



Sri Lanka Institute of Information Technology

## ***Ravishka Mini-Supermarket***

### **Project Report**

Information Technology Project 2022

**Project ID: WD\_B01\_ITP\_G01**

Submitted by:

1. IT20137250 – Rajapaksha R.C.P
2. IT20301972 – De Silva P.R.P
3. IT20648336 – Thilakarathne W.A.S.S
4. IT20621292 – Gunasekara P.G.A.I.P
5. IT20230692 – Arachchi S.C.S
6. IT20263362 – Alahakoon A.M.D.P.S
7. IT20645670 – Umendra T.A.H
8. IT20621056 – Jayasekara H.L

Submitted to:

Mrs. Divya Nanayakkara

27.05.2022

A handwritten signature in black ink, appearing to read "m/s".

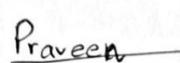
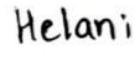
## Declaration

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

### Project Details

<b>Project Title</b>	Ravishka Mini-Supermarket
<b>Project ID</b>	WD_B01_ITP_G01

### Group Members

<b>Reg. No</b>	<b>Name</b>	<b>Signature</b>
IT20137250	Rajapaksha R.C.P	
IT20301972	De Silva P.R.P	
IT20648336	Thilakarathne W.A.S.S	
IT20621292	Gunasekara P.G.A.I.P	
IT20230692	Arachchi S.C.S	
IT20263362	Alahakoon A.M.D.P.S	
IT20645670	Umendra T.A.H	
IT20621056	Jayasekara H.L	

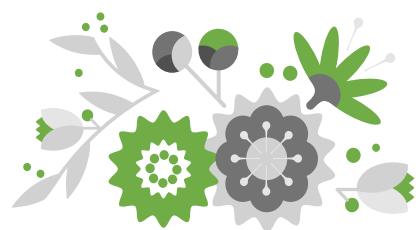
## **Abstract**

The project was developed for the 'Ravishka supermarket,' a Sri Lankan company that provides the highest quality products to thousands of Sri Lankans each week. Their business is to function as a go-between for buyers, sellers, and suppliers. As a result of the country's current conditions, this store planned to develop an online shopping platform that would allow customers to order items online and have them delivered to their homes.

We worked with them as a project team to keep their business running smoothly while adding a range of new technologies. We chose to create a web application so that all of the stakeholders' critical system components could be connected to the same network but with separate interfaces. Our client, 'Ravishka supermarket' was very pleased with our user-friendly interfaces, integrated functionalities, and SQL database which has created a manageable and securable background.

We were successful at providing the final working model, which includes the basic functions requested by the client with certain deviations from the original design, which are both advantages and disadvantages.

We met our client's highest expectations as well as this module's excellent performance. Our goal is to collaborate professionally with 'Ravishka supermarket' to further develop this web application.



## Acknowledgement

First and foremost, we would want to express our heartfelt gratitude to Ms. .... and ...., our dear professors and instructors, for helping us through this difficult work and providing us with excellent advice. We are grateful to the Sri Lankan Institute of Information Technology for allowing us to complete our first project under the supervision of the institution. We would like to give a big thanks to the panel consulting ITP project, the lecturer in charge, and other lecturers. Without your guidance, mentorship, and constant encouragement, this task would not have been a reality. We would like to thank the senior student of SLIIT who supported us by conducting additional sessions. Then, we would like to express our gratitude to the parents of all members who have actively supported us in making this project a success.

We must express our gratitude to ‘Ravishka supermarket’ for their innumerable support and for allowing us to approach them with a new solution for their supermarket. Finally, we want to express our gratitude to each of our group members who worked together in unity and camaraderie to make our plan a success and, more importantly, to show us that we are more than what we believed we were.



# Table of Contents

<b>Declaration.....</b>	<b>2</b>
<b>Abstract.....</b>	<b>3</b>
<b>Acknowledgement.....</b>	<b>4</b>
<b>Table of Contents .....</b>	<b>5</b>
<b>List of Figures.....</b>	<b>6</b>
<b>List of Tables .....</b>	<b>9</b>
<b>List of Acronyms and Abbreviations .....</b>	<b>10</b>
<b>1. Introduction.....</b>	<b>11</b>
1.1    Problem Statement .....	11
1.2    Product Scope .....	13
1.3    Project Report Structure.....	25
<b>2. Methodology .....</b>	<b>26</b>
2.1    Requirements and Analysis.....	26
2.2    Design .....	56
2.3    Implementation .....	114
2.4    Testing.....	117
<b>3. Evaluation.....</b>	<b>158</b>
3.1    Assessment of the Project results.....	158
3.2    Lessons Learned.....	158
3.3    Future Work .....	158
<b>4. Conclusion .....</b>	<b>159</b>
<b>5. References.....</b>	<b>160</b>
<b>6.Appendix C: Selected Code Listings .....</b>	<b>161</b>

## List of Figures

<b>Figure No.</b>	<b>Figure Name</b>
2.1.3.a	Employee Management Use Case Diagram
2.1.3.b	Employee Management Activity Diagram
2.1.3.c	Order Management for Wholesale Buyers Use Case Diagram
2.1.3.d	Order Management for Wholesale Buyers Activity Diagram
2.1.3.e	Transport Management Use Case Diagram
2.1.3.f	Transport Management Activity Diagram
2.1.3.g	Debtor's Management Use Case Diagram
2.1.3.h	Debtor's Management Activity Diagram
2.1.3.i	Financial Management Use Case Diagram
2.1.3.j	Financial Management Activity Diagram
2.1.3.k	Stock Management Use Case Diagram
2.1.3.l	Stock Management Activity Diagram
2.1.3.m	Supplier's management Use Case Diagram
2.1.3.n	Supplier's management Activity Diagram
2.1.3.o	Order management for retail customers Use Case Diagram
2.1.3.p	Order management for retail customers Activity Diagram
2.1.4.a	Employee Management Use Case Scenario
2.1.4.b	Transport Management Use Case Scenario
2.1.4.c	Debtors' Management Use Case Scenario
2.1.4.d	Financial Management Use Case Scenario
2.1.4.e	Order Management for Wholesale Buyers Use Case Scenario
2.1.4.f	Stock Management Use Case Scenario
2.1.4.g	Supplier's Management Use Case Scenario
2.1.4.h	Order management for retail customers Use Case Scenario
2.2.a	High Level Architecture Diagram
2.2.b	High Level Architecture Diagram
2.2.c	ER Diagram
2.2.d	Relational Schema
2.2.e	Relational Schema
2.2.1	All Services Page
2.2.2	Login Page
2.2.3	Sign Up Page
2.2.4	User Profile Page
2.2.5	User Update Page
2.2.6	User Delete Page
2.2.7	Monthly Report Page
2.2.8	Generate Salary Report Page
2.2.9	Employee Salary Detail Page
2.2.10	View Order List Page
2.2.11	Create Order Page
2.2.12	Update Order Page

2.2.13	View Delivery Details List Page
2.2.14	Add Delivery Details Page
2.2.15	Update Delivery Details Page
2.2.16	Wholesale Order Bill Receipt Page
2.2.17	Stock Management Home Page
2.2.18	Add Groceries and Frozen Item Page
2.2.19	Add Vegetable and Fruits Page
2.2.20	Groceries and Frozen Item Search Page
2.2.21	Vegetables and Fruits Search Page
2.2.22	Groceries and Frozen Items View Page
2.2.23	Vegetables and Fruits View Page
2.2.24	Groceries and Frozen Items Update Page
2.2.25	Vegetables and Fruits Update Page
2.2.26	Stock Report Page
2.2.27	Supplier Management Page
2.2.28	Add Supplier Page
2.2.29	View Supplier Page
2.2.30	Update Supplier Page
2.2.31	Add Supplier Payment Page
2.2.32	View Supplier Payment Page
2.2.33	Update Supplier Payment Page
2.2.34	Add Driver Page
2.2.35	Add Vehicle Page
2.2.36	Driver List Page
2.2.37	Driver Report List Page
2.2.38	Parcel Generation Page
2.2.39	Report Generation Page
2.2.40	Search Vehicle List Page
2.2.41	Transport Management Home Page
2.2.42	Update Driver Page
2.2.43	Update Parcel State Page
2.2.44	Vehicle List Page
2.2.45	Wholesale Buyer List Page
2.2.46	Debtors Management Home Page
2.2.47	Insert Regular Customer Details Page
2.2.48	Search Regular Customer Page
2.2.49	View Regular Customer Details and Edit Page
2.2.50	Wholesale Buyers Insert Page
2.2.51	Search Wholesale Buyers Page
2.2.52	View Wholesale Buyers Details and Edit Page
2.2.53	View Current Interest Rate Page
2.2.54	Add New Interest Rate Page
2.2.55	Debt Report Page
2.2.56	Welcome Financial Page
2.2.57	Income Statement Page

2.2.58	View Income Page
2.2.59	Update Income Page
2.2.60	View Pie Chart of Monthly Income
2.2.61	View Pie Chart of Monthly Expenses
2.2.62	Expense Statement Page
2.2.63	View Expense Page
2.2.64	Update Expense Page
2.2.65	Profit Statement Page
2.2.66	Financial Report Page
2.2.67	Welcome Page
2.2.68	Customer Registration Page
2.2.69	Customer Login Page
2.2.70	Customer Home Page
2.2.71	All Vegetables Page
2.2.72	Customer Profile Page
2.2.73	Contact Us Page
2.2.74	Frozen Items Page
2.2.75	Cart Details Page
2.2.76	More Details of a Product Page
2.2.77	Checkout Page
2.3.a	Auto Generate Tracking Number Code
2.3.b	Database Connection
2.3.c	PDF Class Code
2.3.d	Generate PDF Code
2.3.e	Pie Chart Code
6.a	Income Pie Chart
6.b	Income Pie Chart Code
6.c	Expense Pie Chart
6.d	Expense Pie Chart Code
6.e	Monthly Incomes and Expenses Bar Chart
6.f	Bar Chart Code
6.g	Monthly Net Profit Line Chart
6.h	Line Chart Code
6.i	Debtor's Plan Pie Chart
6.j	Debtor's Plan Pie Char Code
6.k	Generate PDF'S
6.l	Generate PDF'S Code

## List of Tables

Table No.	Table Name
1.2.a	Individual Scope
1.2.b	Work Distribution
2.4.1	Test Case 1
2.4.2	Test Case 2
2.4.3	Test Case 3
2.4.4	Test Case 4
2.4.5	Test Case 5
2.4.6	Test Case 6
2.4.7	Test Case 7
2.4.8	Test Case 8
2.4.9	Test Case 9
2.4.10	Test Case 10
2.4.11	Test Case 11
2.4.12	Test Case 12
2.4.13	Test Case 13
2.4.14	Test Case 14
2.4.15	Test Case 15
2.4.16	Test Case 16
2.4.17	Test Case 17
2.4.18	Test Case 18
2.4.19	Test Case 19
2.4.20	Test Case 20
2.4.21	Test Case 21
2.4.22	Test Case 22
2.4.23	Test Case 23
2.4.24	Test Case 24
2.4.25	Test Case 25
2.4.26	Test Case 26
2.4.27	Test Case 27
2.4.28	Test Case 28
2.4.29	Test Case 29
2.4.30	Test Case 30
2.4.31	Test Case 31
2.4.32	Test Case 32
2.4.33	Test Case 33
2.4.34	Test Case 34
2.4.35	Test Case 35
2.4.36	Test Case 36
2.4.37	Test Case 37
2.4.38	Test Case 38
2.4.39	Test Case 39

2.4.40	Test Case 40
2.4.41	Test Case 41

## List of Acronyms and Abbreviations

- ER Diagram - Entity Relationship Diagram
- PHP - Hypertext Preprocessor
- HTML - Hyper Text Markup Language
- CSS - Cascading Style Sheet
- JS - Java Script
- UI - User Interface
- ID - Identification
- XAMPP - Cross-platform, Apache, MySQL, PHP, and Perl
- SQL - Structured Query Language
- RAM - Random Access Memory
- IDE - Integrated Development Environment



# 1. Introduction

## 1.1 Problem Statement

During the second semester of our second year at our university, we had to launch an IT project module strategy to access a logical and effective software solution to a timely problem. As a result, we created a web application for a small supermarket business run by the father of one of our students at the university.

Ravishka Supermarket has been operating in the Rathnapura District for the past 10 years, delivering the latest quality products to thousands of Sri Lankans every week. Their business acts as an intermediary between a buyer, a seller, and a supplier. Bringing over 500 products to its customers, the supermarket has over 15 employees and long-term partners. The supermarket hopes to launch an online shopping platform that will enable customers to bring it home. Under the current system, the company owner must provide all the employee details, sales details, and financial information manually, which is time consuming and time consuming. Therefore, a user-friendly web application with a database will create a more manageable and secure environment.

- Currently ‘Ravishka Supermarket’ is done their process manually. If someone wants to buy products, he needs to come to the shop and buy them. And the suppliers also need to come to the company and supply their goods manually. Therefore, the buyers, sellers and suppliers must waste their time.
- With the current system the company owner must write the supplying details, selling details, buying details manually. Because the owner has not a database to store the details. It wastes his time.
- When customers(buyers) need to know about the products, the owner needs to describe about them manually. Because of that, per a day the owner cannot handle many buyers, he can handle only several customers.
- The owner must handle the all the financial details manually as there is no accountant in the company. So, it is difficult to handle all the employees’ financial details by himself. That consume the time unnecessarily.
- The existing system recording some details which are repeating all over the system.
- Because of the data redundancy, if the company requires update some details that will be more complex.



## Solution

According to the requirements of the company, we decided to divide business operations into some sections. Those sections are mentioned below.

- ✓ **Order management for retail customers**
- ✓ **Order management for wholesale buyers**
- ✓ **Employee management**
- ✓ **Stock management**
- ✓ **Financial management**
- ✓ **Transport management**
- ✓ **Suppliers' management**
- ✓ **Debtors' management**

According to those sections, the company requirements can be fulfilled by developing a web application.

Providing a user-friendly interface for sellers and suppliers. So that, suppliers can easily supply items, sellers can easily sell items.

Providing a database to store all the details of buyers and sellers which help to owner to manage the details easily without time wasting.

Supplying a facility to sellers to write a description about their goods including prices. So that, the owner doesn't need to describe about the items for customers individually.

Supplying the user profiles which can access by the employees (supplier, seller). So that, they see their products summary easily.

Providing a confidential interface with a function to handle financial details of the business, therefore the owner can calculate profits, calculate salaries of the employees easily.

## Benefits of the system

**Efficiency** - Supermarket will be able to provide a fast and efficient service.

**Accuracy** - All records and transactions are reported effectively.

**Helps in avoiding unnecessary paperwork** - Fixed ownership of paper files to records when the system is used to perform normal administrative tasks. It leads to the conservation of natural resources and the keeping of a digital record of data. It also does not confuse the records that need to be kept.

**Flexibility** - Provides simpler, faster, and more flexible.

**Data security** - System ensures that everything is safe and secure over the Internet, important documents can't be stolen and lost. All finance details are secure from unauthorized persons.

**User friendliness** - The stock manager, shop keeper, and customers can easily search view any details.

**Helps in avoiding unnecessary paperwork** - All the unnecessary paper works are avoided. all necessary details are in the database.

**Generate Reports** - Manage records and data easily as Decisions are effective.

## 1.2 Product Scope

This project will be solving the above-mentioned problems by creating a web application. The main functions of our system are Order placement for retail customers, Employee management, Order management for wholesale buyers, Stock management, Supplier's management, Transport management, Debtor's management, and financial management. Many elements are included on the website to help the client's present business overcome its challenges. The technology has enhanced the business's productivity and performance by overcoming irregular record- keeping.

Reg. No	Function Name	Completed Functions
IT20645670	Order placement for retail customers	<ul style="list-style-type: none"><li>- Record customer details.</li><li>- Record order details.</li><li>- Record delivery details (type and mode of the delivery).</li><li>- Issue bills for each retail order.</li><li>- Arrange offers, loyalty points for regular customers</li><li>- Customer can create his profile by inserting his name, mobile number, address etc., Customer can add items to the cart,</li><li>- Customer can search products; Customers can view their profiles by entering their credentials.</li><li>- Customer can update their details in his profile, Customer can update the quantity of items in the cart</li><li>- Customer can delete items from the cart.</li></ul>

IT20263362	Employee management	<ul style="list-style-type: none"> <li>- Add Employee Profile</li> <li>- Update, Delete, Search, View Employee Profile</li> <li>- Update, Delete, Search, View Duty Types</li> <li>- Update, Delete, Search,</li> <li>- Generate report of employee information</li> <li>- Change Password</li> <li>- View Profile</li> </ul>
IT20645670	Order management for wholesale buyers	<ul style="list-style-type: none"> <li>- The Customer is calling the shopkeeper for wholesale order</li> <li>- Shopkeeper can add order details</li> <li>- Update existing orders</li> <li>- Delete completed orders</li> <li>- Shopkeeper can add delivery details</li> <li>- Update existing delivery details by a shopkeeper when the customer wants to change his delivery location</li> <li>- Shopkeeper can delete completed delivery details</li> <li>- Calculate order bill by the shopkeeper</li> </ul>
IT20621292	Stock management	<ul style="list-style-type: none"> <li>- Add Grocery items, Frozen Items to one table.</li> <li>- Add vegetables and fruits to one table.</li> <li>- Search any items available in stock.</li> <li>- View Available grocery and frozen item Stock details.</li> <li>- View Available Vegetable and fruits Stock details.</li> <li>- Update any item when stock is out of stock.</li> <li>- Delete any item if that stock is expired or damaged.</li> <li>- Generate available stock reports with the total amount and what is out of stock, what are the expired stock, and all the stock details.</li> <li>- Get the pdf report of the available stock report.</li> </ul>

IT20301972	Supplier's management	<ul style="list-style-type: none"> <li>- Add Grocery items, Frozen Items to one table.</li> <li>- Add vegetables and fruits to one table.</li> <li>- Search any items available in stock.</li> <li>- View Available grocery and frozen item Stock details.</li> <li>- View Available Vegetable and fruits Stock details.</li> <li>- Update any item when stock is out of stock.</li> <li>- Delete any item if that stock is expired or damaged.</li> <li>- Generate available stock reports with the total amount and what is o</li> </ul>
IT20648336	Transport management	<ul style="list-style-type: none"> <li>- Add new vehicle by shopkeeper.</li> <li>- Add new driver by shopkeeper.</li> <li>- Update/delete/view vehicle by shopkeeper.</li> <li>- Update/delete/view driver by shopkeeper.</li> <li>- Search vehicles with vehicle name (ex: -truck)</li> <li>- Register parcel before delivering.</li> <li>- Generate tracking number.</li> <li>- Calculate delivery cost and add into parcel.</li> <li>- After delivering parcel update parcel state into delivered state.</li> <li>- Get wholesale buyer details using wholesale buyer database.</li> <li>- checking available drivers and assign drivers for delivering items.</li> <li>- Generate monthly report by shopkeeper and download as a pdf.</li> </ul>
IT20230692	Debtors management	<ul style="list-style-type: none"> <li>- Add new regular customers as debtors by shopkeeper.</li> <li>- Add new wholesale buyers' debtors as by shopkeeper.</li> <li>- Add new interest rates by shopkeeper.</li> <li>- Search debtors using relevant name and Debt ID and view by shopkeeper.</li> <li>- Shopkeeper can update relevant debtor's details and extend the loan period.</li> </ul>

		<ul style="list-style-type: none"> <li>- The shopkeeper can delete debtors' records when the debtor finishes theirs's debt amount in the relevant time.</li> <li>- Generate a report with monthly income from interest, and show using graphs to check the most chosen plan.</li> <li>- Monthly total income from interest can be Download as a pdf.</li> </ul>
IT20137250	Financial management	<ul style="list-style-type: none"> <li>- The shopkeeper can register suppliers in the system.</li> <li>- The shopkeeper can view all registered suppliers list.</li> <li>- The shopkeeper can update supplier details and remove supplier in the system.</li> <li>- The shopkeeper can add supplier payment details in the system.</li> <li>- The shopkeeper can view all supplier payment details list.</li> <li>- The shopkeeper can update supplier payment details.</li> </ul>

Table 1.2.a

## **Order placement for retail customers**

Our client request an online ordering platform, where customers can purchase items online, at home. So we developed a web application with super user friendly and attractive interfaces to purchase products.

This is a main function of the system, by this function manages all the orders of retail customer. Here in this function the main goal is to manage the orders of daily customers who are buying something individually rather than wholesales. In this function there are different set of works. Through this function, customers can purchase items by the help of user-friendly interfaces where they can easily add, update delete items from the cart. They can manage their user profiles which can update easily. This function takes place once a customer is log into the system. When an individual buyer(customer) places an order, his details are recorded in the system so that it is easy to access the data again. And this function records all the order details. The most important thing done by this function is issuing bills. When a retail customer places an order, the necessary bills are issued by the system after applying all discounts and promotions. so that the customer can get an idea to do the payments. This function generates reports based on the orders of retail customers.

## **Employee management**

If the register employee or unregistered employee first to the register our system. Register employee view the logging page he inserts the username and password. unregister employee click to the sign-up page to register. He inserts the username, address, contact number, age, email, password, and job role then after the register. Register employee logging to the system he can show his user profile page. He can click the update button to update details. He wants to delete user profile clicking delete button. When he wants to see salary, report go to the monthly report. He can view these details such as total working days, OT hours, number of working days and total salary. After the he can log out our system. Then admin use username and user password logging our system. Admin can generate employee salary report form. Admin input the month, level, total of working days, OT hours, number of working days after auto calculation salary. Then after admin show all employees salary details page. Admin can click the download button to download salary details pdf.

## **Order management for wholesale buyers**

Nowadays ordering goods online is a very popular method. Among these, it is super easy for the customer to order wholesale items through the supermarket. In this mini supermarket system, customers can order their wholesale items in while. Also, we have delivery system to delivery system to get your order to the doorstep. In this mini supermarket system, customers can order their wholesale items in an instant. Also, we have a delivery system to deliver your order to the doorstep. Through the shopkeeper, the customer can order the wholesale order.

What happens is that when the customer calls the shopkeeper and orders the wholesale items he wants, the shopkeeper writes down the required items' quantity and related details in the order list. After entering the order details, the shopkeeper obtains the details required

to deliver the customer's wholesale order. After specifying the delivery details, the shopkeeper prepares the bill for the goods related to the order and sends the order to the customer. This way we can easily place customer wholesale ordering and home delivery through our Ravishka Mini Supermarket.

### **Stock management**

Stock management is keeping a record of stock that comes into the supermarket and stock that leaves the store. In this Function stock manager analyses stock details. After the stock manager goes to the Stock management interface manager can see groceries, frozen items, and vegetables, fruits stock. In this groceries and frozen items have another stock. And vegetables and fruits have another Stock. Because vegetables and fruits does not have Any expire date or else manufacture date. Those stock manually check if damaged or not and then delete from the stock. But groceries and frozen items like shampoo, and ice cream, have manufactured dates and expired dates. Using that manager could see if the stock expired or not. if it expired manager can delete it from the stock. The stock manager can add new grocery and frozen items to the stock. when adding that stock manager must give the Item code, Item name, quantity, item price, manufacture date, and expired date. Then add the item to the database. If the manager clicks a button in the vegetable or fruits area, stock manager can add those without giving expired and manufacture dates. If the manager clicks any of the view buttons stock can see available stock now. Also, there is a button called to get the daily stock report on that page. once stock manager clicks that button manager can generate the pdf report with the current date and time. In that manager can see all the available stock. And stock manager can see what the out-of-stock items are and what are the expired stocks are. Also, manager can see the total amount according to the available quantity. Using these total amounts in the report manager can check losses because of the expired stock. The stock manager can search any items. when clicking the search button, it will redirect to the search page. In that, stock manager can give the item name and search it. then stock manager can get all details according to the item searches for. In there, there is an option bar. in that bar, there is a delete button and an update button. The manager can update if that item is out of stock (if it is out of stock, he can see the "out-of-stock" status in the quantity bar). And delete it if it is expired or damaged (if it is expired, he can see "This stock is expired" in the expire date bar).

### **Supplier's management**

In this function manage all supplier details payments. First Shopkeeper need to login into the system. Therefor shopkeeper give the suppliers details in the system. Such as username, password, NIC no, mobile number, item name, item code, quantity. This allows the owner to register suppliers and stores this detail in the system database. Shopkeeper add the supplier payment information and store in the system database. Shopkeeper can view the list of registered suppliers, update the details of those suppliers and remove suppliers from the list who no longer receive the goods.

And the Shopkeeper has the ability to include all of the suppliers' payment into the system and the Shopkeeper will also be able to view a list of all payments entered and update them if errors are made

Through this function,

- Shopkeeper can store all details of supplier.
- Shopkeeper can store all payment information.
- Shopkeeper can view all details of supplier and payment information.  
Shopkeeper can generate payment information report

## **Transport management**

As the shopkeeper of the RAVISHKA MINI SUPERMARKET institution has decided to deliver necessary grocery items to the customers, this system provides several platforms for the customers and the shopkeeper to perform this task. This function includes few sub functionalities. Shopkeeper can add new driver, add a new vehicle, get delivery items details, generate reports, and get wholesale buyer details. When the shopkeeper adds a new vehicle, this vehicle assigns for delivering items. There next main component of the shopkeeper is the record delivery items details. Here calculates total delivery cost, identify customer location, and generate tracking number. In their shopkeeper can see available driver list then shopkeeper can assign available one driver for delivering items. In their shopkeeper can update item delivery state (pending item, delivered item and cancelled item). As an example, when customer cancelled their order shopkeeper can update pending state to cancelled state. After that save all those details in database. There is an option to generate a monthly driver report. The shopkeeper must click on the "Generate Report" button and the report is downloaded as a pdf file. That report contain driver assign for the delivery purposes. Using this report shopkeeper can see which driver assign most time and which driver assign a smaller number of times. When the Shopkeeper login to the system, there will be an option called "add new driver", shopkeeper can view call the driver details. The driver list is given as an attractive user interface. Clear images are displayed to get a good idea of the driver. There is a button called "View" which displays all information in detail when clicked. Here shopkeeper can see the driver assigned state (Ex: - Assign for a delivery or not). Also, shopkeeper can view all information about the vehicles. In Add driver UI There are three options called update/delete, view and search. Shopkeeper can search vehicle using vehicle name (Ex: - shopkeeper entered truck as a vehicle name then displayed all the registered truck in the shop).

## **Debtors' management**

In Customers and wholesale buyers can borrow goods owing money to the shopkeeper but both of their must be regular customers. Meanwhile the customer and wholesale buyers details must be recorded (name of the debtor, date, price, no of items borrowed etc.). Updating and removing the customer couldn't pay their debts on due date ascertain interest rate will be tallied/added to the previous arrears. Though these function, shopkeeper can take decisions easily and can understand the people who didn't pay the debt of the business

easily. shopkeeper can search debtors' details related to the id. Because of that, data redundancy is very low.

If a debtor has an outstanding payment that has passed the due date, there are a number of steps that the shop keeper can take in order to collect the money owed. invoice list will automatically update to show which payments from debtors are overdue, indicated by a due date that will become red. the shopkeeper can quickly filter the invoices to view only those that are unpaid, and the overdue invoices will be easily viewed. Though this function shopkeeper can take decisions easily and can understand the debtors who paid and non-paid weekly, monthly in the business.

## **Financial management**

Finance Management function will be recorded all the finance performances according to incomes and expenses of the business to get total income and total expense to calculate monthly profit of the business. The main task of this function is to handle all the cash transactions done in a particular day range of the business.

Through this function,

- Manage business income

This function is facilitating the record of all the income. Transactions can be listed down such as Customer income, Sales income from related customer and wholesale customer and Interest income. The shopkeeper receives the relevant monthly income form the reports generated from each type of income. These incomes are inserting to the system with relevant 'income id' number.

- Manage business expenses

This function facilitates recording of all expenses. Expense can be listed down such as Employee payments, Electricity payments, Supplier payments and other utility bills. The shopkeeper receives the relevant monthly expense from the reports generated from each type of expense. The Supply Manager makes the relevant generated report available to the shopkeeper. These expenses are inserting to the system with relevant 'expense id' number.

- Generate monthly financial report

The generating report includes finance transactions that have been done through the month with total income and total expense and monthly net profit. These reports demonstrate increment or decrement of the incomes and expenses related to the financial operations of the business.

Through these functions, the shopkeeper can take decisions easily and can understand the maximization of the profit monthly in the business. Since the payments and incomes inserting into the system with a unique number, the shopkeeper can search finance transactions related to the id.

## **Work Distribution Among Members**

<b>Name and IT number</b>	<b>Work distribution</b>
Jayasekara H.L IT20621056	<p><b>Order management for retail customers</b></p> <ul style="list-style-type: none"> <li>• User interface development</li> <li>• Back-end development</li> <li>• Implementing the related create, update, retrieve and delete functions.</li> <li>• Creating the database</li> <li>• Generating the report</li> </ul> <p><b>Employee management</b></p> <ul style="list-style-type: none"> <li>• User interface development</li> <li>• Back-end development</li> <li>• Implementing the related create, update, retrieve and delete functions.</li> <li>• Creating the database</li> <li>• Generating the report</li> </ul>
Alahakoon A.M.D.P.S IT20263362	
Umendra T.A.H IT20645670	<p><b>Order management for wholesale buyers</b></p> <ul style="list-style-type: none"> <li>• User interface development</li> <li>• Back-end development</li> <li>• Implementing the related create, update, retrieve and delete functions.</li> <li>• Creating the database</li> <li>• Generating the report</li> </ul>

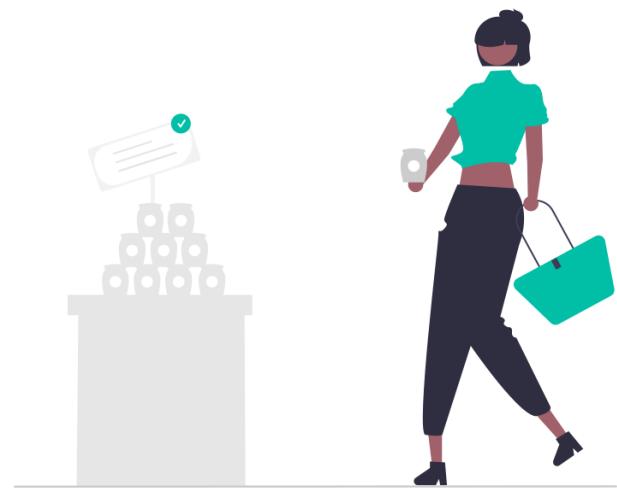
Arachchi S.C.S IT20230692	<b>Debtors' management</b> <ul style="list-style-type: none"> <li>• User interface development</li> <li>• Back-end development</li> <li>• Implementing the related create, update, retrieve and delete functions.</li> <li>• Creating the database</li> <li>• Generating the report</li> </ul>
Gunasekara P.G.A.I.P  IT20621292	<b>Stock management</b> <ul style="list-style-type: none"> <li>• User interface development</li> <li>• Back-end development</li> <li>• Implementing the related create, update, retrieve and delete functions.</li> <li>• Creating the database</li> <li>• Generating the report</li> </ul>
De Silva P.R. P  IT20301972	<b>Supplier management</b> <ul style="list-style-type: none"> <li>• User interface development</li> <li>• Back-end development</li> <li>• Implementing the related create, update, retrieve and delete functions.</li> <li>• Creating the database</li> <li>• Generating the report</li> </ul>
Thilakarathne W.A.S.S  IT20648336	<b>Transport management</b> <ul style="list-style-type: none"> <li>• User interface development</li> <li>• Back-end development</li> <li>• Implementing the related create, update, retrieve and delete functions.</li> <li>• Creating the database</li> <li>• Generating the report</li> </ul>
Rajapaksha R.C.P IT20137250	<b>Finance management</b> <ul style="list-style-type: none"> <li>• User interface development</li> <li>• Back-end development</li> <li>• Implementing the related create, update, retrieve and delete functions.</li> <li>• Creating the database</li> <li>• Generating the report</li> </ul>

### **1.3 Project Report Structure**

With the help of this document, it is possible to get a clear understanding of the whole picture of the System since it contains detailed explanations and diagrams to support every function.

Project report will explain,

- The process our group went through while Requirement Gathering
- The process our group went through while designing the functions
- The process our group went through while connecting to the Database
- The process our group went through while designing the interfaces
- The process our group went through while implementing the functions



## 2. Methodology

### 2.1 Requirements and Analysis

After getting the raw idea of the requirements of the client, we were able to study them and design

- Use Case Diagrams
- Activity Diagrams

To each function to arrive with a fixed list of functions need to be Implemented.

#### ❖ Software Requirements

- ✓ Windows 10 / Windows 11
- ✓ Laravel 9
- ✓ PHP 9
- ✓ XAMPP Server 8
- ✓ Php My admin
- ✓ Visual Studio Code
- ✓ Any Web browser

#### ❖ Hardware Requirements

- ✓ Hard Disk 2GB
- ✓ RAM 1GB



### **2.1.1 Functional Requirements**

System should record customer details.

