**DISTRIBUTOR OPERATOR APPLICIATION**

The project report is submitted for the partial fulfillment of the Bachelor of Science in Software Engineering Degree to the faculty of Software Engineering.



**Supervised by:**

Nauman-UL-Haq

**Submitted by:**

Awais Ahmad

**DEPARTMENT OF SOFTWARE ENGINEERING GOVERNMENT COLLEGE UNIVERSITY FAISALABAD**

**JULY 2023**

**ACKNOWLEDGEMENT**

First and foremost, all praise is due to Almighty Allah alone - The Omnipresent, The Most Merciful, and Compassionate. Allah is the ultimate source of knowledge, rewarding every effort and never letting our endeavors go in vain.We express our sincere gratitude to our Supervisor, Mr. Nouman, whose invaluable contributions, guidance, and motivation were instrumental in the successful completion of our project. Whenever we encountered challenges, he readily offered his support and assistance.We extend our heartfelt appreciation to all the respondents who generously devoted their precious time and provided the necessary resources that greatly aided our project's progress.Special thanks are also due to our beloved family members for their unwavering support throughout this endeavor, ensuring its smooth execution.Lastly, we are thankful to everyone, whether mentioned or not, who played a role in assisting us to accomplish this work. Your contributions have been instrumental, and we are truly grateful for your support.

**DECELARTION**

We hereby declare that the project entitled **"DISTRIBUTOR OPERATOR,"** presented here with as part of the requirements for the completion of the Bachelor of Science in Software Engineering degree at the Faculty of Software Engineering, Government College University, Main Campus, Faisalabad, is the result of our own efforts under the guidance and supervision of **Nauman-UL-Haq.**.We assure that this project has not been submitted elsewhere, nor is it being considered for the award of any other degree or diploma.We take full responsibility for the accuracy and authenticity of the information contained within this report. Any discrepancies found that contradict the above statement shall render us liable for appropriate action by the university.

**Signature of the Candidates:**

|  |  |
| --- | --- |
|  |  |
| Awais Ahmad  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_

**CERTIFICATE OF SUBMISSION**

This certificate is awarded to the candidate mentioned below for the successful completion of the project titled "DISTRIBUTOR OPERATOR" The project has been duly submitted to the Faculty of Software Engineering at Government College University, Main Campus, Faisalabad, fulfilling a significant requirement for the Bachelor of Science in Software Engineering degree.

**Project Candidates Registration Number Roll Number**

AWAIS AHMAD 2019-GCUF-06301 8932

**Project Supervisor Head of Department**

Nauman-UL-Haq Dr. Muhammad Awais

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Abstract**

The "Distributor Operator" project addresses the need for efficient management of beverage distribution centers by introducing a modern and innovative system. The current manual processes in these centers lead to various operational inefficiencies, such as inventory tracking errors, order processing delays, and a lack of real-time monitoring. These issues impact customer satisfaction, increase costs, and hinder business growth. The objective of the project is to develop an automated and optimized system that revolutionizes the management of distribution centers for beverages. The system aims to streamline operations, provide real-time insights, and enhance overall efficiency and profitability.

The key features of " Distributor Operator" include efficient inventory management, real-time sales monitoring, streamlined order processing, comprehensive financial management, user-friendly interface, employee management capabilities, customizable pricing rates, and financial reporting.

The application is developed using modern technologies such as React Native for the Android-based application and React JS for the Web Panel. The back-end database operations are handled using Firebase, a cloud-hosted NoSQL real-time database. The project follows an iterative development model, allowing for continuous improvement and addition of features based on evolving requirements.

In conclusion, " Distributor Operator" is a promising solution that aims to transform the beverage distribution industry, improve customer service, and optimize operational efficiency. The documentation provides an overview of the project's scope, objectives, features, tools/technologies used, and the development process followed to create this innovative system.

**Table Of Content**

**CHAPTER 1: INTRODUCTION ...............................................................................................1**

1.0 Application Introduction  .....................................................................................................2

1.1      Scope of the Project ......................................................................................................................... 2

1.2      Problem Statement ........................................................................................................................... 3

1.3      Problem Description ........................................................................................................................ 3

1.4      Objective .......................................................................................................................................... 4

1.5      Features  ........................................................................................................................................... 5

1.6      Nature of the Project ........................................................................................................................ 5

1.7      Tools/Technology ............................................................................................................................ 5

   1.7.1    React Native .....................................................................................................................6

                 1.7.2    JavaScript/React JS ........................................................................................................ 6

                 1.7.3    Firebase…………........................................................................................................... 6

1.8      Process Model .................................................................................................................................. 6

1.9    Summary ............................................................................................................................................ 6

**CHAPTER 2: REQUIREMENT SPECIFICATION ...............................................................7**

2.1     Interface Requirement  ...................................................................................................................... 7

                 2.1.1    Hardware Interface Requirement ................................................................................... 7

                 2.1.2    Software Interface Requirement .................................................................................... 7

2.2     Database Requirement  ..................................................................................................................... 8

                 2.2.1    Firebase .......................................................................................................................... 8

2.3     Functional Requirement  ................................................................................................................... 8

2.4     Non Functional Requirement ............................................................................................................ 17

                 2.4.1    Usability ......................................................................................................................... 17

                 2.4.2    Scalability  ..................................................................................................................... 17

                 2.4.3    Maintainability ............................................................................................................... 18

                 2.4.4    Performance ................................................................................................................... 18

                 2.4.5    Security .......................................................................................................................... 18

2.5     Use Case Diagram  ............................................................................................................................ 18

2.6     Use Cases Description Tables  .......................................................................................................... 19

2.7     Summary ........................................................................................................................................... 33

**CHAPTER 3: SYSTEM DESIGN AND UML DIAGRAMS ...................................................34**

3.1    System Design  .................................................................................................................................. 35

3.2    Design Methodology  ......................................................................................................................... 35

3.3    Application Interface Design  ............................................................................................................ 35

3.4    Unified Modeling Language Diagrams  ............................................................................................. 42

                 3.4.1    Activity Diagram ........................................................................................................... 42

                 3.4.2    State Machine Diagram  ................................................................................................. 43

                 3.4.3    Sequence Diagram .......................................................................................................... 52

3.5    System Design View Perspective Models ......................................................................................... 59

                 3.5.1    Logical Design View  .................................................................................................... 60

                 3.5.2    Physical Design View  ................................................................................................... 60

3.6    Summary  ........................................................................................................................................... 61

**CHAPTER 4: IMPLEMENTATION  .......................................................................................62**

4.1    Module  .............................................................................................................................................. 63

                 4.1.1    Signup Module ............................................................................................................... 63

                 4.1.2    Login Module  ................................................................................................................ 63

                 4.1.3    Forgot Password Module  .............................................................................................. 63

                 4.1.4    Home Service Module .................................................................................................... 63

                 4.1.5    Register Service Module ................................................................................................. 63

                 4.1.6    Update Service Module  ................................................................................................. 63

4.1.7    Delete Service Module .................................................................................................... 63

                 4.1.8    Pending or Verified Service Module ............................................................................. 64

                 4.1.9    View Service Module  .................................................................................................... 64

                 4.1.10 Search Service Module .................................................................................................... 64

                 4.1.11 Recommend Service Module ........................................................................................... 64

                 4.1.12 Chat Bridge Module ......................................................................................................... 64

                 4.1.13 Admin Panel Module ........................................................................................................ 64

4.2    Framework ........................................................................................................................................... 64

                 4.2.1   React .................................................................................................................................. 64

4.3    Library ............................................................................................................................................... 64

                 4.3.1   React JS .......................................................................................................................... 65

4.4    Database ............................................................................................................................................. 65

                 4.4.1    Firebase  ......................................................................................................................... 65

4.5    External Application Programming Interface (EAPI) ........................................................................ 65

                 4.5.1    Google Maps API  .......................................................................................................... 65

4.6    Summary  ........................................................................................................................................... 65

**CHAPTER 5: SYSTEM TESTING  ..........................................................................................66**

5.1    Testing ............................................................................................................................................... 67

                 5.1.1    Functional Testing ......................................................................................................... 67

                 5.1.2    Performance Testing ...................................................................................................... 67

                 5.1.3    Compatibility Testing  ................................................................................................... 67

5.2    Testing Methodology ......................................................................................................................... 67

5.3    Test Cases .......................................................................................................................................... 67

5.4    Summary ............................................................................................................................................ 75

**CHAPTER 6: CONCLUSION AND FUTURE WORK  .........................................................76**

6.1    Application Overview ........................................................................................................................ 77

6.2    Milestone Achieved ........................................................................................................................... 77

6.3    Limitation  .......................................................................................................................................... 77

                 6.3.1    Domain ........................................................................................................................... 77

                 6.3.2    Transaction System  ....................................................................................................... 78

                 6.3.3    Single – Platform  .......................................................................................................... 78

6.4    Future Position ................................................................................................................................... 78

6.5    Summary ............................................................................................................................................ 78

**REFERENCES ............................................................................................................................79**

***Chapter No. 01***

***Introduction***

1. **Introduction**

In today's world, efficient management of distribution centers is for ensuring the smooth distribution flow of products and services. Distribution centers are key places where beverages are received, stored, and sent to different retail stores. Currently, manual systems used in these centers often have problems like mistakes in tracking of inventory, a lack of real-time monitoring and delays in processing orders. These issues can cause to higher costs, less Customer satisfaction. That's why we need a product / Software that improves the way distribution centers for beverages work.

That's where We need Application like Distributor Operator. It's a advanced system that revolutionizes the management of cold drink distribution centers. By automating tasks , providing real-time information, optimizing operations. Distributor Operator is reliable for businesses to become more efficient and provide better customer service. With its user-friendly and easy-to-use interface, Distributor Operator is ready to transform the bevrages distribution industry and set standards for how distribution centers are managed.

Distributor Operator aims to increase efficiency, reduce mistakes. It does this by simplifying tasks, giving real-time data and insights, and making it easier for businesses to manage their operations. Distributor Operator is a step forward in managing distribution centers for beverages, and help business to make a successful model.

* 1. **Scope of the Project**

The scope of this project is to develop the Distributor Operator application and website specifically for distribution center. The Distributor Operator system will cover almost all aspects of distribution center's operations. It will include features like real-time tracking of inventory, processing of orders, sales monitoring, and finances management. These functionalities will help us have better control over inventory, streamline order processing, and make informed decisions based on up-to-date information.

Our project's main focus is on creating a genuine solution that addresses distribution center's needs. While the application and website are designed for use and adaptation to benefit other distribution centers in the cold drink industry.

* 1. **Problem Statement**

Currently, distribution center's reliance on outdated manual systems for record management ,including difficulties in inventory / Stocks management, delays in order processing and finance management.

* 1. **Problem Description**

The Beverage distribution center's current reliance on manual systems that cause challenges and limitations that affect operational in Distribution centers.

1. **Inefficient Inventory Management:**

Existing manual inventory management processes for Beverage cause problem in tracking of real time stock management. It cause over-Stocking & out Stocking. Moreover, the lack of real-time tracking in stock levels cause in increased costs, product wastage, and customer dissatisfaction.

1. **Order Processing Delays:**

Manual order processing system cause delay in order process and delivering to customer. Manual data entry, and a lack of automation, paper work techniques slow down the order fulfillment process which causes delayed order processing and shipments ,customer dissatisfaction, missed business opportunities, and revenue loss.

1. **Lack of Real-time Monitoring:**

Lack of real-time monitoring of stocks, orders, and overall performance makes it difficult to make final and correct decisions. Real Time monitoring helps management to control operations and take timely decisions.

1. **Inefficient Financial and Employee Management:**

The manual systems affect tracking and management of financial modules such as expenses, sales analysis, and profit and loss tracking’s. which Cause challenges in effective finance & employee management.

* 1. **Objective**

Distributor Operator project is to develop an Revolutionize and automate system that help us in management of distribution centers for beverages, addressing challenges such as inefficient stocks management, order processing delays, a lack of real-time monitoring of distribution center, and inefficient financial and employee management.

* 1. **Features**

Distributor Operator, an innovative system for beverage distribution centers, offers the following features:

* Efficiently manage and track stock levels of beverages, preventing inventory over stock and less stock and optimizing inventory management processes.
* Real-time monitoring of daily sales, enabling businesses to gain insights into sales performance and trends.
* financial management, including expense tracking, profit and loss calculations, and managing company credit/debit.
* User-friendly interface ,easy-to-use, simplifying tasks and providing real-time data and for performing efficient operations.
* Employee management capabilities, including salesperson tracking, commission management, and recovery tracking.
* Customizable pricing rates and promotional management to support dynamic pricing strategies and marketing campaigns.
* Generate various financial reports, such as balance sheets and net profit statements.
  1. **Nature of the Project**

The project is an android-based application registered on Google Play Store and developed by using various modern technologies.

* 1. **Tools/Technology**

Distributor Operator are developed under the following hardware and software technologies and tools.

* + 1. **React Native**

The logical functionalities of Distributor Operator are written in React Native developed by Meta.

* + 1. **JavaScript/React JS**

For Web Panel, Code is written in React Js.

* + 1. **Firebase**

For back-end database operations, the application used cloud-hosted NoSQL Realtime database called firebase.

* 1. **Process Model**

Distributor Operator is developed by using iterative model. In which initial project is developed on basis of first gathered requirement’s then get further requirement’s and perform updating.

* 1. **Summary**

This chapter covered Distributor Operator an android based application & Web Based Panel brief introduction, the scope of this project, why was this application necessary, its objective, the various features this application provided and nature of the project. After explaining this, the following chapter also discuss tools and technologies under which application was developed and process model through which it completed.

***Chapter No. 02***

***Requirement Specification***

* 1. **Interface Requirement**

Interface requirements are defined as those that the system must accomplish. These include the both hardware and software interface requirements. Distributor Operator is an android based application & Web App and mostly be made up of software components, These are the following hardware and software interface requirements of our application.

* + 1. **Hardware Interface Requirement**

These are the various hardware interface requirements that are necessary for development and deployment of an application.

* + - 1. **Desktop Computer/Laptop**

Desktop computer or laptop having high end specifications are required due to modern technologies and tools (react, etc) used in development process of an application. Minimum required specifications are 8 gigabytes of RAM, Core i5 Processor and 512 gigabytes of SSD.

* + - 1. **Android Smart Phone**

Distributor Operator is an android based application that must deploy and run on an android based smart phone. So, an android smart phone having minimum 3 gigabytes of RAM is required with an internet connection.

* + 1. **Software Interface Requirement**

These are the following software interface requirements that are essential for the development of an application.

* + - 1. **React Native**

Hybrid Platform that combines the best parts of native development with React, a best-in-class , Function JavaScript library for building operator interfaces.it also used some other libraries like Code Push.

* + - 1. **Javascript/React JS**

For Web Side Panel, Various JavaScript and its powerful library React JS is required.

* + - 1. **Visual Studio Code**

As a code editor, visual studio code is required because an excellent editor makes the work one-hundred times easier and visual studio code has all the capabilities that declare it an excellent editor.

* 1. **Database Requirement**

Database is a necessary segment of an application, which provide an ability to perform various data operations at back-end. The following database is required in an application.

* + 1. **Firebase**

For back-end database operations, Firebase is required because the application used cloud-hosted NoSQL Realtime database.

* 1. **Functional Requirement**

Functional requirements are actual product features or functions that developers must implement. These are the various functional requirements of Distributor Operator, an android based application.

* + 1. **Signup/Login**

Distributor Operator is very secured and not available for public usage so everyone cannot allow to login into the app handle it from firebase manually that which operator can login and track records.

* + 1. **Delete Request**

The Delete Record Management Process through the application is designed to handle and manage deletion requests for records within the Distribution web panel. This process plays a crucial role in ensuring the efficient removal of records as per user requests. By utilizing this process, admin are able to initiate and track deletion requests, facilitating a streamlined and organized approach to record management within the application.

* + 1. **Sales Man Records**

This Feature Show Real Time Graph Representation into the app which shows Salman recovery remain are other details such as loan empty recovery balance.

* + 1. **Employee**

In this page it include all the records of an employee which includes loan, recovery, remain as well as details is in which includes Credit Debit details and balance details as well managed by dates and show Graphical Representation of Employee Loan, Recovery & Remain.

* + 1. **Daily Activity**

The daily activity section provides access to all records on a daily basis. It includes the starting time of work, a detailed table of records.

* + 1. **Recommend Service**

The application must have a recommend service feature through which the users swipe down to look at the recommend services generated by an algorithm.

* + 1. **Advance Feature**

In advanced feature page you can select delete the bank as well as you can see all the records of the bank in which day includes (date of transaction amount as well as Credit Debit record from to to record etc) and insecurities on session you can add a operator who can access website panel are login into that parallel to manage over distribution section and it also has a feature of security to check password using fingerprint authentication.

**Website Functions**.

* + 1. **Stocks**

In the stock management section, we can effectively handle the inventory of all available categories. Currently, we have RGB 500 ml, 1000 ml, 1500 ml, and other variations. We can submit stock orders to the warehouse and utilize the warehouse function to access comprehensive stock data and calculate the current stock levels available.

* + 1. **Daily Sale**

In the daily sales module, we oversee the management of inventory and stocks that are dispatched by drivers and subsequently returned. Our goal is to ensure efficient handling of these transactions.

* + 1. **Empty**

This module facilitates the management of empty stocks that are returned to the company. Whether it involves pallets returned by our own company or third-party entities, we maintain real-time records of empty stocks within the warehouse.

* + 1. **Sales Man Recovery**

Salesperson recovery encompasses all records associated with individual salespeople, including loan recovery, closing balances, current balances, and the ability to delete records. This module also supports credit and debit functionalities.

* + 1. **Sales Man (Recovery Book)**

.Salesperson recovery encompasses all records associated with individual salespeople, including loan recovery, closing balances, current balances, and the ability to delete records. This module also supports credit and debit functionalities.

* + 1. **Other Staff (Record Book)**

Similar to the Salesperson Record Book, the Other Staff Record Book contains detailed credit and debit transaction records for all staff members, enabling efficient tracking of their performance and financial activities.

