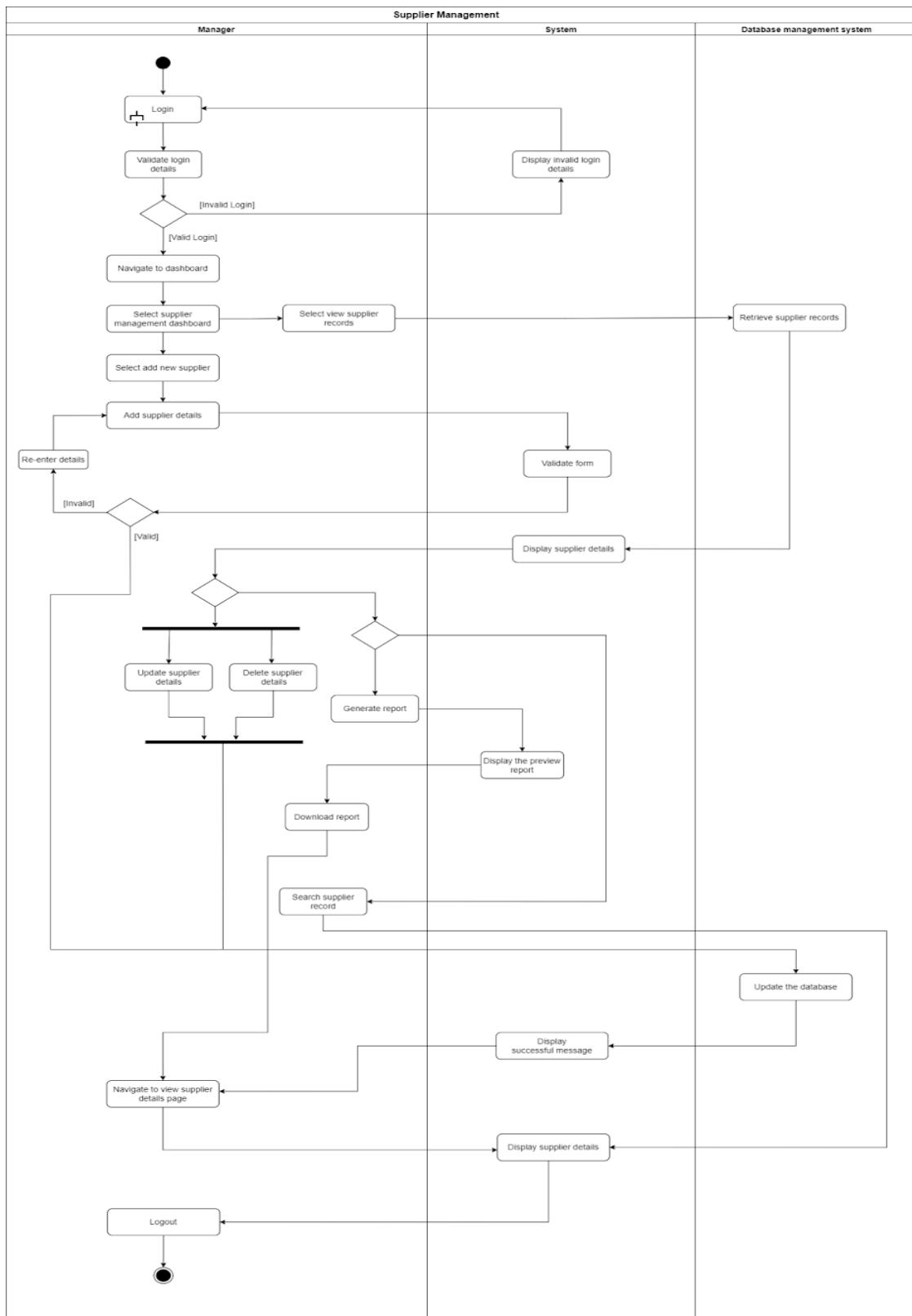


Figure 7.15 : Activity Diagram - Stock Management System

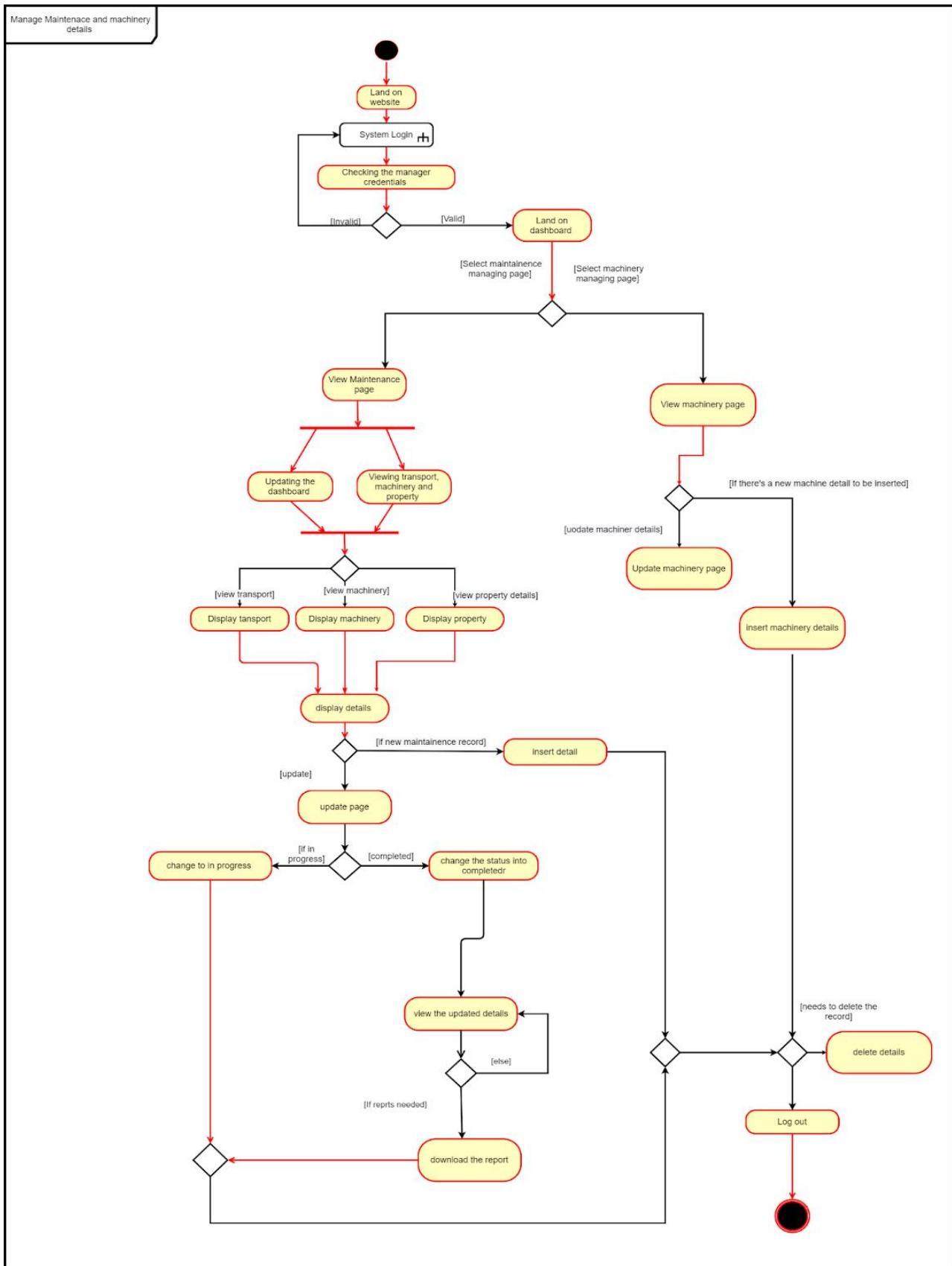


Figure 7.17 : Activity Diagram - Machinery and Maintenance Management System

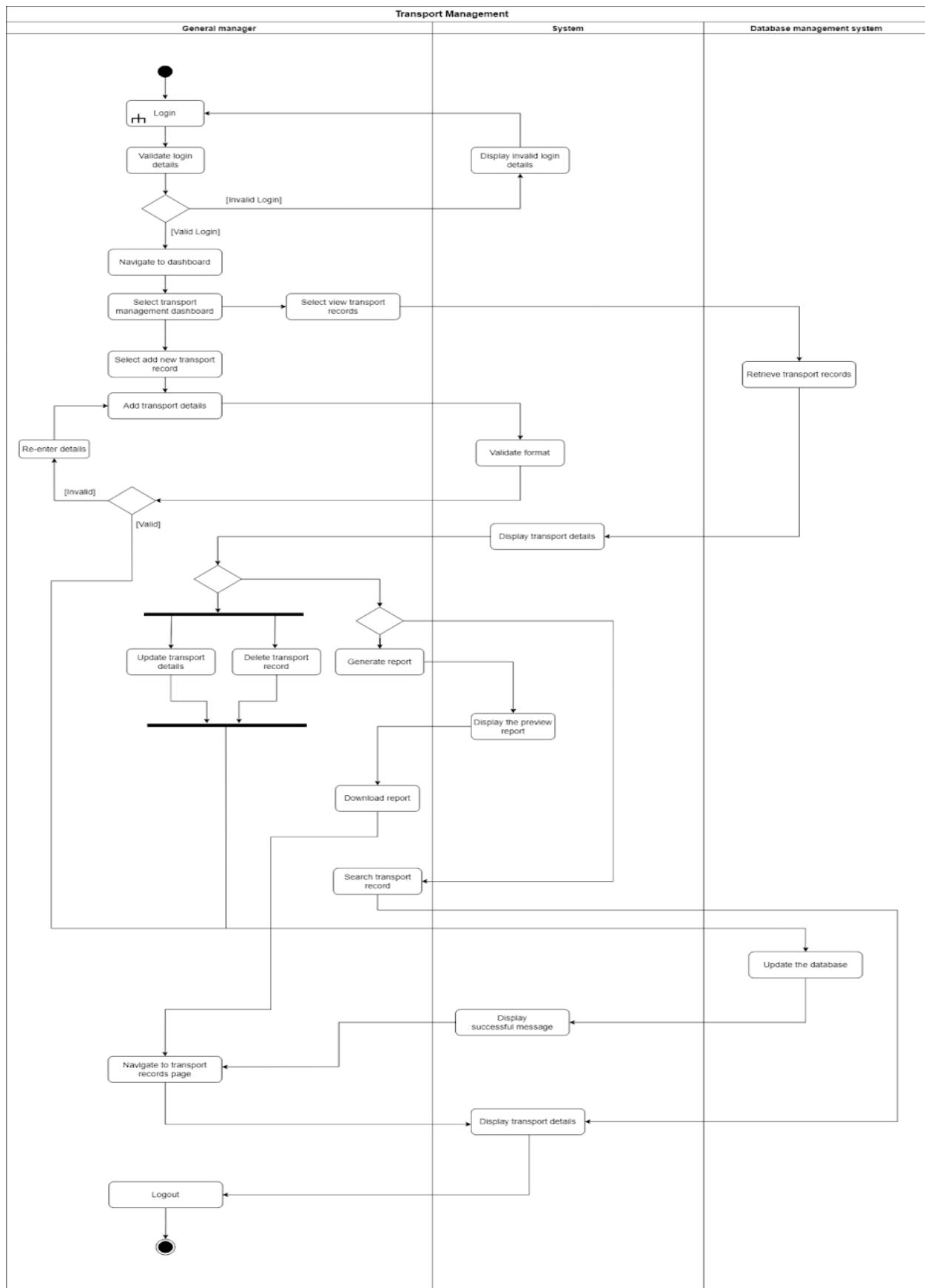


Figure 7.18 : Activity Diagram - Transport Management System