System should record order details.

System should record delivery details (type and mode of the delivery).

System should Issue bills for each retail order.

System should be able to identify loyalty customers.

Customer should be able to create his profile by inserting his name, mobile number, address etc.

Customers should be able to add items to the cart.

Customers should be able to search products.

Customers should be able to view their profiles by entering their credentials.

Customer should be able to update their details in his profile, Customer should be able to update the quantity of items in the cart.

Customer should be able to delete items from the cart.

Employee should be able to register the system first.

Employee should be able to view and edit the details of user account and delete user account.

Admin logging to the system and admins can generate monthly report.

Admin should be able to view, edit monthly report details and delete monthly report.

Wholesale buyers can only make a phone call and place the required order.

The shopkeeper should record the wholesale buyers' order details and delivery details.

The shopkeeper has the access to the system and views the order list and delivery details list.

Also, he can delete and update the delivery, and order details.

Stock manager should be able to get stock from suppliers and add them to grocery items, frozen items, and vegetables, fruit fields. Stock manager should be able to search for any grocery, frozen items or vegetables, and fruits.

Stock manager should be able to see what the expired stocks are. then he can delete it from the stock

Stock manager should be able to see what is out of stock and then he can update that stock.

Stock manager allowed to view available stock for current date and time.

Stock manager can download the available stock report as a pdf with all the stock details.

Shopkeeper should be able to register in the system.

shopkeeper should be able to add new vehicle or driver and view driver/vehicle list.

Shopkeeper should be able to upload driver's profile picture.

Shopkeeper should be able to edit, delete vehicle or driver from the list.

Shopkeeper should be able to search vehicle using vehicle name.

Shopkeeper should be able to view wholesale buyers' details.

Shopkeeper should be able to register a parcel and should be able to calculate total delivery cost.

shopkeeper should be able to generate driver report.

Shopkeeper should be able to view customer's delivery place using google map.

Shopkeeper should be able to assign driver for delivering a parcel.

The shopkeeper should be able to add wholesale buyers and regular customers as a debtor when their requests checking manually and the system.

Shopkeeper should be able to view current interest rates.

Shopkeeper can add new interest rate for the current interest rates.

Shopkeeper can add view wholesale buyers and regular customers using search option.

Shopkeeper should be able to update and remove the customer who couldn't pay their debts on the due date a certain interest rate will be tallied/added to the previous arrears.

Shopkeeper should be able to view report, get the monthly income from debtors, and how many debtors are in each plan using the graph view.

Shopkeeper can download as a pdf monthly income of interest from debtors.

Shopkeeper allowed to add incomes and expenses.

Shopkeeper should be able to generate reports of their incomes and expenses history.

Shopkeeper can view, edit, and delete all incomes and expenses.

Shopkeeper should be able to generate report of all incomes and expenses.

Shopkeeper can deduct the total expenses from the total income and calculate the monthly profit.

Shopkeeper should be able to generate monthly financial report.

Shopkeeper should be able to show charts of all incomes and expenses.

The owner should be able to obtain supply details and payment details from the suppliers and enter those details in the system and register the supplier.

The owner can view registered suppliers list.

The owner should be able to update all the details of that supplier and delete the supplier that is no longer in supply.

## **2.1.2 Non - Functional Requirements**

Quality and Performance characteristic of our system fulfils main non-functional requirements as follows,

### **Security**

User passwords are encrypted by themselves or by the administrator as they are registered in the system. User details are visible only to themselves and the administrator. Users will not be able to access other people's information for any purpose other than the opportunity for the shopkeeper to obtain details of the goods and cash transactions obtained by each customer.

### **Performance**

Any user will be able to register on the network as a customer or employee and will still not have access to other activities on the system. All other users are registered by the shopkeeper.

The response and processing time of each interface are within a few seconds. The system is platform independent and open source.

### **Safety**

Each customer has a unique username and a password to access their respective accounts on the website.

### **Reliability**

The system should be reliable when it comes the making reservations, which is very important module to the customer. It is also necessary that the updating of the database which is done through the system very accurately.

### **Accuracy**

Report generation, data input should be accurate in the system.

Ex: - Employees are never permitted to change their payment information. An attempt of this nature should be reported to the shopkeeper.

### **Speed**

Our system has higher number speed. it takes less time to refresh screen. page loading time is very low and if you search some word result will give quickly.

Ex: - customer search “vegetables” in search bar quickly get data from database and display. It takes a smaller number of times.

## **Availability**

System available for 24 hours and whole week. Any time customers can access to the system.

## **Manageability**

When shopkeeper editing the code for pages backend, frontend working in properly.  
Ex: - shopkeeper when update grocery items, customers can login or search something.

## **Data integrity**

Shopkeeper update every record transaction end of the day.