* + 1. **Employee Record**

The Employee Recovery Book consolidates all data related to employee loan recoveries. It provides insights into loan recovery records, categorized by months and dates, facilitating effective management and analysis.

* + 1. **Daily Expenses and Closing**:

Within the Daily Expenses and Closings module, we can record various expenses that occur during distribution, including bills, vehicle expenses, miscellaneous expenses, and expenditures related to salespeople, employees, and investors. This module provides real-time tracking of total cash, bank transfers, and daily closing records.

* + 1. **Vehicle Expenses**

This section records all vehicle-related expenses, allowing us to track monthly expenditures and apply relevant filters to streamline the data.

* + 1. **Miscellaneous Expenses**

Miscellaneous expenses encompass various expenditure categories, such as distribution, construction, stationery, and food expenses.

* + 1. **Company Credit Debit**

The Company Credit and Debit module enables the addition of debit transactions, such as incentives or discounts, as well as credit transactions, such as uniform expenses or other relevant items.

* + 1. **Tax**

This section maintains records of monthly tax payments and closing balances, with the ability to filter data based on specific months.

* + 1. **Billls**

In the bill management section, we handle all types of distribution-related bills and maintain their records. This includes bills from service providers like PTCL, WAPDA, and gas companies.

* + 1. **Net Profit**

The net profit module allows us to calculate the overall profit by considering total expenses, discounts, company expenses, total sales, current stock levels, total incentives, and net profit.

* + 1. **Rate Manager**

In the rate manager module, we can efficiently manage the prices of various stocks. For example, we can set rates for RGB, 500 ml, and 1500 ml stocks. These rates will be utilized for future calculations within the web portal.

* + 1. **Promo Manager**

The promo manager section enables us to add promotions or discounts for different stock items, providing the flexibility to modify their current prices separately.

* + 1. **Investor**

Within the investor section, we maintain a list of all investors associated with the company or distribution. This includes their details, credit and debit information, as well as transaction history.

* + 1. **Balance Sheet**

The balance sheet module allows us to generate comprehensive balance sheets that include records of both debit and credit transactions.

* + 1. **Daily Activity**

The daily activity section provides access to all records on a daily basis. It includes the starting time of work, a detailed table of records, and the option to print out relevant information.

* + 1. **Profit & Loss**

The Profit and Loss module offers a detailed overview of our current financial status, including profit or loss figures, closing balances, and credit and debit records.

**2.4 Non Functional Requirement**

Non-Functional requirements specify the quality attributes of a product and how well it will operate. These are the following non-functional requirements of an application.

* + 1. **Scalability**

The application is scalable because of an iterative approach used while development. In future, any addition or detached of feature is easily possible.

* + 1. **Maintainability**

The application is written by using write-clean code approach. The code is well-organized according to in order structure inside different files and directories.

* + 1. **Performance**

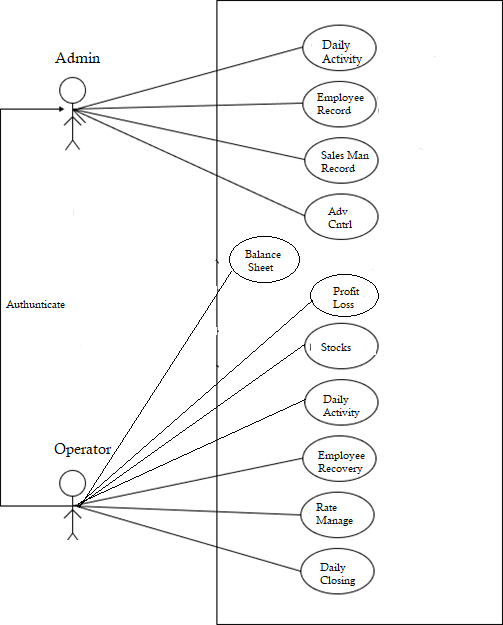
The performance of an application is smooth due to modern front-end and back-end technologies. The users observe the experience of an application overall excellent.

* + 1. **Usability**

Distributor Operator provide effectiveness, efficiency and the overall satisfaction of the user, when they interact with the application.

* 1. **Use Case Diagram**

When a system is examined to gather its functionalities, use cases are created. Below is the following illustrative representation of application use cases.



* 1. **Use Cases Description Tables**

These are the following use cases (illustrate in an above use case diagram) description tables of Distributor Operator.

|  |  |  |  |
| --- | --- | --- | --- |
| App Login Use Case Description Table | | | |
| Name | Login | | |
| ID | UC-01 | | |
| Objective | This use case defines the login process on the application. | | |
| Brief Description | The Admin Can Login in App Which Login is controlled by Firebase | | |
| Pre-Condition | Operator must have to an account in an application or Google. | | |
| Post-Condition | The App is login / Accessed successfully. | | |
| Failed Condition | If you Don’t Have Prmission to Access The App From Firebase | | |
| Primary Actor | Admin | | |
| Dependency | Firebase Approved Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The operator open the application. | The system will show the panel screen. |
| 02 | Admin Can Acess The App if he is already approved From firebase | App Will be Accssible |
| Alternative Flow | Errors | Action | Response |
| 01 | Admin not have acess of App | The system will throw the message That you not have Accessibility |

Use Case Description Table 01: Login

|  |  |  |  |
| --- | --- | --- | --- |
| Delete Record Use Case Description Table | | | |
| Name | Delete Record | | |
| ID | UC-02 | | |
| Objective | This use case outlines the Delete Record Management Process through the application. Essentially, it involves displaying all the deletion requests for records from the Distribution web panel. | | |
| Brief Description | The primary objective of the Delete Record Management Process within the application is to effectively handle and oversee deletion requests for records in the Distribution web panel. | | |
| Pre-Condition | Admin must have Access of an account in an application. | | |
| Post-Condition | Deletion Request Send to Database Successfully | | |
| Failed Condition | Due to Some Inappropriate handling | | |
| Primary Actor | Admin | | |
| Dependency | Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The operator open the application and Click on Delete Request Section | The system will show the Delete Request Screen |
| 02 | Admin Can Delete Details of Deleted Record Request using Delete icon | A Popup Appear For Confirmation of Deletion |
| 03 | The operator select the services form option at bottom navigation bar. | The system will open the services form page. |
| 04 | By Click On Agree Button | System mark the Requst as Deleted |
| Alternative Flow | Errors | Action | Response |
| 01 | By Clicking Disagree | Not Change Anything Deletion Request Will be Cancel |

Use Case Description Table 02: Deletion Service

|  |  |  |  |
| --- | --- | --- | --- |
| Sales Man Record Use Case Description Table | | | |
| Name | Sales Man | | |
| ID | UC-03 | | |
| Objective | This use case defines the real-time graphical representation within the application, showing Salman's recovery progress and other information, such as the remaining loan balance and the amount recovered. | | |
| Brief Description | The real-time graphical representation feature in the application, enabling users to visually track Salman's recovery progress. The feature offers a dynamic display of Salman's ongoing recovery journey, including crucial details such as the remaining balance on his loan and the cumulative amount he has successfully recovered. | | |
| Pre-Condition | Operator must have an account in an application To Access Sales Man Records | | |
| Post-Condition | The service credentials are successfully updated. | | |
| Failed Condition | When you Don’t have access to Admin Account | | |
| Primary Actor | Admin | | |
| Dependency | Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The operator open the application and Click on Sales Man Record. | The system will show the Sales Man Record Scren  And show Graph Representation of Sales man Records |
| 03 | Click on Details. | It will open a modal with all details of Sales Man Including (Loan , Empty , Recovery, Credit / Debit and Balance). |
| Alternative Flow | Errors | Action | Response |
| 01 | The required credentials are missing. | The system will throw the message to must fill the required Crieteria |
| 02 | You Don’t Have Admin Access to Check Details | Show A Popup That you don’t Have Access |

Use Case Description Table 03: Sales Man Record

|  |  |  |  |
| --- | --- | --- | --- |
| Daily Activity Use Case Description Table | | | |
| Name | Daily Activity | | |
| ID | UC-04 | | |
| Objective | The objective of the daily activity section is to provide users with convenient access to comprehensive records on a daily basis. It aims to present relevant information such as the starting time of work and a detailed table of records, enabling users to track their daily activities effectively. | | |
| Brief Description | Daily activity section serves as a centralized hub within the application, offering users a convenient way to access and review their daily records. It prominently displays the starting time of work, allowing users to quickly identify when their workday began. | | |
| Pre-Condition | Admin must have an account in an application Data Base. | | |
| Post-Condition | Visit Daily Basis Activities | | |
| Failed Condition | Internet Connection Lose | | |
| Primary Actor | Admin | | |
| Dependency | Login Use Case Diagram | | |
| Basic Flow | Steps | Action | Response |
| 01 | The operator open the application. | The system will show the Daily Activity Screen. |
| Alternative Flow | Errors | Action | Response |
| None | None | None |

Use Case Description Table 04: Daily Activity

|  |  |  |  |
| --- | --- | --- | --- |
| Employee Record Use Case Description Table | | | |
| Name | Sales Man | | |
| ID | UC-05 | | |
| Objective | The objective of this page is to provide a comprehensive overview of an employee's records, including loan information, recovery progress, remaining balance, as well as detailed credit and debit information. It aims to organize these records by dates and present them in a graphical representation, offering a visual summary of the employee's loan, recovery, and remaining balance | | |
| Brief Description | This page serves as a centralized repository for all relevant records pertaining to an employee. It includes detailed information regarding loans, recovery progress, and remaining balances. The records are organized and managed based on dates, allowing users to easily navigate and access specific entries associated with the employee's account. | | |
| Pre-Condition | Operator must have an account in an application To Access App Functionality | | |
| Post-Condition | Admin Can Access Records | | |
| Failed Condition | When you Don’t have access to Admin Account | | |
| Primary Actor | Admin | | |
| Dependency | Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The operator open the application. And Click on Employee Record | The system will show the Employee Records Screen  And show Graph Representation of Sales man Records |
| 03 | Click on Details. | It will open a modal with all details of Sales Man Including (Loan , Empty , Recovery, Credit / Debit and Balance). |
| Alternative Flow | Errors | Action | Response |
| 01 | The required credentials are missing. | The system will throw the message to must fill the required Crieteria |
| 02 | You Don’t Have Admin Access to Check Details | Show A Popup That you don’t Have Access |

Use Case Description Table 05: Employee Records

|  |  |  |  |
| --- | --- | --- | --- |
| Advance Features Use Case Description Table | | | |
| Name | Advance Feature | | |
| ID | UC-06 | | |
| Objective | Objective of the advanced feature page is to provide users with enhanced functionality for managing their banking activities. This includes the ability to delete a bank, access and review detailed transaction records, add an operator with website panel access for distribution management, and incorporate fingerprint authentication for secure password verification. | | |
| Brief Description | The advanced feature page offers a range of powerful capabilities to facilitate efficient banking management. Users have the option to delete a bank, providing flexibility in managing their accounts and removing unnecessary entries.  The page also grants users access to comprehensive bank records, presenting transaction details such as the date, amount, and credit/debit information. This allows users to track their financial activities and gain a clear understanding of their banking history. | | |
| Pre-Condition | Operator must have an account in an application To Access App Functionality | | |
| Post-Condition | Admin Can Access Records | | |
| Failed Condition | When you Don’t have access to Admin Account | | |
| Primary Actor | Admin | | |
| Dependency | Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The operator open the application.  And Click on Advance Feature | The system will show Advance Feature Screen. |
| 02 | The operator Can Select The Bank From List Which Can be Added in the Allowed List | The system will Update Bank in The Database and used For in Distribution Panel |
| 03 | User Will Fill The Form To Add Operator For Distribution Website Panel | The system will Add Record of Allowed Operator in Data base |
| 04 | Admin Click on Finget Print Icon To Check Password of that Screen | The system will open the Finger Print Scanner and Scan finger if Success it show the Password |
| Alternative Flow | Errors | Action | Response |
| Network Error | While Performing Any Action | Failure of Record |

Use Case Description Table 06: Advance Feature

|  |  |  |  |
| --- | --- | --- | --- |
| View Service Use Case Description Table | | | |
| Name | View Service | | |
| ID | UC-08 | | |
| Objective | This use case defines the view service process. | | |
| Brief Description | The operator views the public and private businesses, marketplaces, institutes and organizations. | | |
| Pre-Condition | Operator must have an account and open the application | | |
| Post-Condition | The services are displayed successfully. | | |
| Failed Condition | No internet connection. | | |
| Primary Actor | Operator | | |
| Dependency | Signup Use Case or Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The operator open the application. | The system will show the home screen. |
| 02 | The operator views all the services portfolios under their specific categories. | The system will show the services portfolios under their specific categories. |
| Alternative Flow | Error | Action | Response |
| 01 | The operator opens the application without internet connection. | The system will throw the message, no internet connection. |

Use Case Description Table 06: View Service

|  |  |  |  |
| --- | --- | --- | --- |
| Search Service Use Case Description Table | | | |
| Name | Search Service | | |
| ID | UC-09 | | |
| Objective | This use case defines the search service process of public and private businesses, marketplaces, institutes and organizations. | | |
| Brief Description | The operator will search the public and private businesses, marketplaces, institutes and organizations according to their categories or nearby feature. | | |
| Pre-Condition | Operator must have an account and open the application. | | |
| Post-Condition | The filtered results shown successfully. | | |
| Failed Condition | When required service is not available. | | |
| Primary Actor | Operator | | |
| Dependency | Signup Use Case or Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The operator open the application. | The system will show the home screen. |
| 02 | The operator searches the require services portfolios by write down the category, apply category filter or nearby feature. | The system will filter the required services portfolios according to the feature and show them. |
| Alternative Flow | Errors | Action | Response |
| None | None | None |

Use Case Description Table 09: Search Service

|  |  |  |  |
| --- | --- | --- | --- |
| Stocks Use Case Description Table | | | |
| Name | Stock Management | | |
| ID | UC-10 | | |
| Objective | The objective of the stock management section is to efficiently handle the inventory of all available categories, specifically RGB 500 ml, 1000 ml, 1500 ml, and other variations. The system allows users to submit stock orders to the warehouse and access comprehensive stock data, enabling them to calculate the current stock levels available. | | |
| Brief Description | The stock management section is a feature within the overall system that deals with inventory control and stock tracking. It provides a centralized platform for managing different product categories, with a particular focus on RGB bottles of varying sizes (500 ml, 1000 ml, 1500 ml, etc.). Users with the appropriate permissions can access this section to perform various stock-related tasks. | | |
| Pre-Condition | The owners must login into the admin panel. | | |
| Post-Condition | The required action is performed successfully. | | |
| Failed Condition | Internet Connectivity Failure | | |
| Primary Actor | Operator | | |
| Dependency | Advance Feature Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The owners of the application open the Distribution operator by given access from admin | The system will show the Menu Screen |
| 02 | From Menu Screen Operator Click The Stock Option | Operator will navigate to Stock Managment Screen |
| 03 | He / She Can Add Items to Stocks | System will update the stocks |
| 04 | The operator also review history of stocks | System show history of Stocks |
| Alternative Flow | Errors | Action | Response |
| None | None | None |

Use Case Description Table 10: Stocks management

|  |  |  |  |
| --- | --- | --- | --- |
| Daily Sale Use Case Description Table | | | |
| Name | Daily Sale | | |
| ID | UC-11 | | |
| Objective | Objective of the daily sales module is to oversee the management of inventory and stocks that are dispatched by drivers and subsequently returned. The primary goal is to ensure the efficient handling of these transactions, which involves tracking sales, managing stock levels, and handling product returns effectively. | | |
| Brief Description | daily sales module is an integral part of the system designed to handle the sales and inventory management process for a business. | | |
| Pre-Condition | The owners must login into the admin panel. | | |
| Post-Condition | The required action is performed successfully. | | |
| Failed Condition | Internet Connectivity Failure | | |
| Primary Actor | Operator | | |
| Dependency | Advance Feature Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The owners of the application open the Distribution operator by given access from admin | The system will show the Menu Screen |
| 02 | From Menu Screen Operator Click The Daily Sale | Operator will navigate to Daily Sale Managment Screen |
| 03 | He / She Can Manage Start and Close of Sale | System will update the Sales Records |
| 04 | The operator also review history of stocks | System show history of Sales |
| Alternative Flow | Errors | Action | Response |
| None | None | None |

Use Case Description Table 11: Daily Sale

|  |  |  |  |
| --- | --- | --- | --- |
| Empty Use Case Description Table | | | |
| Name | Empty | | |
| ID | UC-12 | | |
| Objective | Objective of this module is to streamline and improve the management of empty stocks that are returned to the company. It aims to provide a robust system to handle the handling, tracking, and recording of empty stocks, including pallets, that are either returned by the company's own operations or received from third-party entities. | | |
| Brief Description | Module Empty is to streamline and improve the management of empty stocks that are returned to the company. It aims to provide a robust system to handle the handling, tracking, and recording of empty stocks, including pallets, that are either returned by the company's own operations or received from third-party entities. | | |
| Pre-Condition | The owners must login into the admin panel. | | |
| Post-Condition | The required action is performed successfully. | | |
| Failed Condition | Internet Connectivity Failure | | |
| Primary Actor | Operator | | |
| Dependency | Advance Feature Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The owners of the application open the Distribution operator by given access from admin | The system will show the Menu Screen |
| 02 | From Menu Screen Operator Click The Empty | Operator will navigate to Empty Managment Screen |
| 03 | He / She Can Manage Empty Here | System will update the Empty Records |
| Alternative Flow | Errors | Action | Response |
| None | None | None |