7.3 Screenshots

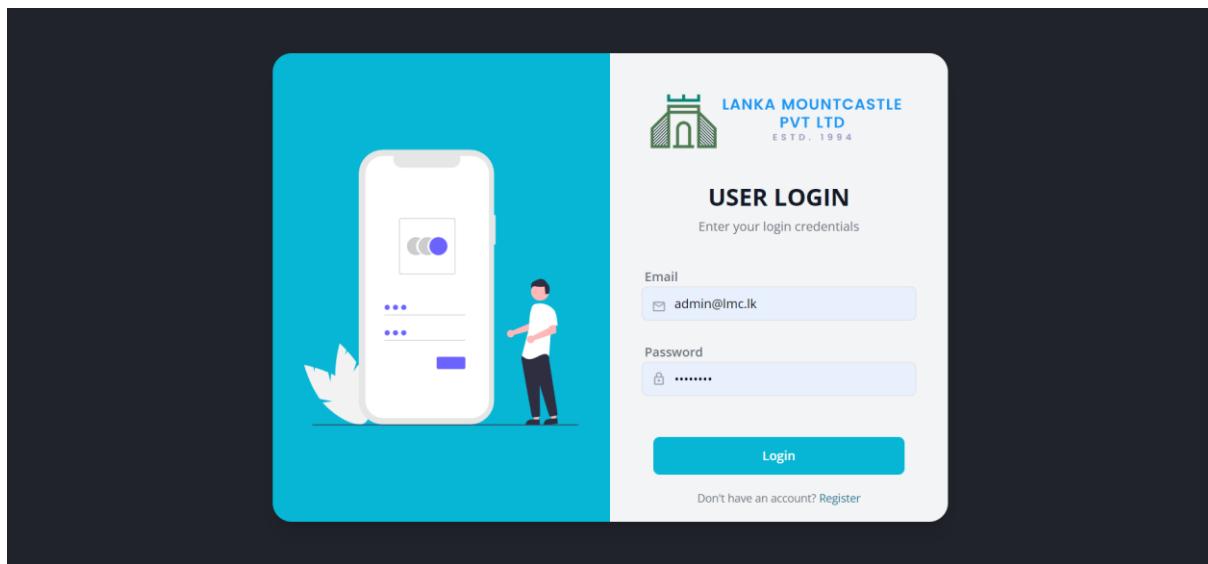


Figure 7.20 : User Login

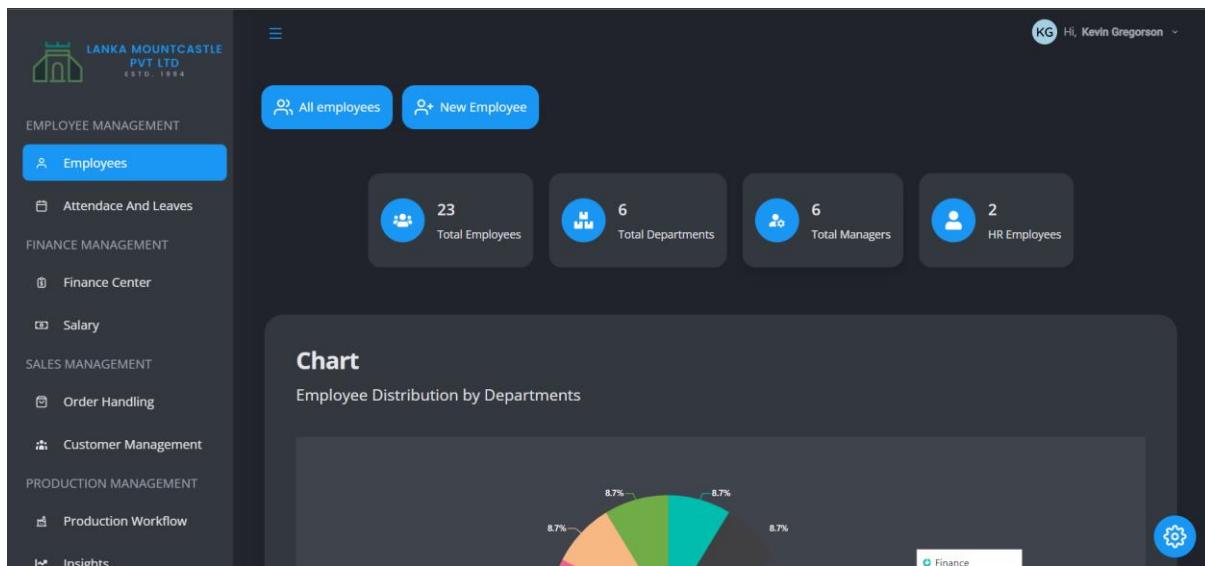


Figure 7.19 : Employee Dashboard

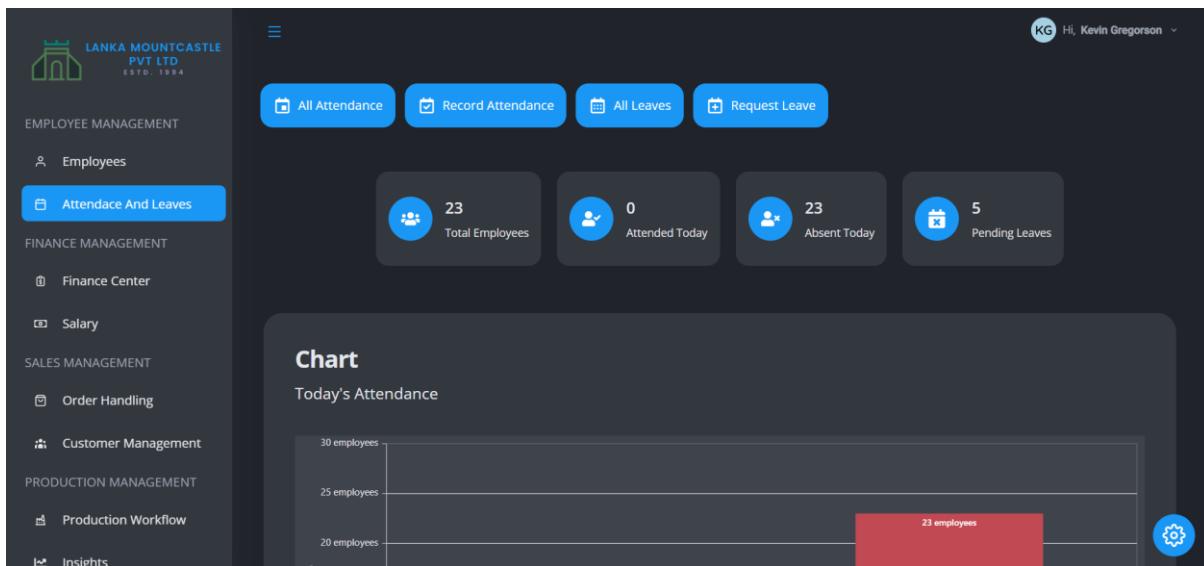


Figure 7.21 : Attendance and Leaves Dashboard