### 2.1.3 Use Case Diagrams and Activity Diagrams of relevant functions

#### Employee Management (IT20263362)

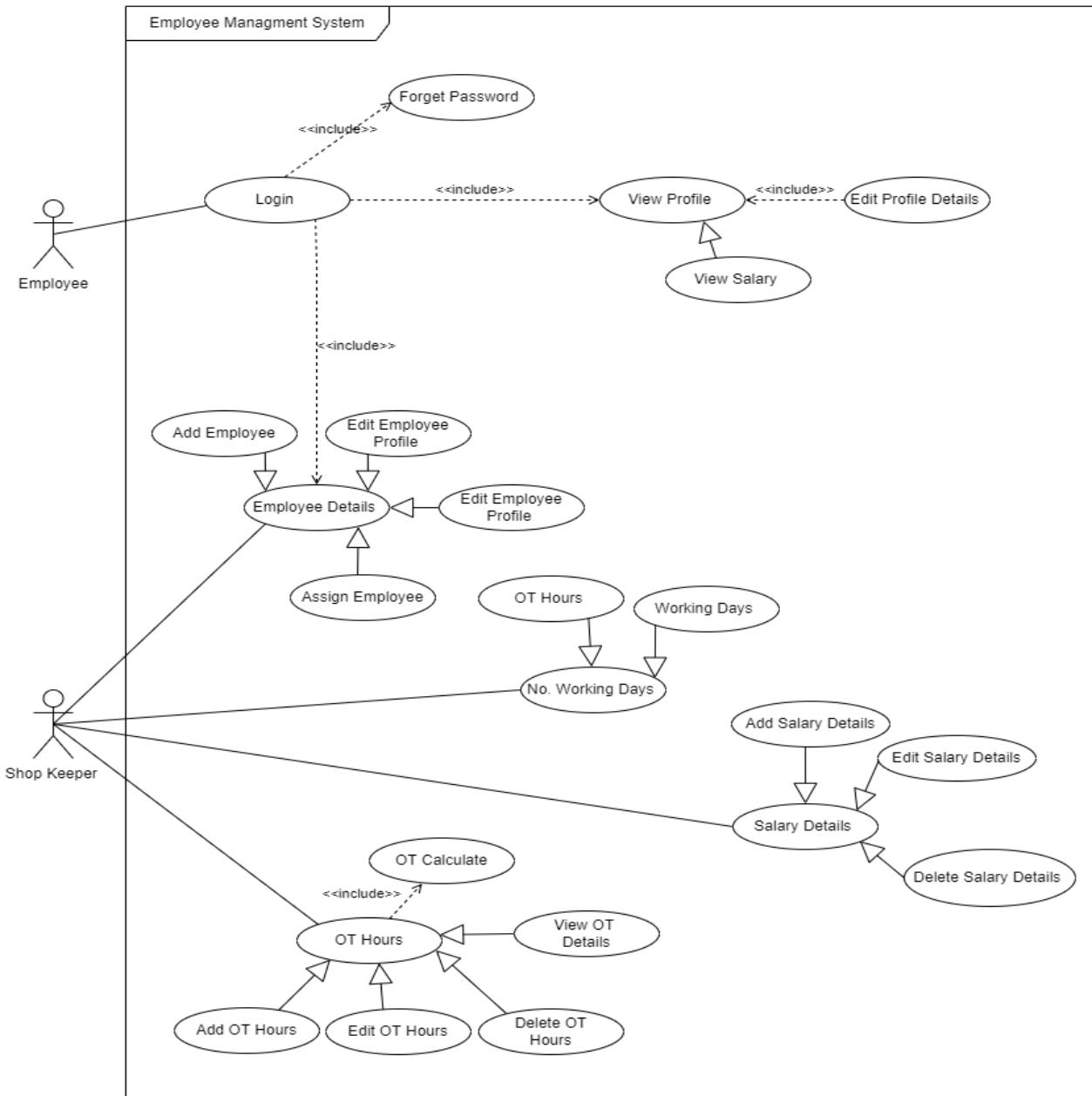


Fig: 2.1.3.a

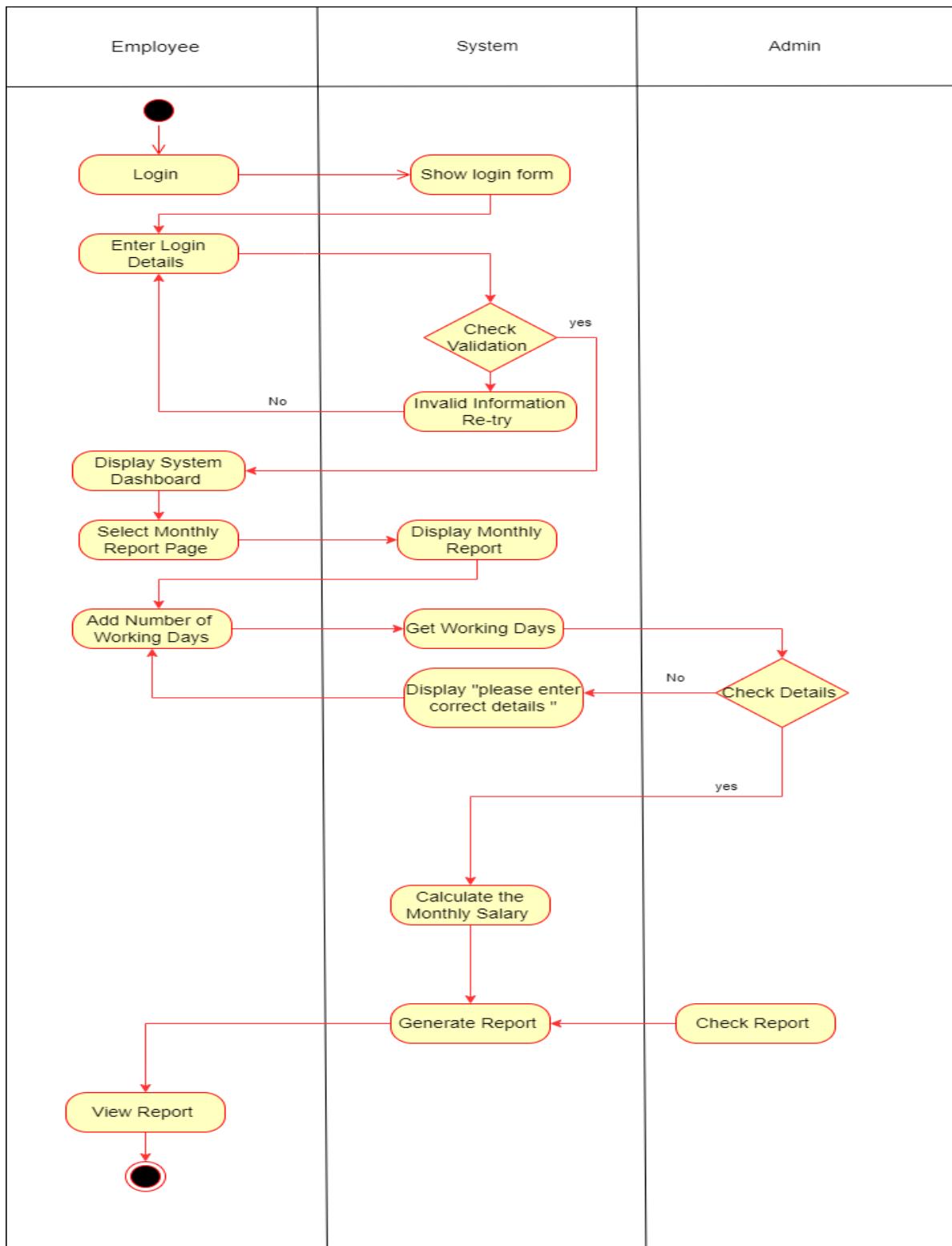


Fig: 2.1.3.b

## Order management for wholesale buyers (IT20645670)

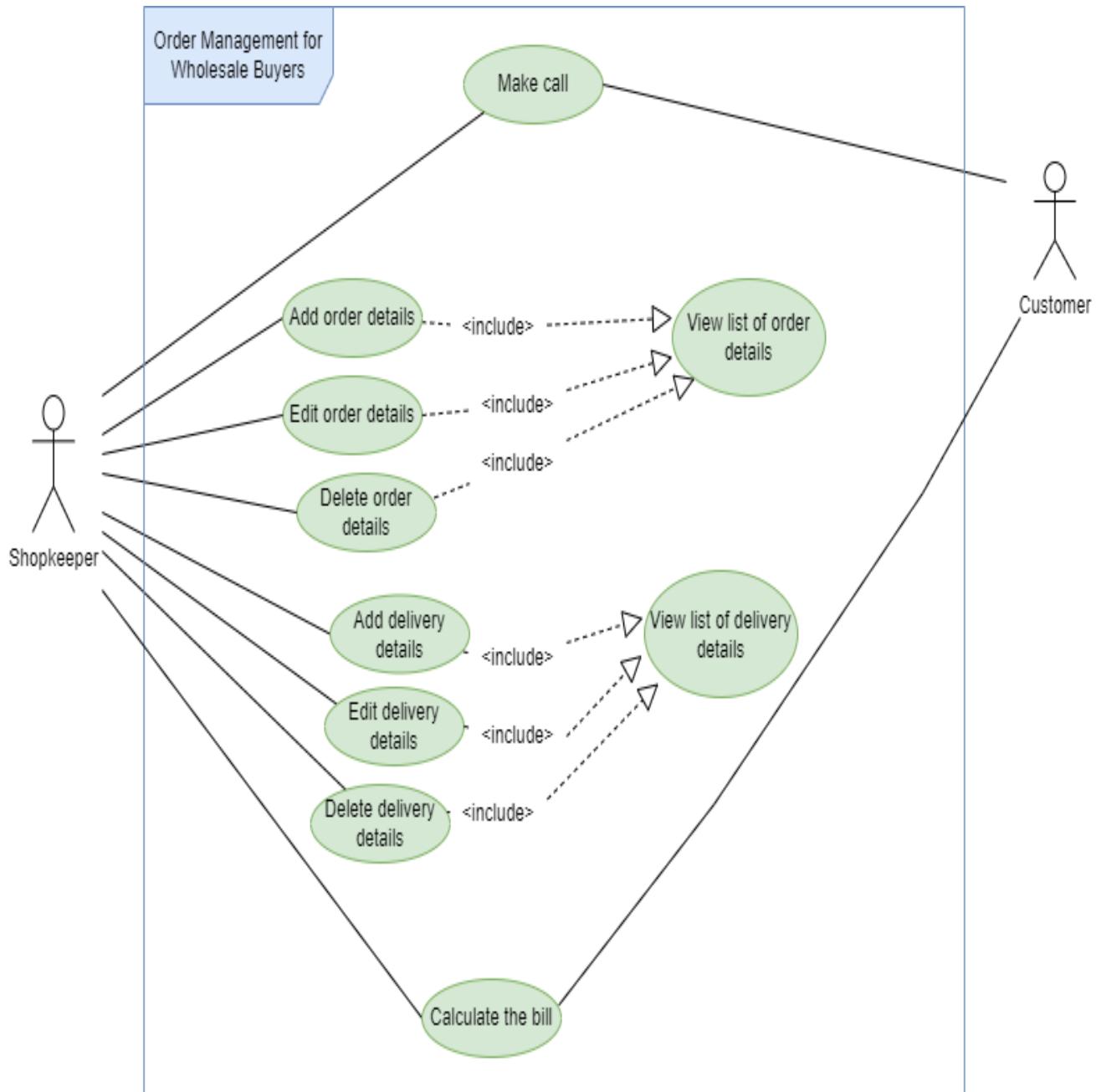


Fig: 2.1.3.c

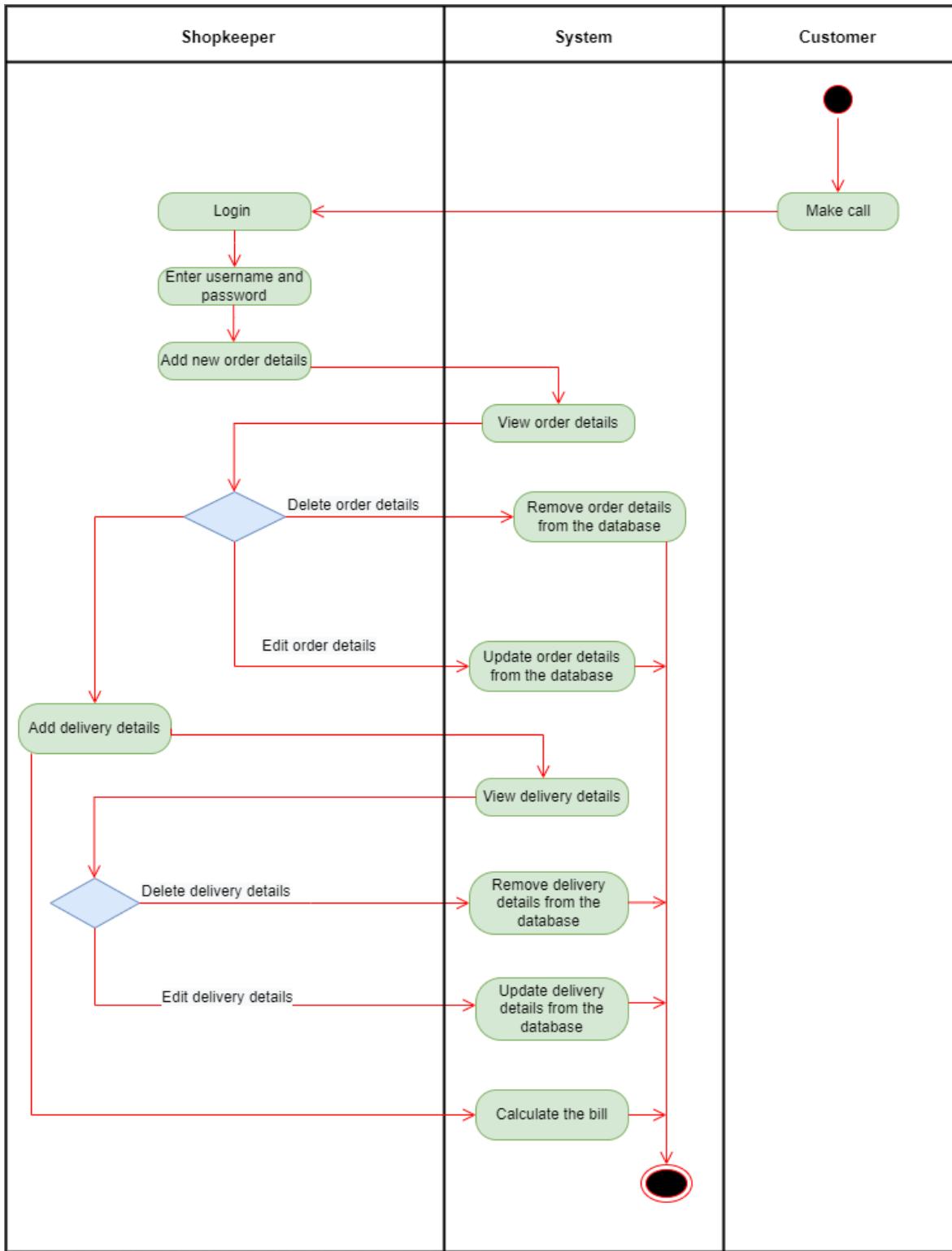


Fig: 2.1.3.d

## Transport management (IT20648336)

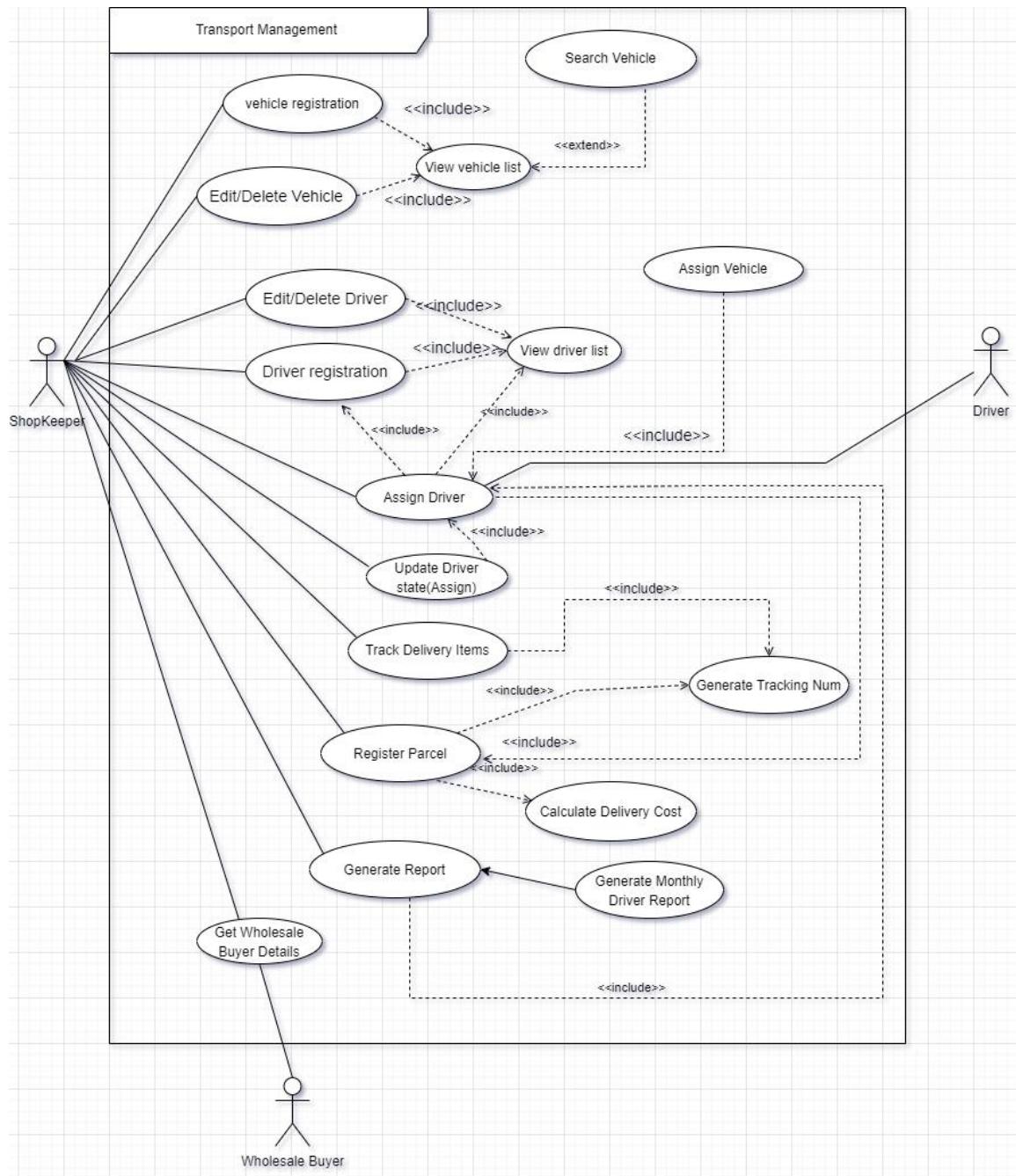


Fig: 2.1.3.e

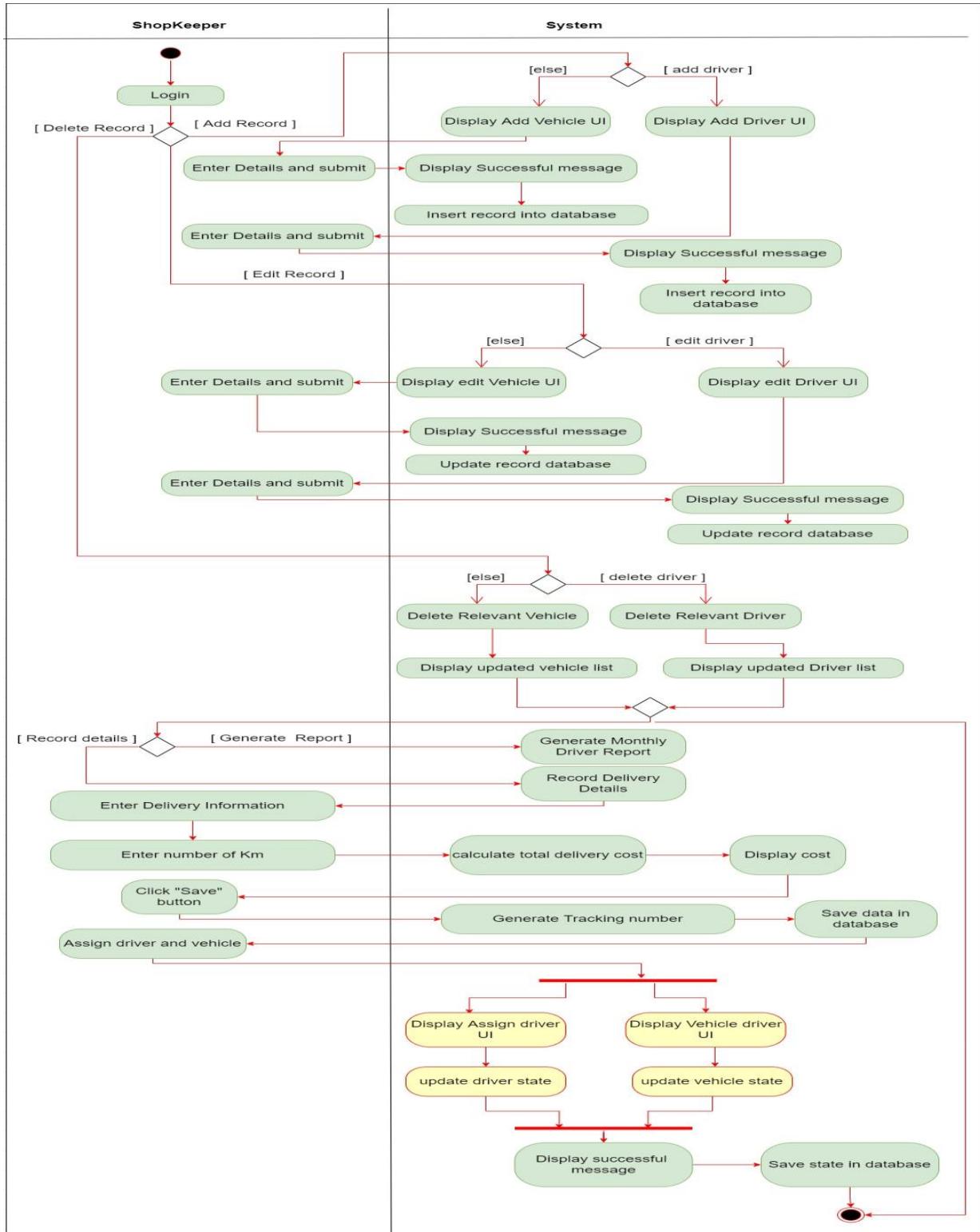


Fig: 2.1.3.f

## Debtor's management (IT20230692)



Fig: 2.1.3.g

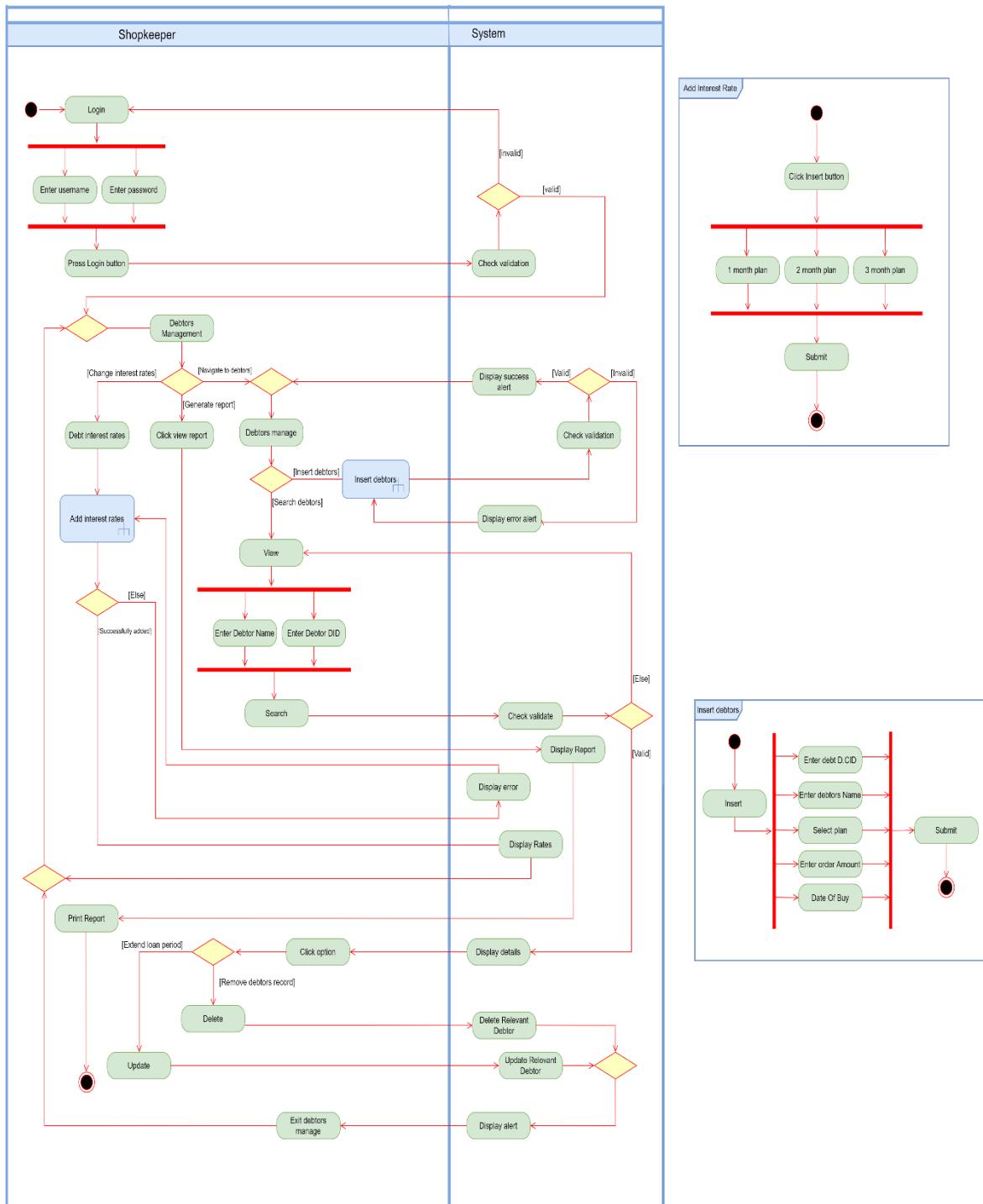


Fig: 2.1.3.h

## Financial management (IT20137250)



Fig: 2.1.3.i

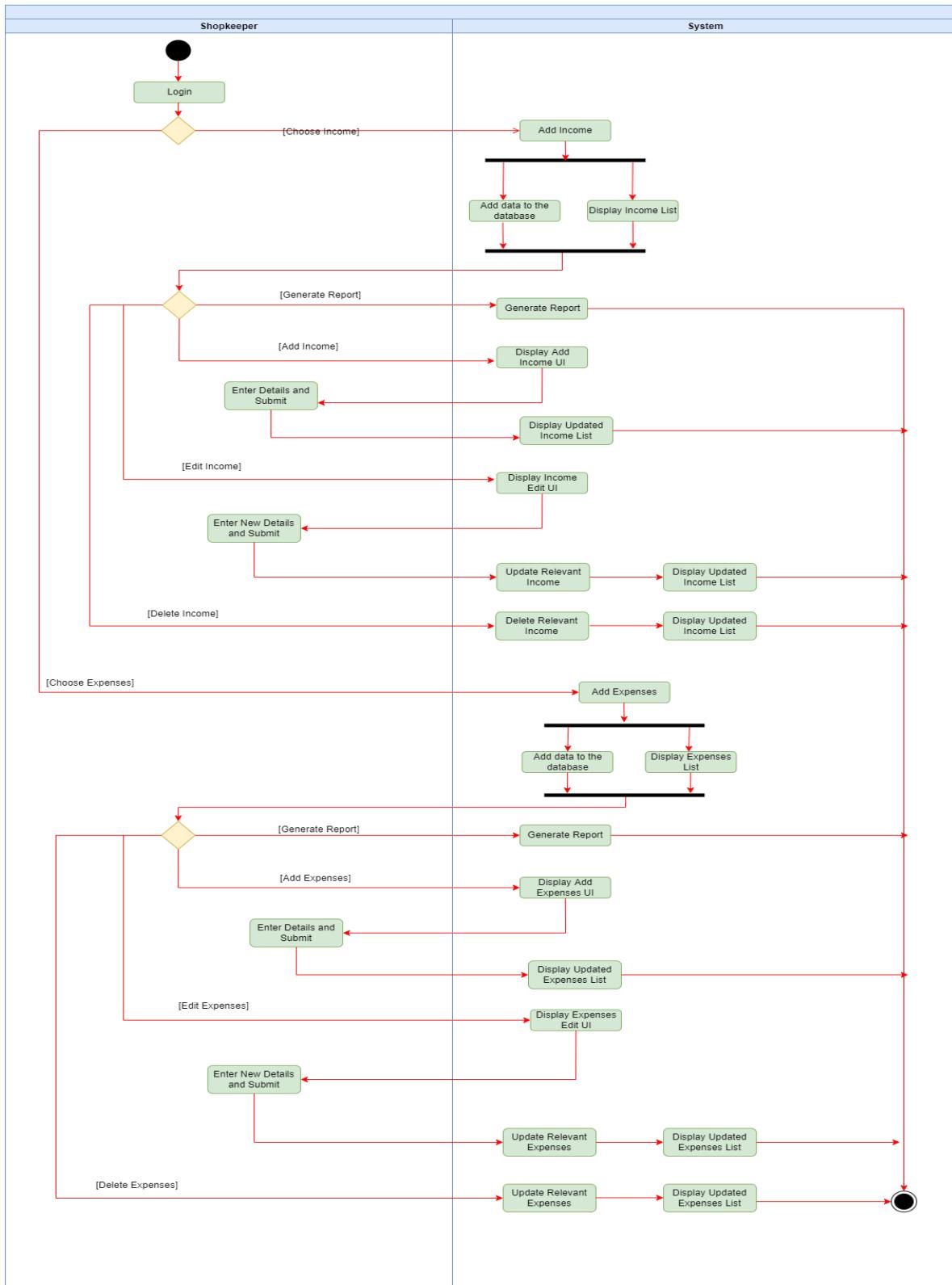


Fig: 2.1.3.j

## Stock management (IT20621292)

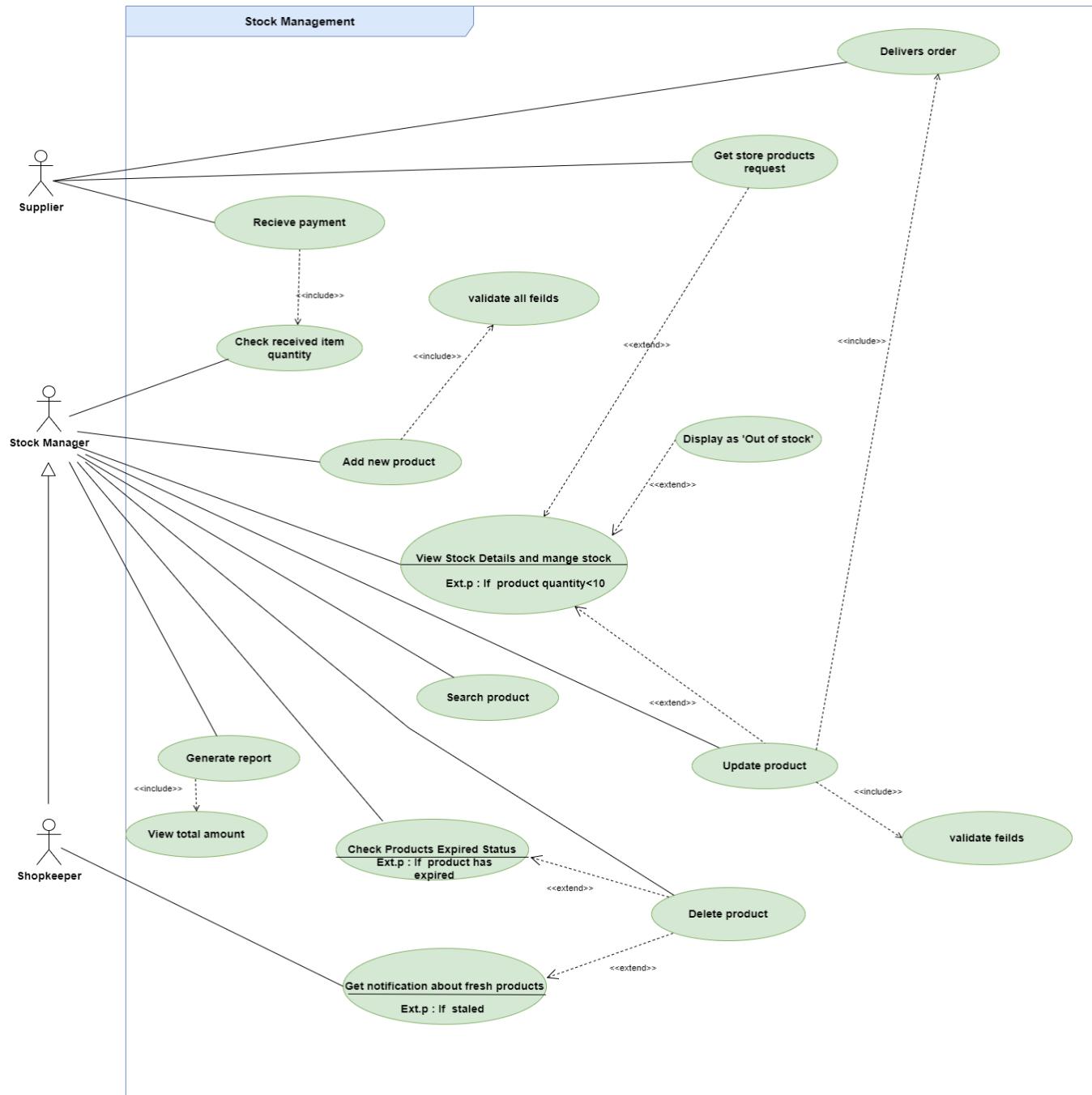


Fig: 2.1.3.k

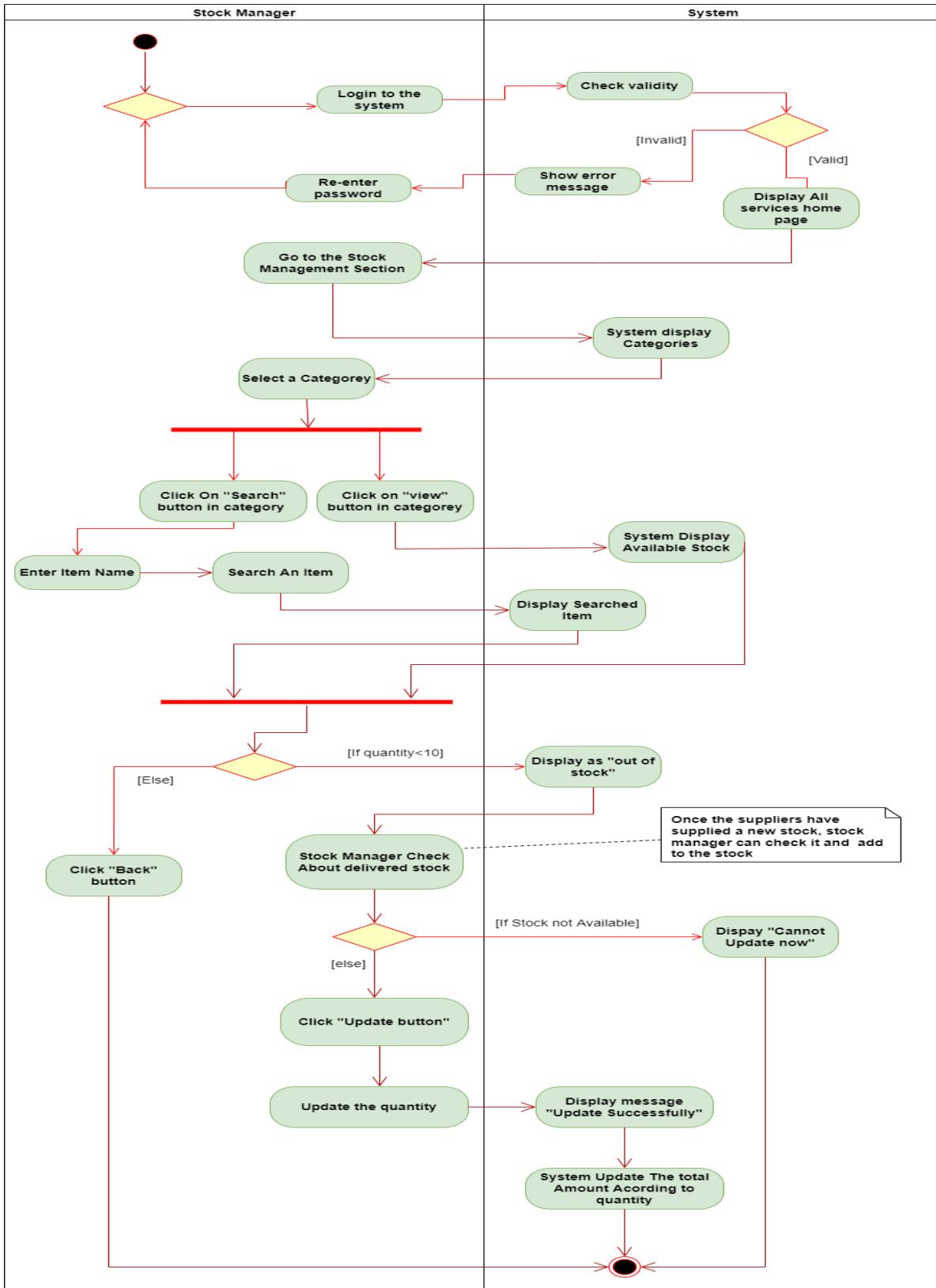


Fig: 2.1.3.l

## Supplier's management (IT20301972)

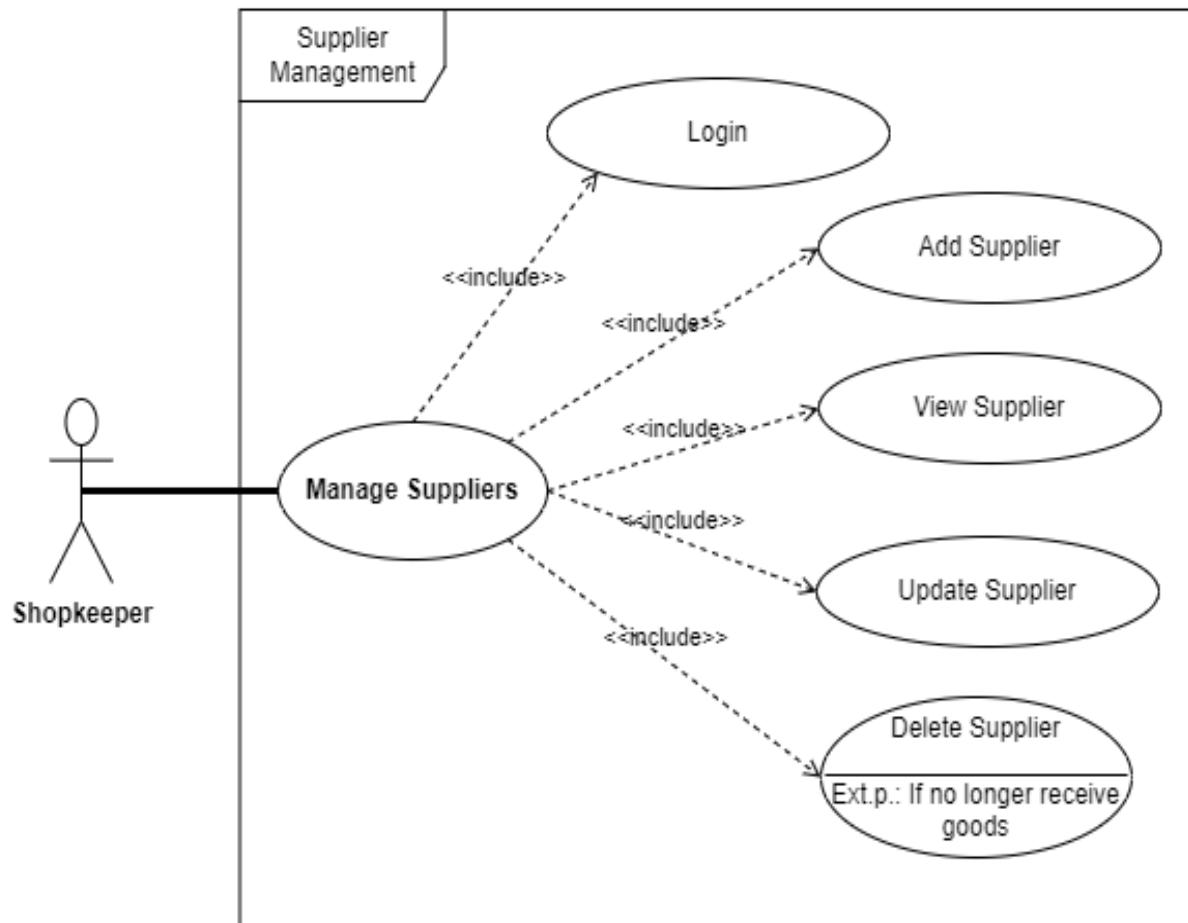


Fig: 2.I.3.m

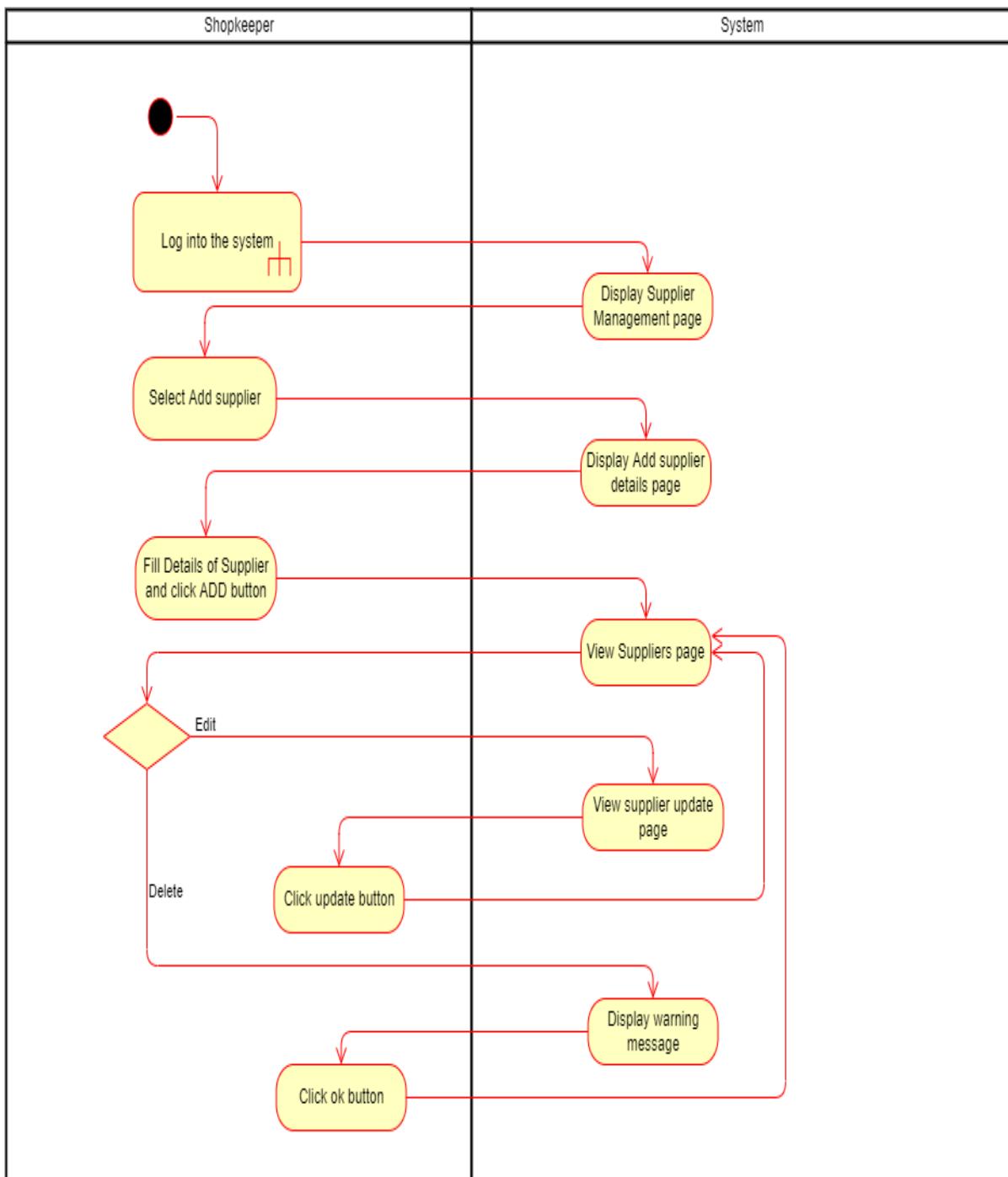


Fig: 2.1.3.n

## Order management for retail customers (IT20621056)

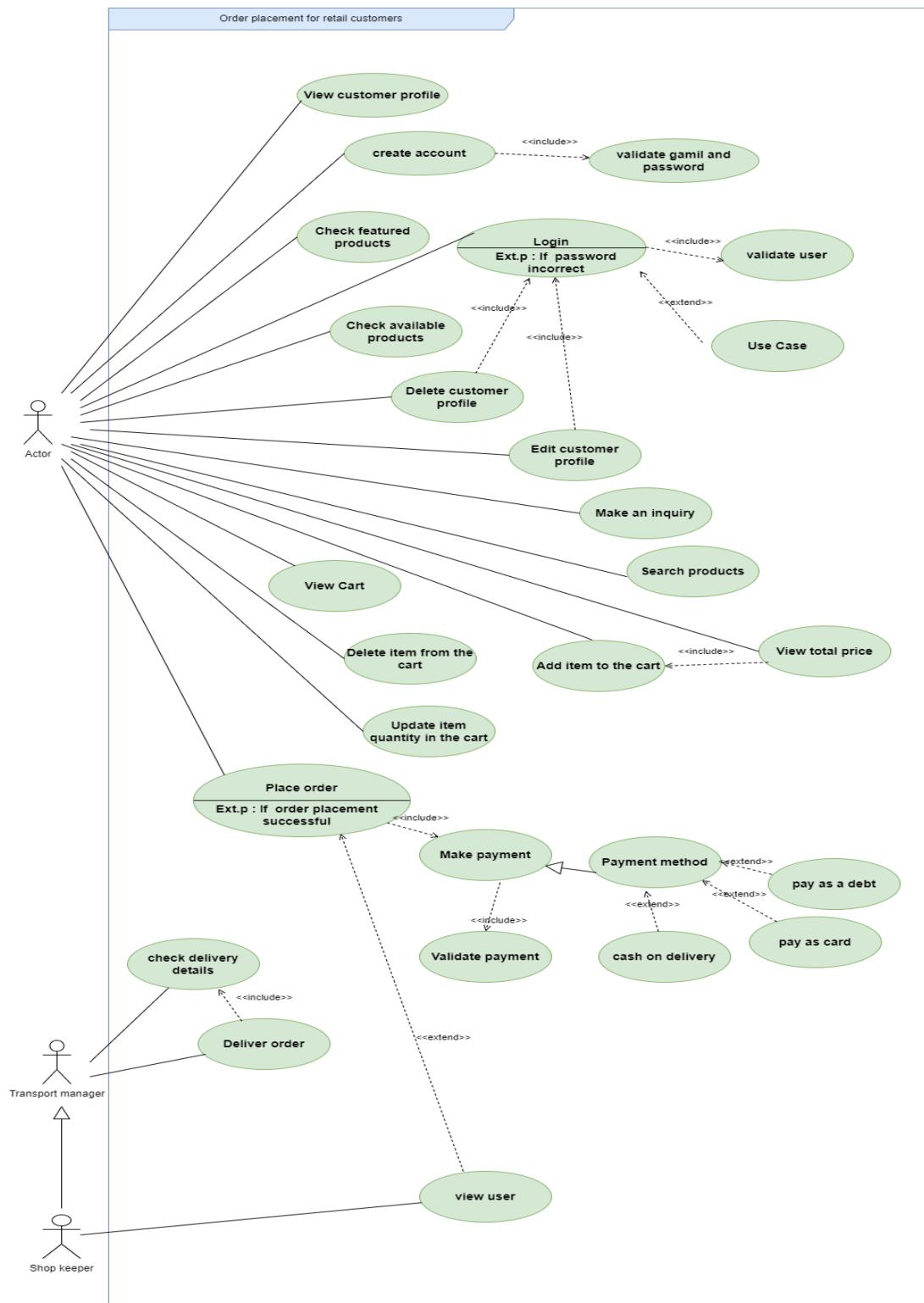


Fig: 2.1.3.o

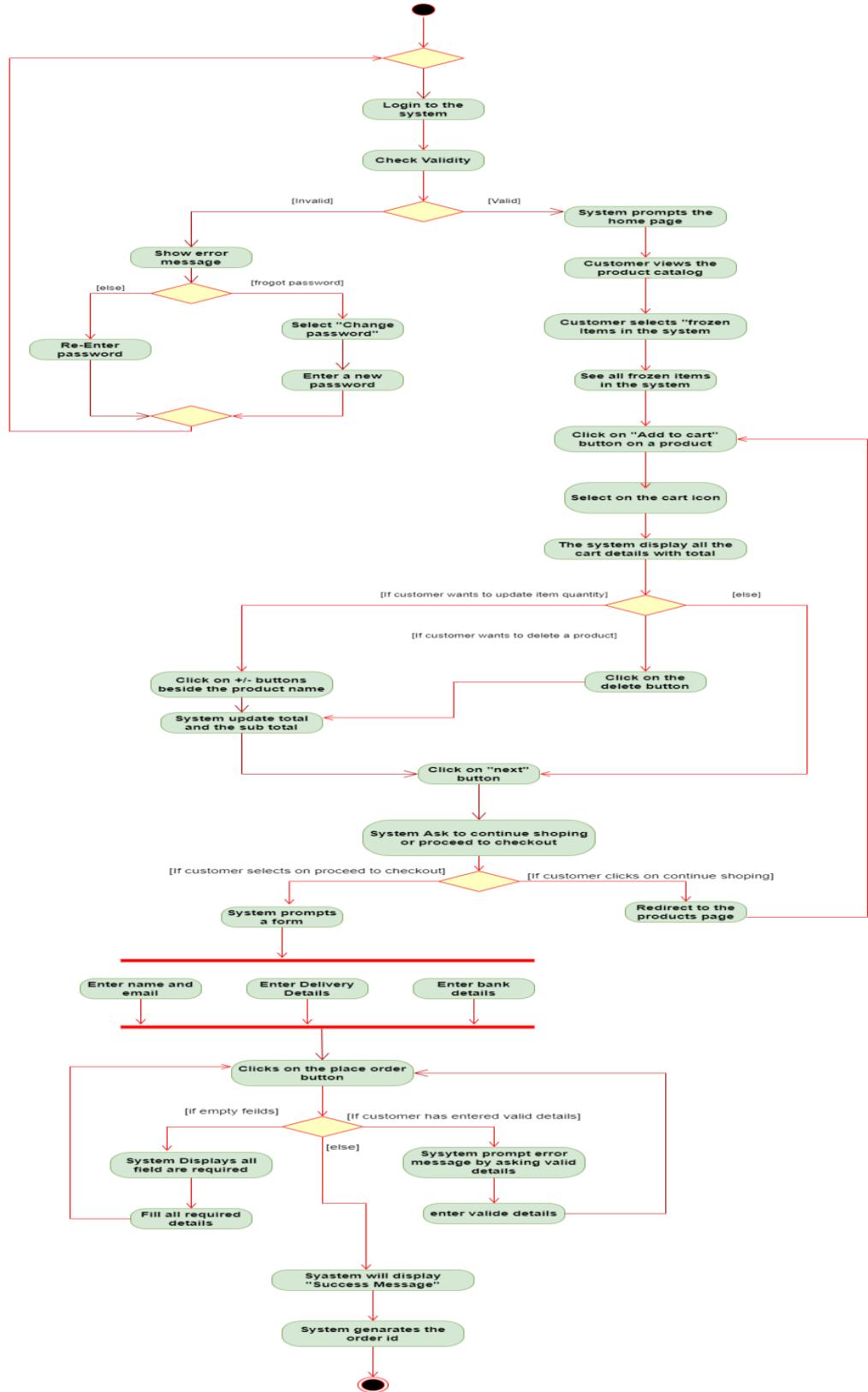


Fig: 2.1.3.p

#### 2.1.4 Use Case Scenario

<b>Use Case ID</b>	S001	
<b>Use Case Name</b>	Salary Report View	
<b>Summary</b>	Shop Keeper view the salary details and verify the salary report	
<b>Priority</b>	3	
<b>Preconditions</b>	Shop Keeper should maintain Report ID, Salary Details, Confirm ID assigned to the Salary Report	
<b>Post Conditions</b>	Shop Keeper can view salary report details in the panel and Employee can access to the system and Employee can see the salary report details.	
<b>Primary Actor</b>	Shop Keeper, Employee	
<b>Secondary Actors</b>	Salary	
<b>Trigger</b>	A new salary report gets into a processing status	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	<b>1</b>	Shop Keeper enters the shop keeper panel
	<b>2</b>	Shop Keeper enters the valid employee credentials
	<b>3</b>	Shop Keeper moves to Salary Report View Page
	<b>4</b>	Shop Keeper keeps a unique salary report id for that new salary report add
	<b>5</b>	Shop Keeper enters new salary report details, OT hours, and assigned default employee id for the salary report viewing
	<b>6</b>	System validates whether salary report id is unique, number of working days, OT hours in correct format, provided and confirmed ID is same, and all required fields are filled
	<b>7</b>	System displays relevant error messages or success messages.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	<b>7a</b>	Shop Keeper can view the salary report details in the shop keeper panel
	<b>5a</b>	Salary report adding to generate is getting created
	<b>5b</b>	Shop Keeper logins with details and change salary report details.
<b>Open issues</b>	<b>1</b>	Can salary report depend on alone? No
	<b>2</b>	Should all the details be required? No

*Fig: 2.1.4.a*

<b>Use case name</b>	Driver Registration	
<b>Use case ID</b>	S002	
<b>Priority</b>	High	
<b>Primary actor</b>	Shopkeeper	
<b>Description</b>	All drivers must be registered with their personal details. They provide their information to the shopkeeper.	
<b>Preconditions</b>	Logged in to the system	
<b>Post conditions</b>	Driver successfully registered and driver ID issued.	
<b>Trigger</b>	This use case is initiated when the shopkeeper selects the option to register details of driver.	
<b>Main scenario</b>	<b>Step</b>	Action
	<b>1</b>	Login to the system using valid credentials.
	<b>2</b>	Shopkeeper enters driver details to the system.
	<b>3</b>	System verifies the information and adds to the database.
	<b>4</b>	System displays “Successful” message.
	<b>5</b>	Auto generated driver registration number generated.
	<b>6</b>	Send registration information view list UI.
	<b>7</b>	Generate driver report.
<b>Extensions</b>	<b>Step</b>	Branching action
	<b>1.1</b>	Invalid login details
	<b>1a</b>	Display re-enter login details
	<b>1.2</b>	Shopkeepers enter valid details
	<b>1a</b>	System prompt to the transport management home page.
	<b>2.1</b>	Shopkeepers enter invalid driver details
	<b>2a</b>	Display error message
	<b>2.2</b>	Shopkeepers enter valid details
	<b>2a</b>	Display “Successful” message

Fig: 2.1.4.b

<b>Number</b>	S003	
<b>Name</b>	Manage Debtors	
<b>Summary</b>	Login and manage Debtors	
<b>Priority</b>	5	
<b>Precondition</b>	User place order as debtor and validated by shopkeeper	
<b>Post condition</b>	Receive debt report as a pdf	
<b>Primary actor</b>	Shopkeeper	
<b>Trigger</b>	shopkeeper has chosen to debt type wholesale or Regular Customer	
<b>Main scenario</b>	<b>step</b>	<b>Action</b>
	<b>1</b>	Login into the system
	<b>2</b>	Shopkeepers check the user type
	<b>3</b>	Check the current Interest rates according to user type
	<b>4</b>	Go to insert debtor's details in relevant type
	<b>5</b>	Enter debtor's D_CID
	<b>6</b>	Enter debtor's name
	<b>7</b>	Select plan according user request
	<b>8</b>	Enter order amount
	<b>9</b>	Enter Date of Buy
	<b>10</b>	Click submit button
	<b>11</b>	Search debtors
	<b>12</b>	update debt details
	<b>13</b>	Delete debt record
	<b>14</b>	Go to view report and print monthly income from debtors
	<b>15</b>	Exit
<b>Extensions</b>	<b>steps</b>	<b>Branching action</b>
	<b>1A</b>	If password or username or both invalid
	<b>a</b>	System shows a notification
	<b>b</b>	Give three attempt to enter password
	<b>1B</b>	If password is forgotten
	<b>a</b>	Allow to create new password
	<b>4A</b>	If user is Regular Customer
	<b>a</b>	Go to regular customer form
	<b>4B</b>	If user is Wholesale Buyer
	<b>a</b>	Go to Wholesale Buyer form
	<b>10A</b>	Display the field error massages
	<b>a</b>	Check each field and enter correct data
	<b>11A</b>	If Display error message
	<b>a</b>	Enter Correct Debt name and D_CID
<b>Open issues</b>	<b>1</b>	Should the system check to see if the shopkeeper entered all the information correctly?

Fig: 2.1.4.c

<b>Use case ID</b>	S004	
<b>Use case Name</b>	Add income details and expenses details	
<b>Summary</b>	Shopkeeper adds a new income and expenses details to the system.	
<b>Priority</b>	O4	
<b>Pre-Condition</b>	Shopkeeper has logged into the admin panel	
<b>Post-Condition</b>	The shopkeeper can view income and expenses details and generate a report	
<b>Primary Actor</b>	Shopkeeper	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	<b>1</b>	Shopkeeper enters his admin panel
	<b>2</b>	Shopkeeper enters his valid user credentials
	<b>3</b>	Shopkeeper moves to income statement page
	<b>4</b>	Select income type
	<b>5</b>	Shopkeeper enters new income details
	<b>6</b>	Presses the add button to confirm and add details to database
	<b>7</b>	Shopkeeper moves to expenditure statement page
	<b>8</b>	Select expense type
	<b>9</b>	Enters new expense details
	<b>10</b>	Presses the add button to confirm and add details to database
<b>Extensions</b>	<b>Step</b>	<b>Branching action</b>
	<b>4a</b>	If the selected income type is an existing income type in the database, it is Disabled.
	<b>6a</b>	If entered details are incorrect, the system asks re-enter correct details.
	<b>8a</b>	If the selected expense type is an existing expense type in the database, it is Disabled.
	<b>10a</b>	If entered details are incorrect, the system asks re-enter correct details.