Use Case Description Table 12: Empty Management

|  |  |  |  |
| --- | --- | --- | --- |
| Sales Man Recovery Case Description Table | | | |
| Name | Sales Man Recovery | | |
| ID | UC-13 | | |
| Objective | Salesperson recovery encompasses all records associated with individual salespeople, including loan recovery, closing balances, current balances, and the ability to delete records. This module also supports credit and debit functionalities. | | |
| Brief Description | The Salesperson Recovery Module is a comprehensive system that manages and tracks all crucial aspects related to individual salespeople. It maintains essential records such as loan recovery, closing balances, and current balances, providing a robust framework for effective salesperson management | | |
| Pre-Condition | The owners must login into the admin panel. | | |
| Post-Condition | The required action is performed successfully. | | |
| Failed Condition | Internet Connectivity Failure | | |
| Primary Actor | Operator | | |
| Dependency | Advance Feature Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The owners of the application open the Distribution operator by given access from admin | The system will show the Menu Screen |
| 02 | From Menu Screen Operator Click The Sales Man Recovery | Operator will navigate to Sales Man Recovery Screen |
| 03 | He Can add the loan and empty recovery | System will update the recovery Database |
|  | 04 |  |  |
| Alternative Flow | Errors | Action | Response |
| None | None | None |

Use Case Description Table 13: Sales Man Recovery

|  |  |  |  |
| --- | --- | --- | --- |
| Sales Man Case Description Table | | | |
| Name | Sales Man Recovery | | |
| ID | UC-13 | | |
| Objective | Objective of the Salesperson Record Book is to maintain a detailed and organized log of credit and debit transactions for individual salespeople | | |
| Brief Description | Salesperson Record Book contains a comprehensive log of credit and debit transactions, allowing us to track their history by date and effectively monitor their sales performance. | | |
| Pre-Condition | The owners must login into the admin panel. | | |
| Post-Condition | The required action is performed successfully. | | |
| Failed Condition | Internet Connectivity Failure | | |
| Primary Actor | Operator | | |
| Dependency | Advance Feature Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The owners of the application open the Distribution operator by given access from admin | The system will show the Menu Screen |
| 02 | From Menu Screen Operator Click The Sales Man | Operator will navigate to Sales Man Recovery Book |
| 03 | He Can Check the Closing Balance | System will Show History of Closing Balance |
|  | 04 |  |  |
| Alternative Flow | Errors | Action | Response |
| None | None | None |

Use Case Description Table 13: Sales Man Record Book

|  |  |  |  |
| --- | --- | --- | --- |
| Other Employee Record Book Case Description Table | | | |
| Name | Other Employee Record Book | | |
| ID | UC-14 | | |
| Objective | Similar to the Salesperson Record Book, the Other Staff Record Book contains detailed credit and debit transaction records for all staff members, enabling efficient tracking of their performance and financial activities. | | |
| Brief Description | Other Staff Record Book contains detailed credit and debit transaction records for all staff members, enabling efficient tracking of their performance and financial activities. | | |
| Pre-Condition | The owners must login into the admin panel. | | |
| Post-Condition | The required action is performed successfully. | | |
| Failed Condition | Internet Connectivity Failure | | |
| Primary Actor | Operator | | |
| Dependency | Advance Feature Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The owners of the application open the Distribution operator by given access from admin | The system will show the Menu Screen |
| 02 | From Menu Screen Operator Click The Sales Man | Operator will navigate to Sales Man Recovery Book |
| 03 | He Can Check the Closing Balance | System will Show History of Closing Balance |
|  | 04 |  |  |
| Alternative Flow | Errors | Action | Response |
| None | None | None |

Use Case Description Table 14: Other Employee Record Book

|  |  |  |  |
| --- | --- | --- | --- |
| Employee Recovery Case Description Table | | | |
| Name | Sales Man Recovery | | |
| ID | UC-15 | | |
| Objective | The Employee Recovery Book is a centralized system that consolidates all data pertaining to employee loan recoveries, offering comprehensive insights into loan recovery records categorized by months and dates, enabling efficient management and analysis. | | |
| Brief Description | Employee Recovery Book serves as a vital repository for managing employee loan recoveries within an organization. By centralizing all relevant data, it provides a clear and organized overview of loan recovery records, arranged by months and date. | | |
| Pre-Condition | The owners must login into the admin panel. | | |
| Post-Condition | The required action is performed successfully. | | |
| Failed Condition | Internet Connectivity Failure | | |
| Primary Actor | Operator | | |
| Dependency | Advance Feature Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The owners of the application open the Distribution operator by given access from admin | The system will show the Menu Screen |
| 02 | From Menu Screen Operator Click The Employee Recovery | Operator will navigate to Employee Recovery Screen |
| 03 | He Can add the loan recovery | System will update the recovery Database |
|  | 04 |  |  |
| Alternative Flow | Errors | Action | Response |
| None | None | None |

Use Case Description Table 15: Employee Recovery

|  |  |  |  |
| --- | --- | --- | --- |
| Daily Expenses & Closing Book Case Description Table | | | |
| Name | Daily Expenses Record Book | | |
| ID | UC-16 | | |
| Objective | Daily Expenses and Closings Module allows recording and monitoring of distribution-related expenses, including bills, vehicle expenses, and miscellaneous costs, along with expenditures for salespeople, employees, and investors. | | |
| Brief Description | Daily Expenses and Closings Module is a vital component of financial management, enabling businesses to efficiently record and manage various expenses incurred during distribution operations. From bills and vehicle expenses to miscellaneous costs, the module comprehensively captures all expenditure details. | | |
| Pre-Condition | The owners must login into the admin panel. | | |
| Post-Condition | The required action is performed successfully. | | |
| Failed Condition | Internet Connectivity Failure | | |
| Primary Actor | Operator | | |
| Dependency | Advance Feature Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The owners of the application open the Distribution operator by given access from admin | The system will show the Menu Screen |
| 02 | From Menu Screen open Daily Expenses and Closing | Operator will navigate to Daily Expenses and Closing Page |
| 03 | He Can Add Expenses here | System will Updated Expenses Records |
|  | 04 |  |  |
| Alternative Flow | Errors | Action | Response |
| None | None | None |

Use Case Description Table 14: Daily Expenses and Closing

|  |  |  |  |
| --- | --- | --- | --- |
| Vehicle & Miscellaneous Expenses Case Description Table | | | |
| Name | Vehicle & Miscellaneous Expenses | | |
| ID | UC-16 | | |
| Objective | There Are Two Seprate Options with Same Functionality : The objective of these section is to maintain a comprehensive record of all vehicle-related expenses, enabling efficient tracking of monthly expenditures and providing relevant filters for data analysis. | | |
| Brief Description | Vehicle and Miscellaneous Expenses work as a centralized system for recording and managing all expenses related to vehicles. It meticulously logs various vehicle-related costs, including fuel, maintenance, repairs, insurance, and other related expenditures. | | |
| Pre-Condition | The owners must login into the admin panel. | | |
| Post-Condition | The required action is performed successfully. | | |
| Failed Condition | Internet Connectivity Failure | | |
| Primary Actor | Operator | | |
| Dependency | Advance Feature Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | application open the Distribution operator by given access from admin | The system will show the Menu Screen |
| 02 | From Menu Screen open (Vehicle or Mislns) Expeneses Section | Operator will navigate to Selected Expenses Page |
| 03 | He Can View All of Expenses as well he can add | System will Updated Expenses Records |
|  | 04 |  |  |
| Alternative Flow | Errors | Action | Response |
| None | None | None |

Use Case Description Table 16: Vehicle and Miscellaneous Expenses

|  |  |  |  |
| --- | --- | --- | --- |
| Credit / Debit Book Case Description Table | | | |
| Name | Credit / Debit Record Book | | |
| ID | UC-17 | | |
| Objective | Company Credit and Debit module facilitates the seamless addition and management of both debit transactions, including incentives and discounts, and credit transactions, such as uniform expenses and other relevant items, providing a comprehensive financial tracking system for the organization. | | |
| Brief Description | Company Credit and Debit module serves as a vital component of the organization's financial management, offering a user-friendly platform to record and monitor both debit and credit transactions. For debit transactions, it allows the easy addition of incentives, discounts, or any other deductions, providing clarity and transparency in financial dealings. | | |
| Pre-Condition | The owners must login into the admin panel. | | |
| Post-Condition | The required action is performed successfully. | | |
| Failed Condition | Internet Connectivity Failure | | |
| Primary Actor | Operator | | |
| Dependency | Advance Feature Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The owners of the application open the Distribution operator by given access from admin | The system will show the Menu Screen |
| 02 | From Menu Screen open Credit / Debit section | Operator will navigate to Credit / Debit page |
| 03 | He Can Add Credit / Debit here | System will Updated Credit / Debit there. |
|  | 04 |  |  |
| Alternative Flow | Errors | Action | Response |
| None | None | None |

Use Case Description Table 17: Credit / Debit

|  |  |  |  |
| --- | --- | --- | --- |
| Tax / Bills Case Description Table | | | |
| Name | Tax / Bills Record Book | | |
| ID | UC-18 | | |
| Objective | The objective of the Tax and Bill Management Section is to efficiently maintain records of monthly tax payments and closing balances while providing the ability to filter data by specific months. | | |
| Brief Description | The Tax and Bill Management Section serves as a comprehensive repository for recording monthly tax payments and closing balances. This efficient system allows for easy data filtering based on specific months, facilitating quick access to tax-related information. Additionally, the section handles the management of various distribution-related bills, meticulously maintaining records for bills received from service providers like PTCL, WAPDA, and gas companies. | | |
| Pre-Condition | The owners must login into the admin panel. | | |
| Post-Condition | The required action is performed successfully. | | |
| Failed Condition | Internet Connectivity Failure | | |
| Primary Actor | Operator | | |
| Dependency | Advance Feature Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The owners of the application open the Distribution operator by given access from admin | The system will show the Menu Screen |
| 02 | From Menu Screen open Tax / Bill Section | Operator will navigate to Selected Section |
| 03 | He Can Add Check his Records there | System will Show The Records |
|  | 04 |  |  |
| Alternative Flow | Errors | Action | Response |
| None | None | None |

Use Case Description Table 18: Tax / Bills

|  |  |  |  |
| --- | --- | --- | --- |
| Rate Managment Case Description Table | | | |
| Name | Rate Management | | |
| ID | UC-19 | | |
| Objective | In the rate manager module, we can efficiently manage the prices of various stocks. For example, we can set rates for RGB, 500 ml, and 1500 ml stocks. These rates will be utilized for future calculations within the web portal. | | |
| Brief Description | The Rate Manager Module is a component within a web portal that facilitates the efficient management of prices for various stocks. Its primary function is to set and maintain rates for different stock items, enabling accurate calculations for future transactions and operations within the portal. The module allows administrators or authorized users to define rates for specific stock items, such as RGB (stock name), 500 ml (stock size), and 1500 ml (stock size).  With the Rate Manager Module, users can easily configure and update the rates for these stocks based on market conditions, costs, or other relevant factors. Once the rates are set, they are utilized by other parts of the web portal for performing calculations related to transactions, pricing, revenue forecasts, and other financial analyses. | | |
| Pre-Condition | The owners must login into the admin panel. | | |
| Post-Condition | The required action is performed successfully. | | |
| Failed Condition | Internet Connectivity Failure | | |
| Primary Actor | Operator | | |
| Dependency | Advance Feature Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The owners of the application open the Distribution operator by given access from admin | The system will show the Menu Screen |
| 02 | From Menu Screen Operator Click The Rate Manager  Option | Operator will navigate to Rate Managment Screen |
| 03 | He / She Can Update Rates of Stocks | System will update the stocks Rates |
| Alternative Flow | Errors | Action | Response |
| None | None | None |

Use Case Description Table 19: Stocks management

|  |  |  |  |
| --- | --- | --- | --- |
| Balance Sheet Case Description Table | | | |
| Name | Balance Sheet | | |
| ID | UC-20 | | |
| Objective | The balance sheet module allows us to generate comprehensive balance sheets that include records of both debit and credit transactions. | | |
| Brief Description | The Balance Sheet Module is a powerful tool that enables the generation of detailed and comprehensive balance sheets within our system. It serves as a central repository for recording and organizing all debit and credit transactions, presenting a holistic view of the company's financial position.  With the Balance Sheet Module, users can efficiently track and manage financial data. | | |
| Pre-Condition | The owners must login into the admin panel. | | |
| Post-Condition | The required action is performed successfully. | | |
| Failed Condition | Internet Connectivity Failure | | |
| Primary Actor | Operator | | |
| Dependency | Advance Feature Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The owners of the application open the Distribution operator by given access from admin | The system will show the Menu Screen |
| 02 | From Menu Screen Operator Click The Balance Sheet  Option | Operator will navigate to Balance Sheet Screen |
| 03 | He / She Can Get Details of Currently Balance in Form of Credit / Debit by Click here | Records will be Generated |
| Alternative Flow | Errors | Action | Response |
| None | None | None |

Use Case Description Table 20: Balance Sheet management

|  |  |  |  |
| --- | --- | --- | --- |
| Daily Activity Case Description Table | | | |
| Name | Daily Activity | | |
| ID | UC-21 | | |
| Objective | The daily activity section provides access to all records on a daily basis. It includes the starting time of work, a detailed table of records, and the option to print out relevant information. | | |
| Brief Description | Daily Activity Section is a key component of the system that offers users convenient access to a comprehensive record of daily activities. It serves as a centralized hub for storing and displaying essential information related to daily work routines and tasks. | | |
| Pre-Condition | The owners must login into the admin panel. | | |
| Post-Condition | The required action is performed successfully. | | |
| Failed Condition | Internet Connectivity Failure | | |
| Primary Actor | Operator | | |
| Dependency | Advance Feature Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The owners of the application open the Distribution operator by given access from admin | The system will show the Menu Screen |
| 02 | From Menu Screen Operator Click The Daily Activity Option | Operator will navigate to Daily Activity Screen |
| 03 | He / She Can Get Details of All Records on Daily Basis | He / She Can Get Record on Daily Basis |
| Alternative Flow | Errors | Action | Response |
| None | None | None |

Use Case Description Table 21: Daily Activity

Summary

This chapter discussed the requirement specification of the application, which includes interface requirements (hardware and software interface requirements), database requirements, functional requirements and non-functional requirements. It also talks over use case diagram and description table of main use cases.

***Chapter No. 03***

***System Design and UML Diagrams***

* 1. **System Design**

System design is a mechanism to transform user requirements into some suitable form, which helps the programmer in software coding and implementation. It deals with representing the requirement as described in software requirement specification (SRS). Furthermore, make its activity, state machine, sequence and deployment architectures by illustrated various diagrams that discussed below in detail.

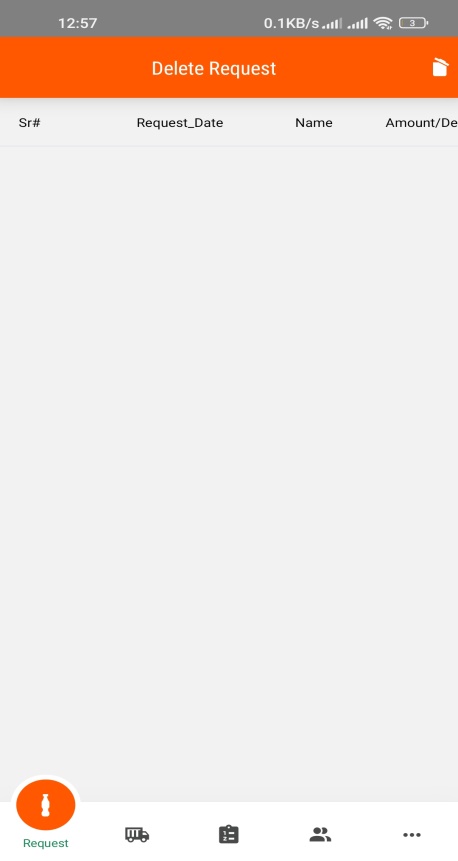
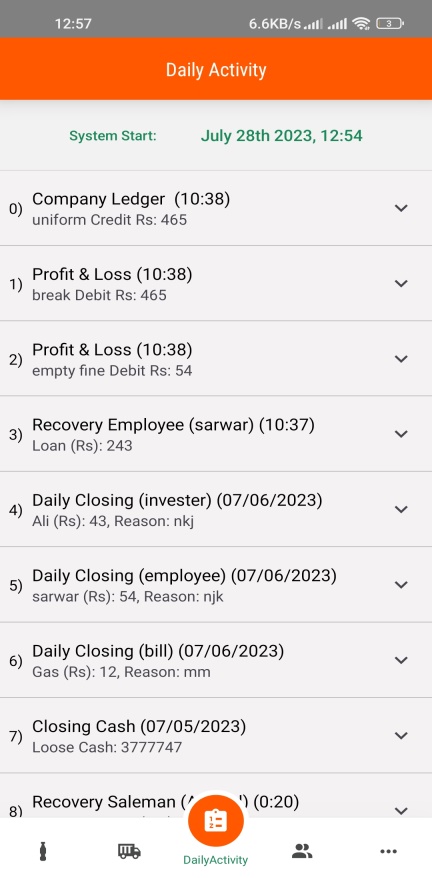
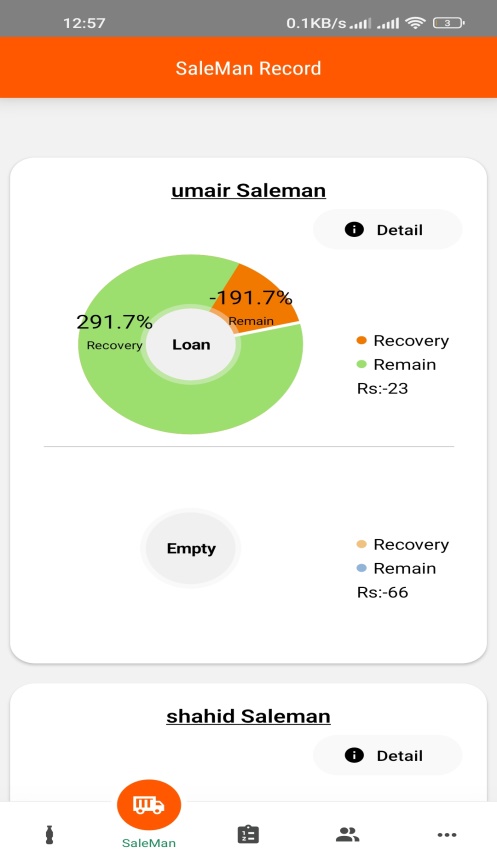
* 1. **Design Methodology**