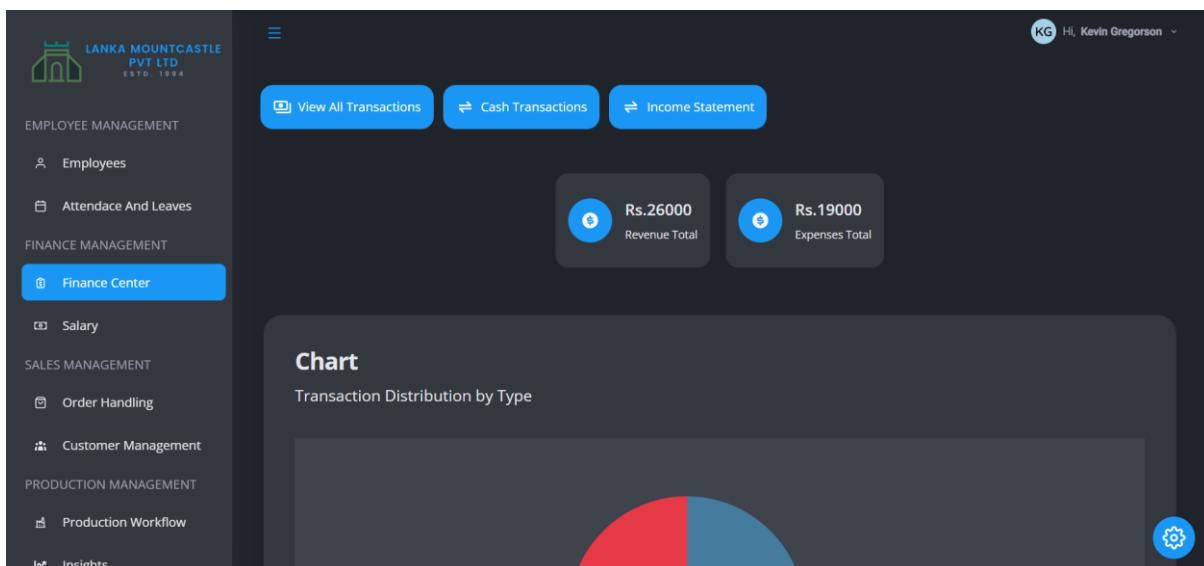


Figure 7.22 : Finance Center



Figure 7.23 : Salary Dashboard

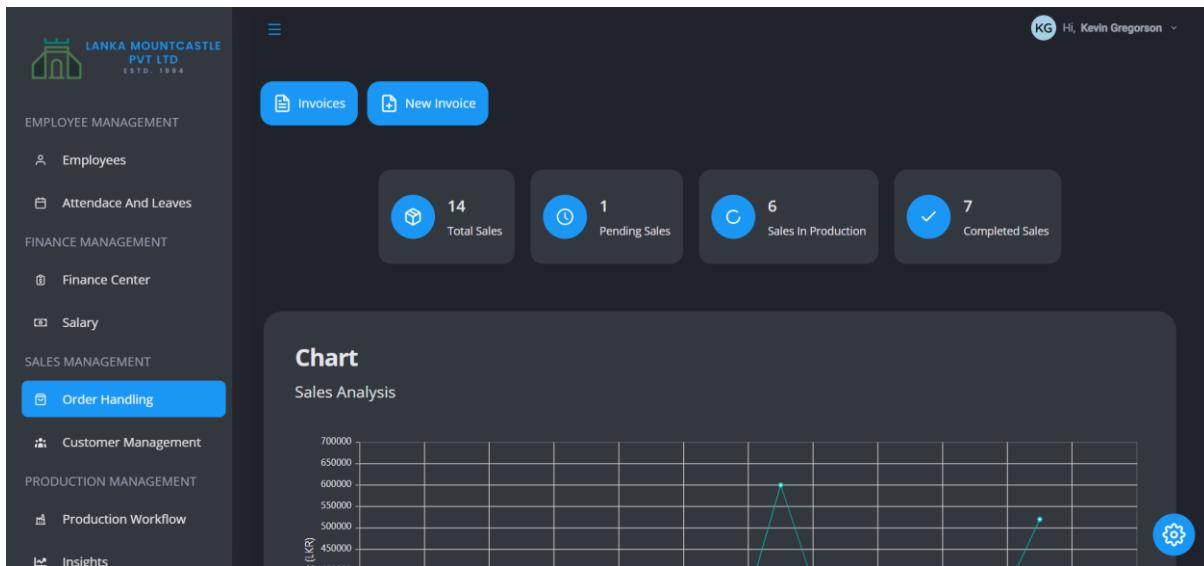


Figure 7.24 : Orders Dashboard

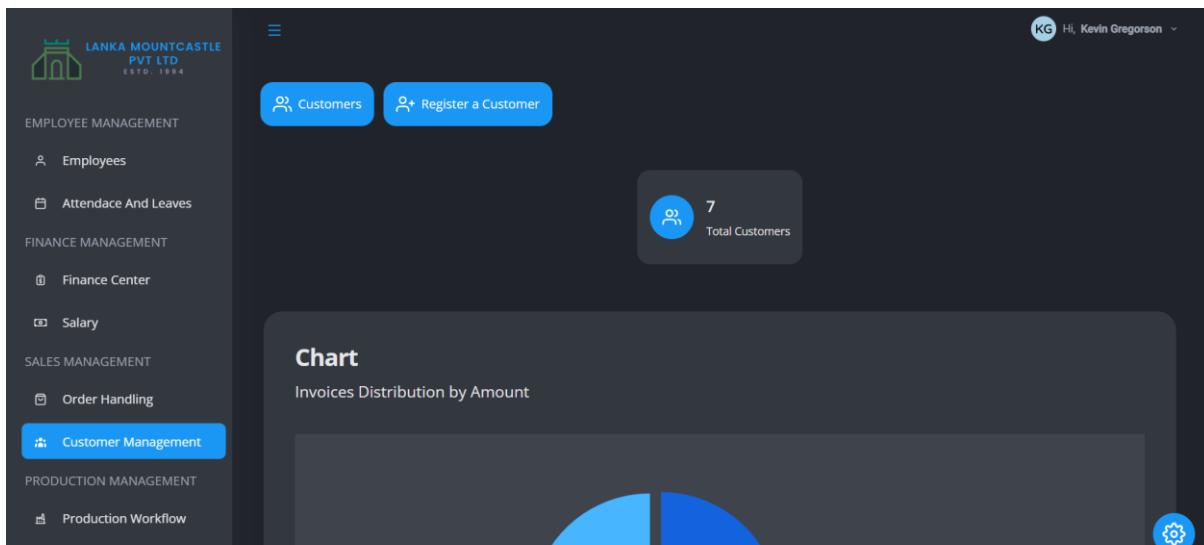


Figure 7.25 : Customer Dashboard

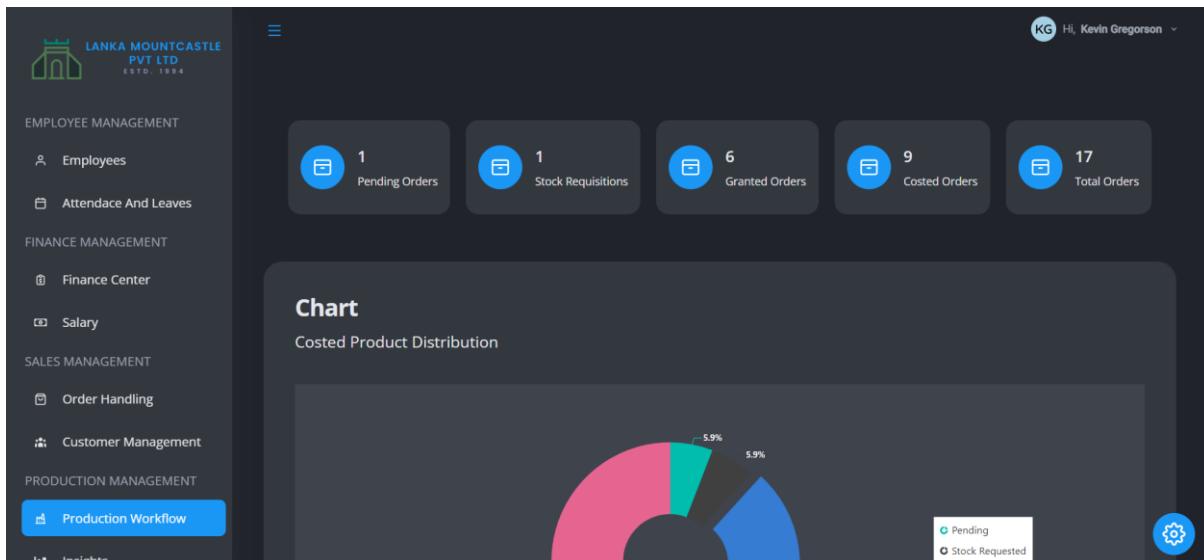