*Fig: 2.1.4.d*

<b>Use case ID</b>	S005	
<b>Use case Name</b>	Manage wholesale details	
<b>Priority</b>	03	
<b>Primary Actor</b>	Customer, Shopkeeper	
<b>Pre-condition</b>	Shopkeeper must login to the system and submit wholesale details	
<b>Post-condition</b>	Shopkeeper must calculate the bill and complete the order	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	<b>1</b>	Customer is calling to the shopkeeper
	<b>2</b>	Shopkeeper login to the system and view wholesale order service page
	<b>3</b>	Then he adds the customer wholesale order details (Product, Quantity and Description) and submit it.
	<b>4</b>	After the customer record the order details, he should add the delivery details (Customer Name, Email, Address, Contact No and Delivery Instructions) and save it.
	<b>5</b>	Finally, Shopkeeper calculate the bill of the wholesale order
	<b>6</b>	Then order sent to checkout
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	<b>3.1</b>	If the customer wants to delete or update his order, Shopkeeper can <ul style="list-style-type: none"> <li>• Delete order</li> <li>• Update order</li> </ul>
	<b>4.1</b>	When the customer wants to change his delivery location, or a shopkeeper wants to delete the delivery details after the order is completed, he can <ul style="list-style-type: none"> <li>• Update delivery details</li> <li>• Delete delivery details</li> </ul>

Fig: 2.1.4.e

<b>Number</b>	S006	
<b>Use Case Name</b>	View Stock details and mange stock	
<b>Summary</b>	Stock manager view stock details and update stock.	
<b>Priority</b>	High	
<b>Preconditions</b>	Stock manager should log into the system	
<b>Postconditions</b>	Stock manager manages stock details	
<b>Primary Actor</b>	Stock Manager	
<b>Trigger</b>	Stock manager Stock manager view stock details and update stock.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Stock manager login to the system using their login credential
	2	Stock managers move on to the All-service home page
	3	Stock manager should click on the view button of the stock management section.
	4	System displays stock management home page
	5	Four categories are shown on the stock management home page. each category has added, view, search button
	6	Stock manager goes through the view path in the groceries category
	7	Stock manager will navigate to the Available stock page.
	8	Stock manager can see all the details according to the added items.
	9	System display status “This Item out of stock”.
	10	Stock manager clicks the update button.
	11	System checks another stock deliver or not.
	12	Stock manager redirects update product page.
	13	Stock managers update the quantity more than 10.
	14	Then clicks the Update button in the product page.
	15	Stock manager again redirect to the Available stock page.
	16	System will remove the “this item out of stock” status.
	17	System will Update the quantity.
	18	Then system will display total amount according to the quantity
<b>Extensions</b>	<b>Step</b>	<b>Action</b>
	1.1	Username or password incorrect
		1.1a. Display message to re-login
	11.1	If another stock is not available
		11.1a. Display message “You Can’t Update now”
	12.1	Stock manager do not enter quantity and put empty field and clicks update button
		12.1a. Display message “quantity field is required”.
	12.2	Stock manager tries to edit item name and item code
		12.2a. Display message as these fields are read only

Fig: 2.1.4.f

<b>Number</b>	S007	
<b>Use Case Name</b>	Manage Suppliers	
<b>Summary</b>	Shopkeeper view supplier details and update.	
<b>Priority</b>		
<b>Preconditions</b>	Shopkeeper should log into the system	
<b>Postconditions</b>	Shopkeeper manages supplier details	
<b>Primary Actor</b>	Shopkeeper	
<b>Trigger</b>	Shopkeeper view supplier details and update.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Shopkeeper login to the system using their login credential
	2	Shopkeeper move on to the All-service home page
	3	Shopkeeper should click on the view button of the supplier management section.
	4	System displays supplier management home page
	5	Four categories are shown on the supplier management home page.
	6	Shopkeeper goes through the add supplier button will navigate to the add supplier page.
	7	Shopkeeper fill the supplier details and click Add supplier button.
	8	Shopkeeper return to the view supplier page
	9	Shopkeeper can update any supplier
	10	Shopkeeper will redirect to the supplier update page
	11	Shopkeeper can update the supplier details and click Update supplier button.
	12	Shopkeeper can delete any supplier
	13	Shopkeeper goes through the add supplier payment button will navigate to the add supplier page.
	14	Shopkeeper fill the supplier payment details and click Add supplier payment button.
	15	Shopkeeper return to the view supplier payment page
	16	Shopkeeper can update any supplier payment
	17	Shopkeeper will redirect to the supplier payment update page
	18	Shopkeeper can update the supplier payment details and click Update supplier payment button.
<b>Extensions</b>	<b>Step</b>	<b>Action</b>
	1.1	Username or password incorrect
		1.1a. Display message to re-login
	7.1	If Shopkeeper put invalid input
		7.1a. Display error message as invalid input
	7.2	If Shopkeeper click add button with empty fields
		7.2a. Display error message as empty fields
	12.1	System will display warning message

Fig: 2.1.4.h

<b>USE CASE ID</b>	01	
<b>USE CASE NAME</b>	Place Order	
<b>SUMMARY</b>	The customer selects products and places an order.	
<b>PRIORITY</b>	High	
<b>PRECONDITIONS</b>	The customer should have registered to the system	
<b>POSTCONDITIONS</b>	The customer has placed an order successfully	
<b>PRIMARY ACTOR(S)</b>	Customer	
<b>TRIGGER</b>	Customer views products in the “ <del>Ravishka MiniSupermarket</del> ” web application	
<b>MAIN SCENARIO</b>	<b>Step</b>	<b>action</b>
	01	Customer login to the system using his login credential
	02	Customer lands on the home page
	03	The customer views the product catalog on the home page.
	04	The system displays the customer home page
	05	Four categories are shown on the customer home page. Each category holds products under that category
	06	The customer goes through the view path in the “frozen items” category
	07	The customer can see all frozen items
	08	The customer can add items to the cart from the “Add to cart” button
	09	The customer can view the items in the cart by clicking the cart icon.
	10	The system <u>renders</u> the total of each product type and the subtotal of the order.
	11	The system asks to continue shopping or proceed to checkout.
	12	The customer clicks on the <u>proceed</u> to checkout button.
	13	The system prompts a form with all customer details and delivery details.
	14	The customer fills in the details of the form.
	15	Finally, the customer clicks on the “Place Order” button.
	16	The system will display an Order Id with an “order success” message
<b>EXTENSIONS</b>	<b>Step</b>	<b>Branching action</b>
	01.a	The system notifies the username or password is incorrect
	01.b	The system displays an error message and gives three times to re-enter

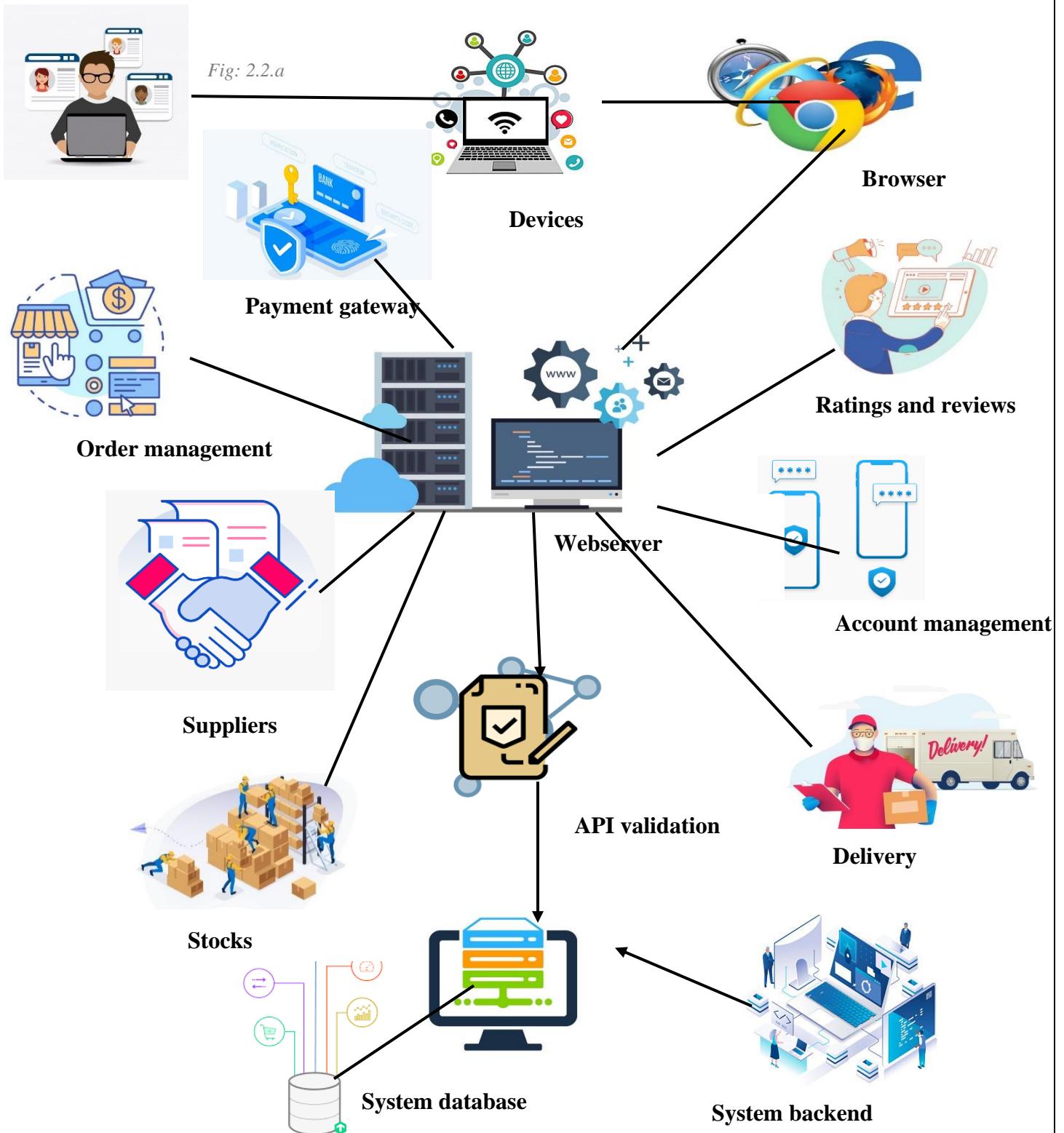
Fig: 2.1.4.h

	01.c	If the customer has forgotten the current password system asks to change the password
	01.d	Ask to enter the verification code
	01.e	Allows to create a new password
	07.a	If the customer wants to see more details of a product he can click on the product
	07.b	The system prompts the more details page of the particular product.
	09.a	If the customer wants to update the product quantity, he can use the +,- buttons.
	09.b	If the customer wants to remove a product from the cart, he can click on the <u>delete</u> button.
	11.a	The customer selects the ‘continue shopping’ option.
	11.b	The system will redirect to the products page.
	15.a	The customer has clicked on the “Place Order” button with empty fields in the form or wrong details.
	15.b	The system prompts error messages.

*Fig: 2.1.4.h*

## 2.2 Design

High level architecture diagram



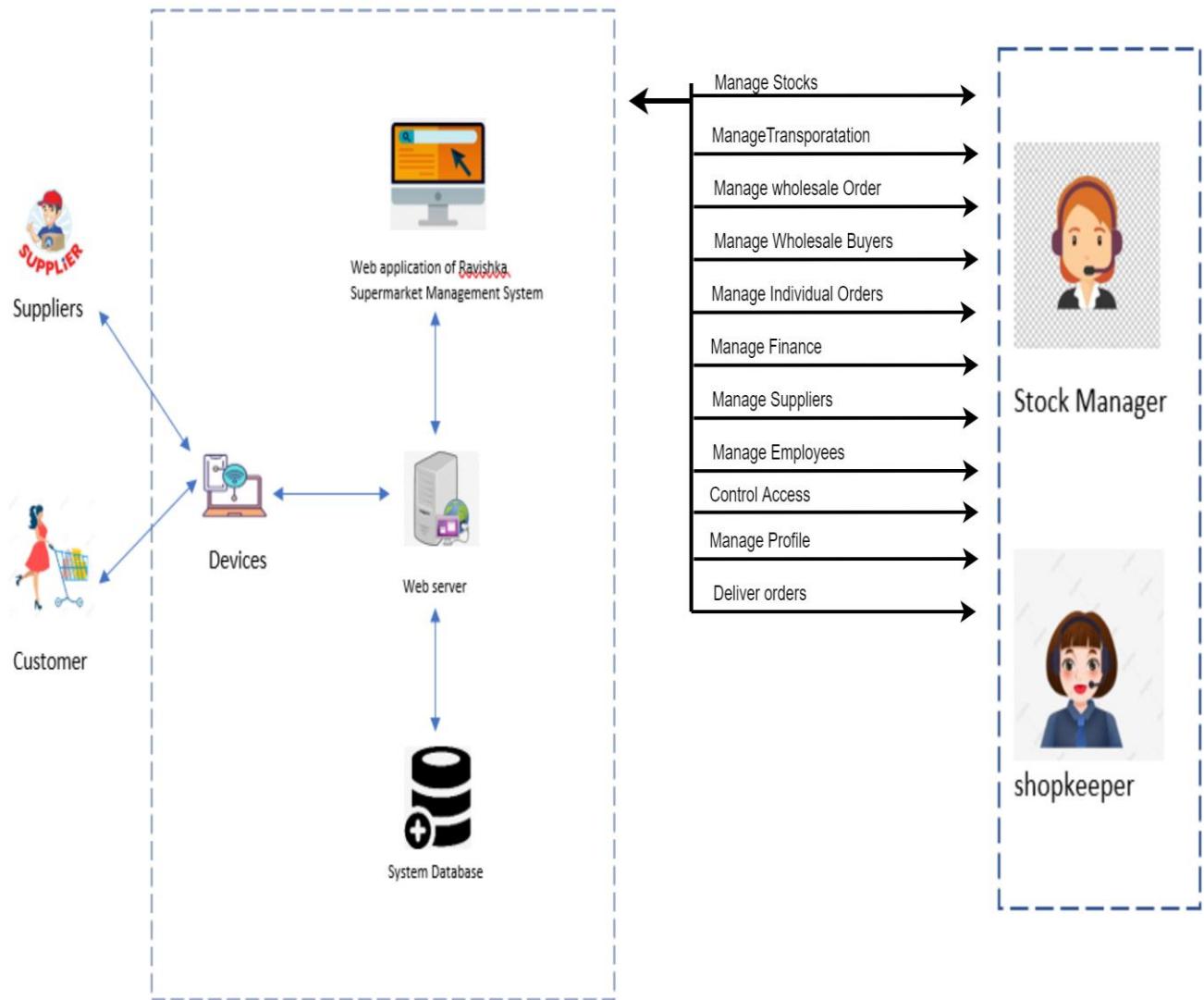
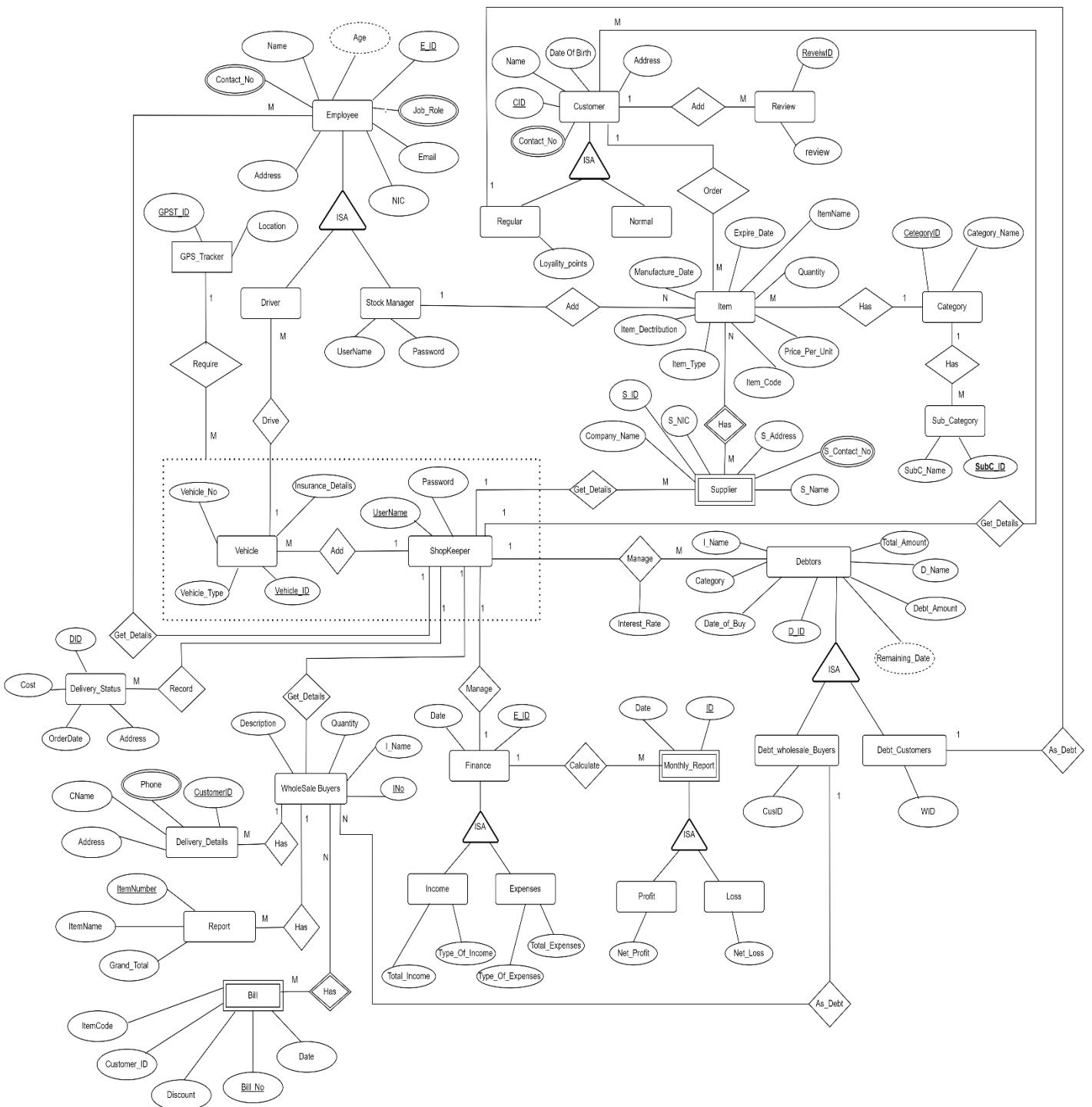


Fig: 2.2.b

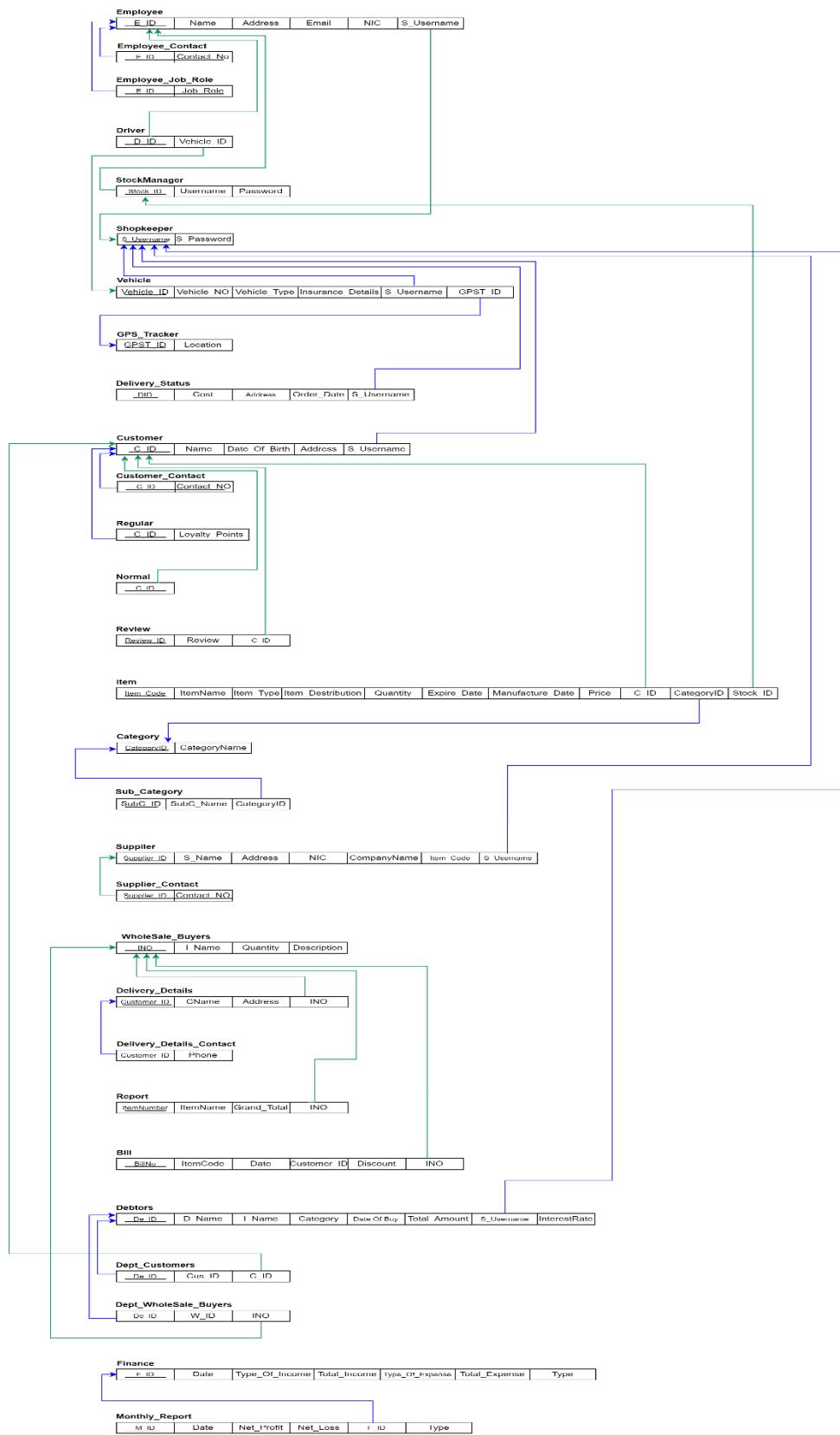
## ER diagram



*Fig: 2.2.c*

## Relational Schema

Fig: 2.2.d



**Shopkeeper**

<u>S_Username</u>	<u>S_Password</u>
-------------------	-------------------

**Employee**

<u>E_ID</u>	<u>E_Name</u>	Address	Email	NIC	<u>S_Username</u>
-------------	---------------	---------	-------	-----	-------------------

**Employee\_Contact**

<u>E_ID</u>	<u>Contact_No</u>
-------------	-------------------

**Employee\_Job\_Role**

<u>E_ID</u>	<u>Job_Role</u>
-------------	-----------------

**GPS\_Tracker**

<u>GPST_ID</u>	Location
----------------	----------

**Vehicle**

<u>Vehicle_ID</u>	<u>Vehicle_No</u>	<u>Vehicle_Type</u>	<u>Insurance_ID</u>	<u>S_Username</u>	<u>GPST_ID</u>
-------------------	-------------------	---------------------	---------------------	-------------------	----------------

**Driver**

<u>D_ID</u>	<u>Vehicle_ID</u>
-------------	-------------------

**Stockmanager**

<u>Stock_ID</u>	Username	Password
-----------------	----------	----------

**Delivery\_Status**

<u>D_ID</u>	Cost	Address	OrderDate	<u>S_Username</u>
-------------	------	---------	-----------	-------------------

**Customer**

<u>C_ID</u>	<u>C_Name</u>	<u>Date_Of_Birth</u>	Address	<u>S_Username</u>
-------------	---------------	----------------------	---------	-------------------

**Customer\_Contact**

<u>C_ID</u>	<u>Contact_No</u>
-------------	-------------------

**Regular**

<u>C_ID</u>	Loyalty_Points
-------------	----------------

**Normal**

<u>C_ID</u>
-------------

**Review**

<u>Review_ID</u>	Review	C_ID
------------------	--------	------

**Category**

<u>Category_ID</u>	CategoryName
--------------------	--------------

**Sub\_Category**

<u>Subc_ID</u>	Subc_Name	Category_ID
----------------	-----------	-------------

**Item**

<u>Item_Code</u>	ItemName	Item_Type	Description	Quantity	Expire_Date
Manufacture_Date	Price	C_ID	Category_ID	Stock_ID	

**Supplier**

<u>Supplier_ID</u>	S_Name	Address	NIC	CompanyName	<u>Item_Code</u>
S_Username					

**Supplier\_Contact**

<u>Supplier_ID</u>	<u>Contact_No</u>
--------------------	-------------------

**WholeSale\_Buyers**

<u>W_ID</u>	W_Name	Location
-------------	--------	----------

**Delivery\_Details**

<u>C_ID</u>	C_Name	Address	W_ID
-------------	--------	---------	------

**Delivery\_Details\_Contact**

<u>C_ID</u>	Phone
-------------	-------

**Report**

<u>Report_No</u>	ItemName	Grand_Total	W_ID
------------------	----------	-------------	------

**Bill**

<u>Bill_No</u>	ItemCode	Date	C_ID	Discount	<u>W_ID</u>
----------------	----------	------	------	----------	-------------

**Debtors**

<u>De_ID</u>	De_Name	I_Name	Category	Date_Of_Buy	Debt_Amount
Total_Amount	S_Username	Interest_Rate			

**Dept\_Customers**

<u>De_ID</u>	C_ID
--------------	------

**Dept\_Wholesale\_Buyers**

<u>De_ID</u>	W_ID
--------------	------

**Finance**

<u>F_ID</u>	Date	Type_Of_Income	Total_Income	Type_Of_Expense
Total_Expense	Type			

**Monthly\_Report**

<u>M_ID</u>	<u>F_ID</u>	Date	Net_Profit	Net_Loss	Type
-------------	-------------	------	------------	----------	------

*Fig: 2.2.e*

Foreign key (E\_ID) references Employee (E\_ID) ---(Employee\_Contact)  
Foreign key (E\_ID) references Employee (E\_ID) ---(Employee\_Job\_Role)

Foreign key (S\_Username) references Shopkeeper (S\_Username)  
Foreign key (GPST\_ID) references GPS\_Tracker (GPST\_ID)

Foreign key (D\_ID) references Employee (D\_ID)

Foreign key (S\_Username) references Shopkeeper (S\_Username) ---(Delivery\_Status)

Foreign key (S\_Username) references Shopkeeper (S\_Username) ---(Customer)

Foreign key (C\_ID) references Customer (C\_ID) ---(Customer\_Contact)

Foreign key (C\_ID) references Customer (C\_ID) ---(Regular-ISA Relationship)

Foreign key (C\_ID) references Customer (C\_ID) ---(Normal-ISA Relationship)

Foreign key (C\_ID) references Customer (C\_ID) ---(Review)

Foreign key (CategoryID) references Category (CategoryID)

Foreign key (C\_ID) references Customer (C\_ID)  
Foreign key (CategoryID) references Category (CategoryID)  
Foreign key (Stock\_ID) references Stockmanager (Stock\_ID)

Foreign key (Item\_Code) references Item (Item\_Code)  
Foreign key (S\_Username) references Shopkeeper (S\_Username)

Foreign key (Supplier\_ID) references Customer (Supplier\_ID) ---(Supplier\_Contact)

Foreign key (W\_ID) references Wholesale\_Buyers (W\_ID)

Foreign key (C\_ID) references Delivery\_Details (C\_ID) ---(Delivery\_Details\_Contact)

Foreign key (W\_ID) references Wholesale\_Buyers (W\_ID) ---(Report)

Foreign key (W\_ID) references Wholesale\_Buyers (W\_ID) ---(Bill)

Foreign key (S\_Username) references Shopkeeper (S\_Username) ---(Debtors)

Foreign key (De\_ID) references Debtors (De\_ID) ---(Dept\_Customers-ISA Relationship)  
Foreign key (C\_ID) references Customer (C\_ID)

Foreign key (De\_ID) references Debtors (De\_ID) ---(Dept\_Wholesale\_Buyers-ISA Relationship)

Foreign key (W\_ID) references Wholesale\_Buyers (W\_ID)

Foreign key (F\_ID) references Monthly\_Report (F\_ID)



## User interfaces

Our web application consists of two parts.

### ✓ Admin Part:

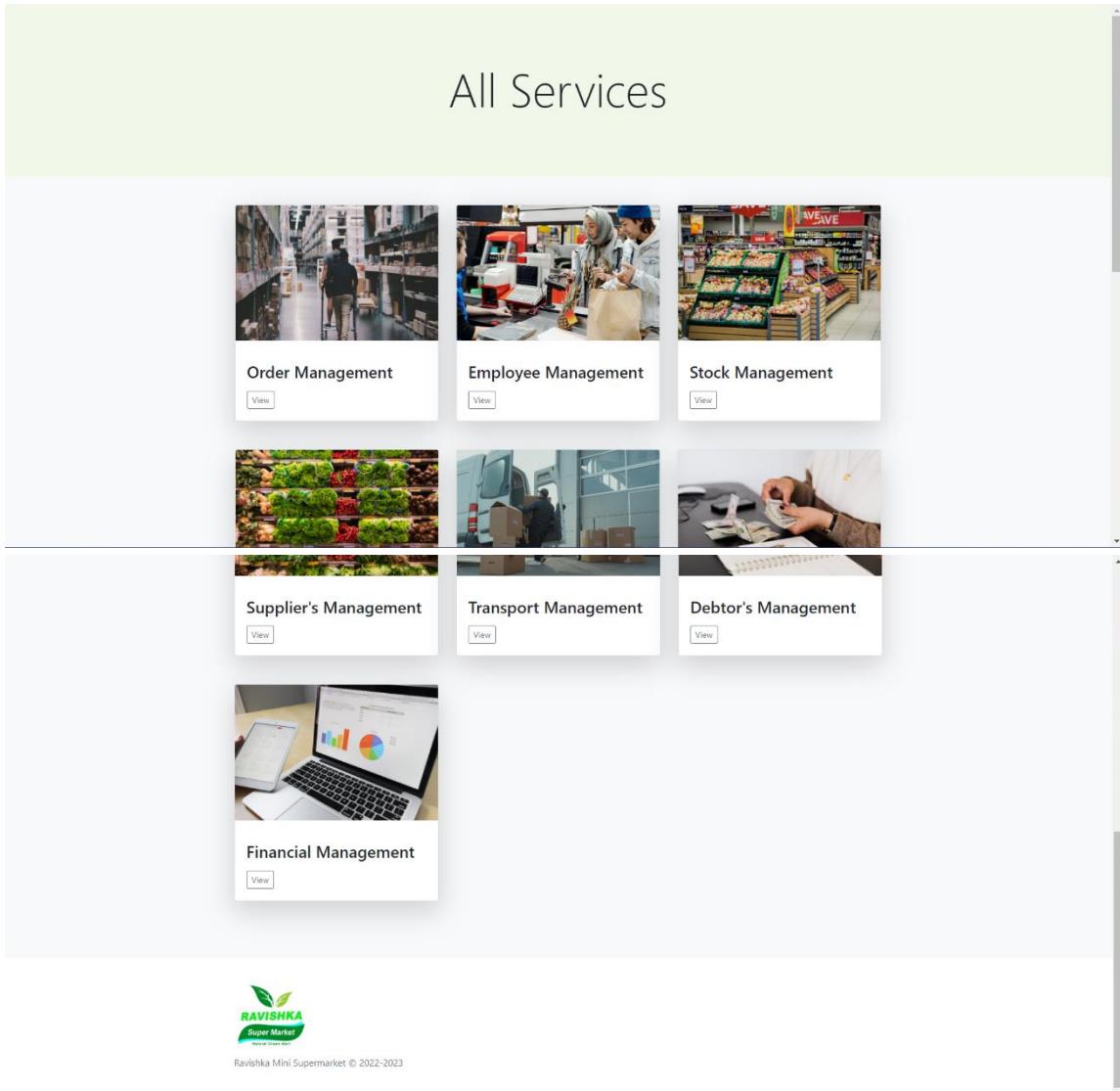


Fig: 2.2.1

## Employee Management (IT20263362)

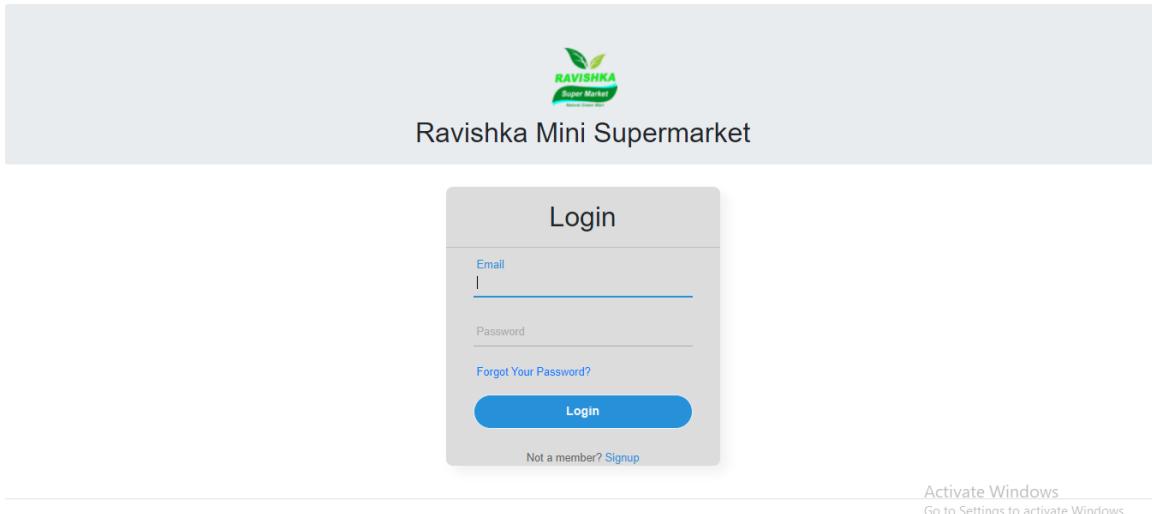


Fig: 2.2.2

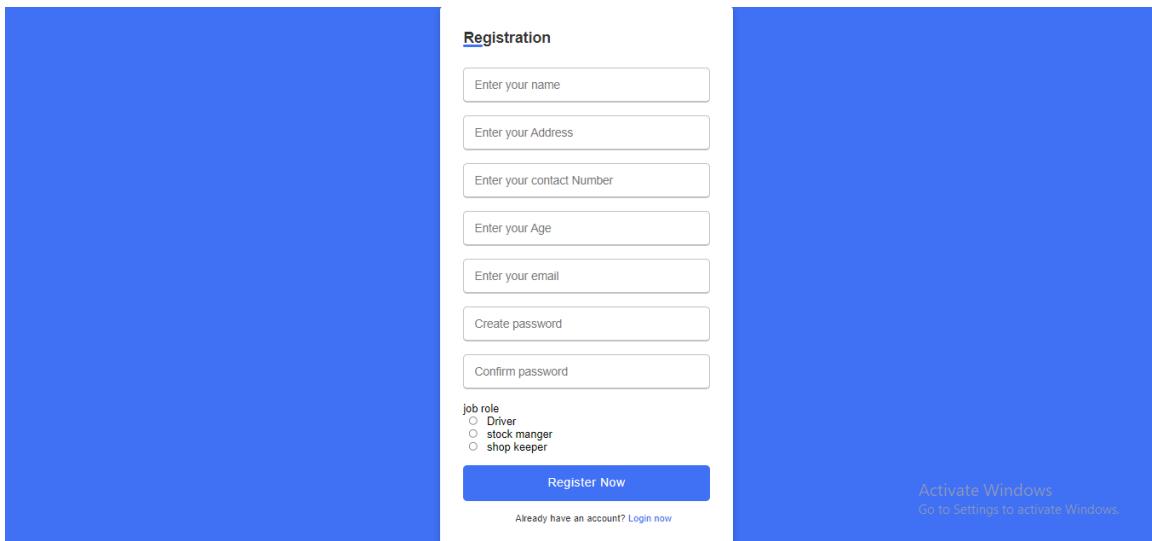


Fig: 2.2.3

Ravishka Mini Supermarket

User Profile

User Name: praveen

Address: No.182,Nawodya Mavatha.