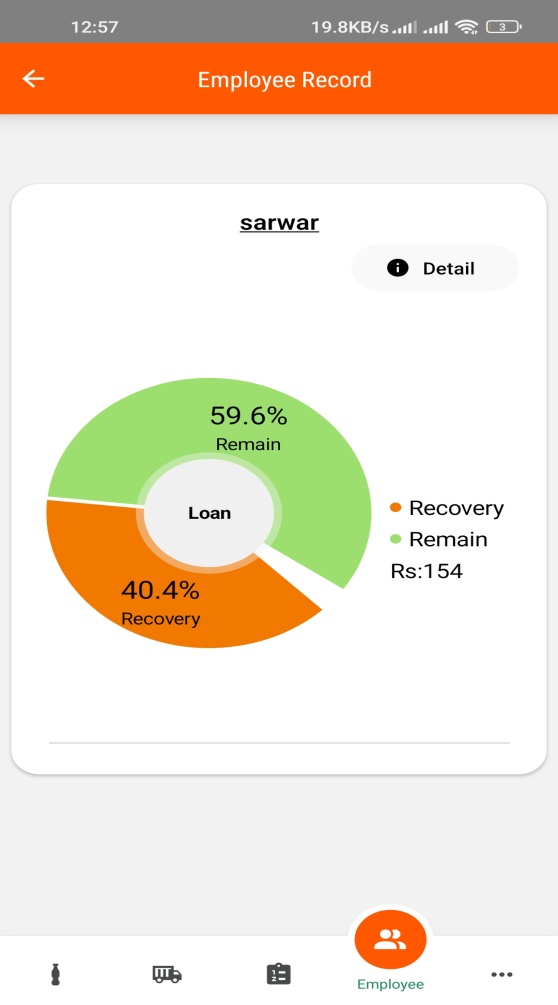
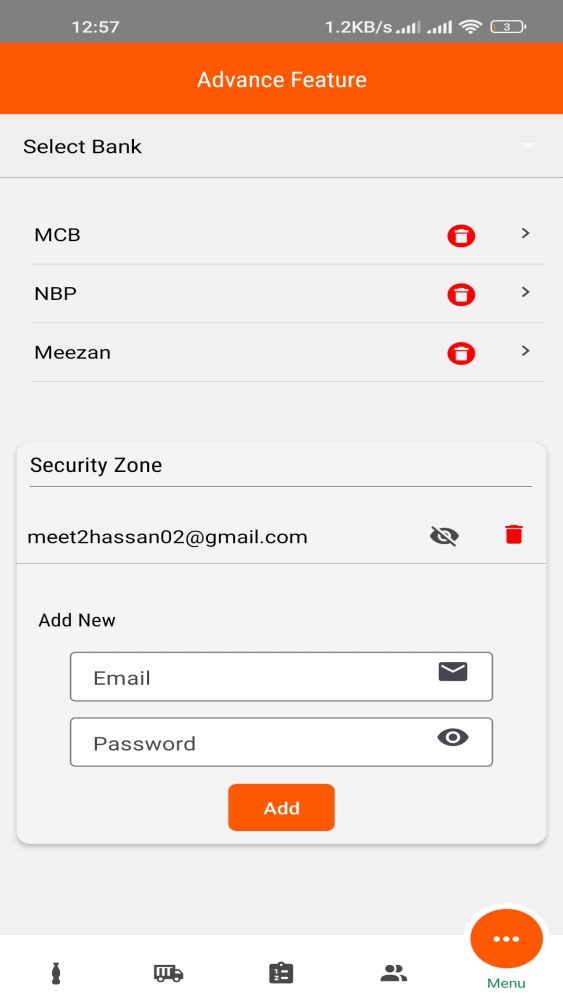
The application is designed using bottom up methodology, in which we solved smaller problems and integrate it as complete. This methodology is really suitable for Functional programming approach and as we known earlier our application is build by using React Native & React JS.

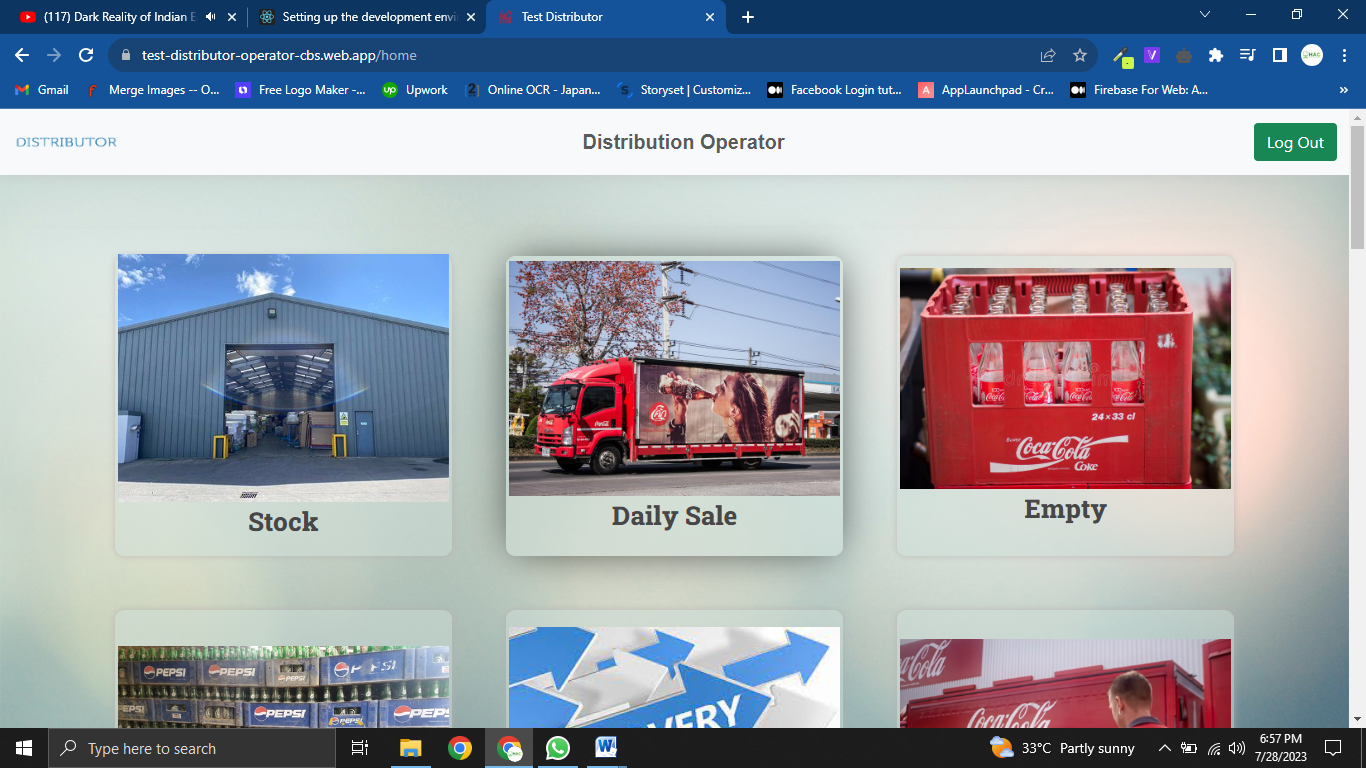
* 1. **Application Interface Design**

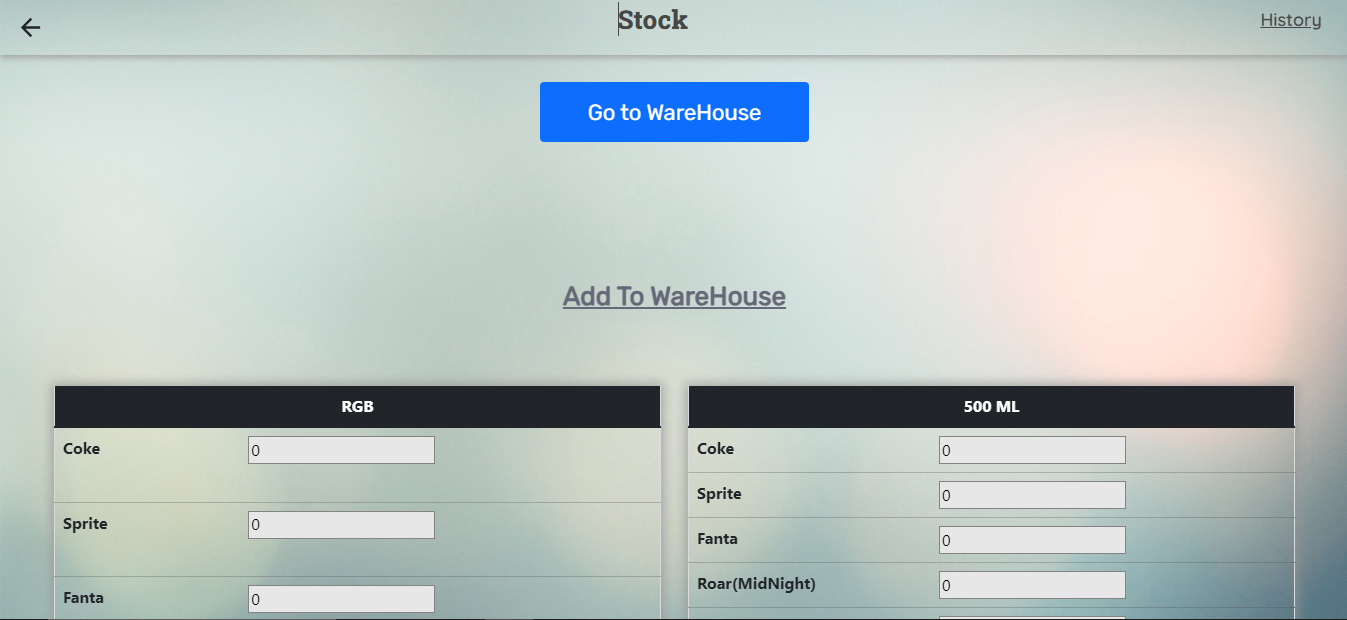
In designing, application interface is a significant segment. The interaction of the user with an application is important and a non-conformed design leave a bad impression. Taking everything in mind, we designed the interface of Distributor Operator as much interactive possible and user-friendly, a collaborative web application for interface

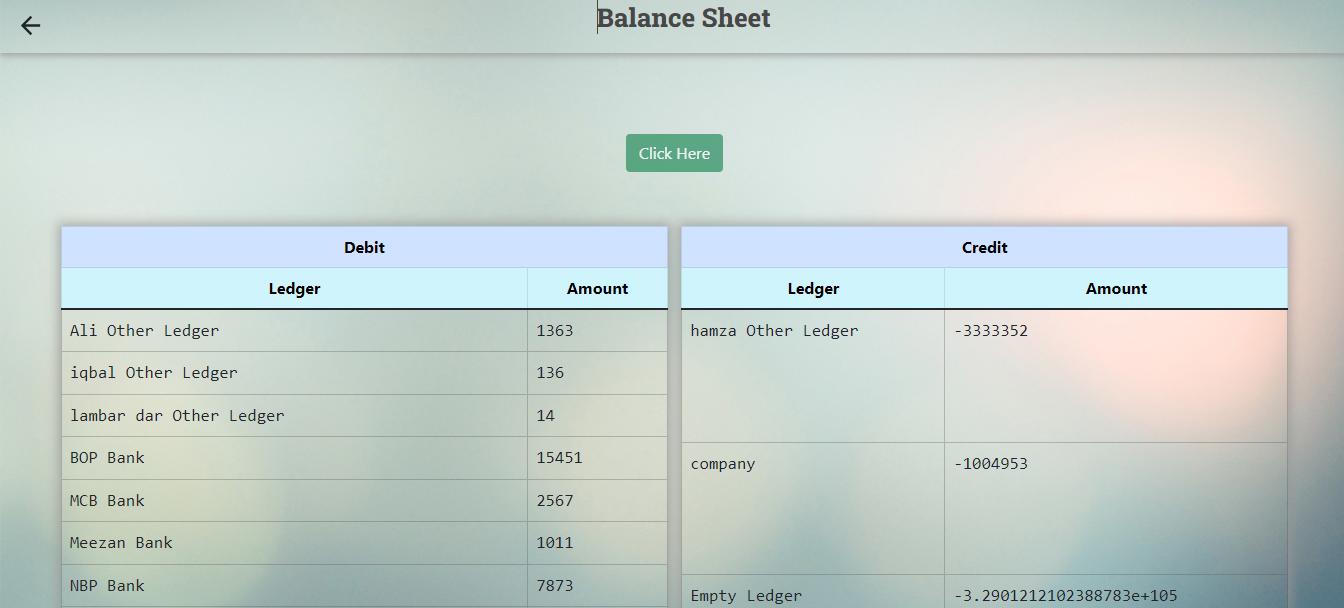
design. These are the following prototype images of the application.

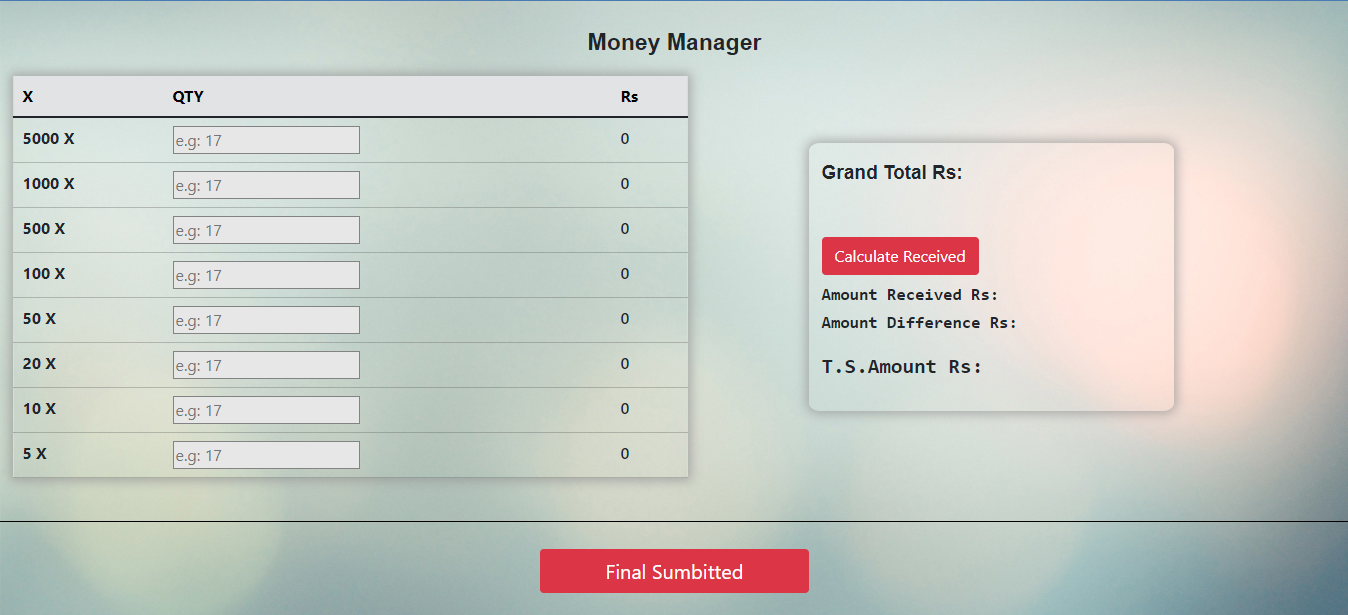
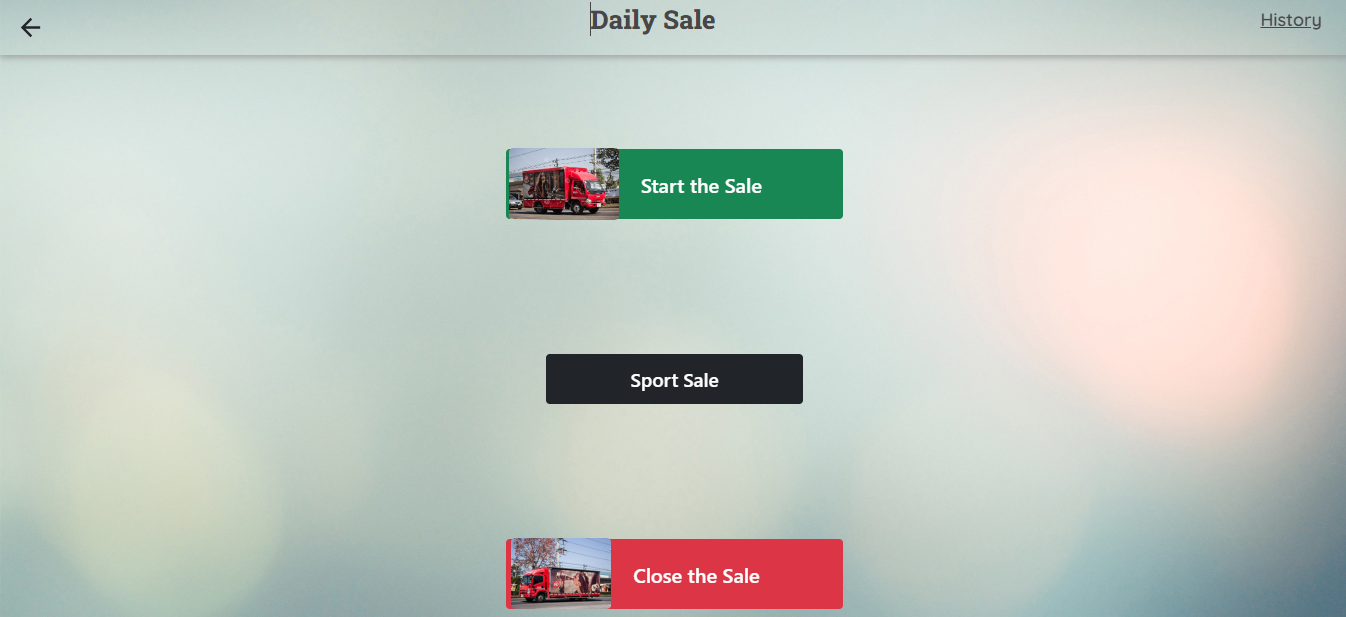