Figure 7.26 : Production Workflow

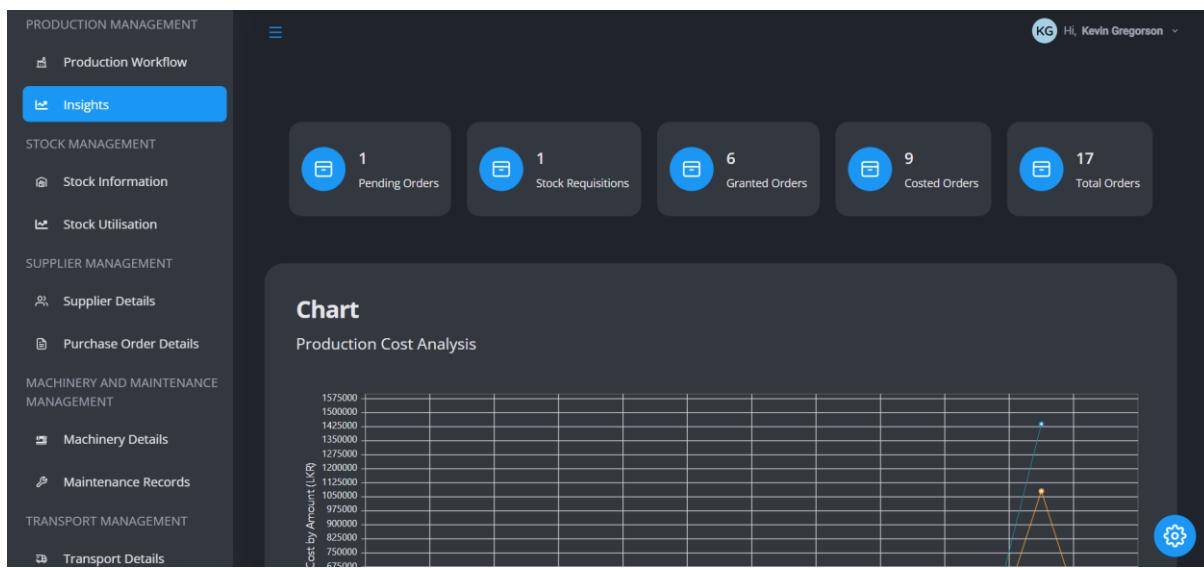


Figure 7.27 : Production Insights

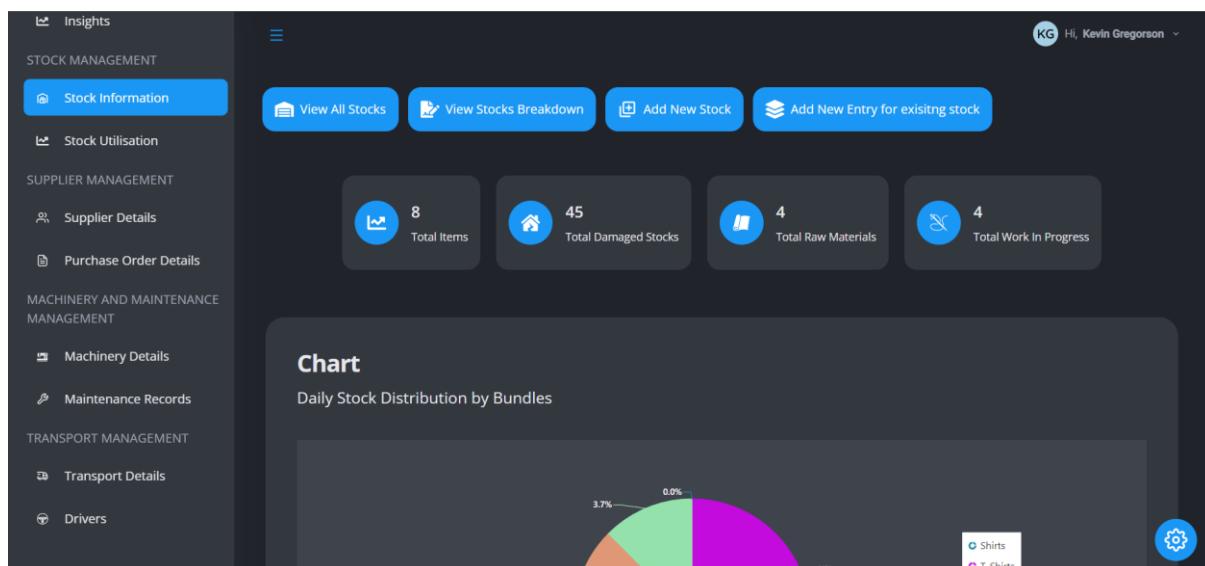


Figure 7.28 : Stock Information

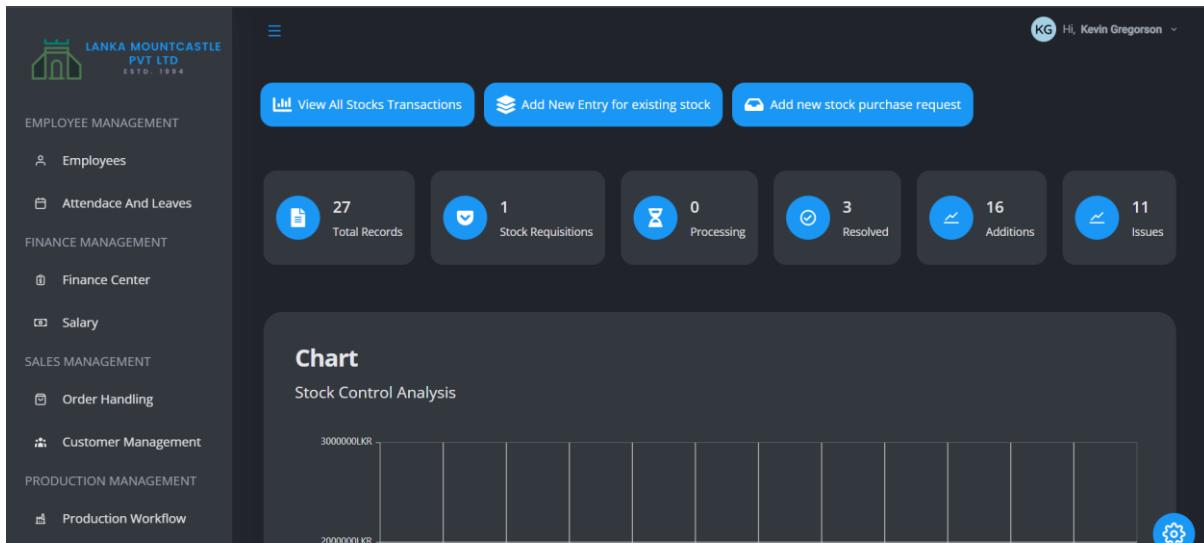


Figure 7.30 : Stock Utilization

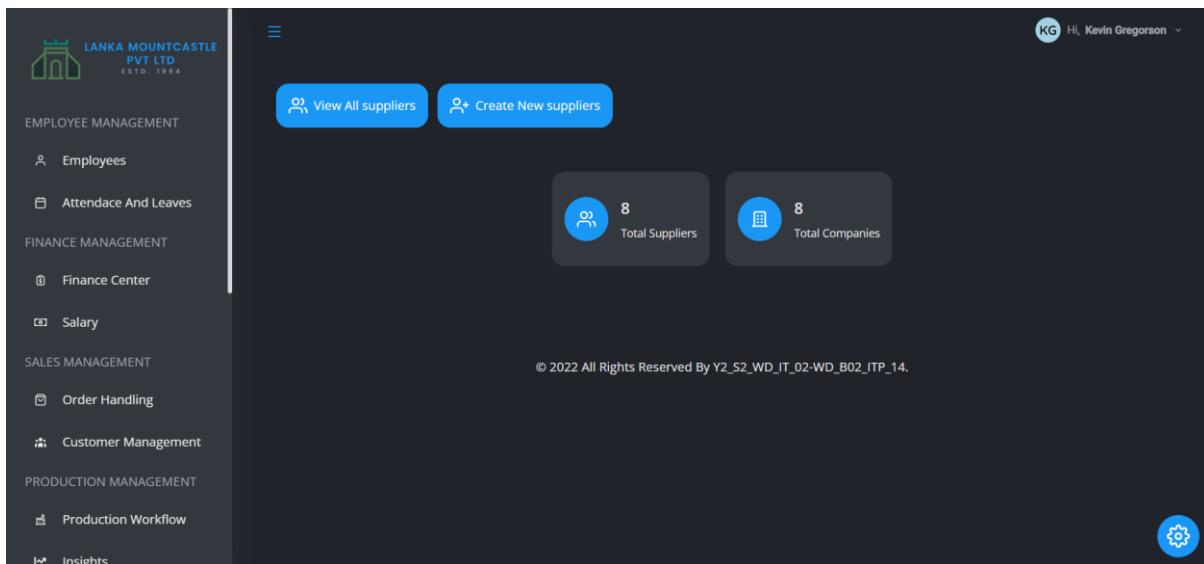