Contact Number: 0715874433

Age: 25

Email: 98@gmail.com

Job Role: Driver

Update Monthly Report Delete

Logout

Activate Windows  
Go to Settings to activate Windows.

This screenshot shows the 'User Profile' section of the Ravishka Mini Supermarket application. It contains input fields for User Name (praveen), Address (No.182,Nawodya Mavatha.), Contact Number (0715874433), Age (25), Email (98@gmail.com), and Job Role (Driver). Below the form are three buttons: 'Update', 'Monthly Report', and 'Delete'. A 'Logout' button is located at the bottom left. At the bottom right, there is a watermark for 'Activate Windows' with the text 'Go to Settings to activate Windows.'

Fig: 2.2.4

Ravishka Mini Supermarket

Update page

User Name: praveen

Address: No.182,Nawodya Mavatha.

Contact Number: 0715874433

Age: 25

Email: 98@gmail.com

Job Role:  Driver  
 stock manger  
 shop keeper

Back Update

Activate Windows  
Go to Settings to activate Windows.

This screenshot shows the 'Update page' section of the application. It includes all the same fields as the User Profile page: User Name (praveen), Address (No.182,Nawodya Mavatha.), Contact Number (0715874433), Age (25), Email (98@gmail.com), and Job Role (Driver). The 'Job Role' field has a radio button group where 'Driver' is selected. Below the form are 'Back' and 'Update' buttons. A watermark for 'Activate Windows' is present at the bottom right.

Fig: 2.2.5



This screenshot shows the services offered by Ravishka Mini-Supermarket. On the left, the Ravishka logo and the text "Ravishka Mini-Supermarket" are visible. In the center, under the heading "Services", there are two options: "Order management for wholesale buyers" and "Employee management". On the right, there is an "Activate Windows" message with a link to "Go to Settings to activate Windows".

Fig: 2.2.6

This screenshot shows a "Delete User" form for Ravishka Mini Supermarket. At the top right is the Ravishka logo. Below it, the text "Ravishka Mini Supermarket" is displayed. A central box titled "Delete User" contains the following fields:

- User Name: praveen
- Address: No.182, Navodya Mawatha,
- Contact Number: 0715874433
- Age: 25
- Email: 98@gmail.com
- Job Role: Driver

A "Delete" button is located at the bottom right of this box. To the right of the box, there is an "Activate Windows" message with a link to "Go to Settings to activate Windows".

Fig: 2.2.7

**Ravishka Mini Supermarket**

**form**

User name:

Month:  Level:

Total of work days:  
 OT hours:  
 Number of work days:  
 Total Salary:

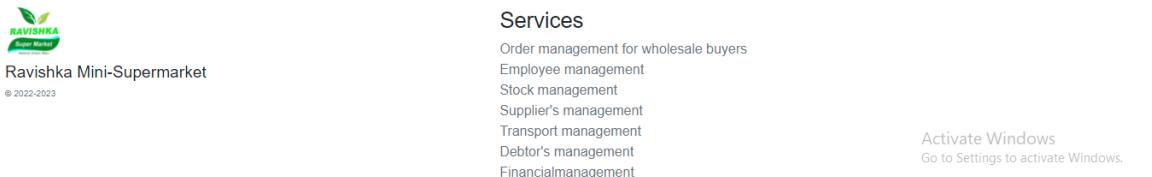


*Fig: 2.2.8*

**Ravishka Mini Supermarket**

[Download as PDF](#)

User id	Month	Level	Total of work days	OT hours	Number of work days	Total Salary
5	1	Level 1	30	10	20	21000.00
10	1	Level 1	30	20	20	26000.00
5	1	Level 1	0	0	0	0.00
9	1	Level 2	31	31	31	52700.00



*Fig: 2.2.9*

## Order management for wholesale buyers (IT20645670)



### Wholesale Buyer Order Form

[Create Order](#)

Order ID	Product	Quantity	Description	Action
9	Milk Powder	1500kg	1kg packs	<a href="#">Edit</a> <a href="#">Delete</a>
8	Tea	1200kg	400 g	<a href="#">Edit</a> <a href="#">Delete</a>
7	Lux Soap	200	Five Boxes	<a href="#">Edit</a> <a href="#">Delete</a>
6	Milk Powder	100 Boxes	400 g	<a href="#">Edit</a> <a href="#">Delete</a>
5	Flour	1400kg	250g Packets	<a href="#">Edit</a> <a href="#">Delete</a>

[« Previous](#)

[Next »](#)

Showing 1 to 5 of 9 results

Fig: 2.2.10

### Ravishka Mini Supermarket

#### Add Order

[Back](#)

**Product**

**Quantity**

**Description**

[Submit](#)

Fig: 2.2.11

# Ravishka Mini Supermarket

## Edit Order

[Back](#)

**Product**

Tea

**Quantity**

1200kg

**Description**

400 g

[Submit](#)

Fig: 2.2.12



# Ravishka Mini Supermarket

## Wholesale Buyer Delivery Details Form

[Add Details](#)

ID	Customer Name	Customer Email	Customer Address	Contact No	Delivery Instructions	Action
12	Jani Gunarathna	janis@gmail.com	164A,Jaffna	742319328	Putthalamai Rd	<a href="#">Update</a> <a href="#">Delete</a>
11	Sadaru Silva	sadaru@gmail.com	Kandy Rd, Malabe	772243987	Kandy Colombo Bus Route	<a href="#">Update</a> <a href="#">Delete</a>
10	sandu perera	sandu@mail.com	231B, wiharamawatha, weligama	762243235	Galle Rd	<a href="#">Update</a> <a href="#">Delete</a>
9	Piyumi Madusha	piu@gmail.com	123A,Pannipitiya,Colombo	772243235	Malabe Road	<a href="#">Update</a> <a href="#">Delete</a>
8	Hirushi Umendra	hirushiumendra1@gmail.com	106/A, Bandarawaththa, Makavita, Navimana, Matara.	712390998	matara	<a href="#">Update</a> <a href="#">Delete</a>

Fig: 2.2.13

# Ravishka Mini Supermarket

## Add Delivery Details

[Back](#)

**Customer Name:**

Customer Name

**Customer Email:**

Customer Email

**Customer Address:**

Customer Address

**Contact No:**

Contact No

**Delivery Instructions:**

Delivery Instructions

[Save](#)

Fig: 2.2.14

# Ravishka Mini Supermarket

## Edit Delivery Details

[Back](#)

**Customer Name:**

Hirushi Umendra

**Customer Email:**

hirushiumendra1@gmail.com

**Customer Address:**

106/A, Bandarawaththa, Makavita, Navimana, Matara.

**Contact No:**

0712390998

**Delivery Instructions:**

matara

[Update](#)

Fig: 2.2.15

No.	Product	Qty	Price
1	Milk Powder	25	880

Receipt				
No.	Product Name	Quantity	Price	Total
1	Brown	16	330	5280
2	Prima	19	400	7600
3	Milk	25	880	22000
			<b>Sub Total: Rs.</b>	<b>34880</b>
			<b>Tax (5%) : Rs.</b>	<b>1744.00</b>
			<b>Grand Total: Rs.</b>	<b>36624.00</b>

Fig: 2.2.16

## Stock management (IT20621292)

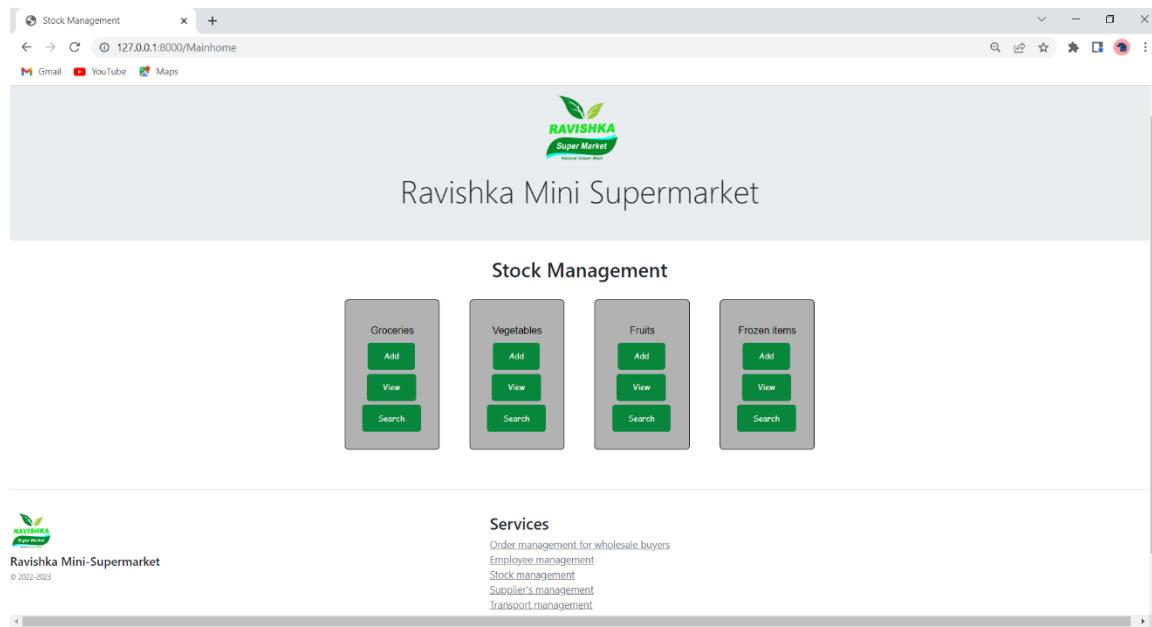


Fig: 2.2.17

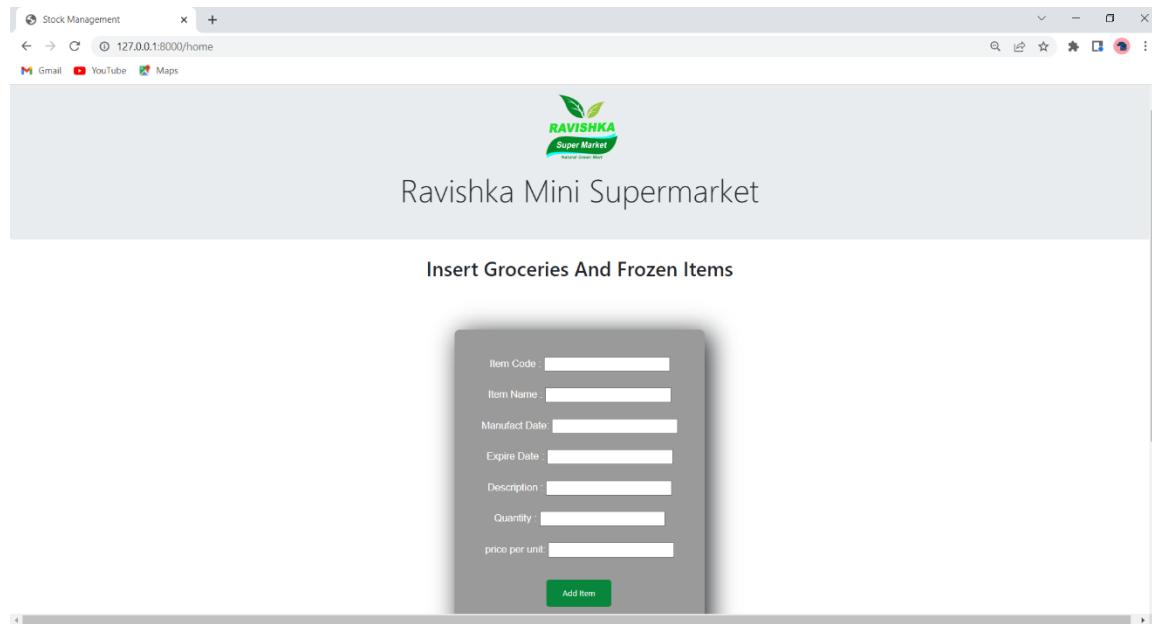


Fig: 2.2.18

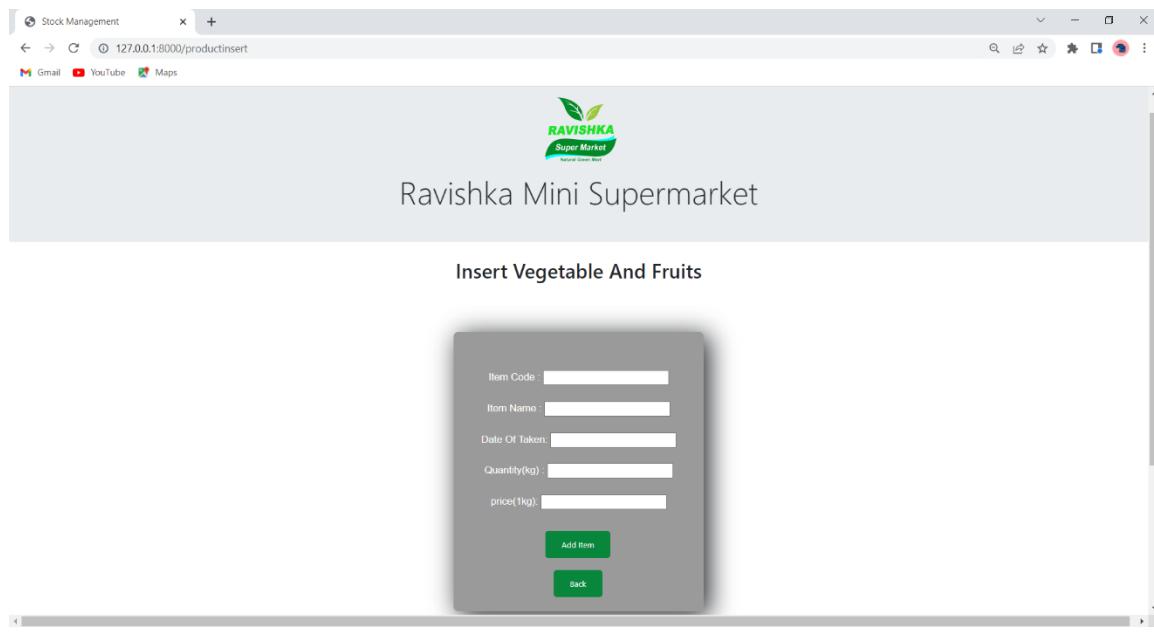


Fig: 2.2.19

Ravishka Mini Supermarket

Search Groceries And Frozen Items

Item Code	Item Name	Manufacture Date	Expire Date	Description	Quantity	Price per unit	Total Amount	Option
G011	Shampoo	2022-02-05	2022-05-30	sunsilk thick and long(300ml)	9 - Out Of Stock	150	1350	<input type="button" value="Update"/> <input type="button" value="Delete"/>
G012	Shampoo	2022-02-05	2022-05-24 - This Stock is Expired	Dove (250ml)	30	300	9000	<input type="button" value="Update"/> <input type="button" value="Delete"/>

**Services**

- Order management for wholesale buyers
- Employee management
- Stock management
- Supplier's management
- Transport management
- Debtor's management
- Financial management

Fig: 2.2.20

Ravishka Mini Supermarket

Search Vegetables And Fruits

Item Code	Item Name	Date Of Taken	Quantity (kg)	Price per unit (1kg)	Total Amount	Option
V002	Leeks	2022-02-17	20	56	1120	<button>update</button> <button>Delete</button>

Back

**Services**

- Order management for wholesale buyers
- Employee management
- Stock management
- Supplier's management
- Transport management
- Debtor's management
- Financial management

Fig: 2.2.21

Ravishka Mini Supermarket

Groceries and frozen items stock

Item Code	Item Name	Manufacture Date	Expire Date	Description	Quantity	Price per unit	Total Amount	Option
G001	sanitizer spray	2022-02-05	2022-12-17	Lifeboy hand sanitizer	20	100	2000	<button>Update</button>
G009	cheese	2022-02-05	2022-05-27	Abewela	30	10	300	<button>Update</button>
G011	Shampoo	2022-02-05	2022-05-30	sunilk thick and long(500ml)	9 - Out Of Stock	150	1350	<button>Update</button>
G020	tea bag pack	2022-02-05	2022-05-29	flora	30	30	900	<button>Update</button>
G012	Shampoo	2022-02-05	2022-05-24: This Stock is Expired	Dove (250ml)	30	300	9000	<button>Update</button>
G002	Soya	2022-02-05	2023-02-05	Chiken flavor	8 - Out Of Stock	150	1200	<button>Update</button>

Get Stock Report Back

**Services**

- Order management for wholesale buyers
- Employee management
- Stock management
- Supplier's management
- Transport management
- Debtor's management
- Financial management

Fig: 2.2.22

Ravishka Mini Supermarket

Vegetables and fruits stock

Item Code	Item Name	Date Of Taken	Quantity (kg)	Price per unit(kg)	Total Amount	option
V001	beans	2022-02-17	50	100	5000	<button>Update</button>
V003	Pumkin	2022-02-17	3 - Out Of Stock	280	840	<button>Update</button>
V004	Tommatos	2022-02-17	25	123	3075	<button>Update</button>
F012	Guwava	2022-02-17	24	123	2952	<button>Update</button>
V002	Leeks	2022-02-17	2 - Out Of Stock	56	112	<button>Update</button>
V010	Orange	2022-02-17	20	100	2000	<button>Update</button>

Get Stock Report Back

**Services**  
Order management for wholesale buyers

Fig: 2.2.23

Ravishka Mini Supermarket

Update Grocery Details And Frozen Item Details

Item Code :

Item Name :

Manufacture Date :

Expire Date :

Description :

Quantity :

price per unit :

update Back

Fig: 2.2.24

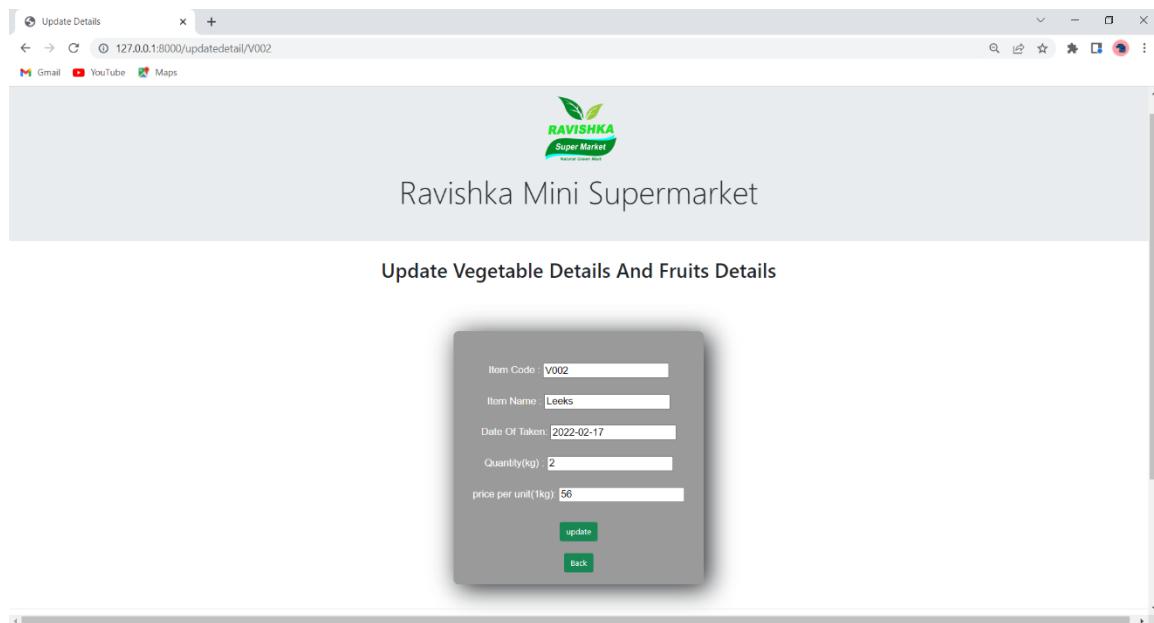


Fig: 2.2.25

Item Code	Item Name
G001	sanitizer spray
G009	cheese
G011	Shampoo
G020	tea bag pack
G012	shampoo
G002	Soya

Fig: 2.2.26

## Supplier's management

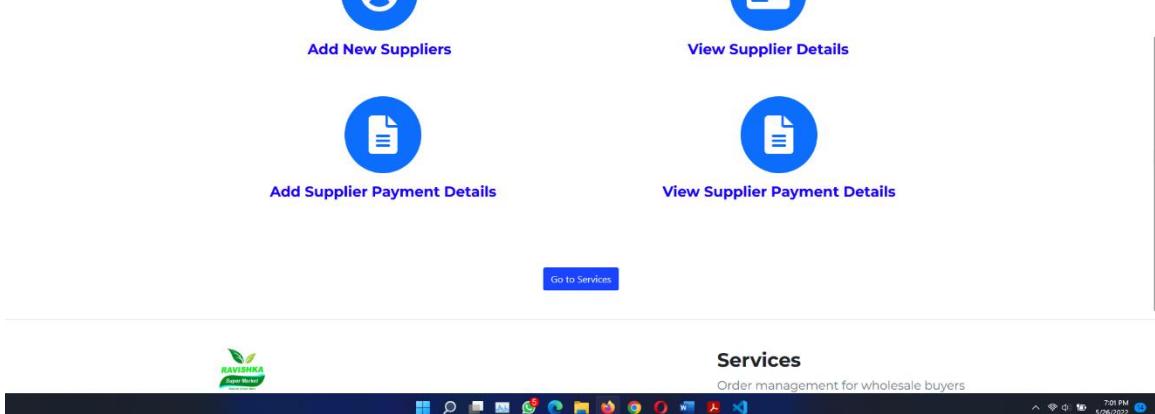
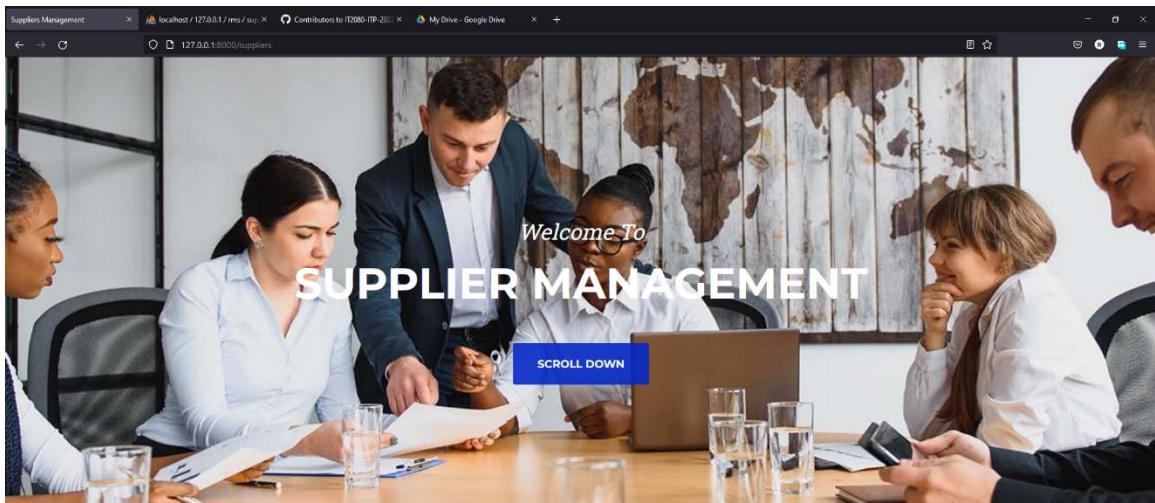


Fig: 2.2.27

**Ravishka Mini Supermarket**

**Add Supplier**

Supplier Name:	<input type="text"/>
Supplier ID:	<input type="text"/>
Company Name:	<input type="text"/>
NIC:	<input type="text"/>
Item Name:	<input type="text"/>
Contact Number:	<input type="text"/>

[Home](#) [Add Supplier](#) [CLEAR ENTRY](#)

---

**Services**

- Order management for wholesale buyers
- Employee management
- Stock management
- Supplier's management
- Transport management
- Debtor's management
- Financial management

Fig: 2.2.28

**Ravishka Mini Supermarket**

**View Suppliers**

Index	Supplier Name	Supplier ID	Company Name	NIC	Item Name	Contact Number	Actions
1	Ravishka Pravindika	s0001	RMS company	992854359V	Biscuit	0765339505	<a href="#">Edit</a> <a href="#">Delete</a>
2	Samindya	s0002	SSN	2001452187	Shampoo	0715448762	<a href="#">Edit</a> <a href="#">Delete</a>
5	Chandima	s0003	SCS	992468753V	Laptop	0754298563	<a href="#">Edit</a> <a href="#">Delete</a>
22	Iod harsana	s0023	KVD	994368752V	bag	0719387657	<a href="#">Edit</a> <a href="#">Delete</a>

[Home](#)

---

**Services**

- Order management for wholesale buyers
- Employee management
- Stock management
- Supplier's management
- Transport management
- Debtor's management
- Financial management

Fig: 2.2.29

Ravishka Mini Supermarket  
Update Supplier

Supplier Name:

Supplier ID:

Company Name:

NIC:

Item Name:

Contact Number:

**Services**

- Order management for wholesale buyers
- Employee management
- Stock management
- Supplier's management
- Transport management

Fig: 2.2.330

Ravishka Mini Supermarket  
Add Suppliers Payment

Supplier Name:

Supplier ID:

Item Name:

Quantity:

Date:

Payment Amount:

**Services**

- Order management for wholesale buyers
- Employee management
- Stock management
- Supplier's management
- Transport management

Fig: 2.2.31

The screenshot shows a web browser window with the title 'View Suppliers Payment'. The URL is '127.0.0.1:8000/viewsupplierpayment'. The page features the Ravishka Super Market logo at the top. Below it, the heading 'Ravishka Mini Supermarket' and 'View Suppliers Payment' are displayed. A table lists seven supplier payment entries with columns for Index, Supplier Name, Supplier ID, Item Name, Quantity, Date, Payment Amount, and Actions (Edit button). At the bottom of the table is a blue 'Home' button.

Index	Supplier Name	Supplier ID	Item Name	Quantity	Date	Payment Amount	Actions
1	Ravishka	s0001	Biscuit	100	2022-04-28	10000	Edit
2	Sanju	s0005	Shampoo	200	2022-04-29	50000	Edit
3	Kavindu	s0003	Bug	500	2022-05-06	100000	Edit
4	Kavishka	s0052	Drink	5000	2022-04-30	100000	Edit
5	DSBTW4	ERTE4V	ERTVE	3432	2022-04-25	3253	Edit
6	adasa	adasas	adasa	323	2022-05-16	23445	Edit
7	ravishka Pravindika	s0001	BUscuits	500	2022-05-23	427632	Edit

[Home](#)

**Ravishka Mini-Supermarket**  
© 2022-2023

**Services**  
Order management for wholesale buyers  
Employee management  
Stock management  
Supplier's management  
Transport management  
Debtor's management  
Financialmanagement

7:03 PM 5/26/2022

Fig: 2.2.32

The screenshot shows a web browser window with the title 'Update Suppliers Payment Details'. The URL is '127.0.0.1:8000/updatesupplierpayment/view/1'. The page features the Ravishka Super Market logo at the top. Below it, the heading 'Ravishka Mini Supermarket' and 'Update Suppliers Payment Details' are displayed. A form is present with fields for Supplier Name (Ravishka), Supplier ID (s0001), Item Name (Biscuit), Quantity (100), Date (04/28/2022), and Payment Amount (10000). At the bottom are 'Update Supplier Payment Details' and 'Cancel' buttons. To the right, a 'Services' sidebar lists various management functions. The system status bar at the bottom shows '7:03 PM 5/26/2022'.

**Ravishka Mini Supermarket**  
**Update Suppliers Payment Details**

Supplier Name:

Supplier ID:

Item Name:

Quantity:

Date:

Payment Amount:

[Update Supplier Payment Details](#) [Cancel](#)

**Ravishka Mini-Supermarket**  
© 2022-2023

**Services**  
Order management for wholesale buyers  
Employee management  
Stock management  
Supplier's management  
Transport management

7:03 PM 5/26/2022

Fig: 2.2.33

## Transport management (IT20648336)

Add Driver [BACK](#)

Driver Name

Vehicle

Driver Profile Image  
 Choose File | No file chosen

Assign

[Save Driver](#)



### Services

Order management for wholesale buyers  
Employee management  
Stock management  
Supplier's management  
Transport management  
Debtor's management  
Financialmanagement

Fig: 2.2.34

Ravishka Mini Supermarket

[VIEW](#) [UPDATE](#) [DELETE](#)

Add New Vehicle

Vehicle Number  
 Enter Vehicle Number

Vehicle Type  
 Enter Vehicle Type

License ID  
 Enter license ID

Renew License Details  
 Enter renewe

[SAVE](#)

Fig: 2.2.35

ID	Driver Name	vehicle	Assign	Profile Image	Edit	Delete
55	Amal sampath	Truck	Assign		<a href="#">Edit</a>	<a href="#">Delete</a>
56	Kamal Peries	Truck	not		<a href="#">Edit</a>	<a href="#">Delete</a>
57	Nimal Jayasooriya	Truck	not		<a href="#">Edit</a>	<a href="#">Delete</a>
58	W.S.Perera	Truck	New		<a href="#">Edit</a>	<a href="#">Delete</a>
59	Kamal Sampath	Truck	Assign		<a href="#">Edit</a>	<a href="#">Delete</a>
60	shashini Sankalpana	Truck	not		<a href="#">Edit</a>	<a href="#">Delete</a>

Fig: 2.2.36

#	Name	Assign	
1	Amal sampath	Assign	<b>2022-05-21 12:51:06</b>
2	Kamal Peries	not	<b>2022-05-22 06:22:38</b>
3	Nimal Jayasooriya	not	<b>2022-05-22 18:31:08</b>
4	W.S.Perera	New	<b>2022-05-22 18:34:53</b>
5	Kamal Sampath	Assign	<b>2022-05-22 18:40:37</b>
6	shashini Sankalpana	not	<b>2022-05-23 11:16:33</b>

Fig: 2.2.37

### New Parcel

[VIEW](#) [UPDATE/DELETE](#) [CHECK DELIVERY ITEM](#)

Sender Name <input type="text"/>	Recipient Name <input type="text"/>
Sender Address <input type="text"/>	Recipient Address <input type="text"/>
Sender Contact Number <input type="text"/>	Recipient Contact Number <input type="text"/>

**Delivery information**

Weight <input type="text"/> Enter Weight	Height <input type="text"/> Enter Height	Length <input type="text"/> Enter Length	width <input type="text"/> Enter Width
price <input type="text"/> price	Status <input type="text"/> Enter Delivery Status	Driver ID <input type="text"/> Enter Driver ID	<a href="#">SAVE</a>

Total(km) :   
1km-rs.50.00:   
[Calculate](#) [Divide](#)  
Total Amount :



Fig: 2.2.38



## Ravishka Mini Supermarket

### Select Dates For Report Generate

Start Date

mm/dd/yyyy [Calendar](#)

End Date

mm/dd/yyyy [Calendar](#)

[Submit](#)

Fig: 2.2.39

The Search results for your query **truck** are :

### Sample vehicle Details

Vehicle Type	Vehicle Number
truck	AB12365
truck	AB12345
Truck	AB0025689
Truck	AB0025689
Truck	AB0025689

Fig: 2.2.40

The screenshot shows the main dashboard of the Ravishka Mini Supermarket application. At the top center is the logo "RAVISHKA Super Market" with the tagline "Natural Green Market". Below the logo, the text "Ravishka Mini Supermarket" is displayed. The dashboard features several green rectangular buttons with white text: "MANAGE DRIVERS", "REPORTS", "GET WHOLESALE BUYERS", "ADD NEW VEHICLE", and "RECORD & CHECK DELIVERY ITEMS". On the left side, there is a sidebar with the "RAVISHKA Super Market" logo, the text "Ravishka Mini-Supermarket", and the copyright notice "© 2022-2023". To the right of the sidebar, under the heading "Services", is a list of management services: Order management for wholesale buyers, Employee management, Stock management, Supplier's management, Transport management, Debtor's management, and Financial management.