****

****

****

****

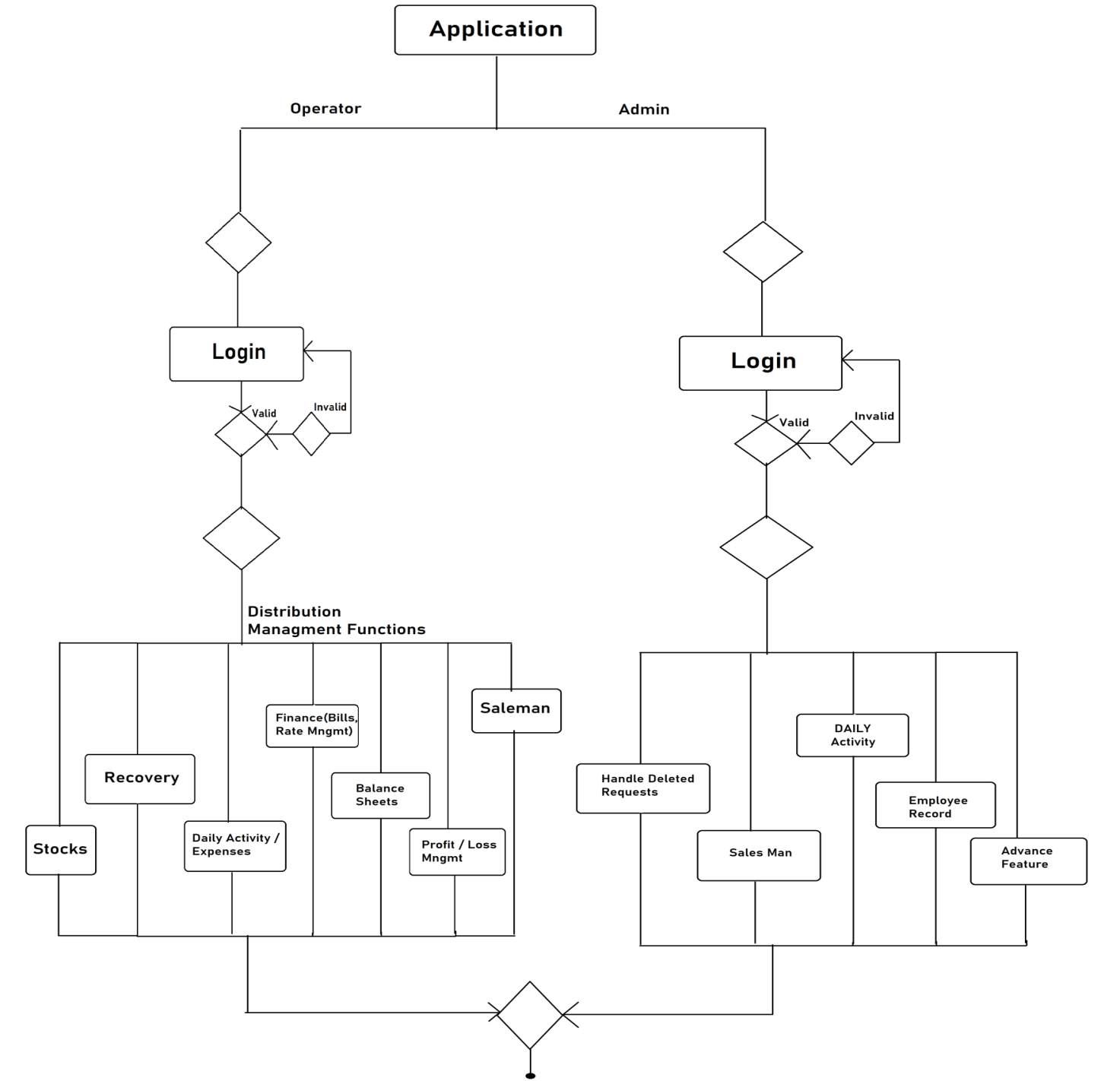
****

* 1. **Unified Modeling Language Diagrams**

These are the following unified modeling language (UML) diagrams used to visually represent the architecture design of an android based application called Distributor Operator.

* + 1. **Activity Diagram**

This is essentially a flowchart that represent the activities performed by the application. The illustration of the following diagram is given below.

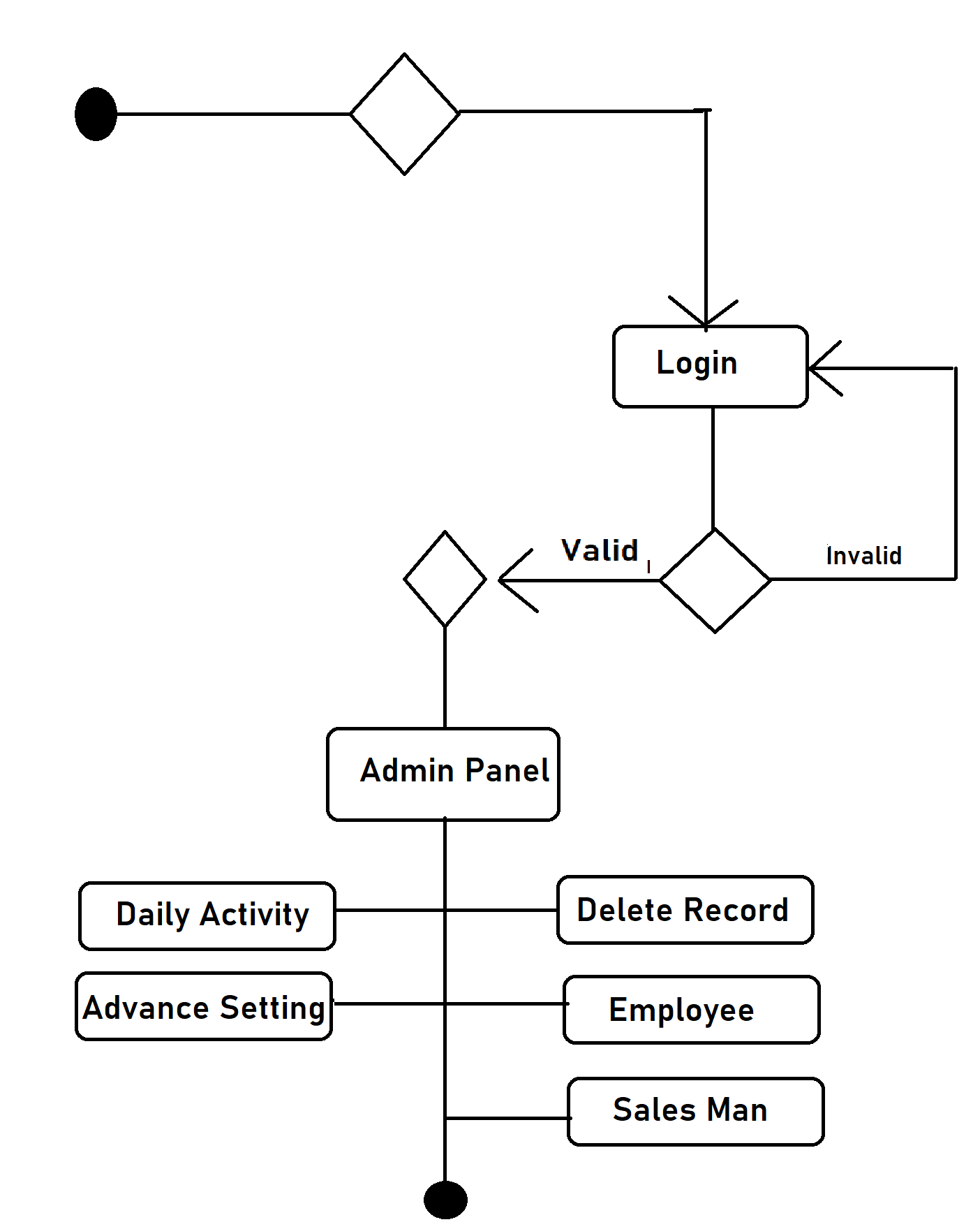


* + 1. **State Machine Diagram**

This is almost similar to the activity diagram but describes in-depth all the states and transitions for a single object. It represents application behavior upon the user and other external actions. These are the following illustrative state machine diagrams of the application.

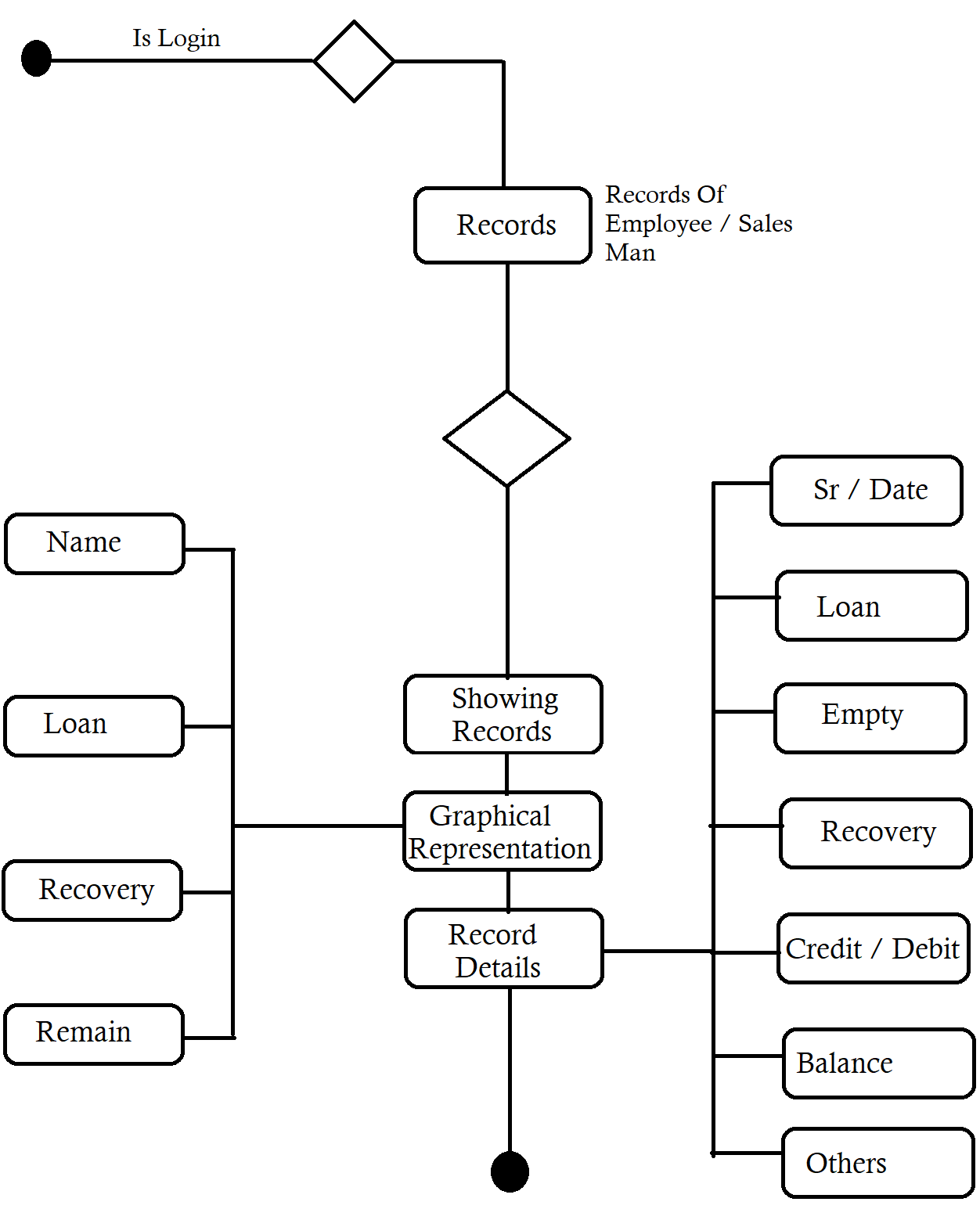
* + - 1. **Login State Machine Diagram**

Given below is the following illustrative representation of login panel state machine diagram.



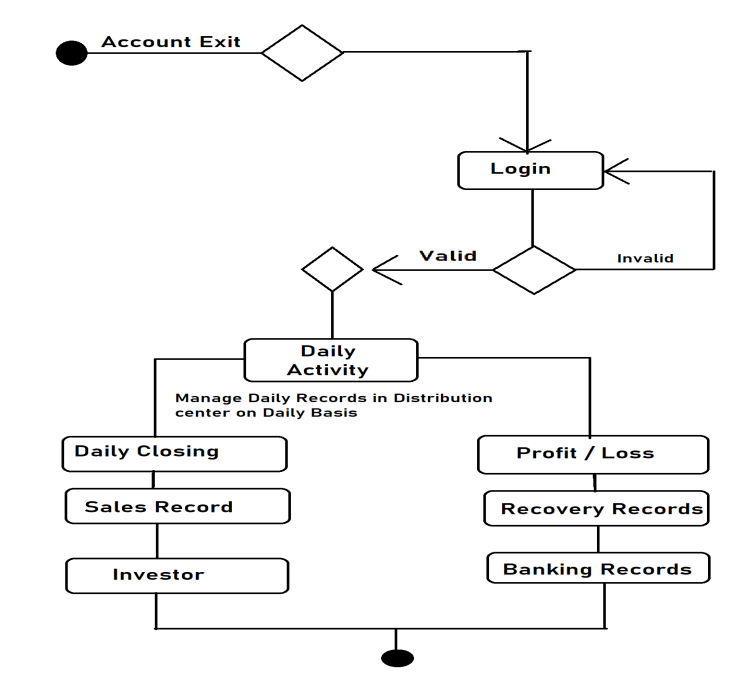
* + - 1. **Employee and Sales Man Record Diagram**

Given below is the following illustrative representation of employee and sales man record state machine diagram.



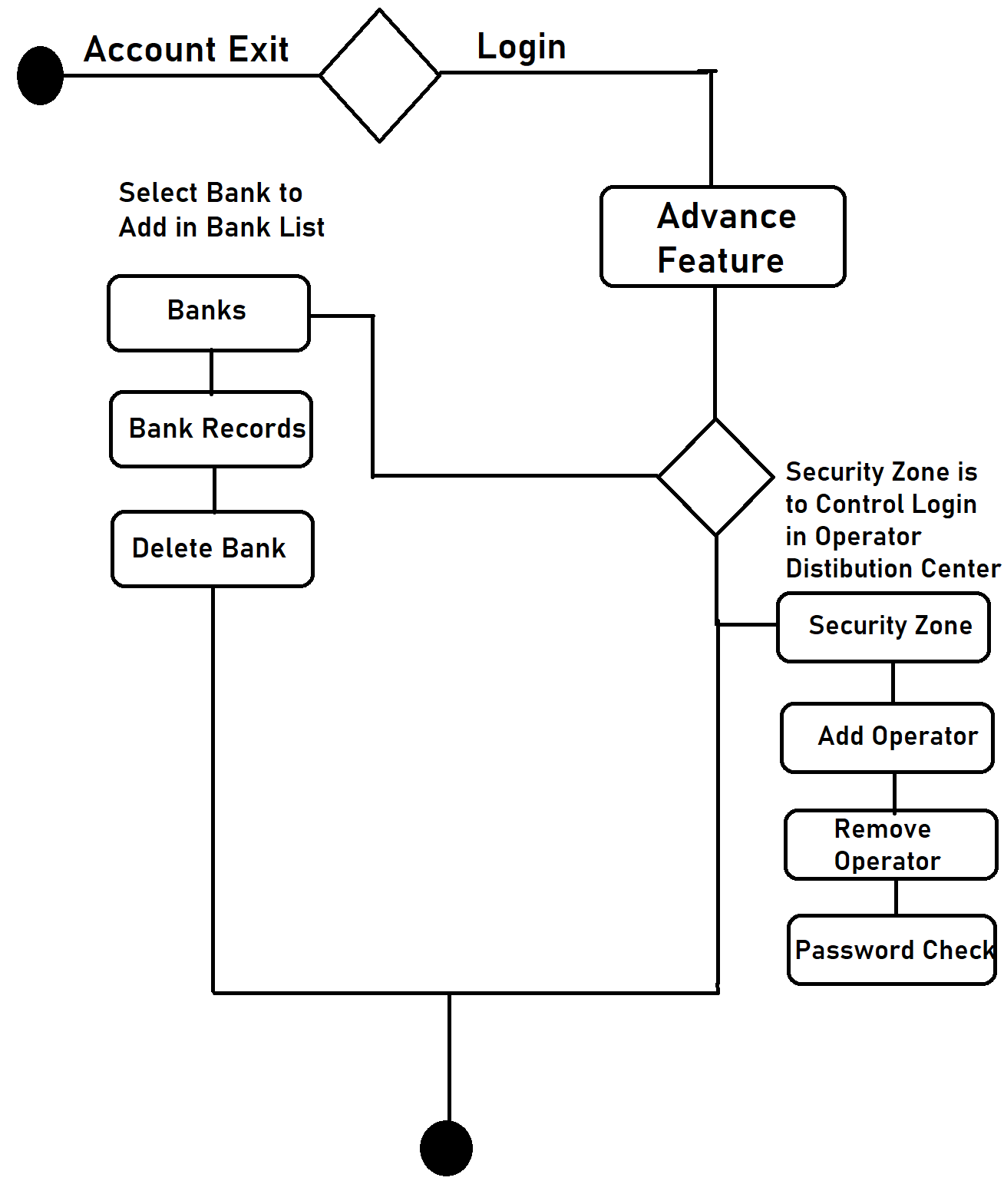
* + - 1. **Daily Activity State Machine Diagram**

Given below is the following illustrative representation of daily activity state machine diagram.