Figure 7.29 : Supplier Dashboard

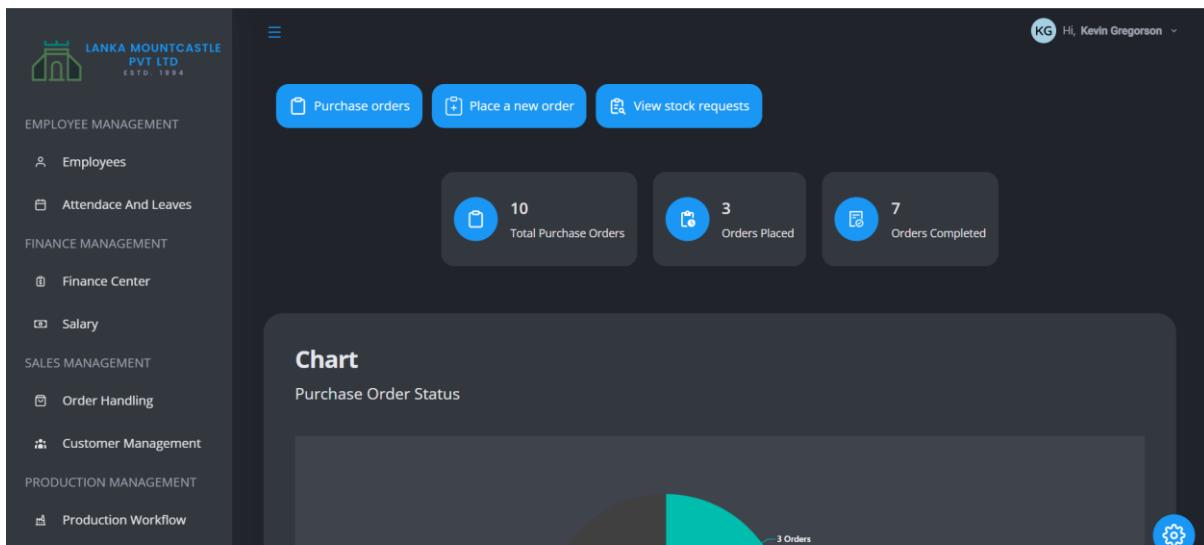


Figure 7.31 : Purchase Order Dashboard

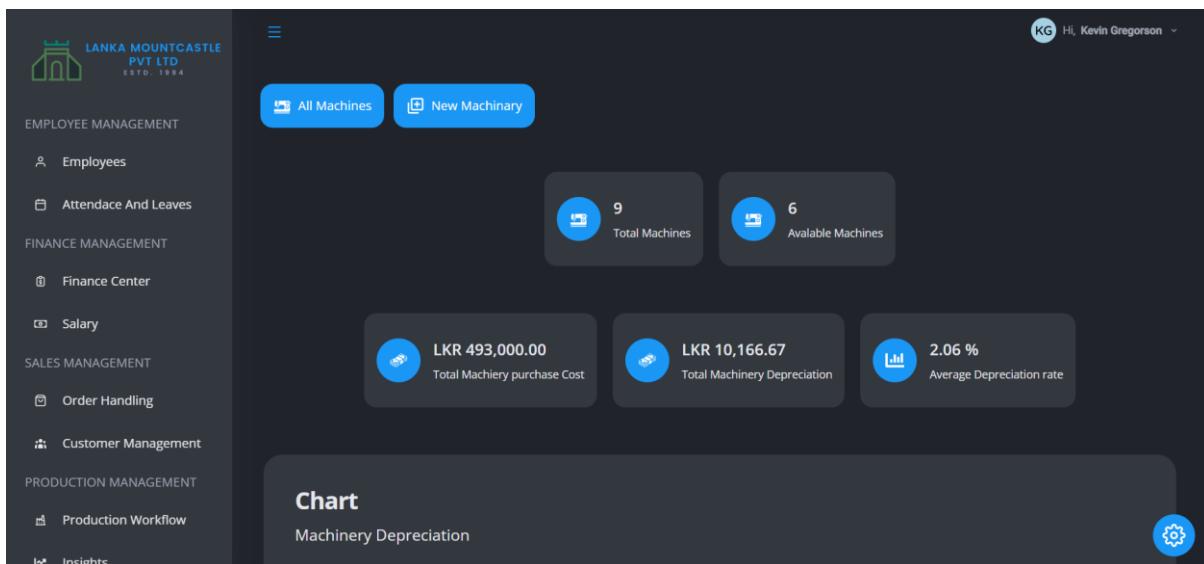


Figure 7.32 : Machinery Dashboard

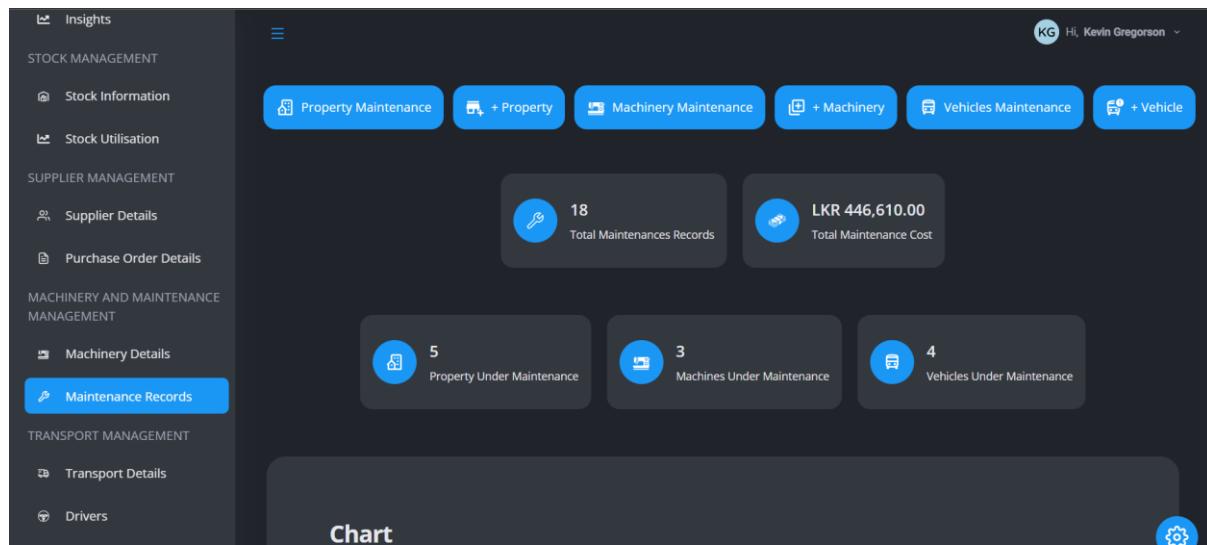


Figure 7.33 : Maintenance Dashboard