Fig: 2.2.41

Update Driver Details [BACK](#)

Driver Name

Vehicle Type

Assign

Driver Profile Image  
 No file chosen  


[Update Driver](#)

Fig: 2.2.42

Update Status

Select a status:

[Update](#)

Fig: 2.2.43



Fig: 2.2.44

The screenshot shows a table titled "Wholesale Buyer Details". The table has columns for Wholesale Buyer ID, Wholesale Buyer Name, Email, and Address. The data includes:

Wholesale Buyer ID	Wholesale Buyer Name	Email	Address
C001	Shashini Sankalpana	shashini@gmail.com	No50, Veyangoda, nittambuwa
002	Amal sunil	amal@gmail.com	No10, Nittambuwa, Gampaha

**Services**

- Order management for wholesale buyers
- Employee management
- Stock management
- Supplier's management
- Transport management
- Debtor's management
- Financialmanagement

Fig: 2.2.45

## Debtor's management (IT20230692)



Fig: 2.2.46

This screenshot displays the 'Insert Regular Customer Details' form. The form is titled 'Insert Regular Customer Details' and includes the following fields:

- D\_CID (input field)
- Regular Customer Debtor's Name (input field)
- Select Plan (radio buttons for Months 1, 2, or 3)
- Order Amount (input field)
- Interest Rate Per Month (input field)
- Debt Amount (input field)
- Total Amount (input field)
- Date Of The Buy (input field with placeholder mm/dd/yyyy)

A large green 'Submit' button is located at the bottom of the form. Below the form, there is a link 'Go to Debtors management GO ->'.

Fig: 2.2.47


  
**Ravishka Mini Supermarket**

**Search Regular Customers Details**

---

Regular Customer's Name  
sanju

---

Regular Customer's D\_CID  
rd001

---

**Search**

Go to Debtors management [GO](#)

Record No	Customer D_ID	Customer Name	Date of The Buy	Option
1	rd001	sanju	2022-04-20	<a href="#">view</a>
10	rd001	sanju	2022-05-26	<a href="#">view</a>

Fig: 2.2.48


  
**Ravishka Mini Supermarket**

**View Regular Customer's Details**

---

recordNo :  
1

---

Debtor's D\_CID  
rd001

---

Debtor's Name  
sanju

---

Select Plan  
 Months 1    Months 2    Months 3

Interest Rate  
3.5

---

Amount  
40000

---

Debt Amount  
1400

---

Total Amount  
41400

---

Date Of The Buy  
04/20/2022

---

Date of the Remaining  
53 Days

---

**update**

**delete**

Go to Debtors management [GO](#) >

Fig: 2.2.49



## Ravishka Mini Supermarket

**Insert Wholesale Buyers Details**

---

D\_CID

---

Wholesale Debtors' Name

---

Select Plan

Months 1    Months 2    Months 3

Order Amount

---

Interest Rate Per Month

---

Debt Amount

---

Total Amount

---

Date Of The Buy  
mm/dd/yyyy

---

**Submit**

Go to Debtors management [GO ->](#)

Fig: 2.2.50



## Ravishka Mini Supermarket

**Search Wholesale Buyers Details**

---

Wholesale Debtors' Name  
chandima

---

D\_CID  
d001

---

**Search**

Go to Debtors management [GO](#)

Record No	Wholesale Buyer's ID_DCID	Wholesale Buyer's Name	Date of Buy	Option
1	d001	chandima	2022-03-20	<a href="#">View</a>
4	d001	chandima	2022-05-22	<a href="#">View</a>
7	d001	chandima	2022-06-24	<a href="#">View</a>

Fig: 2.2.51



## Ravishka Mini Supermarket

### View Wholesale Buyers Details

recordNo:  
1

Debtor's D\_CID  
d001

Debtor's Name  
chandima

Select Plan  
 Months 1  Months 2  Months 3

Interest Rate  
1

Amount  
500000

Debt Amount  
5

Total Amount  
505

Date Of The Buy  
03/20/2022

Date of the Remaining  
7 Days

[update](#)

[delete](#)

[Go to Debtors management GO ->](#)

Fig: 2.2.52

The screenshot shows a software interface for managing interest rates. At the top is the Ravishka Super Market logo. Below it is the title "Ravishka Mini Supermarket". A main section titled "View Current Interest Rates" displays a table of monthly interest rates for different customer types. A green button labeled "Change Rates" is at the bottom, and a link "Go to Debtors management GO ->" is at the very bottom.

	Monthly	Interest	Rate
	1st Month	2nd Month	3rd Month
Regular Customers	1.7	2.8	3.5
Wholesale Buyers	1.2	2.5	3

Fig: 2.2.53

The screenshot shows a software interface for changing debtors' interest rates. At the top is the Ravishka Super Market logo. Below it is the title "Ravishka Mini Supermarket". A main section titled "Change Debtors' Interest Rates" contains two parts: "Regular Customers Interest Rates" and "Wholesale Buyers Interest Rates", each with input fields for 1st, 2nd, and 3rd months. A green "Submit" button is at the bottom, and a link "Go to Debtors management GO ->" is at the very bottom.

Fig: 2.2.54



# Ravishka Mini Supermarket

## Debtors Report

### Monthly Income from debtors' Interest :

Total Income from Wholesale Buyers Interst :

11905

Total Income from Regular Customer Interst :

8464

Total amount from Interest :

20369

[Download as PDF](#)

### Number Of Wholesale\_Debtors :

Number Of Wholesale\_Debtors :

Pie Chart

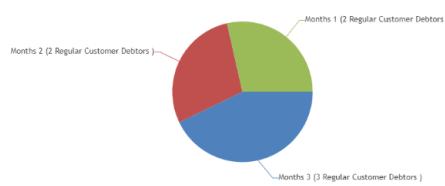
No of Wholesale Buyer Debtors for each plan



### Number Of Regular Customers\_Debtors :

Pie Chart

No of Regular Customer Debtors for each plan



[Go to Debtors management](#)

Fig: 2.2.55

## Financial management (IT20137250)



Welcome To Ravishka Mini Supermarket!

# IT'S TIME TO CALCULATE THE FINANCIAL REPORT

[TELL ME MORE](#)



## FINANCIAL SERVICES

*Lead a comfortable life, take care of your finances.*

**Income Statement**

Represents revenue for the goods sold or services provided during a given period of time.

**Expenditure Statement**

An expenditure is defined as the purchase of goods or services that are expected to have an economic benefit during a specified period.

**Report Statement**

A financial statement is a written record of the activities and financial performance of a company during a set period.



Ravishka Mini-Supermarket

Fig: 2.2.56

  
RAVISHKA Mini Supermarket

### Income Statement

Type of income

Income ID

Income Date

Income Date

Monthly Income

**ADD** **View Income Page**

[\*\*< View all Services\*\*](#) [\*\*Go Expenditure Statement Page >\*\*](#)

Fig: 2.2.57

**Ravishka Mini Supermarket**

**Income Page**

Type of Income	<input type="button" value="Customer Income"/>	<input type="button" value="Interest Income"/>	<input type="button" value="Sales Income"/>
Income ID	<input type="text" value="C05"/>	<input type="text" value="I05"/>	<input type="text" value="S05"/>
Income Date	<input type="text" value="2022-05-23"/>	<input type="text" value="2022-05-23"/>	<input type="text" value="2022-05-23"/>
Monthly Income(Rs.)	<input type="text" value="250000"/>	<input type="text" value="125000"/>	<input type="text" value="1000000"/>
<input type="button" value="UPDATE"/> <input type="button" value="DELETE"/> <input type="button" value="UPDATE"/> <input type="button" value="DELETE"/> <input type="button" value="UPDATE"/> <input type="button" value="DELETE"/>			

Fig: 2.2.58

**Ravishka Mini Supermarket**

**Update Customer Income**

Type of Income	<input type="button" value="Customer Income"/>
Income ID	<input type="text" value="C05"/>
Income Date	<input type="text" value="2022-05-23"/>
Monthly Income(Rs.)	<input type="text" value="250000"/>
<input type="button" value="UPDATE"/>	

Fig: 2.2.59

Pie chart of monthly income

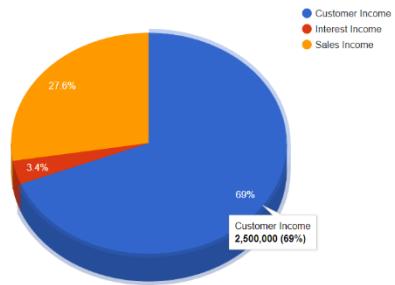


Fig: 2.2.60

Pie chart of monthly expenses

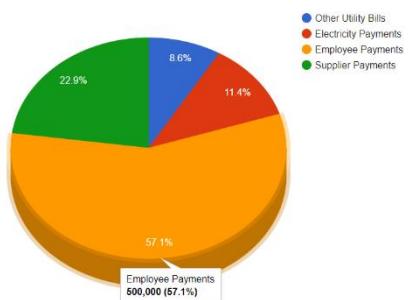


Fig: 2.2.61

  
RAVISHKA  
Super Market  
Natural Groceries Mart

## Ravishka Mini Supermarket

### Expenditure Statement

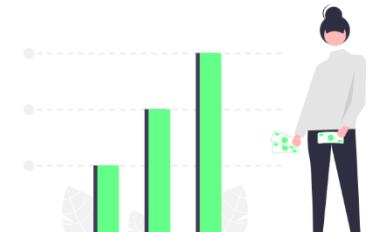
Type of expense

Expense ID

Expense Date

Monthly Expense

**ADD** **View Expense Page**



< Back Next >

  
RAVISHKA  
Super Market

**Services**  
Order management for wholesale buyers  
Employee management

Fig: 2.2.62

**Ravishka Mini Supermarket**

**Expenditure Page**

Type of Expense	Other Utility Bills	Electricity Payments	Employee Payments	Supplier Payments
Expense ID	005	E05	EM05	S05
Expense Date	2022-05-23	2022-05-23	2022-05-23	2022-05-23
Monthly Expense(Rs.)	75000.75	100000.34	500000	200000
UPDATE    DELETE		UPDATE    DELETE	UPDATE    DELETE	UPDATE    DELETE

*Fig: 2.2.63*

**Ravishka Mini Supermarket**

**Update Employee Payments**

Type of Expense	Employee Payments
Expense ID	EM05
Expense Date	2022-05-23
Monthly Expense(Rs.)	500000
UPDATE	

*Fig: 2.2.64*

Profit Statement for the period ending December 31,2022



Select Month

Select...

**Revenue**

Customer Income	2500000
Sales Income	1000000
Interest Income	125000

Total Incomes **3625000**

**Expenses**

Employee Payments	500000
Electricity Payments	1000000.34
Supplier Payments	200000
Other Utility Bills	75000.75

Total Expenses **875001.09**

**ADD**

Net Profit **2749998.91**

[< Back](#)

[Next >](#)



Ravishka Mini-Supermarket

© 2022-2023

**Services**

- Order management for wholesale buyers
- Employee management
- Stock management
- Supplier's management
- Transport management
- Debtor's management
- Financial management

*Fig: 2.2.65*

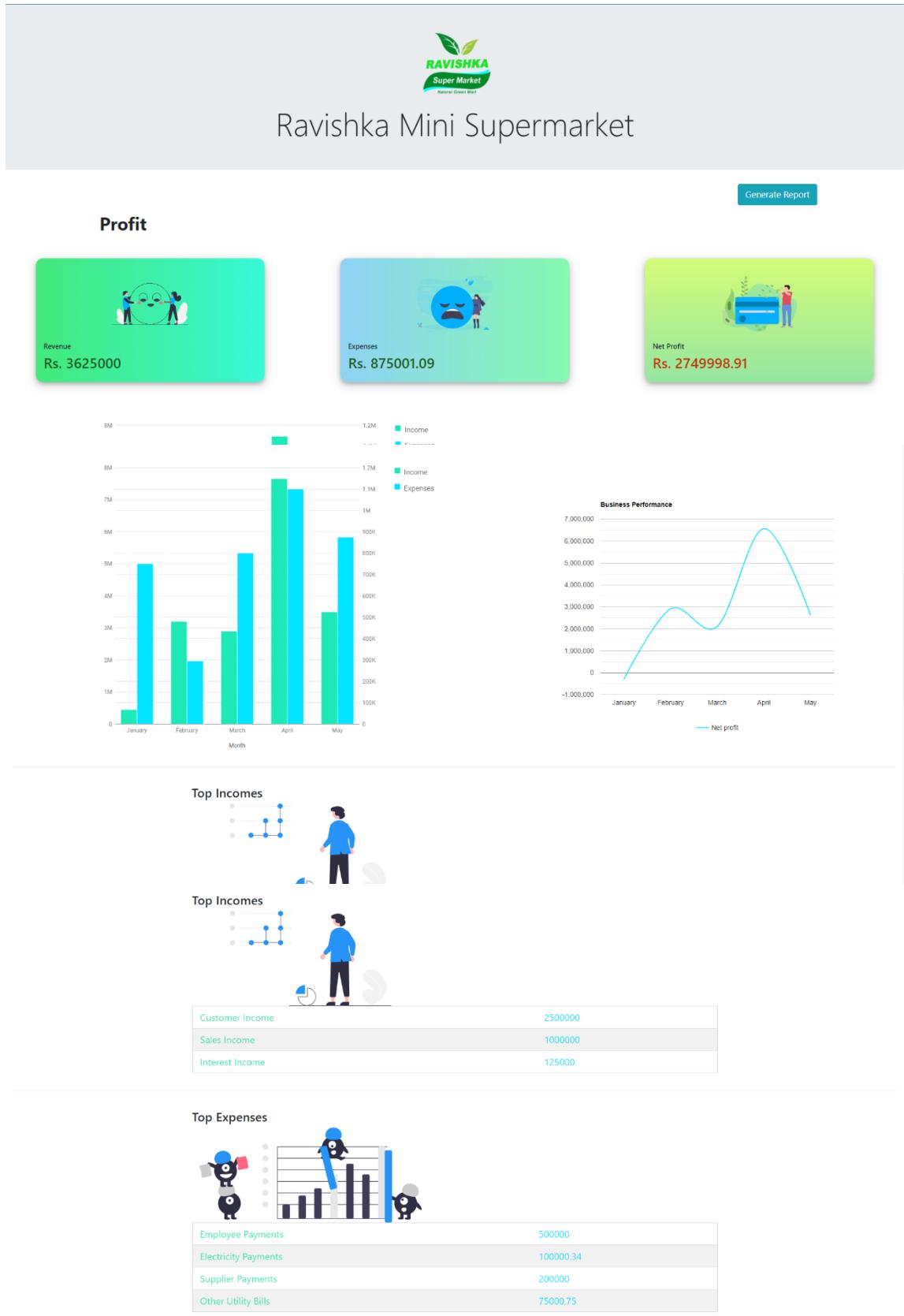
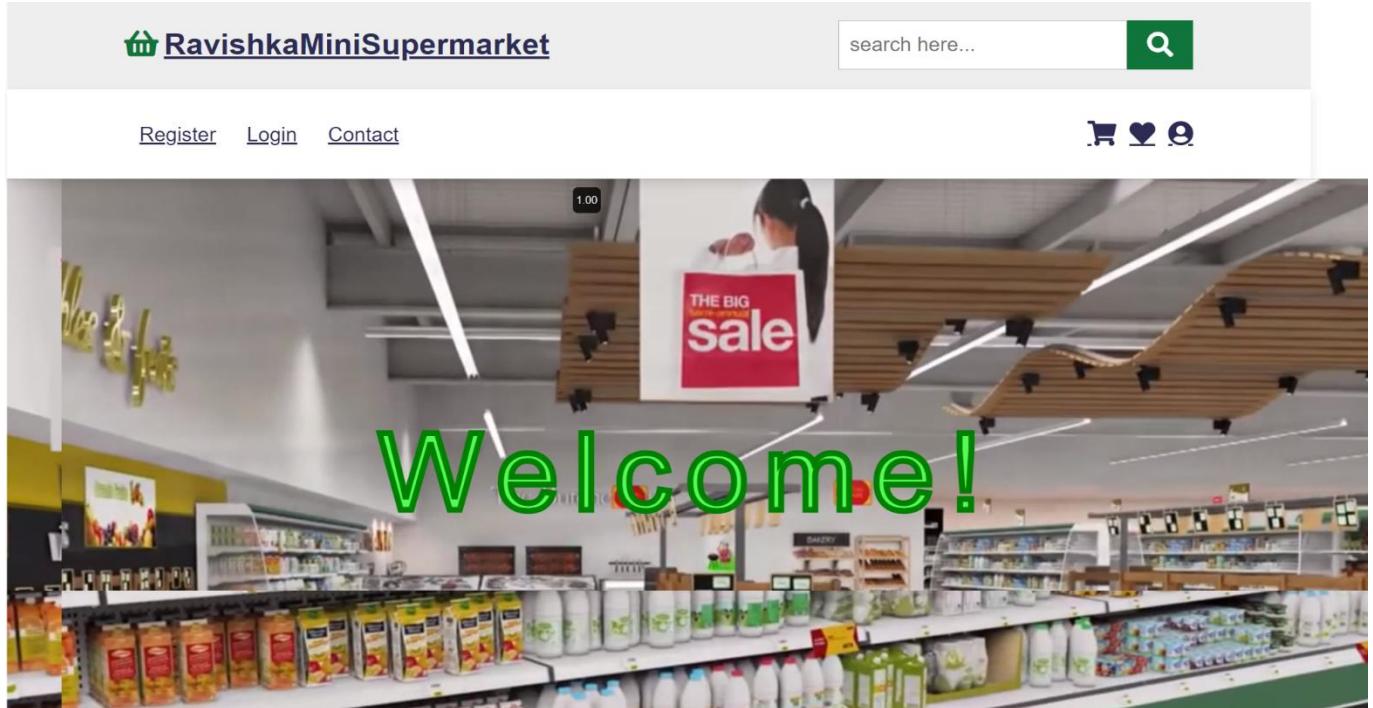


Fig: 2.2.66

✓ User Part:

Order placement for retail customers (IT20621056)



Subscribe Us For Latest Updates

enter your email

Subscribe

**RavishkaMiniSupermarket**

Lorem Ipsum Dolor Sit, Amet Consectetur Adipisicing Elit. Ullam Culpa Sit Enim Nesciunt  
Rerum Laborum Illum Quam Error Ut Alias!

f

Quick Links

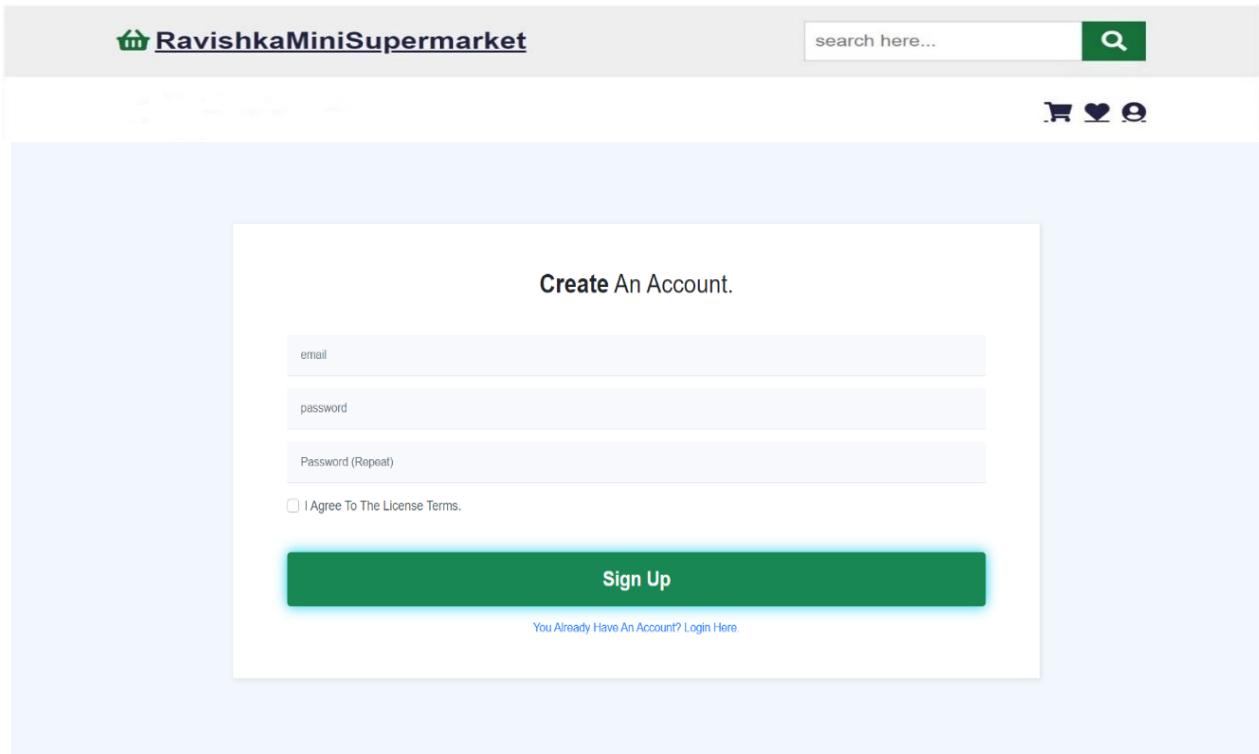
[Category](#)

[Product](#)

[Contact](#)

Created By **Ravishka Mini Supermarket** | All Rights Reserved!

Fig: 2.2.67



### Subscribe Us For Latest Updates

### RavishkaMiniSupermarket

Lorum Ipsum Dolor Sit, Amet Consectetur Adipisicing Elit. Utiam Culpa Sit Enim Nesciunt  
Rerum Laborum Illum Quam Error Ut Alias!



### Quick Links

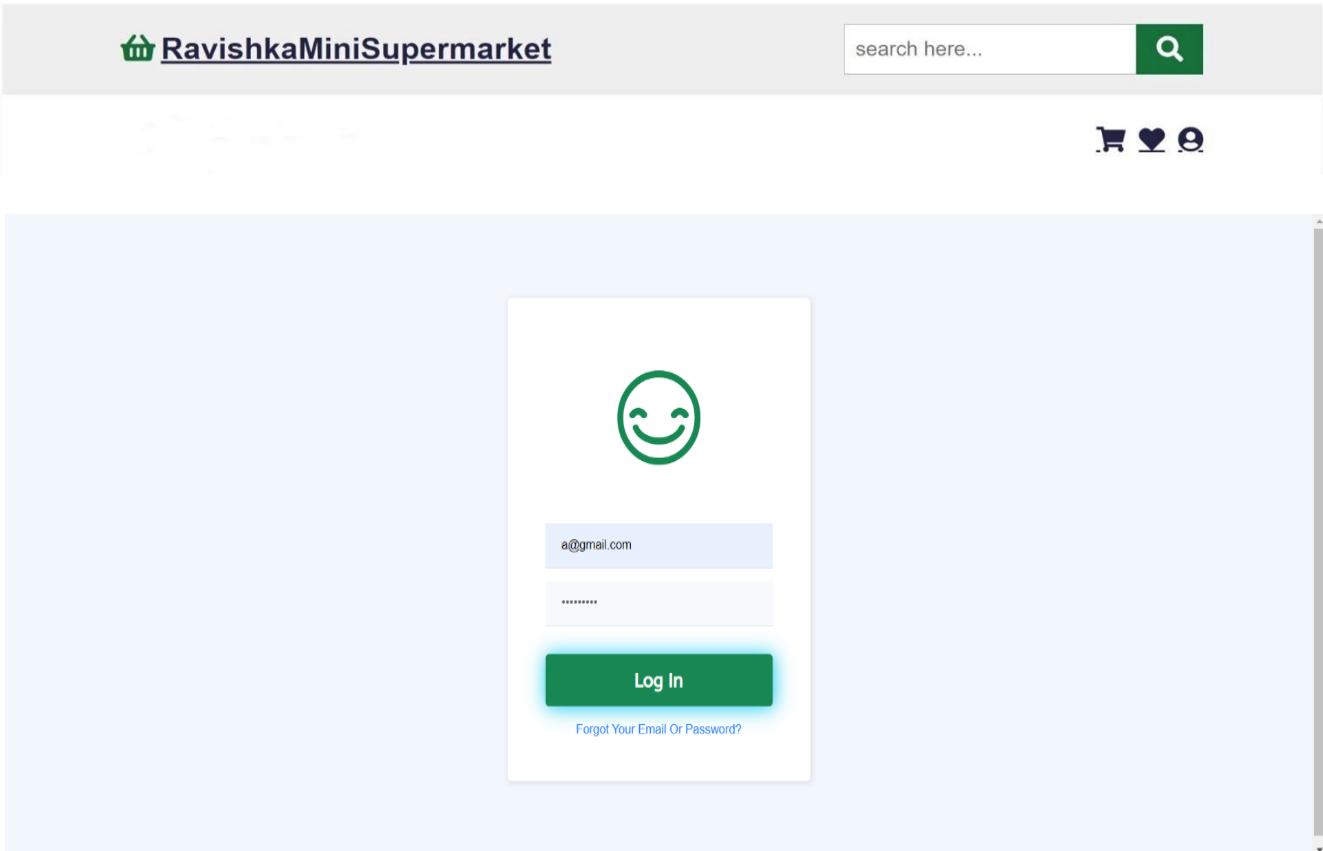
[Category](#)

[Product](#)

[Contact](#)

Created By [Ravishka Mini Supermarket](#) | All Rights Reserved!

*Fig: 2.2.68*



### Subscribe Us For Latest Updates

### RavishkaMiniSupermarket

Lorem Ipsum Dolor Sit, Amet Consectetur Adipisicing Elit. Ullam Culpa Sit Enim Nesciunt  
Rerum Laborum Illum Quam Error Ut Alias!



### Quick Links

[Category](#)

[Product](#)

[Contact](#)

Created By [Ravishka Mini Supermarket](#) | All Rights Reserved!

*Fig: 2.2.69*

search here...



[My Profile](#) [Contact](#)



Grocery ▾

Vegetables ▾

Fruits ▾

Frozen Items ▾

All Frozen Items

Ice Cream

Youghurt Drink



## Featured Products

Maggi Noodles Chicken



[More Details](#)

Broccoli



[More Details](#)

Red Melon



[More Details](#)

## Enjoy The Benifits For Our Regular Customers

Lorem Ipsum Dolor Sit Amet, Consectetur Adipiscing Elit. Maecenas Porttitor Congue Massa. Fusce Posuere, Magna Sed Pulvinar Ultricies, Purus Lectus Malesuada Libero, Sit Amet Commodo Magna Eros Quis Urna  
Nunc Viverra Imperdiet Enim. Fusco Est. Vivamus A Tellus.  
Pellentesque Habitant Morbi Tristique Senectus Et Netus Et Malesuada Fames Ac Turpis Egestas. Proin Pharetra Nonummy Pede. Mauris Et Orci.



## RavishkaMiniSupermarket

Loreum Ipsum Dolor Sit, Amet Consectetur Adipiscing Elit. Utiam  
Culpa Sit Enim Nesciunt Rerum Laborum Illum Quam Error Ut  
Alias!

f

## Quick Links

[Category](#)

[Products](#)

[Contact](#)

Created By Ravishka Mini Supermarket | All Rights Reserved!

Fig: 2.2.70

 **RavishkaMiniSupermarket**

search here... 

[My Profile](#) [Contact](#)   

[Home](#) / [Products](#)

 <p>Broccoli Per 100 G(S)Rs 46.50</p> <p><a href="#">Add to cart</a></p>	 <p>Bell Pepper Yellow Per 100 G(S)Rs 109.50</p> <p><a href="#">Add to cart</a></p>	 <p>Bell Pepper Red Per 100 G(S)Rs 109.50</p> <p><a href="#">Add to cart</a></p>
 <p>Red Cabbage Per 100 G(S)Rs 65.50</p> <p><a href="#">Add to cart</a></p>	 <p>Parsley Per 100 G(S)Rs 99.50</p> <p><a href="#">Add to cart</a></p>	 <p>Carrot Per 300 G(S)Rs 145.50</p> <p><a href="#">Add to cart</a></p>

#### Subscribe Us For Latest Updates

[Subscribe](#)

#### RavishkaMiniSupermarket

Lorum Ipsum Dolor Sit, Amet Consectetur Adipisicing Elit. Utiam Culpa Sit Enim Nesciunt  
Rerum Laborum Illum Quam Error Ut Alias!



#### Quick Links

[Category](#)

[Product](#)

[Contact](#)

Created By [Ravishka Mini Supermarket](#) | All Rights Reserved!

*Fig: 2.2.71*



[Home](#) / [My Profile](#)

Hello Helani !



[My Orders](#)

[Loyalty Points](#)

[Update Information](#)

[Delete Account](#)

### Profile Settings

Name	<input type="text" value="abc"/>
Surname	<input type="text" value="perea"/>
Mobile Number	<input type="text" value="0741314424"/>
Land Line	<input type="text" value="011234355"/>
Email Address	<input type="text" value="a@gmail.com"/>
Address Line 1	<input type="text" value="homagama,kottawa"/>
Address Line 2	<input type="text" value="36/2/4,wekada rod,homagana"/>
City	<input type="text" value="colombo"/>

[Submit](#)

### Subscribe Us For Latest Updates

enter your email

[Subscribe](#)

**RavishkaMiniSupermarket**

Lorem Ipsum Dolor Sit, Amet Consectetur Adipisicing Elit. Ut enim laborum illum quam error ut alias!

### Quick Links

[Category](#)

[Product](#)

[Contact](#)

Created By [Ravishka Mini Supermarket](#) | All Rights Reserved!

Fig: 2.2.72

[Home](#)   [My Profile](#)

## CONTACT US

## Subscribe Us For Latest Updates



### Quick Links

[Category](#)[Product](#)[Contact](#)

## RavishkaMiniSupermarket

search here... 

[Home](#) [My Profile](#) [Contact](#)



 Cart 0



Ice Cream  
Chocolate Ice Cream  
**Price:** Rs600.00



Sausages  
Cheese Chicken Sausages  
**Price:** Rs550.00



Yoghurt  
Orange Flavored Drinking Yoghurt  
**Price:** Rs120.00



Green Peas  
Frozen Green Peas 1Kg  
**Price:** Rs1100.00

[Add To Cart](#)



French Fries  
Frozen Potatoes  
**Price:** Rs700.00

[Add To Cart](#)



Meats  
Fresh Chicekn  
**Price:** Rs900.00

[Add To Cart](#)



Fish  
Fresh Thalpath Fish 1kg  
**Price:** Rs1800.00

[Add To Cart](#)



Frozen Berries  
Fresh Frozen Rassbery  
**Price:** Rs500.00

[Add To Cart](#)

[Add To Cart](#)

[Add To Cart](#)

[Add To Cart](#)

Subscribe Us For Latest Updates

enter your email

[Subscribe](#)

## RavishkaMiniSupermarket

Ipsum Dolor Sit, Amet Consectetur Adipiscing Elit. Ullam Culpa Sit Enim Nesciunt  
Rerum Laborum Illum Quam Error Ut Alias!



### Quick Links

[Category](#)

[Product](#)

[Contact](#)

Created By [Ravishka Mini Supermarket](#) | All Rights Reserved!

*Fig: 2.2.74*

# RavishkaMiniSupermarket

[Cart](#)

[Home](#)
[My Profile](#)
[Contact](#)

[Facebook](#)
[Twitter](#)
[Instagram](#)

Cart 4

Cart Updated Successfully

Product	Price	Quantity	Subtotal	
	Rs1100.00	<input type="text" value="2"/>	Rs2200	<a href="#"><span>Remove</span></a>
	Rs600.00	<input type="text" value="1"/>	Rs600	<a href="#"><span>Remove</span></a>
	Rs120.00	<input type="text" value="1"/>	Rs120	<a href="#"><span>Remove</span></a>
	Rs900.00	<input type="text" value="1"/>	Rs900	<a href="#"><span>Remove</span></a>

Total Rs3820

[< Continue Shopping](#)
[Checkout>](#)

### Subscribe Us For Latest Updates

[Subscribe](#)

### RavishkaMiniSupermarket

Loreum Ipsum Dolor Sit, Amet Consectetur Adipisicing Elit. Illum Culpa Sit Enim Nesciunt  
Rerum Laborum Illum Quam Error Ut Alias!