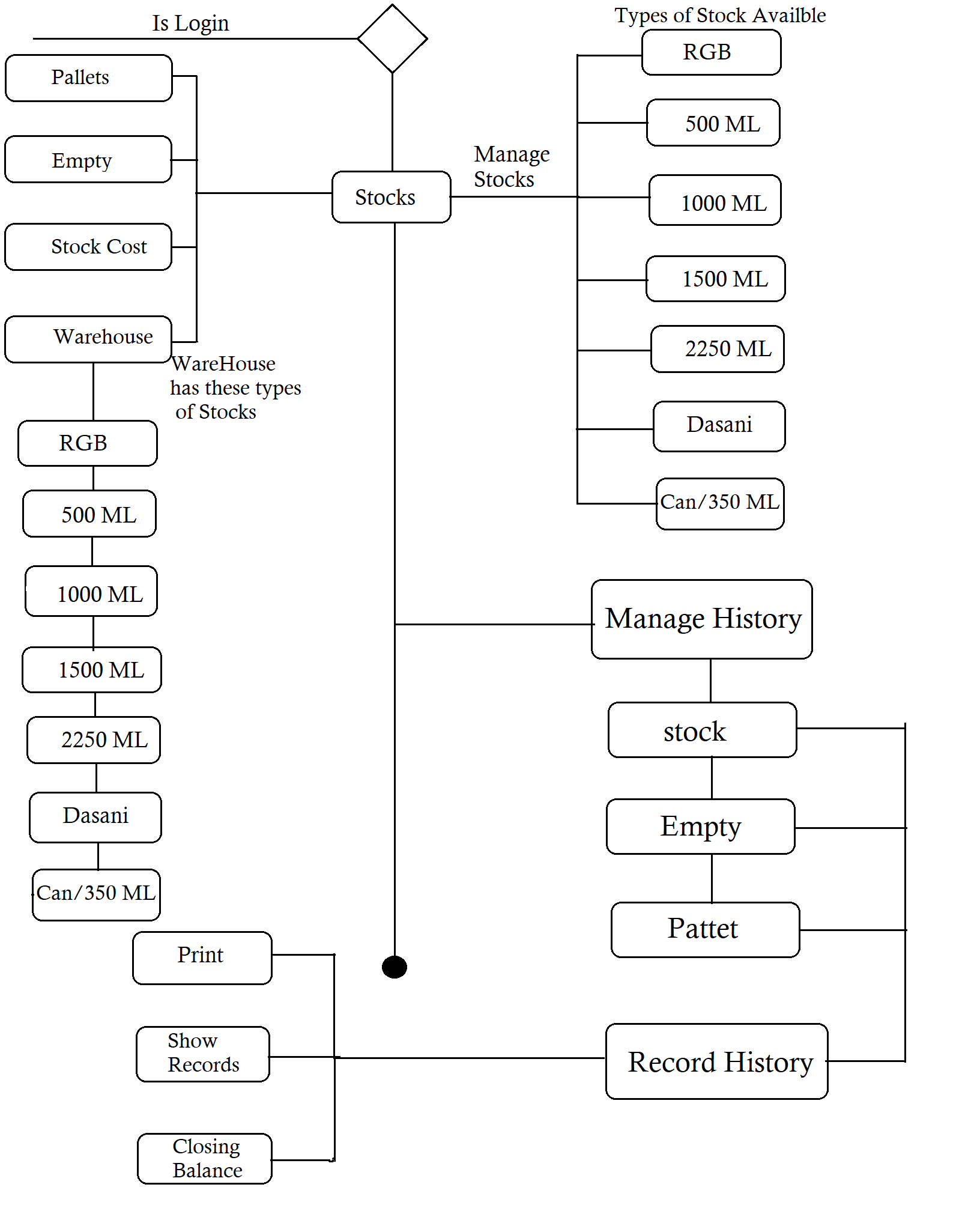
* + - 1. **Advance Feature State Machine Diagram**

Given below is the following illustrative representation of Advance Feature state machine diagram.



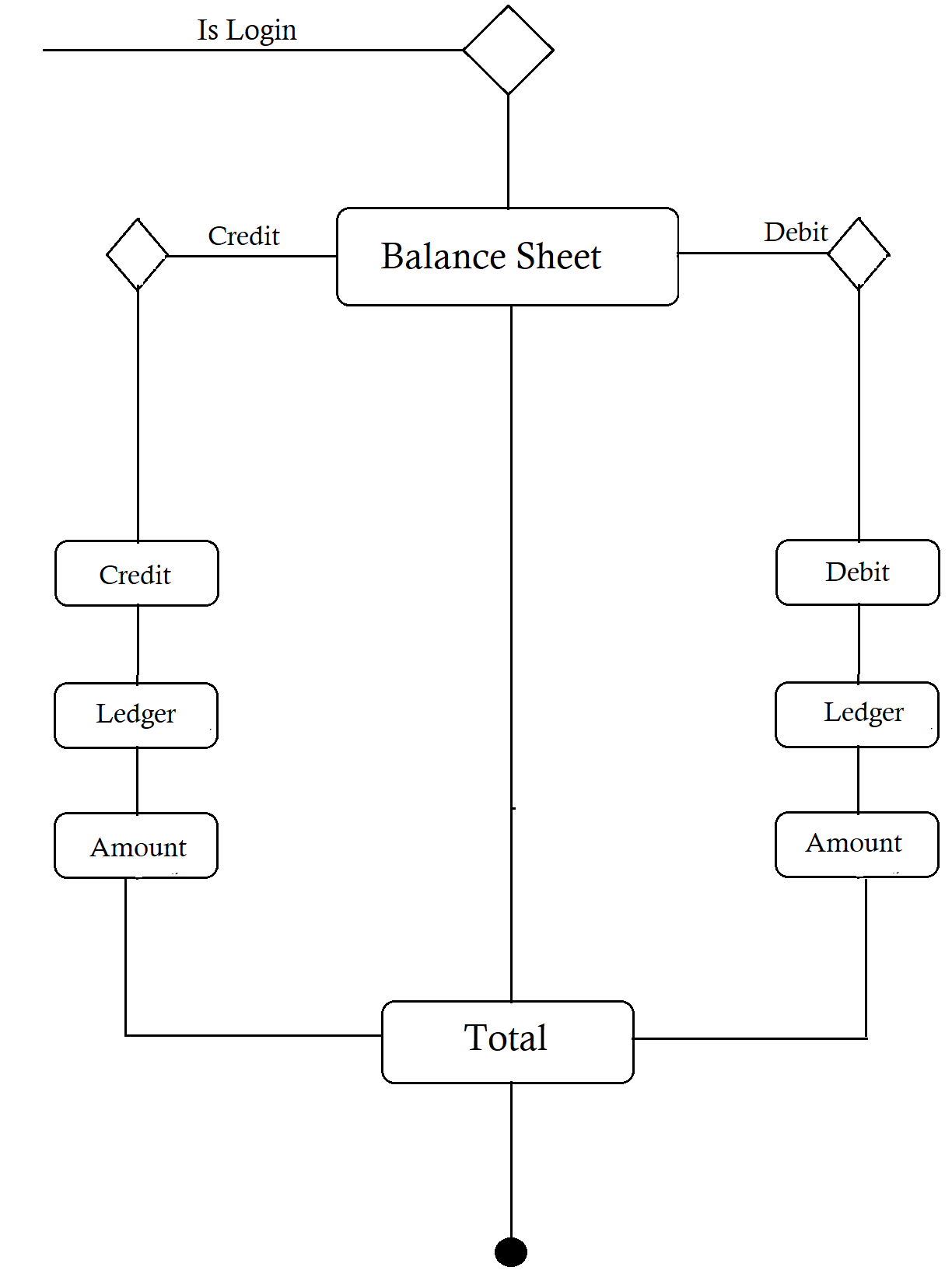
* + - 1. **Stocks State Machine Diagram**

Given below is the following illustrative representation of Stocks state machine diagram.



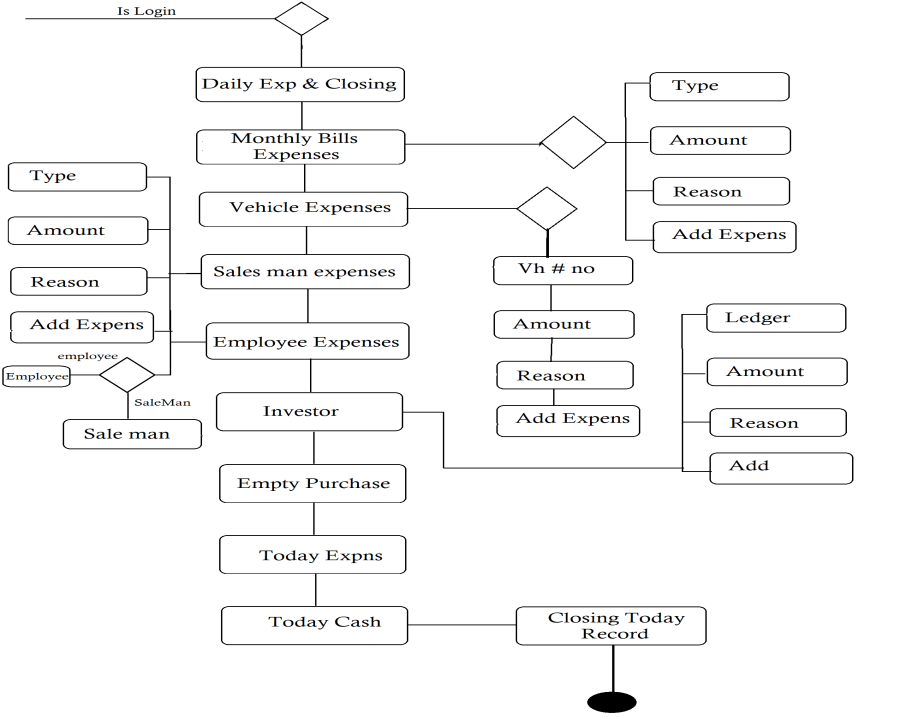
* + - 1. **Balance Sheet State Machine Diagram**

Given below is the following illustrative representation of Balance Sheet state machine diagram.



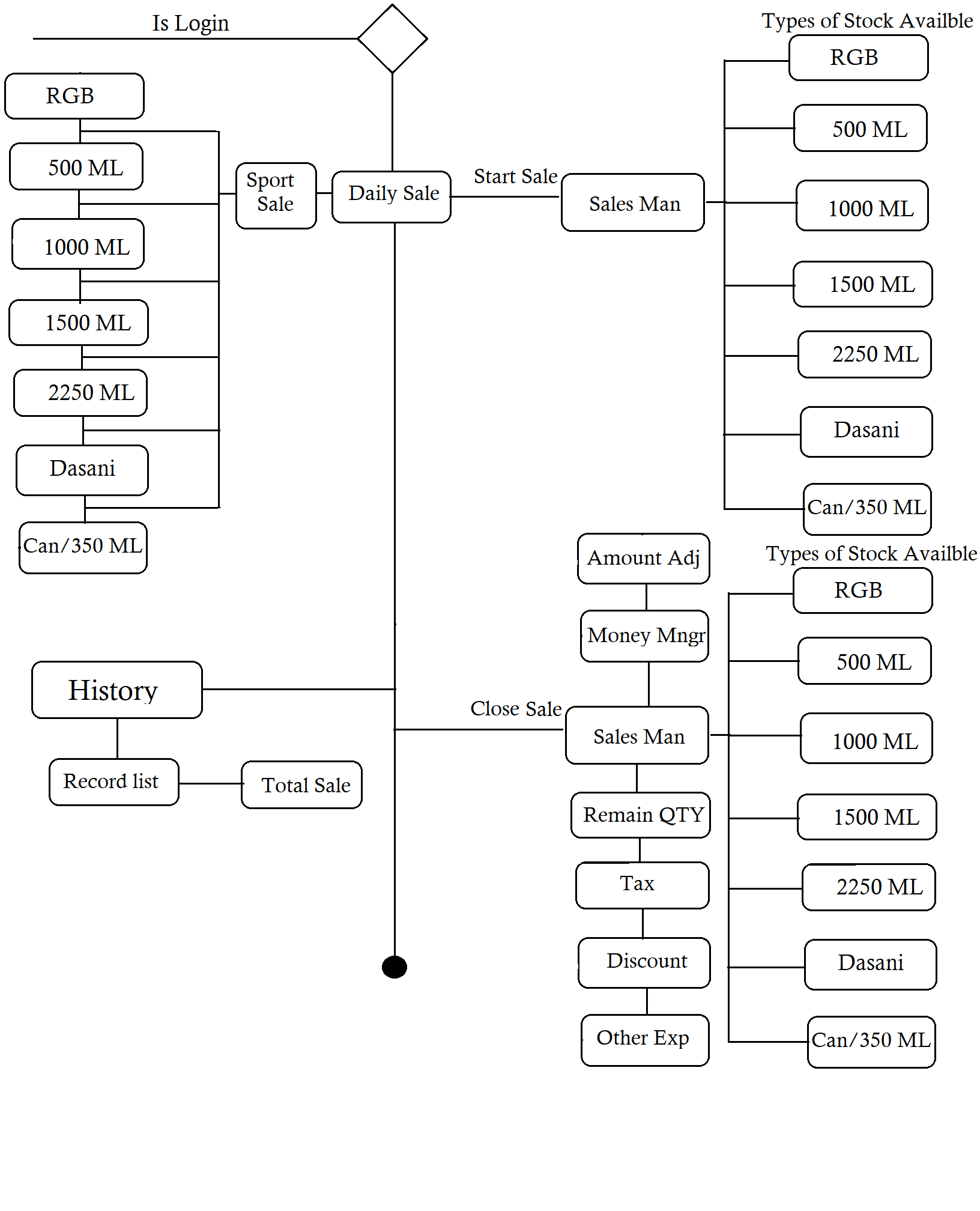
* + - 1. **Daily Expenses & Closing Service State Machine Diagram**

Given below is the following illustrative representation of Daily Expenses & Closing Services state machine diagram.



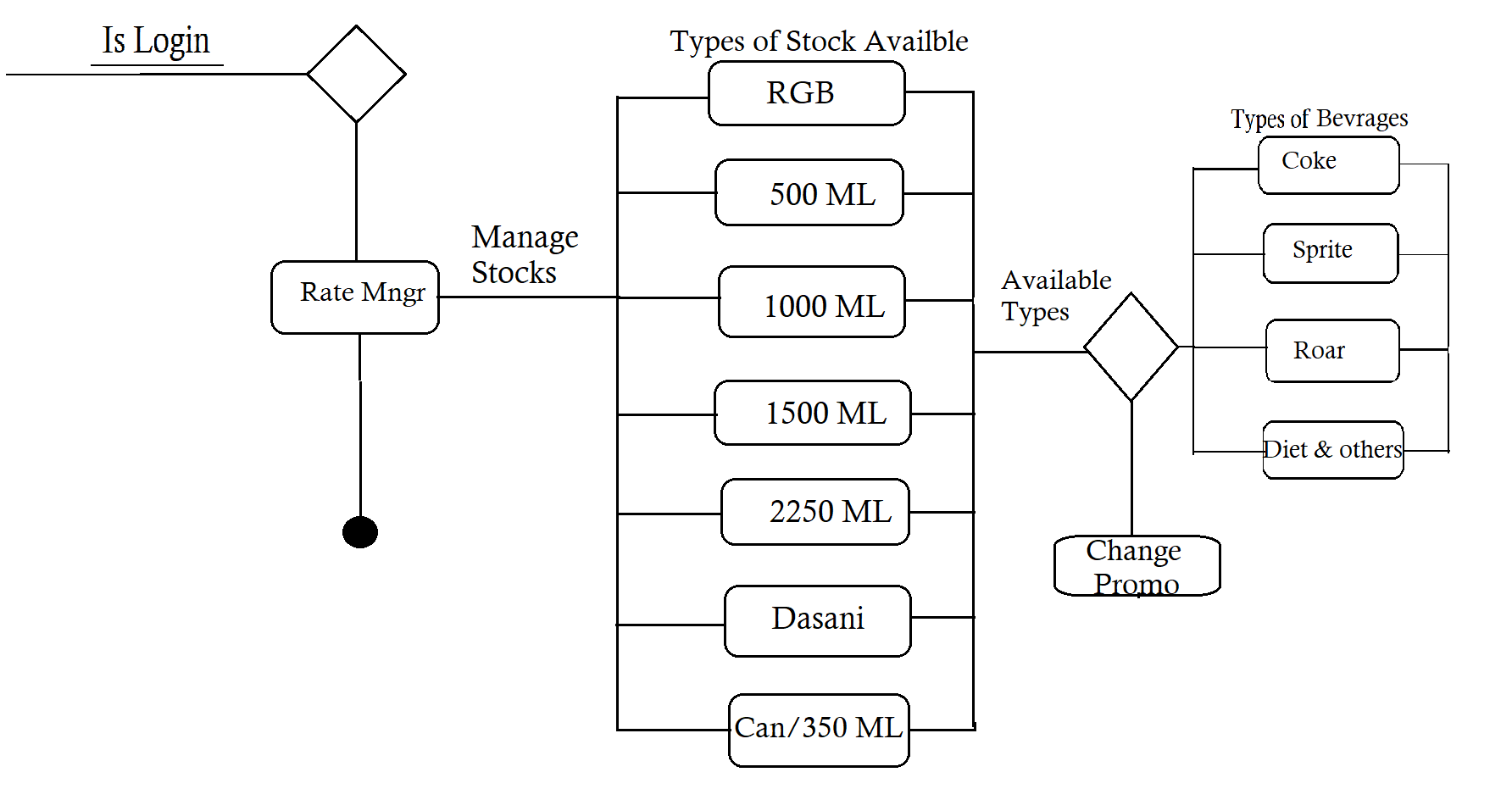
* + - 1. **Daily Sale State Machine Diagram**

Given below is the following illustrative representation of Daily Sale state machine diagram.