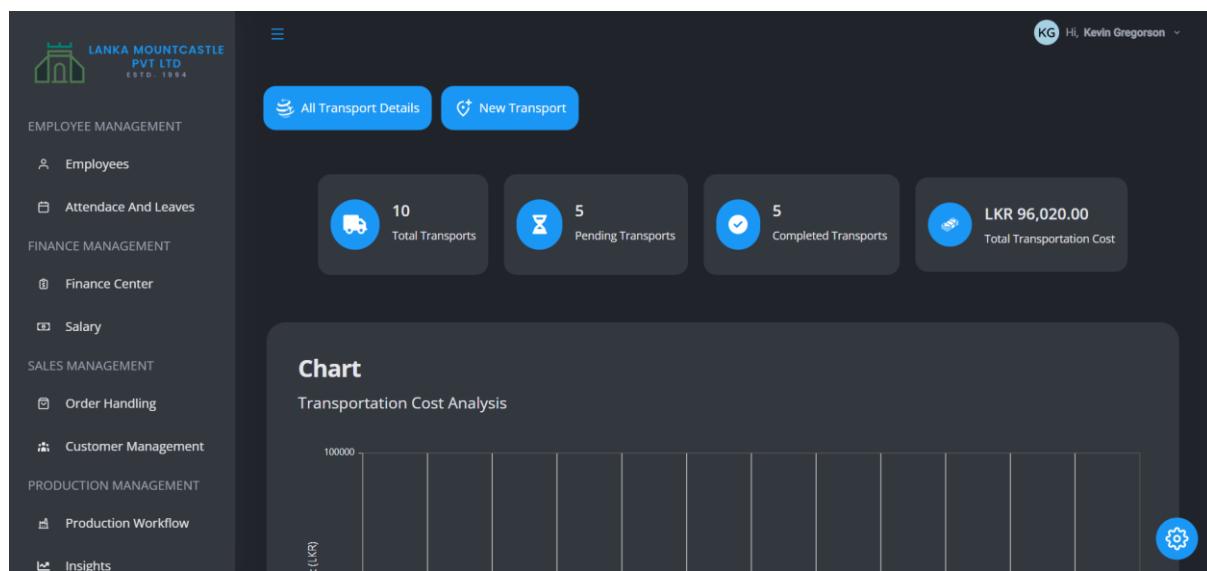


Figure 7.34 : Transport Dashboard

The screenshot shows a dashboard for 'LANKA MOUNTCASTLE PVT LTD' with a dark theme. On the left, a sidebar lists various management categories: Employee Management, Finance Management, Sales Management, and Production Management. The main area displays three summary cards: 'Total Drivers' (8), 'Available Drivers' (6), and 'Unavailable Drivers' (2). Below these is a table titled 'Available Drivers' with columns: NIC, Name, Driving License Number, Contact Number, Vehicle Number, and Vehicle Model. Two rows of data are shown.