### Quick Links

[Category](#)
[Product](#)
[Contact](#)

Created By Ravishka Mini Supermarket | All Rights Reserved!

Fig: 2.2.75



Broccoli

Per 100 G(S) **Rs 46.50**[Add to cart](#)[More About This Product](#)

Lore ipsum Dolor Sit Amet, Consectetur Adipiscing Elit. Maecenas Porttitor Congue Massa. Fusce Posuere, Magna Sed Pulvinar Ultricies, Purus Lectus Malesuada Libero, Sit Amet Commodo Magna Eros Quis Urna. Nunc Viverra Imperdiet Enim. Fusce Est. Vivamus A Tellus. Pellentesque Habitant Morbi Tristique Senectus Et Netus Et Malesuada Fames Ac Turpis Egestas. Proin Pharetra Nonummy Pede. Mauris Et Orci.

## Subscribe Us For Latest Updates

[Subscribe](#)

### Quick Links

[Category](#)[Product](#)[Contact](#)

Cart 4

Total: Rs 3820

	Green Peas	Rs100.00	Quantity:2
	Ice Cream	Rs600.00	Quantity:1
	Yoghurt	Rs120.00	Quantity:1
	Meats	Rs900.00	Quantity:1

[View All](#)

**BILLING ADDRESS**

Full Name :

Email :

Address :

City :

District :  Zip Code :

**PAYMENT**

Cards Accepted :

Name On Card :

Credit Card Number :

Exp Month :

Exp Year :  CVV :

[Place Order](#)

## Subscribe Us For Latest Updates

[Subscribe](#)

## 2.3 Implementation

The implementation allows you to see your objectives materialize. In the development phase of our web application, we implemented for “[Ravishka Mini-Supermarket](#)”. We used **Laravel** for our backend implementation. Laravel is one of the most popular and efficient PHP frameworks in the web development market. Its advantages are outlined below.

- Advanced Security
- Improved Performance
- Better Authentication
- Enhanced URL routing
- Built in error and exception handling system
- Friendly support
- Easy to learn

We used **HTML, CSS, Bootstrap** and **JS** for our frontend and **Visual Studio Code** as our IDE as it had some themes where we can easily identify parameters, arguments, functions, variables, constants, and methods separately. We used some VS code extensions to make our codes more decent and clearer for us to find something to change instantly.

Here we can see which parcel is being carried by which driver by the tracking code. You can also see where the parcel is by the tracking code. also, then we can see driver details and parcel details. A 6-digit code is automatically generated for each parcel. after registration parcel, this code generates automatically. it's very useful to secure our items.

Fig: 2.3.a

```
do {
    $code = random_int(100000, 999999);
} while (parcel::where("code", "=", $code)->first());

$query = DB::table('parcels')->insert([
    'name'=> $request->name,
    'address'=> $request->address,
    'rname'=> $request->rname,
    'raddress'=> $request->raddress,
    'rcontact' =>$request->rcontact,
    'weight' =>$request->weight,
    'height' =>$request->height,
    'length' =>$request->length,
    'width'=>$request->width,
    'price'=>$request->price,
    'status'=>$request->status,
    'vehicle_type'=>$request->vehicle_type,
    'code'=>$code,
]);
if($query){
    return back()->with('success','Data have been successfully inserted');
}else{
    return back()->with('fail','something went wrong');
}

function trackingmNumber(){
    Model::select('column')
        ->where('column', 'value')
        ->intTrackingNum()
        ->limit(2)
        ->get();
}
```

The implemented modules are Order placement for retail customers, Employee management, Order management for wholesale buyers, Stock management, Supplier's management, Transport management, Debtor's management, and financial management.

The MySQL operations were carried out using the PHP-based phpMyAdmin application, which is designed to perform MySQL administration over the Internet. The user interface was used to manage databases, tables, columns, relations, indexes, users, and permissions, with the ability to execute any SQL command directly over it. Ensured that Apache and MySQL were operating in the XAMPP control panel by clicking the start buttons beneath the Actions column for access to installed hosting software XAMPP on machines and to work with phpMyAdmin from XAMPP.

Code segments related to the initialization to the Database are as follows.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD=
```

Fig: 2.3.b

The Dompdf method was used to generate reports. Requires the following package for composer.json; It was updated and downloaded this package and the dompdf + fontlib library.

```
composer require barryvdh/laravel-dompdf
```

After updating composer add the following lines to register provider in app.php

```
/*
 * Package Service Providers...
 */
Barryvdh\DomPDF\ServiceProvider::class,
```

Fig: 2.3.c

Code segments related to the generate reports are as follows.

```
public function export_report_pdf() {
    $incomes = Income::all();
    $tasks = Task::all();

    $pdf = PDF::loadView('report_pdf', ['incomes' => $incomes, 'tasks' => $tasks]);

    return $pdf->download('report_pdf.pdf');
}
```

Fig: 2.3.d

Different types of graphs were used to graphically show the details of the reports generated by each service.

```
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
    google.charts.load("current", {"packages":["corechart"]});
    google.charts.setOnLoadCallback(drawChart);
    function drawChart() {
        var data = google.visualization.arrayToDataTable([
            ['Task', 'Hours per Day'],
            <?php
            echo $chartData;
            ?>
        ]);
        var options = {
            is3D: true,
        };
        var chart = new google.visualization.PieChart(document.getElementById('piechart_3d'));
        chart.draw(data, options);
    }
</script>
```

Fig: 2.3.e

## 2.4 Testing

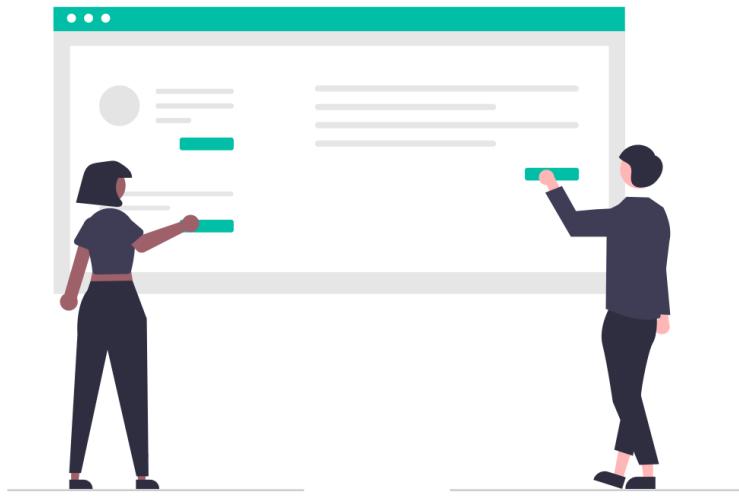
To ensure that the system operates more reliably and is easier to maintain, testing was done in two ways. This section's major goal is to offer a good, user-friendly, and high-performing system.

### Integration Testing

- ✓ This level of testing is designed to reveal flaws in the interaction of integrated units.
- ✓ When a single function is completed, unit testing is used. The goal is to ensure that each component of the software works as intended.
- ✓ Integration testing will be performed to ensure that all parts and pages in a given function properly.

### System Testing

- ✓ System testing is a type of testing that verifies a software product's completeness and integration.
- ✓ A system test is used to assess the end-to-end system requirements.
- ✓ This will be the system's last and most essential testing since it will check for faults in the system quickly as well as how the system performs test situations.
- ✓ In this part, the major focus is on system performance. This will be accomplished by manually entering selected data into corporate papers. The group will check to see if the output value matches the manual manuals.
- ✓ System testing is a collection of tests designed to put a computer-based system through its paces.



**Group ID: WD\_B01\_ITP\_G01**

**Project title: Supermarket Management System**

**Testing Function 1: Test the Customer Registration**

**Test case ID: SM01**

**Test designed by:  
Reg. No- IT20621056  
Name- Jayasekara H L**

**Test priority (High/Medium/Low):**

**High**

**Test Description:** Verify registration with valid username and password

**Preconditions (if there are any):** The customer should access the registration page.

**Dependencies (if there are any):**

**Test steps:**

- Navigate to the Register page.
- Enter a valid username in the “Username” field.
- Enter a valid password in the “Password” field.
- Enter the password again.
- Click the “Register” button.

**Pass-condition:**

- Email and password fields should not be empty.
- Passwords should be matched.

*Table: 2.4.1*

**Test executed Group ID: WD\_B01\_ITP\_G05**

**Test Executed members details:**

**Reg. No- IT20643836 / IT20636692**

**Name- Ariyarathnna B M N D S / Perera L A M D**

Test ID	Test Inputs	Expected Output	Actual Output	Result (Pass/Fail)	Comments
SM_01	<ul style="list-style-type: none"> <li>• Email:</li> <li>• Password:</li> <li>• Confirm Password:</li> </ul>	Error messages showing "Please enter the email, Please enter the password"	Error messages showing "Please enter the email, Please enter the password"	Pass	Email and Password are required fields.
SM_01	<ul style="list-style-type: none"> <li>• Email: helani@gmail.com</li> <li>• Password:</li> <li>• Confirm Password:</li> </ul>	Error messages showing "Please enter the password"	Error messages showing "Please enter the password"	Pass	Email and Password are required fields.
SM_01	<ul style="list-style-type: none"> <li>• Email: helani@gmail.com</li> <li>• Password: helani1234</li> <li>• Confirm Password: helani</li> </ul>	Error messages showing "Password must match"	Error messages showing "Password must match"	Pass	Customer must enter the password twice when registers in the system and they should be matched.
SM_01	<ul style="list-style-type: none"> <li>• Email: helani@gmail.com</li> <li>• Password: helani1234</li> <li>• Confirm Password: helani1234</li> </ul>	Redirect to the Login page.	Redirect to the Login page.	Pass	If the customer is redirected to the login page, he has successfully registered with the system.

Table: 2.4.2

## Testing Function 2: Test the calculation in the shopping cart

Test case ID: SM_02	Test designed by: Reg. No- IT20621056 Name- Jayasekara H L
Test priority (High/Medium/Low):	Medium
<b>Test Description:</b> Check that the total amount calculated on the shopping cart is correct	

**Preconditions (if there are any):** Customer should login to the system.

**Dependencies (if there are any):**

### Test steps:

- Login to the system with valid credentials.
- Add products to the shopping cart.

### Pass-condition:

- Calculate the correct total amount according to the products in the shopping cart.

Table: 2.4.3

**Test executed Group ID: WD\_B01\_ITP\_G05**

**Test Executed members details:**

**Reg. No- IT20654030**

**Name- Dasanayake K D H S**

Test ID	Test Inputs	Expected Output	Actual Output	Result (Pass/Fail)	Comments
SM_02	Added "Drinking yoghurt and Chocolate ice cream" to the shopping cart and check the total amount	Total Amount = \$6.00	Total Amount = \$6.00	Pass	The value of the two items is correctly calculated as the total amount.
SM_02	Added "Drinking yoghurt and two Chocolate ice cream" to the shopping cart and check the total amount	Total Amount = \$10.00	Total Amount = \$10.00	Pass	The correct total amount is calculated according to the products in the shopping cart.
SM_02	Added "Drinking yoghurt and two Chocolate ice cream and Sausages" to the shopping cart. Later remove one "Chocolate ice cream" and check the total amount	Total Amount = \$6.00	Total Amount = \$6.00	Pass	The correct total amount is calculated according to the products in the shopping cart.

*Table: 2.4.4*

<b><u>Testing Function 3: Registration</u></b>					
Logic in Interfaces	Validation and proper error messages	Connected to the database	Easy to use, User friendliness,	Easy navigation	Function is complete (%)
Register to the system with a strong password	Generate proper error messages until get a strong password from the user	Successfully connected to the database.	1. creative and Attractive UI 2. user friendly  3. error messages and success messages are properly displayed in pop-up windows.	Easy navigation ways with breadcumbs.	Overall function was completed and working well.  100% complete

Table: 2.4.5

**Testing Function 1:** Calculate Salary function in Employee Management System

**Test case ID: SM\_03**

<b>Test priority (High/Medium/Low):</b>	Low
---	-----

**Test Description:** Entered Overtime Hours and Number of Working days in numeric format

**Preconditions (if there are any):** Admin should have logged in to the system with valid user credentials

**Dependencies (if there are any):** Not Applicable

**Test steps:**

Input: Without Selecting the Employee category from the drop-down list

Input: Enter the Provider name in the requested format

Input: Enter Overtime Hours and Number of Working days in numeric format

Press Add button

**Pass-condition:**

Employee Level (job role) should be selected

Table: 2.4.6

Test ID	Test Inputs	Expected Output	Actual Output	Result (Pass/Fail)	Comments
SM_03	Admin Login  Provider Email- (Empty)  Password – (Empty)	Warning message “Please fill out these fields”	Warning message “Please fill out this field”	Pass	<b>Correctly worked</b>  <b>A warning message is shown</b>
SM_03	Admin Login  Provider Email- Praveen.subhash98@gmail.com  Password – 12345678	Display user Profile	Display user Profile	Pass	<b>Correctly worked</b>  <b>Success message and database insert operation</b>

Table: 2.4.7

<b>Group ID: WD_B01_ITP_G01</b>	
<b>Project title: Supermarket Management System</b>	
<b><u>Testing Function 1: Test the wholesale buyers' order details</u></b>	
<b>Test case ID: SM_04</b>	<b>Test designed by: Reg. No- IT20645670 Name- T.A.H.Umendra</b>
<b>Test priority (High/Medium/Low):</b>	High
<b>Test Description:</b> Check all fields filled in the order information form.	
<b>Preconditions (if there are any):</b> The shopkeeper must enter all fields in order details form for wholesale order.	
<b>Dependencies (if there are any):</b>	
<b>Test steps:</b>	
<ul style="list-style-type: none"> <li>• Go to the wholesale buyers' order details page.</li> <li>• Enter create order information.</li> <li>• Click the "Submit" button.</li> </ul>	

*Table: 2.4.8*

**Test executed Group ID: WD\_B01\_ITP\_G01**

**Test Executed members details:**

**Reg. No- IT20645670**

**Name- T.A.H. Umendra**

<b>Test ID</b>	<b>Test Inputs</b>	<b>Expected Output</b>	<b>Actual Output</b>	<b>Result (Pass/Fail)</b>	<b>Comments</b>
<b>SM_04</b>	Fill Required Fields	Successful Message	Order Added Successfully	Pass	Created order details
<b>SM_04</b>	Empty	Error in create order	Error message in each field	Pass	Errors and required all fields

*Table: 2.4.9*

**Testing Function 2: Validate the email and contact number field when entering the delivery details for wholesale buyers.**

Test case ID: SM_05	Test designed by: Reg. No- IT20645670 Name- T.A.H. Umendra
Test priority (High/Medium/Low):	High
<b>Test Description:</b> Validate email address and contact number	

**Preconditions (if there are any):** The shopkeeper must enter the email address with '@' sign and mobile number in 10 numbers.

**Dependencies (if there are any):**

**Test steps:**

- Go to the wholesale buyers' delivery details page.
- Enter the email with '@' sign and contact number with 10 numbers.
- Then click the "Save" button.

**Pass-condition:**

- Details added to the system

Table: 2.4.10

**Test executed Group ID: WD\_B01\_ITP\_G01**

**Test Executed members details:**

**Reg. No-IT20645670**

**Name-T.A.H. Umendra**

Test ID	Test Inputs	Expected Output	Actual Output	Result (Pass/Fail)	Comments
<b>SM_05</b>	Email: hiru@gmail.com  Contact No: 07732431	Error Message in email field and Contact No field	“Please include an ‘@’ in the email address” message and “Contact number should be 10 digits” Message in each field	Pass	Contact number field should be 10 digits. The email field should be filled with the ‘@’ sign
<b>SM_05</b>	Email: <a href="mailto:hiru@gmail.com">hiru@gmail.com</a>  Contact No: 0774321860	“Successfully added delivery details” message	“Successfully entered delivery details” message	Pass	Added delivery details to the database.

*Table: 2.4.11*

<b><u>Testing Function 3: Order Management for Wholesale Buyers</u></b>					
<b>Logic in Interfaces</b>	<b>Validation and proper error messages</b>	<b>Connected to the database</b>	<b>Easy to use, User friendliness ,</b>	<b>Easy navigation</b>	<b>Function is complete (%)</b>
Cannot added invalid data and invalid characters. Insertion data validation successfully created.	Proper error messages for email validation part. Proper error messages for phone number validation part.	Successfully connected with database. Successful database insert, update, delete operation and success message	Separate edit and delete buttons for each record	Navigation is easy because navigation buttons are set to go to the relevant path for each page	Overall function was completed and working well. 100% complete

*Table: 2.4.12*

### Testing Function 1: Test the calculation of the report

Test case ID: SM_06	Test designed by: Reg. No- IT20621292 Name- Gunasekara P.G.A.I.P
Test priority (High/Medium/Low):	Medium
<b>Test Description:</b> Check that the total amount according to current products calculated on the report is correct	

**Preconditions (if there are any):** Stock Manager should Enter quantity and price per unit.

**Dependencies (if there are any):**

#### Test steps:

- Login to the stock management.
- Enter quantity when adding groceries, frozen items or vegetables, and fruits.
- Enter the price per unit for each.

#### Pass-condition:

- Calculate the correct total amount according to the products.

Table: 2.4.13

**Test executed Group ID: WD\_B01\_ITP\_G05**

**Test Executed members details:**

**Reg. No- IT20654030**

**Name- Dasanayake K D H S**

Test ID	Test Inputs	Expected Output	Actual Output	Result (Pass/Fail)	Comments
<b>SM_06</b>	Added “shampoo” with “30 “quantity and price unit is “100.00” To the grocery Items to check the total amount	Total Amount = 3000	Total Amount = 3000	Pass	The total amount of grocery item is correctly calculated.
<b>SM_06</b>	Added “Cheese” with “100 “quantity stock and price per unit is “200.00” To the frozen Items to check the total amount	Total Amount = 20000	Total Amount = 20000	Pass	The total amount of frozen items is correctly calculated.
<b>SM_06</b>	Added “orange” with quantity(kg)50 and price per unit(1kg) is “100.00” to the fruits. after some days customers buy oranges and the quantity is 40.	Total Amount = 4000	Total Amount = 4000	Pass	The correct total amount of fruit is calculated according to the quantity

*Table: 2.4.14*

## Testing Function 2: validate fields when inserting and update

<b>Test case ID:</b> SM_07	<b>Test designed by:</b> Reg. No- IT20621292 Name- Gunasekara P.G.A.I. P
<b>Test priority (High/Medium/Low):</b>	Medium
<b>Test Description:</b> when updating stock manager can't update the item code and item name. Also, when inserting items, you can't put any empty fields. all fields are required.	

**Preconditions (if there are any):**

**Dependencies (if there are any):**

### **Test steps:**

- Login to the stock management.
- Click the add button and add items.
- Click the search button and go to the search page, search for any item.
- Then click the update button in there.

### **Pass-condition:**

- Can't Add the items to stock.
- Item name and Item code cannot be changed.

*Table: 2.4.15*

**Test executed Group ID: WD\_B01\_ITP\_G05**

**Test Executed members details:**

**Reg. No- IT20654030**

**Name- Dasanayake K D H S**

Test ID	Test Inputs	Expected Output	Actual Output	Result (Pass/Fail)	Comments
<b>SM_07</b>	Click “Add” button with all empty fields in Groceries.	All fields are required. Can’t Add.	All fields are required. Can’t Add.	Pass	Validating was done well.
<b>SM_07</b>	Click “Add” button with all empty fields in fruits.	All fields are required.	All fields are required.	Pass	Validating was done well.
<b>SM_07</b>	Search “cheese” in search bar. Then click the Update button	You can’t click or type Item name and Item code fields	Can’t click or type Item name and Item code fields	Pass	As I think the technique used, is very practical for the given scenario.

*Table: 2.4.16*

<b><u>Testing Function 1: Test the calculation in the report</u></b>					
<b>Logic in Interfaces</b>	<b>Validation and proper error messages</b>	<b>Connected to the database</b>	<b>Easy to use, User friendliness,</b>	<b>Easy navigation</b>	<b>Function is complete (%)</b>
Add Quantity and price per unit when inserting items	Total update according to the number of quantities	Successfully connected to the database.	1. creative and Attractive UI 2. user friendly 3. Easily view and can see Total Amount and Using that check the loss.	Easy navigation ways to users. .	Overall function was completed and working well. 100% complete

Table: 2.4.17

<b><u>Testing Function 2: validate fields when inserting and update</u></b>					
<b>Logic in Interfaces</b>	<b>Validation and proper error messages</b>	<b>Connected to the database</b>	<b>Easy to use, User friendliness ,</b>	<b>Easy navigation</b>	<b>Function is complete (%)</b>
cannot insert invalid data types and Cannot add products with empty fields.  Cannot edit Item code field and Item name field	Proper warning messages to when submitting form with empty fields.	Successfully connected to database.	Properly aligned select options to execute actions.  2.Attractive UI  3.High user friendliness.	Navigation is easy because of page redirecting functions and backward navigation buttons.	Overall function was completed and working well.  100% complete

Table: 2.4.18

<b>Group ID:</b> WD_B01_ITP_G01	
<b>Project title:</b> Supermarket Management System	
<b>Testing Function 1: Test the Customer Registration</b>	
<b>Test case ID:</b> SM_08	<b>Test designed by:</b> Reg. No- IT20648336 Name- Thilakarathne W A S S
<b>Test priority (High/Medium/Low):</b>	High
<b>Test Description:</b> Enter delivery item details using valid conditions	
<b>Preconditions (if there are any):</b> The customer must order a item.	
<b>Dependencies (if there are any):</b>	
<b>Test steps:</b> <ul style="list-style-type: none"> <li>• Go to the record and check delivery item details page.</li> <li>• Enter a parcel and delivery information.</li> <li>• Click the “Save” button.</li> </ul>	
<b>Pass-condition:</b> <ul style="list-style-type: none"> <li>• All fields required.</li> </ul>	

Table: 2.4.19

**Test executed Group ID: WD\_B01\_ITP\_G05**

**Test Executed members details:**

**Reg. No- IT20658472**

**Name- Weerapperuma D I**

<b>Test ID</b>	<b>Test Inputs</b>	<b>Expected Output</b>	<b>Actual Output</b>	<b>Result (Pass/Fail)</b>	<b>Comments</b>
<b>SM_08</b>	Contact number 0712456780  Delivery Status	Displays an error message in the relevant fields.	Displays an error message in the relevant fields	Pass	Contact number and delivery status field is required.
<b>SM_08</b>	Contact number 0761733667  Delivery status	Displays an error message “Contact number should be 10 digits” and “Delivery status is required”	Displays an error message “Contact number should be 10 digits”	Pass	Contact number field should be 10 digits.  Delivery status must require field.
<b>SM_08</b>	Contact number 0761733651  Delivery status pending	Display “Successful” popup message.	Display “Successful” popup message.	Pass	Then shopkeeper can track parcel.

*Table: 2.4.20*

## Testing Function 2: Test the calculation in the shopping cart

Test case ID: SM_09	Test designed by: Reg. No- IT20648336 Name- Thilakarathne W A S S
Test priority (High/Medium/Low):	Heigh
Test Description: create tracking number after entering delivery information	

**Preconditions (if there are any):** The shop owner must have included delivery information

**Dependencies (if there are any):**

### Pass-condition:

- Track delivery item

### Test steps:

- Entered a delivery information
- Go the tracking delivery item page.

Table: 2.4.21

**Test executed Group ID: WD\_B01\_ITP\_G05**

**Test Executed members details:**

**Reg. No- IT20648190**

**Name- Ahmed M N T**

<b>Test ID</b>	<b>Test Inputs</b>	<b>Expected Output</b>	<b>Actual Output</b>	<b>Result (Pass/Fail)</b>	<b>Comments</b>
<b>SM_09</b>	1 <sup>st</sup> delivery item	Tracking number: - 2564853	Tracking number: - 2564853	Pass	Create unique tracking number And insert into database.
<b>SM_09</b>	2 <sup>nd</sup> delivery item	Tracking number: - 2389456	Tracking number: - 2389456	Pass	Create unique tracking number And insert into database.
<b>SM_09</b>	3rd delivery item	Tracking number: - 2653698	Tracking number: - 2653698	Pass	Create unique tracking number And insert into database.

*Table: 2.4.22*

<b><u>Testing Function 3: vehicle and delivery item management</u></b>					
Logic in Interfaces	Validation and proper error messages	Connected to the database	Easy to use, User friendliness,	Easy navigation	Function is complete (%)
1.Tracking number created successfully.  2.Insert data validation properly created.	1.display proper error messages in data validation part.  2.tracking number successful message show properly.	proper connection with database.	user friendly interface created.	easy to navigation.	overall function good. 100% completed.

Table: 2.4.23

### Testing Function 1: Test the select plan of the wholesale buyers insert as a debtor

<b>Test case ID:</b> SM_10	<b>Test designed by:</b> Reg. No- IT20230692 Name- Arachchi S C S
<b>Test priority (High/Medium/Low):</b>	High
<b>Test Description:</b> Check that the total amount calculated with the current interest rate	

**Preconditions (if there are any):** the shopkeeper should check the current interest rate of the buyers.

**Dependencies (if there are any):** use view current interest rates and check rates.

#### Pass-condition:

- Calculate the correct total amount with the total debt amount according to the chosen plan.

#### Test steps:

- Check the relevant interest plan using view current interest Rates.
- Navigate to wholesale buyer details and go to insert.
- Enter valid Debt\_id and Name D\_CID and "Wholesale Debtors' Name" fields.
- Select a plan.
- Enter correct order amount "Order Amount" field. (When you enter the correct order amount it will calculate and display the Debt amount and total amount)
- Select today date.
- Press the "submit" button to submit details.

Table: 2.4.24

**Test executed Group ID: WD\_B01\_ITP\_G05**

**Test Executed members details:**

**Reg. No- IT20644680 / IT20653118**

**Name- Aathif M A M / Wishvajith Y W S H**

<b>Test ID</b>	<b>Test Inputs</b>	<b>Expected Output</b>	<b>Actual Output</b>	<b>Result (Pass/ Fail)</b>	<b>Comments</b>
<b>SM_10</b>	<ul style="list-style-type: none"><li>• D_CID: D001</li><li>• Wholesale Debtors' Name : chandima</li><li>• Select Plan:</li><li>• Order Amount:</li><li>• Date Of The Buy:05/11/2022</li></ul>	Error messages showing "Please select interest plan", "Please Enter Order Amount".	Error messages showing "Please select interest plan", "Please Enter Order Amount".	Pass	Select Plan, Order Amount, required fields.
<b>SM_10</b>	<ul style="list-style-type: none"><li>• D_CID: D001</li><li>• Wholesale Debtors' Name : chandima</li><li>• Select Plan: Months 2</li><li>• Order Amount:</li><li>• Date Of The Buy:05/11/2022</li></ul>	Error messages showing "Please Enter Order Amount".	Error messages showing "Please Enter Order Amount".	Pass	Order Amount required fields
<b>SM_10</b>	<ul style="list-style-type: none"><li>• D_CID: D001</li><li>• Wholesale Debtors' Name : chandima</li><li>• Select Plan:: Months 2</li><li>• Order Amount: AMOUNT</li><li>• Date Of The Buy:05/11/2022</li></ul>	Error messages showing "The wh amount must be a number".	Error messages showing "The wh amount must be a number".	Pass	Order Amount required fields and it must be a numeric
<b>SM_10</b>	<ul style="list-style-type: none"><li>• D_CID: D001</li><li>• Wholesale Debtors' Name : chandima</li><li>• Select Plan:: Months 2</li><li>• Order Amount: 200000</li><li>• Date Of The Buy:05/11/2022</li></ul>	redirected to the Debtors Management page	redirected to the Debtors Management page	Pass	The shopkeeper is redirected to the Debtors Management page and display the "success!", Wholesale Buyer added Successfully "

Table: 2.4.25

**Testing Function 3: The debt of wholesale buyer details inserting**

Logic in Interfaces	Validation and proper error messages	Connected to the database	Easy to use, User friendliness,	Easy navigation	Function is complete (%)
cannot insert invalid data types and Cannot submit without a selected interest plan.	display proper error messages in data validation part.	Successfull y connected to database.	user friendly interface created.	Easy navigation ways to users	Overall function was completed and working well. 100% complete

Table: 2.4.26

<b>Testing Function 2: Test the wholesale buyers search option</b>	
<b>Test case ID:</b> SM_11	<b>Test designed by:</b> Reg. No- IT20230692 <b>Name-</b> Arachchi S C S
<b>Test priority (High/Medium/Low):</b>	Medium
<b>Test Description:</b> Check that the view all records search-relevant debtor's details.	

<b>Preconditions (if there are any the shopkeeper should insert wholesale debtors before the search.)</b>
<b>Dependencies (if there are any):</b> use wholesale buyer's details.

<b>Pass-condition:</b>
<ul style="list-style-type: none"> <li>• Show all available records relevant debtor's details.</li> </ul>

<b>Test steps:</b>
<ul style="list-style-type: none"> <li>• Navigate to wholesale debtor view option</li> <li>• Then Enter the correct wholesale debtor's name.</li> <li>• Enter the correct wholesale debtor's ID.</li> <li>• Press search button.</li> <li>• If the shopkeeper wants to view full details, click the View option on the relevant record of the debtor's record number.</li> </ul>

Table: 2.4.27

**Test executed Group ID: WD\_B01\_ITP\_G05**

**Test Executed members details:**

**Reg. No- IT20644680 / IT20653118**

**Name- Aathif M A M / Wishvajith Y W S H**

<b>Test ID</b>	<b>Test Inputs</b>	<b>Expected Output</b>	<b>Actual Output</b>	<b>Result (Pass/Fail)</b>	<b>Comments</b>
<b>SM_11</b>	<ul style="list-style-type: none"> <li>• Wholesale Debtors' Name:</li> <li>• D_CID:</li> </ul>	Error messages showing "Please enter correct debt name", "Please Enter D_CID".	Error messages showing "Please enter correct debt name", "Please Enter D_CID".	Pass	Wholesale Debtors' Name, D_CID are required fields.
<b>SM_11</b>	<ul style="list-style-type: none"> <li>• Wholesale Debtors' Name: chandima</li> <li>• D_CID:</li> </ul>	Error messages showing "Please Enter D_CID".	Error messages showing "Please Enter D_CID".	Pass	D_CID is required field.
<b>SM_11</b>	<ul style="list-style-type: none"> <li>• Wholesale Debtors' Name: chandima</li> <li>• D_CID: x123(Invalid D_CID)_</li> </ul>	Error messages showing "Invalid name or _CID"	Error messages showing "Invalid name or _CID"	pass	NO results for wrong filling.
<b>SM_11</b>	<ul style="list-style-type: none"> <li>• Wholesale Debtors' Name: chandima</li> <li>• D_CID: d001</li> </ul>	View relevant debtor's details below the search window	View relevant debtor's details below the search window	pass	View debtor's details. Work successfully.

*Table: 2.4.28*

### **Testing Function 3: Debtors Management**

Logic in Interfaces	Validation and proper error messages	Connected to the database	Easy to use, User friendliness,	Easy navigation	Function is complete (%)
1. cannot insert invalid data types and Cannot submit without a selected interest plan.	1. display proper error messages in data validation part. 2. display proper error messages when submitting.	1. Successfully connected to database.	1. user friendly interface created. 2. properly aligned button to execute actions. 3. have helpful warning messages.	1. Easy navigation ways to users and Backward navigation buttons use	1. Overall function was completed and working well. 2. 100% complete

*Table: 2.4.29*

<b>Group ID: WD_B01_ITP_G01</b>	
<b>Project title: Supermarket Management System</b>	
<b>Testing Function 1: The shopkeeper is not allowed to add income and expenses with the same type</b>	
<b>Test case ID: SM_12</b>	<b>Test designed by:</b> Reg. No- IT20137250 Name- Rajapaksha R C P
<b>Test priority (High/Medium/Low):</b>	High
<b>Test Description:</b> When adding a new income and expenses shopkeeper cannot add a new income and expenses with a type that exists in the income and expenses database. Because it will be challenging to find the correct income and expenses.	
<b>Preconditions (if there are any):</b> An income and expense must be added previously with the same type.	
<b>Dependencies (if there are any):</b>	
<b>Test steps:</b> <ul style="list-style-type: none"> <li>• Navigate to the Income statement page.</li> <li>• Select the form with the income type that is already in the database.</li> <li>• Navigate to the Expenditure statement page.</li> <li>• Select the form with the expense type that is already in the database.</li> </ul>	
<b>Pass-condition:</b> <ul style="list-style-type: none"> <li>• The shopkeeper is not allowed to add an income and expense with a type that is already in the database.</li> </ul>	

Table: 2.4.30

**Test executed Group ID: WD\_B01\_ITP\_G05**

**Test Executed members details:**

**Reg. No- IT20636524**

**Name- Ariyasena P P S**

Test ID	Test Inputs	Expected Output	Actual Output	Result (Pass/Fail)	Comments
SM_12	Selected "Sales Income" in the type of income.	Sales income is inserted.	Sales income is inserted.	Pass	Worked successfully.
SM_12	Selected same income type.	The relevant income type has become invisible.	The relevant income type has become invisible.	Pass	Useful method to improve user friendliness of the web application.
SM_12	Selected "Employee Payments" in the type of expense.	Employee payment is inserted.	Employee payment is inserted.	Pass	Worked successfully.
SM_12	Selected same expense type.	The relevant expense type has become invisible.	The relevant expense type has become invisible.	Pass	Useful method to improve user friendliness of the web application.

*Table: 2.4.31*

<b><u>Testing Function 3: Add income and Expense type</u></b>					
<b>Logic in Interfaces</b>	<b>Validation and proper error messages</b>	<b>Connected to the database</b>	<b>Easy to use, User friendliness,</b>	<b>Easy navigation</b>	<b>Function is complete (%)</b>
The drop-down option created disabled to add income and expense type.	Same income and expense types are invisibly selective.	Successfully connected to database.	1.Properly aligned select options to execute actions. 2.Attractive UI 3.High user friendliness	Easy navigation ways to users.	Overall function was completed and working well.  100% complete

Table: 2.4.32

## Testing Function 2: Test the calculation in the monthly net profit

Test case ID: SM_13	Test designed by: Reg. No- IT20137250 Name- Rajapaksha R C P
Test priority (High/Medium/Low):	High
<b>Test Description:</b> Check the monthly net profit calculated with the current total income and total expense	

**Preconditions (if there are any):** An income and expense details must be added previously.

**Dependencies (if there are any):**

### Test steps:

- Navigate to the Income statement page.
- Enter correct income details.
- Press the “add” button to submit details.
- Navigate to the Expenditure statement page.
- Enter correct expense details.
- Press the “add” button to submit details.
- Navigate to the Profit statement page.
- Check monthly net profit.

### Pass-condition:

- Calculate the correct monthly net profit according to the added income and expenses details.

Table: 2.4.33

**Test executed Group ID: WD\_B01\_ITP\_G05**

**Test Executed members details:**

**Reg. No- IT20636524**

**Name- Ariyasena P P S**

Test ID	Test Inputs	Expected Output	Actual Output	Result (Pass/Fail)	Comments
SM_13	<b>Type of income:</b> Customer Income <b>Income ID:</b> C01 <b>Income Date:</b> 2022/05/12 <b>Monthly Income:</b> 2500000  <b>Type of expense:</b> Employee Payments <b>Expense ID:</b> EM01 <b>Expense Date:</b> 2022/05/12 <b>Monthly Expense:</b> 500000	<b>Total Net Profit:</b> 2624998.91	<b>Total Net Profit:</b> 2624998.91	Pass	Worked successfully.
SM_13	<b>Type of income:</b> Sales Income <b>Income ID:</b> S02 <b>Income Date:</b> 2022/05/12 <b>Monthly Income:</b> 1000000  <b>Type of expense:</b> Electricity Payments <b>Expense ID:</b> E02 <b>Expense Date:</b> 2022/05/12 <b>Monthly Expense:</b> 100000	<b>Total Net Profit:</b> 3210000.00	<b>Total Net Profit:</b> 3210000.00	Pass	Worked successfully.

Table: 2.4.34

<b><u>Testing Function 3: Add income and Expense type</u></b>					
<b>Logic in Interfaces</b>	<b>Validation and proper error messages</b>	<b>Connected to the database</b>	<b>Easy to use, User friendliness,</b>	<b>Easy navigation</b>	<b>Function is complete (%)</b>
The drop-down option created disabled to add income and expense type.	Same income and expense types are invisibly selective.	Successfully connected to database.	1. Properly aligned select options to execute actions. 2. Attractive UI 3. High user friendliness	Easy navigation ways to users.	Overall function was completed and working well.  100% complete

Table: 2.4.35

### Testing Function 1: validate fields when inserting

<b>Test case ID:</b> SM_14	<b>Test designed by:</b> Reg. No- IT20301972 Name- P. R. P. De Silva
<b>Test priority (High/Medium/Low):</b>	Medium
<b>Test Description:</b> When shop keeper inserting items you can't put any empty fields. All fields are required. Shop keeper can't insert any string value into the contact number field.	
<b>Preconditions (if there are any):</b>	
<b>Dependencies (if there are any):</b>	
<b>Test steps:</b> <ul style="list-style-type: none"><li>• Login to the system</li><li>• Select supplier management</li><li>• Select Add new suppliers button in the suppliers management page</li><li>• Fill all fields about supplier correctly, then click Add Supplier button</li></ul>	

#### Pass-condition:

- Fill all fields are required.

Table: 2.4.36

**Test executed Group ID: WD\_B01\_ITP\_G05**

**Test Executed members details:**

**Reg. No- IT20654030**

**Name- Dasanayake K D H S**

<b>Test ID</b>	<b>Test Inputs</b>	<b>Expected Output</b>	<b>Actual Output</b>	<b>Result (Pass/Fail)</b>	<b>Comments</b>
<b>SM_14</b>	Click “Add Supplier” button with all empty fields.	All fields are required. Can’t Add.	All fields are required. Can’t Add.	Pass	Validating was done well.
<b>SM_14</b>	Click “Add Supplier” button with contact field empty.	Contact Number is required.	Contact Number is required.	Pass	Validating was done well.

*Table: 2.4.37*

## Testing Function 2: validate fields when updating

<b>Test case ID:</b> SM_15	<b>Test designed by:</b> Reg. No- IT20301972 Name- P. R. P. De Silva
<b>Test priority (High/Medium/Low):</b>	Medium
<b>Test Description:</b> when updating shop keeper can't update the item code and item name. Shop keeper can't put any string value into the contact number field. Also, all fields are required.	

**Preconditions (if there are any):**

**Dependencies (if there are any):**

### **Test steps:**

- Login to the system
- Select supplier management
- Select View Supplier Details button in the suppliers management page
- Click the appropriate Edit button for the supplier you want to update
- Then click Update Supplier button

### **Pass-condition:**

- Fill all fields are required.
- Item name and Item code cannot be changed.

*Table: 2.4.38*

**Test executed Group ID: WD\_B01\_ITP\_G05**

**Test Executed members details:**

**Reg. No- IT20654030**

**Name- Dasanayake K D H S**

Test ID	Test Inputs	Expected Output	Actual Output	Result (Pass/Fail)	Comments
SM_15	Click “Update Supplier” button with contact field empty.	Contact Number is required.	Contact Number is required.	Pass	Validating was done well.
SM_15	Click “Update Supplier” button with contact field with string value.	Contact Number is numeric field.	Contact Number is numeric field.	Pass	Validating was done well.

*Table: 2.4.39*

<b><u>Testing Function 2: validate fields when inserting</u></b>					
<b>Logic in Interfaces</b>	<b>Validation and proper error messages</b>	<b>Connected to the database</b>	<b>Easy to use, User friendliness</b>	<b>Easy navigation</b>	<b>Function is complete (%)</b>
cannot insert invalid data types and any empty fields.	Proper warning messages to when deleting suppliers.	Successfully connected to database.	Properly aligned select options to execute Actions. 2. Attractive UI	Easy navigation ways to users and Backward navigation buttons use	Overall function was completed and working well. 100% complete

Table: 2.4.40

<b><u>Testing Function 2: validate fields when updating</u></b>					
<b>Logic in Interfaces</b>	<b>Validation and proper error messages</b>	<b>Connected to the database</b>	<b>Easy to use, User friendliness</b>	<b>Easy navigation</b>	<b>Function is complete (%)</b>
cannot update invalid data types and any empty fields.  Cannot edit Item code field and Item name field	Proper warning messages to when deleting suppliers.	Successfully connected to database.	Properly aligned select options to execute Actions. 2. Attractive UI	Easy navigation ways to users and Backward navigation buttons use	Overall function was completed and working well. 100% complete

Table: 2.4.41

### **3. Evaluation**

#### **3.1 Assessment of the Project work**

This system is used by purely low-level technicians. Therefore, we implement the system at a high level. We use simple navigation headers, sidebars, footers, buttons, color concepts, tables, and charts to provide the best user experience and interface. User input fields are validated at the user level and annotated with appropriate messages about the format of the user input, as well as user-friendly confirmation alerts, success messages, and error messages.

The search for data from the database is organized in tabular form and is made easier by the menu function. Automatic identifiers, search functions and reporting functions have been implemented to add value to the system. No programming knowledge is required to use the system. Images and plain language are used to show functionality and are neatly arranged to avoid cluttering. All relevant navigation, notifications and alerts have been carefully checked to provide the best experience. We are guaranteed by various types of tests. Unit testing was done by the developers themselves, and integration testing was done by a delegated testing team. In addition, we hired an external testing agency to test the entire component at user level. Finally, the customer can carry out an acceptance test.

#### **3.2 What we learned**

You have learned to plan, evaluate, design, develop, test, and document your entire IT project. I was under pressure to meet the deadline in a short amount of time. We have learned to communicate with each of the teams, to accept the differences and difficulties of other members and to understand the different perspectives and opinions of other members. Through collaboration and evaluation from other perspectives, many impossible and difficult tasks may have been accomplished. I learned time management skills, networking skills, internet skills, delegation skills and was able to become a great programmer in the real world. We learned the basics of do-it-yourself development through this non-academic IT project.

#### **3.2 Future Work**

The development of the system never ends. It's a continuous development. We task the customer to implement further functions that the system or customers require in the future. A specific team will be equipped remotely at any time. We will monitor security aspects of server deployment and password security policies. It continuously monitors and evaluates the system flow and handles all problem and error reports immediately to ensure that the system is not disrupted.

We have accepted a request from the customer regarding the integration of the database into the cloud and are analyzing the feasibility. We are looking for the prerequisites to work with cloud integration. Management is negotiating the introduction of a biometric attendance system for employees and the integration of the biometric system into the web

application for automatic attendance calculation. Today it is managed manually by the customer. Also, they discuss the launch of Android and IOS mobile applications for their system. We want to accept the contract. The system takes the system to the next level.

## 4. Conclusion

The goal of this project was to automate the manual processes of handling the **Ravishka mini supermarket's** daily responsibilities for the client who owns the above-mentioned store. During the process of gathering and analyzing requirements, some potential issues with the existing supermarket system were discovered, and the main goal of developing this system was to automate the entire system while providing users with an interactive experience and allowing management to complete their day-to-day tasks efficiently and easily. Furthermore, the system was created in accordance with the company's customer-client criteria. When the final system was shown to the client, the advocates concluded that the produced online system did a sufficient job of meeting the client's needs.

However, various flaws in the system, such as a lack of professionalism, might be found. Future plans have been implemented as a team to overcome them. Moreover, the system is fully automated with possesses few limitations too. This system was built with PHP and the Laravel framework, with MVC design, and MySQL was utilized for the database because of its flexibility, security, and performance. However, these technologies are constantly evolving and are prone to becoming obsolete. Furthermore, creating a completely foolproof system is a difficult process. As a result, continuous maintenance is essential to overcome such constraints.

Overall, it can be concluded that the created system meets all of the client's functional and non-functional needs, adds capabilities that were not requested by the client, and aids the organization's processes through automation, which has reduced errors and fraudulent activities. As a result, the company will be more efficient and productive. Furthermore, this mini supermarket system has simple interfaces, Increased Accuracy, Increased Efficiency, and Accurate data Storing as the benefits of the system.



## 5. References

1. Report Guidelines Lectures from Sri Lanka Institute of Information Technology
2. IEEE Citation Guidelines  
[Publish with IEEE - IEEE Author Center](#)
3. First steps to the Laravel from the official website  
<https://laravel.com/docs/9.x/installation>
4. GitHub repository management Tutorials in the You Tube  
<https://youtu.be/xuB1Id2Wxak>
5. Bootstrap official website  
<https://getbootstrap.com/>
6. Google Charts  
<https://developers.google.com/chart>
7. Stack Overflow  
<https://stackoverflow.co/>
8. Git hub  
[https://github.com/IT2080-ITP-2022-Feb/itp\\_project\\_group\\_b1-wd\\_b01\\_itp\\_g01](https://github.com/IT2080-ITP-2022-Feb/itp_project_group_b1-wd_b01_itp_g01)
9. Draw.io  
<https://www.diagrams.net/>
10. w3school  
<https://www.w3schools.com/>
11. XAMPP  
<https://www.apachefriends.org/download.html>
12. A DOMPDF Wrapper for Laravel  
<https://github.com/barryvdh/laravel-dompdf>
13. unDraw  
<https://undraw.co/illustrations>
14. MockFlow  
<https://mockflow.com/wireframing/>

## 6. Appendix C: Selected Code Listings

Display a pie chart about Income and Expenses in Financial Management

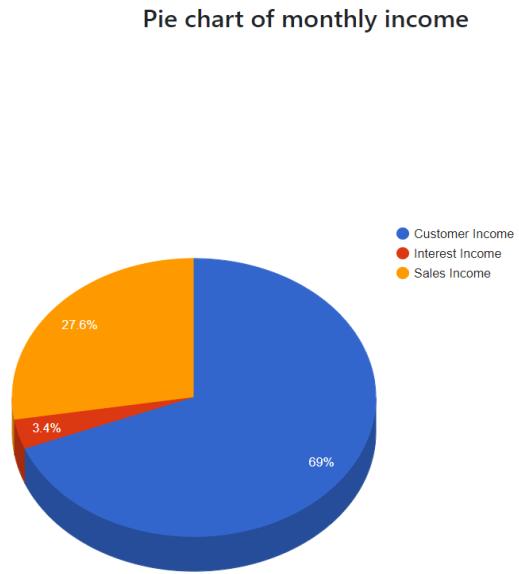


Fig: 6.a