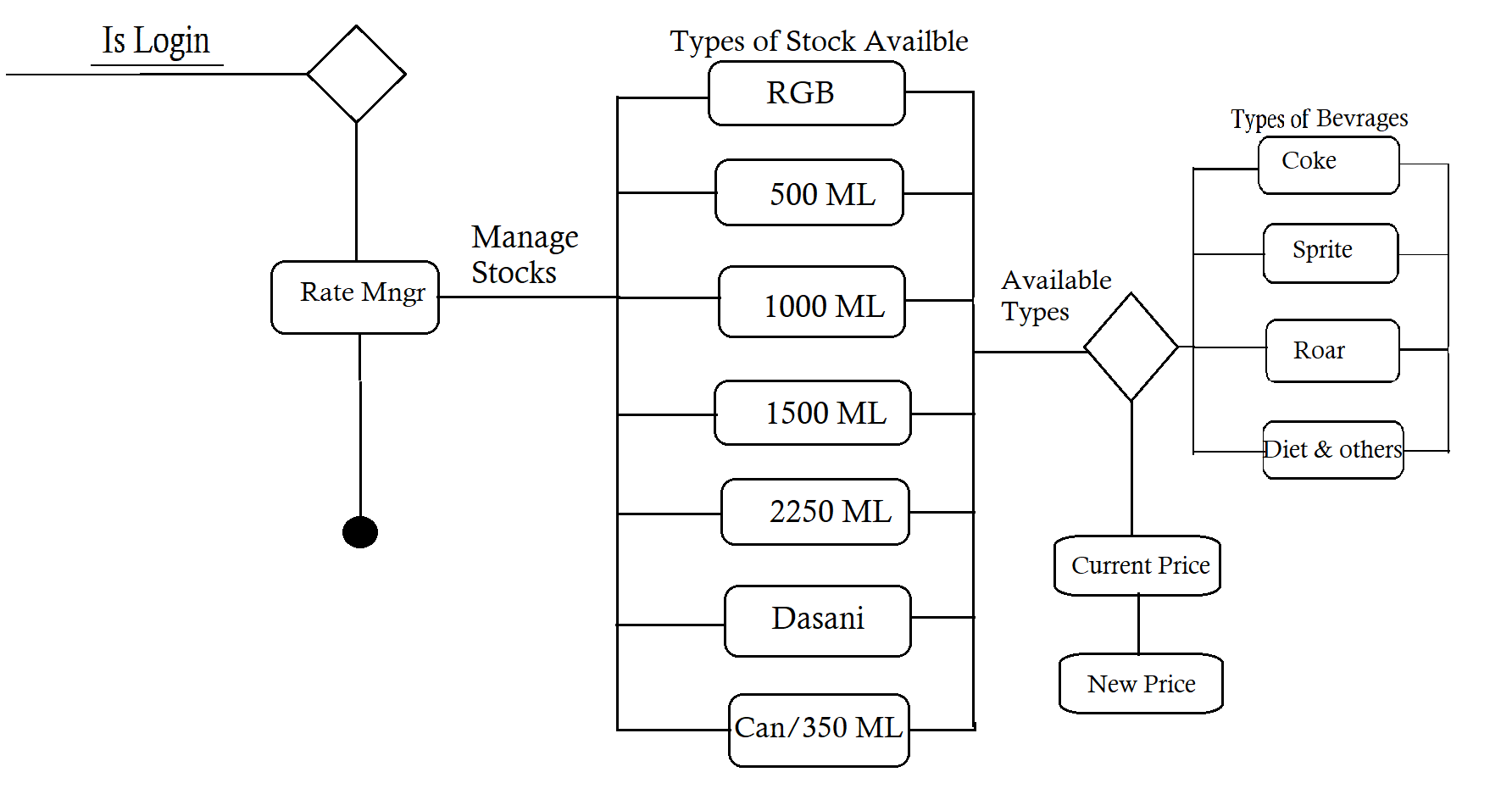
* + - 1. **Promo Manager State Machine Diagram**

Given below is the following illustrative representation of Promo Manager state machine diagram.



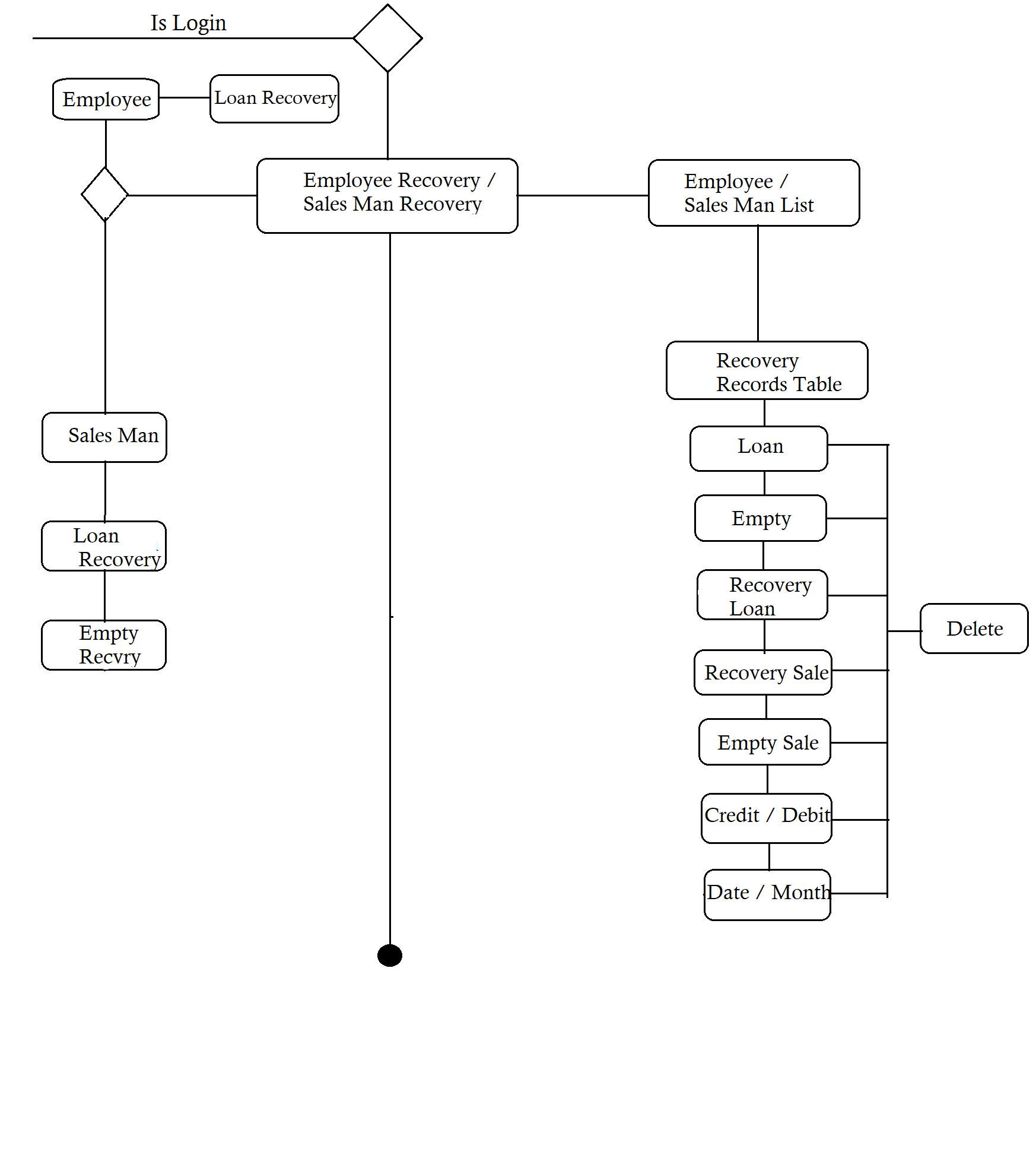
* + - 1. **Rate Manager State Machine Diagram**

Given below is the following illustrative representation of Rate Manager state machine diagram.



* + - 1. **Sales Man / Employee Recovery State Machine Diagram**

Given below is the following illustrative representation of Sales man / Employee Recovery state machine diagram.



* 1. **System Design View Perspective Models**

There are two main types of system design models in the perspective of the view. The logical design view and the physical design view. The further explanation of these view with respect to the application is discussed below.

* + 1. **Logical Design View**

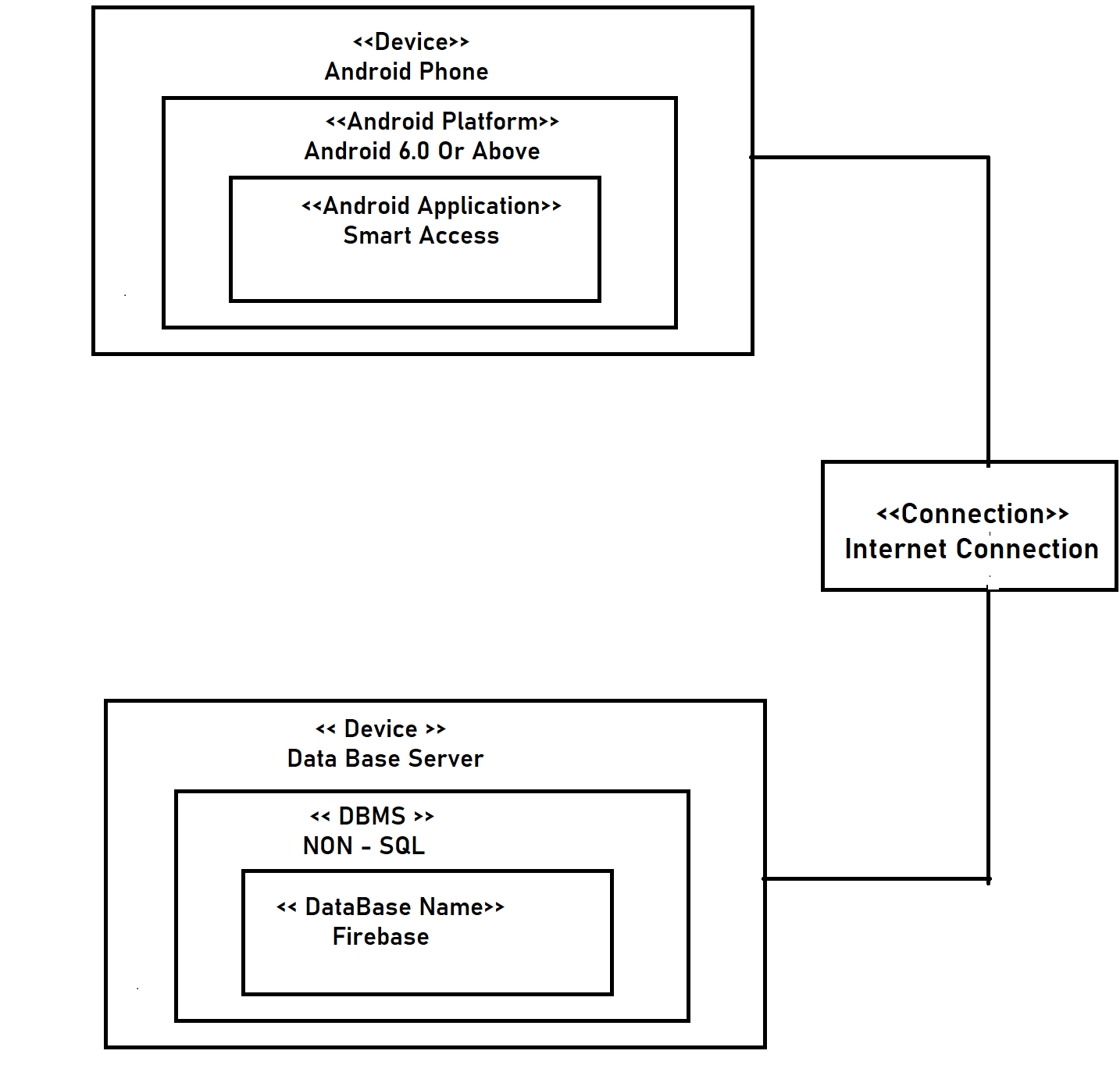
The application is used Firebase, a cloud-hosted NoSQL realtime database that stored data in the form of Javascript Object Notation (JSON). That’s why the logical design view is not possible because it consists of class and entity relationship diagrams.

* + 1. **Physical Design View**

This design view represents the end deployment of the application. It helps the engineers as a visual guide to deployed the system accordingly. In this the deployment diagram is used which shows the execution architecture with software and hardware of the application.

* + - 1. **Deployment Diagram**

Given below is the following deployment illustration of the Smart Access an android based application.



* 1. **Summary**

This chapter talks about the system design, the methodology used in the designing process, the application interface prototype and various unified modeling language diagrams like activity, state machine and sequence diagrams. Furthermore, the chapter also discussed about the system design view perspective models like logical view and physical view models with deployment diagram.

***Chapter No. 04***

***Implementation***

* 1. **Module**

This segment shows the integration of the modules with each other to execute the functionalities and workflow of the application. These are the following module wise description.

* Login Module

Login Module enables secure access to the app, with Smart Access ensuring only authorized operators can login and track records. Access is managed manually through Firebase, restricting public usage.

* Sales Man Records

Sales Man Records module displays real-time graph representations of salesperson recovery, remaining balances, loan details, empty recovery, and other relevant information.

* Employee

Employee module maintains comprehensive records of employees, including loans, recoveries, remaining balances, credit/debit details, and balance information. Records are managed and organized by dates.

* Advance Features

Advanced Features module offers additional functionalities such as deleting bank records and viewing transaction details, including date, amount, credit/debit records, and other relevant information. It also includes session security and supports fingerprint authentication for operator login and distribution management.

* 1. **Website Modules** 
     1. **Stocks Module**

Manage inventory of RGB 500 ml, 1000 ml, 1500 ml, and other variations. Submit stock orders, access comprehensive stock data, and calculate current stock levels.

* + 1. **Daily Sale**

Oversee inventory management and transactions by drivers. Ensure efficient handling of sales and returns.

* + 1. **Empty**

Manage returned empty stocks, including pallets from the company and third-party entities. Maintain real-time records in the warehouse.

* + 1. **Sales Man Recovery:**

Track individual salespeople's records, including loan recovery, balances, and deletion. Support credit and debit functionalities.

* + 1. **Sales Man (Record Book):**

Comprehensive log of credit and debit transactions for salespeople. Monitor sales performance and history.

* + 1. **Other Staff (Record Book):**

Detailed credit and debit records for all staff members. Track performance and financial **activities.**

* + 1. **Employee Recovery:**

Consolidate employee loan recovery data for effective management and analysis by month and date.

* + 1. **Daily Expenses and Closing:**

Record distribution expenses, vehicle expenses, and miscellaneous expenses. Track cash, bank transfers, and daily closing records.

* + 1. **Vehicle Expenses**

Track monthly vehicle-related expenditures with filtering options.

* + 1. **Miscellaneous Expenses**

Categorize various expenditure types like distribution, construction, stationery, and food expenses.

* + 1. **company Credit Debit**

Add debit transactions (e.g., incentives) and credit transactions (e.g., uniform expenses). Manage company finances.

* + 1. **Tax**

Maintain monthly tax s and closing balances with filtering capabilities.

* + 1. **Billls**

Manage distribution-related bills and keep records of service providers' invoices.

* + 1. **Net Profit:**

Calculate overall profit considering expenses, discounts, sales, stock levels, incentives, and net profit.

* + 1. **Rate Manager:**

Efficiently manage prices for different stocks. Set rates for future calculations.

* + 1. **Promo Manager:**

Add promotions or discounts for stock items separately.

* + 1. **Investor:**

Maintain a list of investors with their details, credit/debit information, and transaction history.

* + 1. **Balance Sheet:**

Generate comprehensive balance sheets with debit and credit records.

* + 1. **Daily Activity:**

Access daily records, including work start time, detailed table, and printable information.

* + 1. **Profit & Loss:**

Detailed overview of financial status, including profit/loss, balances, and credit/debit records.

* 1. **Framework**

Below was the given framework description that take part in the building of the application.

* + 1. **React**

React is a JavaScript library for building user interfaces. ReactJS is used for developing web applications, allowing developers to create reusable UI components and efficiently manage application state. React Native, on the other hand, is a framework for building mobile applications using JavaScript, providing a cross-platform solution with native-like performance. Both ReactJS and React Native leverage React's component-based approach for creating dynamic and interactive user interfaces.

* 1. **Library**

This was the following library description that participate in the workflow of the application.

* + 1. **React JS**

Our website, powered by ReactJS, offers a range of modules designed to streamline inventory management and financial activities. From managing stock levels and overseeing sales to tracking employee loans and generating balance sheets, our website provides comprehensive tools for efficient business operations. With a user-friendly interface and real-time data tracking, businesses can easily monitor their daily activities, track expenses, and calculate net profit. ReactJS enables us to deliver a responsive and intuitive website experience, ensuring smooth navigation and effective management of crucial business functions.

* + 1. **React Native**

Our app, built with React Native, offers a range of modules for enhanced functionality. The Login Module ensures secure access, with Smart Access and Firebase integration providing restricted usage. Sales Man Records module displays real-time graphs of salesperson metrics. The Employee module maintains comprehensive records, including loans and balances. Advanced Features module enables deletion of bank records, viewing transactions, and session security with fingerprint authentication. With React Native's cross-platform capabilities, our app delivers a seamless user experience on both iOS and Android platforms.

* 1. **Database**

This was the following database used at the back-end of the application, Distributor Operator.

* + 1. **Firebase**

This is a cloud based realtime no structure query language (NoSQL) database provided by Google, that stored data in Javascript Object Notation (JSON) format and synchronize it.This was used in the application as back-end database.

* 1. **Summary**

This chapter explained the modules that integrates with each other to functional the application. Furthermore, it discussed the library, framework, database that participate in the workflow of the application.

***Chapter No. 05***

***System Testing***

* 1. **Testing**

Essential part of the application development is testing before release it in the public. These are the following subject of testing, we used after the development of an android based application called Distributor Operator.

* + 1. **Functional Testing**

In this subject, we test the application according to does its meet the exact need, what it’s supposed to do? The application main objective and all functional requirements are fulfilled after this testing.

* + 1. **Performance Testing**

In this subject, we test the efficiency and quickness of the application. After the testing process, we conclude the overall efficiency and timely response of the application is very good.

* + 1. **Compatibility Testing**

In this subject, we test the compatibility of the application on android based smartphones currently available in the market OR for website any type of laptop Available in market. The result is positive and also work well on the application programming interface level.

* 1. **Testing Methodology**

For android based application called Distributor Operator, we used the instrumented testing methodology as post-development. According to this methodology, the application is run on an android device either physical or emulated as test application. We launched an application and interact with it for user interface testing and also testing the logical operations are working well. Furthermore, we also used unit, integration and end-to-end testing methodologies under development or sometime at post-development phases (when more features are integrated).