NIC	Name	Driving License Number	Contact Number	Vehicle Number	Vehicle Model
199912354678	Hans Christ	6679940	0725836914	LPH6561	Daihatsu Hijet
123456123546	Natalie James	3581558	0123456123	7DIT508	Ford F-450

Figure 7.36 : Driver Dashboard

The screenshot shows a dashboard for 'LANKA MOUNTCASTLE PVT LTD' with a dark theme. The sidebar includes categories like Employee Management, Finance Management, Sales Management, and Production Management. The main area features a table titled 'Leaves' with columns: Employee ID, Employee Name, Leave Type, Start Date, End Date, Reason, Status, and Manage. A search bar and a 'Generate Report' button are at the top of the table area. A 'Filter' button and a date range selector ('Select a month') are also present. The table contains several rows of leave requests.

Employee ID	Employee Name	Leave Type	Start Date	End Date	Reason	Status	Manage
0006	Natalie James	Regular	2022-10-18	2022-10-18	Other	Approved	
0006	Natalie James	Short Leave	2022-09-30	2022-09-30	Other	Approved	
0001	Mary Gregorson	Short Leave	2022-08-31	2022-08-31	Other	Approved	
0001	Mary Gregorson	Short Leave	2022-09-01	2022-09-01	Other	Approved	
0002	janet Peterson	Short Leave	2022-09-04	2022-09-04	Other	Approved	

Figure 7.35 : Tables with search and filter

EMPLOYEE MANAGEMENT

- Employees
- Attendance And Leaves

FINANCE MANAGEMENT

- Finance Center
- Salary

SALES MANAGEMENT

- Order Handling
- Customer Management

PRODUCTION MANAGEMENT

- Production Workflow
- Insights

Form

Create New Stock

Stock Code:

Stock Name

Category:

Description:

Date:

Quantity Purchased:

KG Hi, Kevin Gregorson

Figure 7.38 : Forms with validations

EMPLOYEE MANAGEMENT

- Employees
- Attendance And Leaves

FINANCE MANAGEMENT

- Finance Center
- Salary

SALES MANAGEMENT

- Order Handling
- Customer Management

PRODUCTION MANAGEMENT

- Production Workflow
- Insights

LANKA MOUNTCASTLE
PVT LTD
ESTD. 1994

Lanka MountCastle (Pvt) Ltd,
No.124, Hendala, Wattala
011 2942 672

INVOICE

Invoice No: INV00001
Invoice Date: 2022-1-14
Due Date: 2022-11-13

Billed to:

Customer ID: 00001
Name : Johnny English
Contact No: 0114589632

Invoice No.	Item Name	Quantity	Total Amount
INV00001	Caps	200	LKR 120.000.00

Figure 7.37 : Sales Invoices

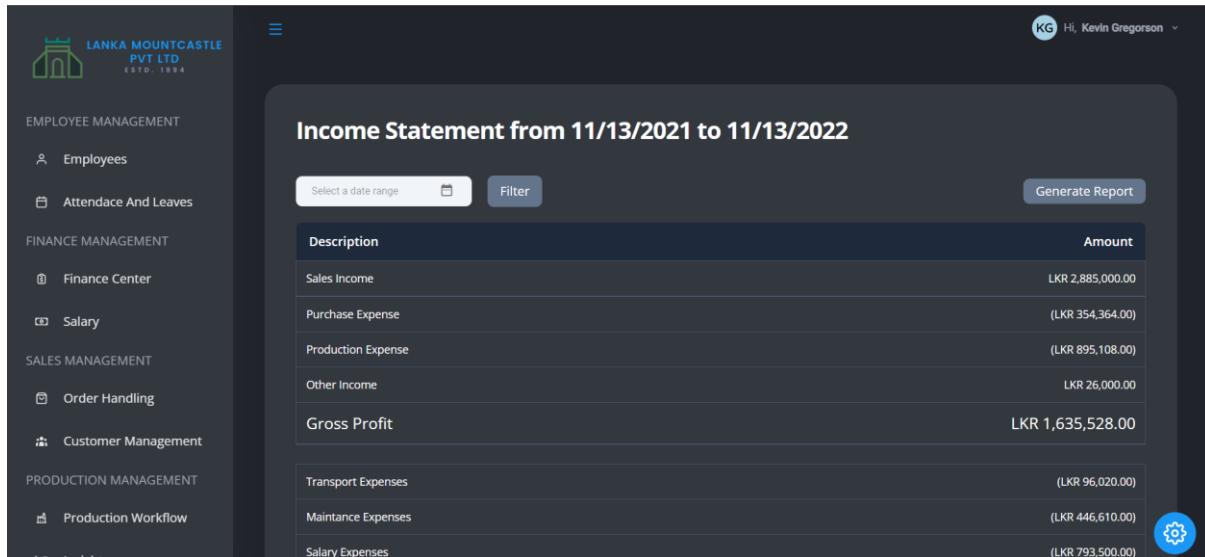


Figure 7.40 : Income statements



Figure 7.39 : Theming Options

Add New Existing Stock

Figure 7.41 : Form with date validations

Stock Name: Caps

Date: 11/11/2022

November 2022

Su	Mo	Tu	We	Th	Fr	Sa
23	24	25	26	27	28	29
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3

Clear Today

0

Add entry for existing stock

LANKA MOUNTCASTLE PVT LTD ESTD. 1994

EMPLOYEE MANAGEMENT

Employees

Attendace And Leaves

FINANCE MANAGEMENT

Finance Center

Salary

SALES MANAGEMENT

Order Handling

Customer Management

PRODUCTION MANAGEMENT

Production Workflow

Insights

KG Hi, Kevin Gregorson

Table Employees

Search Here

Generate Report

Employee ID

0000

0001

0002 Janet Peterson 123456789014 Female Manager

0003 Ben Peterson 123456789015 Male Assistant Manager

0004 Jane Green 123456789016 Female Manager

0005 Devin Sherlock 123456789789 Male Assistant Manager

0006 Natalie James 123456123546 Female Driver

Department

Manage

Sales

Maintenance

Finance

Production

Human Resources

Finance

Driver

Transportation

Data Successfully Updated

Figure 7.42 : Alerts interface