* 1. **Test Cases**

In software engineering, a test case is a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular [software testing](https://en.wikipedia.org/wiki/Software_testing) objective, such as to exercise a particular program path or to verify compliance with a specific requirement. [1] These are the following test cases given below to test the functional working of the application.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Operator Web Login Test Case | | | | | |
| Name | Login | | | | |
| ID | TC-01 | | | | |
| Use Case Reference | UC-01 | | | | |
| Objective | Check Login module is working properly. | | | | |
| Environment | Any Device Have internet browser | | | | |
| Assumption | The user is on Login panel. | | | | |
| Pre-Requisite | The user Can Login Through Google | | | | |
| Test Procedure | Steps | Action | Input | Expected Result | Actual Result |
| 01 | Press signup button | N/A | Input required | Input required |
| Invalid syntax | Must Have Access Of Distribution Account | Login with Google |
| Valid input | Home screen | Home screen |
| Google account already login in device | Home Screen | Home Screen |
| Status | Approved | | | | |

Table 01: Login Distribution Center Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stock Management Test Case | | | | | |
| Name | Stock Managment Service | | | | |
| ID | TC-02 | | | | |
| Use Case Reference | UC-02 | | | | |
| Objective | Check Login module is working properly. | | | | |
| Environment | Any Device with Web browser | | | | |
| Assumption | The user is on Stock Management Screen. | | | | |
| Pre-Requisite | The user must have a registered account already. | | | | |
| Test Procedure | Steps | Action | Input | Expected Result | Actual Result |
| 01 | Open Stock Button | Click Stock Button | Stock Management open | We Can Manage Stocks Here |
| Status | Sucess | | | | |

Table 04: Stock Managment Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Daily Sale Management Test Case | | | | | |
| Name | Daily Management Service | | | | |
| ID | TC-03 | | | | |
| Use Case Reference | UC-03 | | | | |
| Objective | Check Login module is working properly. | | | | |
| Environment | Any Device with Web browser | | | | |
| Assumption | The user is on Daily Sale Management Screen. | | | | |
| Pre-Requisite | The user must have a registered account already. | | | | |
| Test Procedure | Steps | Action | Input | Expected Result | Actual Result |
| 01 | Open Daily Sale Button | Click Daily Sale Button | Daily Sale open | We Can Manage Daily Sale (Start , close & End Sale) Here |
| Status | Sucess | | | | |

Table 03: Daily Sale Management Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sale Man / Employee Recovery Test Case | | | | | |
| Name | Sales Man / Employee Recovery Service | | | | |
| ID | TC-04 | | | | |
| Use Case Reference | UC-04 | | | | |
| Objective | Check Login module is working properly. | | | | |
| Environment | Any Device with Web browser | | | | |
| Assumption | The user is on Sales Man / Employee Recovery Screen. | | | | |
| Pre-Requisite | The user must have a registered account already. | | | | |
| Test Procedure | Steps | Action | Input | Expected Result | Actual Result |
| 01 | Open Sales Man / Employee Recovery Button | Click Sales Man / Employee Recovery Button | Daily Man / Employee Recovery Open | We Can Manage Recovery Records Here |
| Status | Sucess | | | | |

Table 03: Daily Sale Management Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Daily Expenses & Closing / Vehicle Expenses / Miscellaneous Expenses Test Case | | | | | |
| Name | Daily Expenses & Closing / Vehicle Expenses / Miscellaneous Expenses Service | | | | |
| ID | TC-06 | | | | |
| Use Case Reference | UC-06 | | | | |
| Objective | Check Login module is working properly. | | | | |
| Environment | Any Device with Web browser | | | | |
| Assumption | The user is on Daily Expenses & Closing / Vehicle Expenses / Miscellaneous Expenses Screen. | | | | |
| Pre-Requisite | The user must have a registered account already. | | | | |
| Test Procedure | Steps | Action | Input | Expected Result | Actual Result |
| 01 | Open Daily Expenses & Closing / Vehicle Expenses / Miscellaneous Expenses Button | Click Daily Expenses & Closing / Vehicle Expenses / Miscellaneous Expenses Button | Daily Expenses & Closing / Vehicle Expenses / Miscellaneous Expenses Opened | We Can Manage Expenses Here |
| Status | Sucess | | | | |

Table 06: Daily Expenses & Closing / Vehicle Expenses / Miscellaneous Expenses Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rate Management Test Case | | | | | |
| Name | Rate Management Service | | | | |
| ID | TC-07 | | | | |
| Use Case Reference | UC-07 | | | | |
| Objective | Check Login module is working properly. | | | | |
| Environment | Any Device with Web browser | | | | |
| Assumption | The user is on Daily Expenses & Closing / Vehicle Expenses / Miscellaneous Expenses Screen. | | | | |
| Pre-Requisite | The user must have a registered account already. | | | | |
| Test Procedure | Steps | Action | Input | Expected Result | Actual Result |
| 01 | Open Rate Management Button | Click Rate Management Button | Rate Management Open | Manage Rates According to out Available Stocks |
| Status | Sucess | | | | |

Table 07: Rate Management Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Daily Activity Test Case | | | | | |
| Name | Daily Activity Service | | | | |
| ID | TC-08 | | | | |
| Use Case Reference | UC-08 | | | | |
| Objective | Check Login module is working properly. | | | | |
| Environment | Any Device with Web browser | | | | |
| Assumption | The user is on Daily Activity Screen. | | | | |
| Pre-Requisite | The user must have a registered account already. | | | | |
| Test Procedure | Steps | Action | Input | Expected Result | Actual Result |
| 01 | Open Daily Activity Button | Click Daily Activity Button | Daily Activity Open | We Can View Daily Activity here |
| Status | Sucess | | | | |

Table 08: Daily Activity Management Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Profit & Loss Test Case | | | | | |
| Name | Profit & Loss Service | | | | |
| ID | TC-09 | | | | |
| Use Case Reference | UC-09 | | | | |
| Objective | Check Login module is working properly. | | | | |
| Environment | Any Device with Web browser | | | | |
| Assumption | The user is on Profit / Loss Screen. | | | | |
| Pre-Requisite | The user must have a registered account already. | | | | |
| Test Procedure | Steps | Action | Input | Expected Result | Actual Result |
| 01 | Open Profit / Loss Button | Click Profit / Loss Button | Profit / Loss Open | We Can View Profit / Loss here |
| Status | Sucess | | | | |

Table 09: Profit / Loss Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Employee Record / Sales Man Record Test Case | | | | | |
| Name | Employee Record / Sales Man Record | | | | |
| ID | TC-10 | | | | |
| Use Case Reference | UC-10 | | | | |
| Objective | Check Login module is working properly. | | | | |
| Environment | Android App is Opened | | | | |
| Assumption | The user is Login in Android app | | | | |
| Pre-Requisite | The user must have a Valid account already. | | | | |
| Test Procedure | Steps | Action | Input | Expected Result | Actual Result |
| 01 | Open Employee Record / Sales Man Records | Click Employee Record / Sales Man Records Button | Employee Record / Sales Man Records Open | We Can View Graphical Representation of Recovery / Loan here |
| Status | Sucess | | | | |

Table 10: Employee Record / Sales Man Record Test Case

Summary

This chapter discussed the testing with its subjects as well as testing methodologies used in the testing of an application. Furthermore, we talk about the test cases and portray each of the test case in the form of table according to the functionality

***Chapter No. 06***

***Conclusion and Future Work***

* 1. **Application Overview**

In industry of today, efficient management of distribution centers is essential for ensuring a smooth flow of products and services. However, the current manual systems used in these centers often lead to mistakes in inventory management, delays in order processing, and a lack of real-time monitoring. To address these challenges and revolutionize distribution center management, our Distributor Operator application provides an automated and optimized solution. By utilizing advanced technologies such as React Native and React JS, our app simplifies operations through features like real-time inventory tracking, optimized order processing, and comprehensive financial management. It offers a user-friendly interface and intuitive features, empowering businesses to make informed decisions based on up-to-date data and insights. Moreover, Distributor Operator enables efficient employee tracking and allows for customizable pricing strategies, supporting dynamic marketing campaigns. With a cloud-hosted Firebase backend, the app ensures seamless database operations. By employing an iterative development process model, we continually enhance the app's functionalities to meet the evolving needs of the industry.

**Milestones Achieved**

**App Milestones**

1. Login: Achieved milestone by implementing a secure login system that allows authorized operators to access the app. Distributor Operator ensures strict authentication and restricts public usage.

1. Sales Man Records: Accomplished milestone by developing a module that presents real-time graphical representations of salesperson recovery, remaining balances, loan details, empty recovery, and other relevant information.
2. Employee: Successfully achieved milestone by creating a module that manages comprehensive records of employees, including loans, recoveries, remaining balances, credit/debit details, and balance information. Records are efficiently organized by dates.
3. Advanced Features: Milestone accomplished by incorporating advanced functionalities, including the ability to delete bank records, view transaction details, ensure session security, and enable fingerprint authentication for operator login and distribution management.

**Website Milestones**

* Stocks Module: Achieved milestone by implementing a module that effectively manages inventory of RGB 500 ml, 1000 ml, 1500 ml, and other variations. The module enables users to submit stock orders, access comprehensive stock data, and calculate current stock levels.
* Daily Sale: Successfully accomplished milestone by developing a module that oversees inventory management and driver transactions. The module ensures efficient handling of sales and returns, resulting in streamlined operations.
* Empty: Accomplished milestone by creating a module that efficiently manages returned empty stocks, including pallets from the company and third-party entities. Real-time records are maintained in the warehouse, ensuring accurate tracking.
* Sales Man Recovery: Milestone achieved by implementing a module that tracks individual salespeople's records, including loan recovery, balances, and deletion. The module supports credit and debit functionalities, enabling efficient management of salesperson accounts.
* Sales Man (Record Book): Successfully achieved milestone by developing a comprehensive log of credit and debit transactions for salespeople. This module allows for effective monitoring of sales performance and provides a historical record.
* Other Staff (Record Book): Accomplished milestone by creating a module that maintains detailed credit and debit records for all staff members. This module tracks performance and financial activities, ensuring accurate record-keeping.
* Employee Recovery: Milestone achieved by developing a module that consolidates employee loan recovery data, allowing for effective management and analysis by month and date.
* Daily Expenses and Closing: Successfully accomplished milestone by creating a module that records distribution expenses, vehicle expenses, and miscellaneous expenses. The module tracks cash, bank transfers, and daily closing records for accurate financial management.
* Vehicle Expenses: Achieved milestone by implementing a module that tracks monthly vehicle-related expenditures with filtering options, providing users with a comprehensive view of expenses.
* Miscellaneous Expenses: Successfully accomplished milestone by developing a module that categorizes various types of expenditure, such as distribution, construction, stationery, and food expenses. This module helps manage and track different types of expenses accurately.
* Company Credit Debit: Milestone achieved by creating a module that allows users to add debit transactions (e.g., incentives) and credit transactions (e.g., uniform expenses). This module effectively manages company finances and tracks credit/debit activities.
* Tax: Successfully accomplished milestone by developing a module that maintains monthly tax payments and closing balances. The module provides filtering capabilities for efficient tax management.
* Bills: Achieved milestone by implementing a module that manages distribution-related bills and maintains records of service providers' invoices. This module ensures organized bill management and easy access to relevant information.
* Net Profit: Successfully accomplished milestone by creating a module that calculates overall profit considering expenses, discounts, sales, stock levels, incentives, and net profit. This module provides users with an accurate financial overview.
* Rate Manager: Milestone achieved by implementing a module that efficiently manages prices for different stocks. Users can easily set rates for future calculations, ensuring effective pricing management.
* Promo Manager: Successfully accomplished milestone by developing a module that allows users to add promotions or discounts for stock items separately. This module supports dynamic marketing campaigns and enables effective promotional management.
* Investor: Achieved milestone by creating a module that maintains a list of investors with their details, credit/debit information, and transaction history. This module ensures transparent investor management and accurate record-keeping.
* Balance Sheet: Successfully accomplished milestone by developing a module that generates comprehensive balance sheets with debit and credit records. Users can access detailed financial information for informed decision-making.
* Daily Activity: Milestone achieved by implementing a module that provides access to daily records, including work start time, detailed tables, and printable information. This module ensures efficient tracking of daily activities.
* Profit & Loss: Successfully accomplished milestone by creating a module that offers a detailed overview of financial status, including profit/loss, balances, and credit/debit records. Users can analyze financial performance and make informed decisions based on accurate data.
  1. **Limitation**

Every product has some limitations and Distributor Operator has these following given below.

* + 1. **Public Access**

As you know Our Web and App Cover Distribution Center which deal with very Secured Data. So its not available for Public Use.

* + 1. **Future Position**

The future position with respect to the application is to terminates the limitations as much as possible and must available the application on hybrid platforms with transaction system and much more.

* + 1. **Reliance On Internet Connectivity**

The app and web modules require a stable internet connection for real-time data synchronization and functionality. Poor or unreliable internet connectivity can hinder the proper functioning of the system.

* + 1. **Single-Platform**

The application presently marks the community of only android users and not available for the users of other operating systems.

* 1. **Future Position**

The future position with respect to the application is to terminates the limitations as much as possible and must available the application on hybrid platforms with Single Platform and much more.

# 6 Postmortem report

# 6.1 Description

## **6.1.1 Project Name**

DISTRIBUTOR OPERATOR

## **6.1.2 Client**

* Distribution Center Operator

## **6.1.3 Project Manager**

Supervisor **Mr.Numan-Ul-Haq** lecturer Software engineering GC University Faisalabad

## 6.1.4 Solutions Architect

Awais Ahmad 8932

## 6.1.5 Start Date

1 May 2023

## 6.1.6 Completion Date

1 August 2023

# 6.2 Project Overview

The main purpose of **Frugal Healthcare Collaboration System for Doctor and Patient** is to provide its consumer with reliable and easy appointment getting that save their time and money as well. The patient uses the system to find a doctor and view their schedule. The schedules shows information such as the doctor’s working hours, lunch breaks, holidays, vacations, unavailable appointment times, etc… The patient then uses the system to request an appointment at an available time. The doctor looks at a master view of the schedule and approves the requested appointments. An email message is sent to the patient to confirm the appointment that they requested.

* In first chapter complete overview of the system is defined and all steps reqired to develop an application are described.
* In second chapter all the management plans are discussed the model used in developing the application and all other management criteria’s.
* In third chapter the design and user interface of the system is defined.
* In fourth chapter all the implementation plans are discussed coding style structures and database structure.
* In fifth chapter quality criteria’s of the system are defined including testing and problem solving issues.

## 6.2.1 Project success criterion?

## The project success criteria is to develop and complete an application by using whom doctor and patient can connect to each other providing an application that help patient find and make appointment with the doctor in given time.

# 6.3 Performance

**Table 6.1** Table for checking performance of system by scoring the criterias

|  |  |
| --- | --- |
| Criteria | Score |
| Performance against project goals/objectives | 7 |
| Performance against planned schedule | 8 |
| Performance against quality goals | 9 |
| Performance against planned budget | 10 |
| Adherence to scope | 8 |
| Project planning | 8 |
| Resource management | 8 |
| Project management | 7 |
| Development | 10 |
| Communication | 6 |
| Project deliverable(s) | 8 |

## 6.3.1 Key Accomplishments

* + - What went right?

The appointment getting part went right and showing the booked appointment on calendar.

Send message to doctor and patient as appointment conformation by using ozeki application.

* + - What worked well?

Registering patient and doctor and hospital was easy and managing their information managing appointments.

* + - What was found to be particularly useful?

Scheduling calendar with the help of JavaScript. Code reusability coding in a sequence it was easy to understand.

* + - Project highlights
* Created admin panel
* Doctor panel
* Patient panel
* Scheduling calendar
* Creating appointment
* Conformation message sending

## **6.3.2 Key Problem** Areas

* + - What went wrong?

Find some difficulties in creating doctors schedule in was pretty difficult for me to simulate the doctors timing in different day and different hospitals.

* + - What project processes didn’t work well?

The doctors schedule doesn’t work well there are some issue I am dealing with in updating the doctors schedule.

* + - What specific processes caused problems?

Developing calendar with the help of doctor’s schedule caused a little problem

* + - What were the effects of key problems areas?

Lesser knowing about Using JavaScript

## 6.3.3 Risk Management

* + - I have used JavaScript in my project that response according to the windows environment the application fails in working if the time date or the system is not right.

## 6.3.4 Overall Project Assessment

All the development work went excellent.

## 6.3.5 Additional Comments

Have a good experience of developing an application professionally.

# 6.4 Key Lessons Learned

At the end of the project, I have learned many things and smarter than when the project started. I have experienced issues, probably several that were complete surprises, and I hopefully worked my way around them. I have learned some things simply by performing the project management practices throughout the project and watching the life-cycle processes play out. Now is the time to document those issues and workaround and to enlist my stockholders in that process as well. My goal is to be able to operate more effectively and more efficiently on future projects and also can share my experience with colleagues so that they may use this information wisely as well.

# 6.5 Post Project Tasks/Future Considerations

The application require ongoing development and maintenance there is a lot of work that is required to do in this project as

* Taking records of patient’s history.
* Paying appointment charges by some reliable means.
* Providing a mean like messenger between doctor and patient.
  1. **Summary**

This chapter talks about the overview of the application called Distributor Operator and the milestones that application achieved within the duration. Furthermore, the chapter also discussed in detail about the limitations application faced and its future position.

# 

# Summary of the project

# Distributor Operator

Main purpose of Creating application Distributor operator is a system that revolutionizes the management of cold drink distribution centers. It will cover almost all aspects of distribution center's operations. It will include features like real-time tracking of inventory, processing of orders, sales monitoring, and finances management. These functionalities will help us have better control over inventory, streamline order processing, and make informed decisions based on up-to-date information. Distributor Operator is reliable for businesses to become more efficient and provide better customer service. With its user-friendly and easy-to-use interface, Distributor Operator is ready to transform the beverages distribution industry and set standards for how distribution centers are managed.