```
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
  google.charts.load("current", {"packages":["corechart"]});
  google.charts.setOnLoadCallback(drawChart);
  function drawChart() {
    var data = google.visualization.arrayToDataTable([
      ["Task", "Hours per Day"],
      <?php
      echo $chartData;
      ?>
    ]);

    var options = {
      is3D: true,
    };

    var chart = new google.visualization.PieChart(document.getElementById('piechart_3d'));
    chart.draw(data, options);
  }
</script>
```

Fig: 6.b

Pie chart of monthly expenses

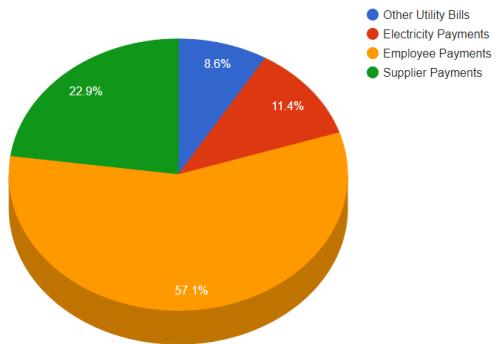


Fig: 6.c

```
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
  google.charts.load("current", {packages:["corechart"]});
  google.charts.setOnLoadCallback(drawChart);
  function drawChart() {
    var data = google.visualization.arrayToDataTable([
      ['Task', 'Hours per Day'],
      <?php
      echo $chartData;
      ?>
    ]);

    var options = {
      is3D: true,
    };

    var chart = new google.visualization.PieChart(document.getElementById('piechart_3d'));
    chart.draw(data, options);
  }
</script>
```

Fig: 6.d

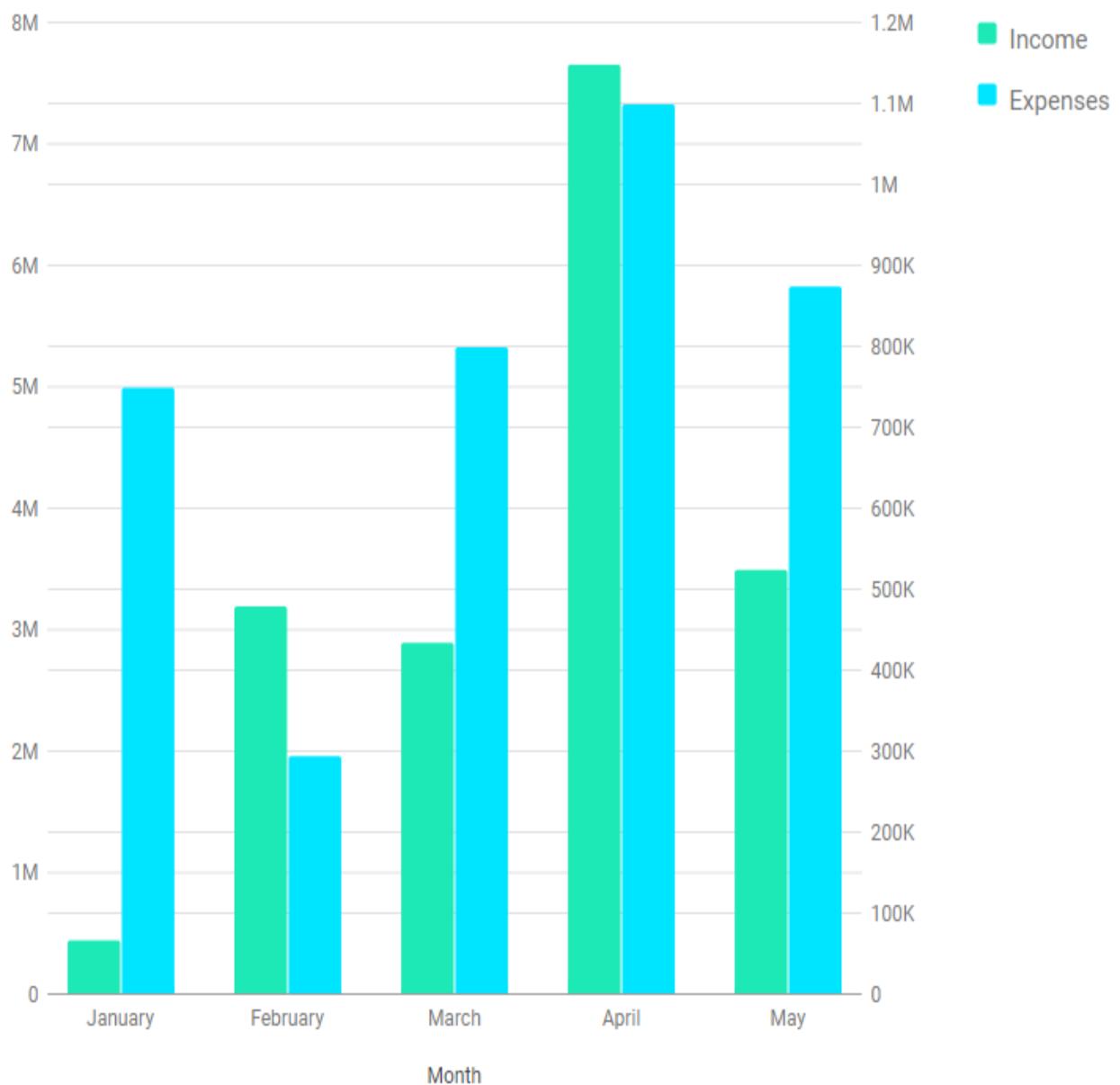


Fig: 6.e

```

<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
  google.charts.load('current', {'packages':['bar']});
  google.charts.setOnLoadCallback(drawStuff);

  function drawStuff() {
    var data = new google.visualization.arrayToDataTable([
      ['Month', 'Income', 'Expenses'],
      <?php
        $query = "select month, total_income, total_expense from profits";
        $res = mysqli_query($conn,$query);
        while($data = mysqli_fetch_array($res)) {
          $month = $data['month'];
          $total_income = $data['total_income'];
          $total_expense = $data['total_expense'];
        }
        [<'<?php echo $month; ?>', <?php echo $total_income; ?>, <?php echo $total_expense; ?>,
        <?php
        |   |
        ?>
    ]);
  }

  var options = {
    width: 700,
    colors: ['#1DE9B6', '#00E5FF'],
    bars: 'vertical', // Required for Material Bar Charts.
    series: [
      0: { axis: 'distance' }, // Bind series 0 to an axis named 'distance'.
      1: { axis: 'brightness' } // Bind series 1 to an axis named 'brightness'.
    ],
    axes: {
      x: {
        distance: {label: 'parsecs'}, // Bottom x-axis.
        brightness: {side: 'top', label: 'apparent magnitude'} // Top x-axis.
      }
    }
  };

  var chart = new google.charts.Bar(document.getElementById('dual_x_div'));
  chart.draw(data, options);
}

```

Fig: 6.f

Display a line chart about Monthly Net Profit in Financial Management

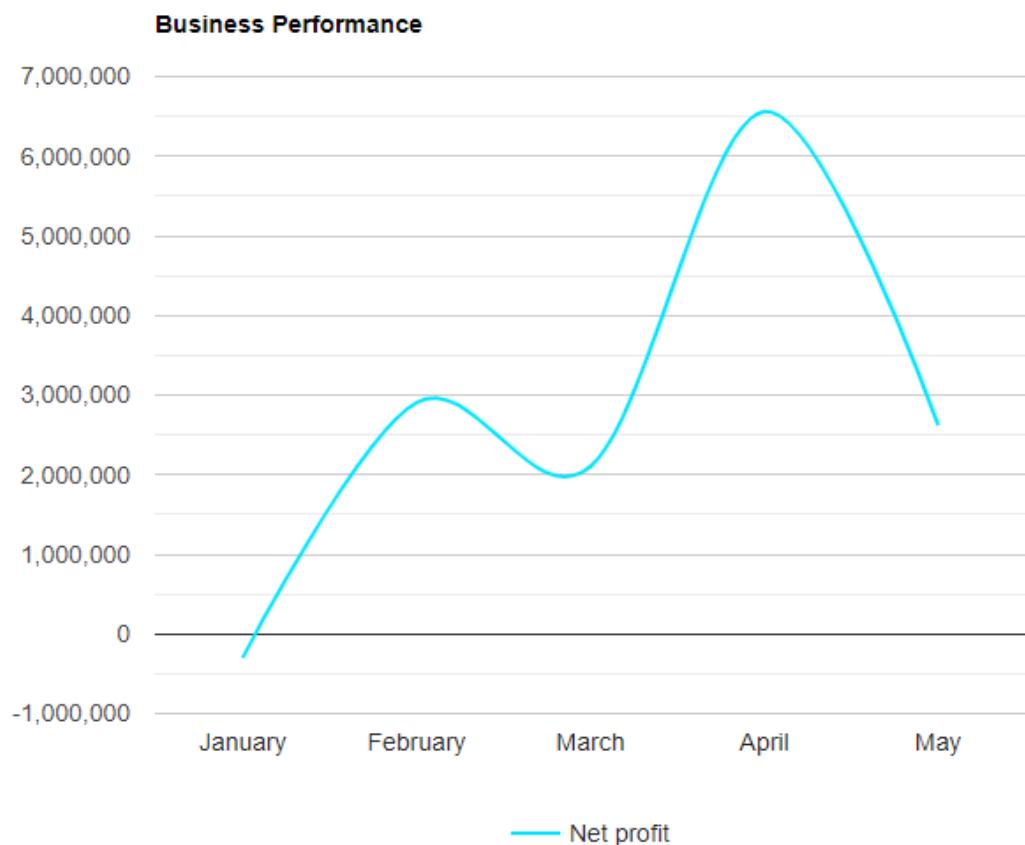


Fig: 6.g

```

google.charts.load('current', {'packages':['corechart']});
google.charts.setOnLoadCallback(drawChart);

function drawChart() {
  var data = google.visualization.arrayToDataTable([
    ['Month', 'Net profit'],
    <?php
      $query = "select month, net_profit from profits";
      $res = mysqli_query($conn,$query);
      while($data = mysqli_fetch_array($res)) {
        $month = $data['month'];
        $net_profit = $data['net_profit'];
      }
    ?>
    ['<?php echo $month; ?>', '<?php echo $net_profit; ?>'],
    <?php
    ?
  ?>
]);
}

var options = {
  width: 800,
  colors: ['#00E5FF'],
  title: 'Business Performance',
  curveType: 'function',
  legend: { position: 'bottom' }
};

var chart = new google.visualization.LineChart(document.getElementById('curve_chart'));

chart.draw(data, options);
}

</script>

```

Fig: 6.h

Display a pie chart about No of Wholesale Buyer Debtors for each plan and No of Regular Customer Debtors for each plan in Debtor's Management

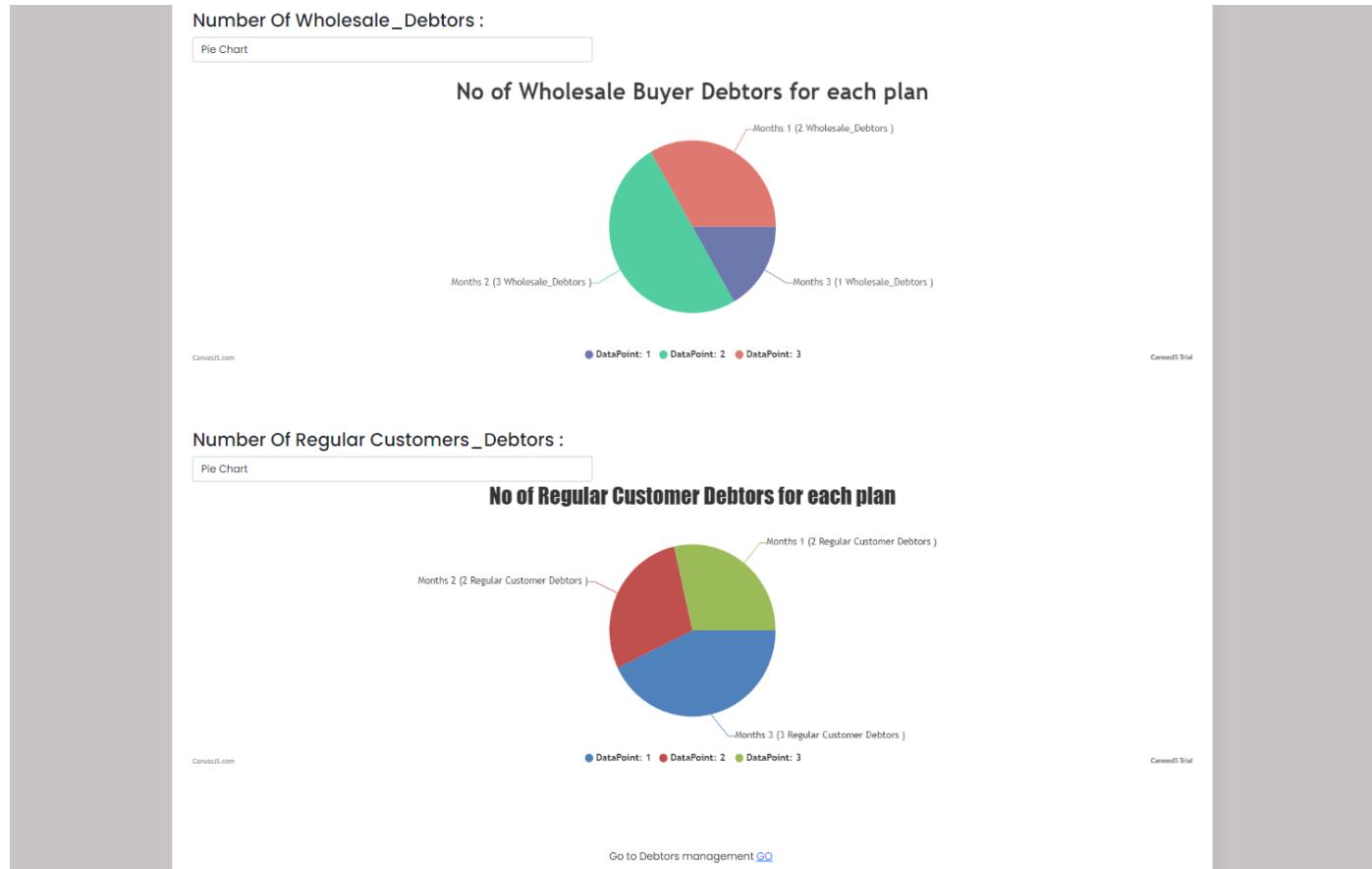


Fig: 6.i

Wholesale Buyer Debtors for each plan and No of Regular Customer Debtors for each plan in Debtor's Management connect database and retrieve data for each queries.

```
<?php

$conn =mysqli_connect('localhost','root','','supermarket');
$sql="select Count(D_CID) as NO_of_wholesale_Debtors,whPlan from whbuyers GROUP BY whPlan ORDER BY whPlan DESC";
$res=mysqli_query($conn,$sql);
while($row= mysqli_fetch_assoc($res)){
    $datapoints[] =array("label"=>$row['whPlan'],"y"=>$row['NO_of_wholesale_Debtors']);
}

$sql2="select Count(rcus_CID) as NO_of_wholesale_Debtors,rcusPlan from debtrcustomers GROUP BY rcusPlan ORDER BY rcusPlan DESC";
$res2=mysqli_query($conn,$sql2);
while($row2= mysqli_fetch_assoc($res2)){
    $datapoints2[] =array("label"=>$row2['rcusPlan'],"y"=>$row2['NO_of_wholesale_Debtors']);
}
?>
```

```
<section class="content" style ="margin-left:2%;margin-right:2%;margin-top:5%;">
<h3>Number Of Wholesale_Debtors :</h3>
<select name ="chart" onchange ="myfunction()" class ="form-control" id="chart" style ="width:40%;">
    <option value="null">Select Chart</option>
    <option value="pie">Pie Chart</option>
    <option value="column">Column chart</option>
    <option value="bar">Bar chart</option>
    <option value="doughnut">doughnut chart</option>
</select>
<br>
<div class="product-index " alignn="right" style=" width:100%;>
<div id="chartContainer" style= "height :auto; width:100%;> <br> <br> <br>
</div>
</div>
<h3>Number Of Regular Customers_Debtors : </h3>
<select name ="chart" onchange ="myfunction2()" class ="form-control" id="chart2" style ="width:40%;">
    <option value="null">Select Chart</option>
    <option value="pie">Pie Chart</option>
    <option value="column">Column chart</option>
    <option value="bar">Bar chart</option>
    <option value="doughnut">doughnut chart</option>
</select>
<div class="product-index " alignn="right" >
<div id="chartContainer2" style= "height :auto; width:100%;>
</div>
</div>
</section>
```

Fig: 6.j

```

<script>
    function myfunction()
    {
        var chartType =document.getElementById("chart").value;
        var chart = new CanvasJS.Chart("chartContainer", {
            animationEnabled: true,
            theme: "light2", // "light1", "light2", "dark1", "dark2"
            title:{ 
                text: "No of Wholesale Buyer Debtors for each plan"
            },
            axisY: {
                title: "NO of Debtors"
            },
            axisX: {
                title: "Wholesale Buyer Plan"
            },
            data: [{
                type: chartType,
                showInLegend: true,
                yValueFormatString: "#,##0\"\"",
                indexLabel:"{label} ({y} Wholesale_Debtors )",
                dataPoints: <?php echo json_encode($datapoints,JSON_NUMERIC_CHECK); ?>
            }]
        });
        chart.render();
    }
    function myfunction2()
    {
        var chartType =document.getElementById("chart2").value;
        var chart = new CanvasJS.Chart("chartContainer2", {
            animationEnabled: true,
            theme: "light1", // "light1", "light2", "dark1", "dark2"
            title:{ 
                text: "No of Regular Customer Debtors for each plan"
            },
            axisY: {
                title: "NO of Debtors"
            },
            axisX: {
                title: "Regular Customer Plan"
            },
            data: [{
                type: chartType,
                showInLegend: true,
                yValueFormatString: "#,##0\"\"",
                indexLabel:"{label} ({y} Regular Customer Debtors )",
                dataPoints: <?php echo json_encode($datapoints2,JSON_NUMERIC_CHECK); ?>
            }]
        });
        chart.render();
    }
</script>

```

Fig: 6.j

The generation of PDFs takes place in financial management, debtor's management, stock management and transport management

### Monthly Financial Report PDF

**Income Statement**

Type of incomes	Monthly income
1 Customer income	Rs.2500000
2 Sales income	Rs.1000000
3 Interest income	Rs.125000
<b>Total income</b>	<b>Rs.3625000</b>

**Expenses Statement**

Type of expenses	Monthly expenses
1 Employee Payments	Rs.500000
2 Electricity Payments	Rs.100000.34
3 Supplier Payments	Rs.200000
4 Other Utility Bills	Rs.75000.75
<b>Total expense</b>	<b>Rs.875001.09</b>

**Financial Statement**

Total income	Total expense	Net Profit / Loss
1 Rs.3625000	Rs.875001.09	Rs.2749998.91

  
 Ravishka Mini-Supermarket

### Debtors Report

Monthly Income from debtors' interest :

Total income from Wholesale Buyers Interet:	11905
Total income from Regular Customer Interet:	8464
Total amount from interest:	20369

Available Stock details report							
Item Code	Item Name	Manufacture Date	Expire Date	Description	Quantity	Price per unit	Total Amount
G001	sanitizer spray	2022-02-05	2022-12-17	Lifeboy hand sanitizer	20	100	2000
G009	cheese	2022-02-05	2022-05-27	Abewela	30	10	300
G011	Shampoo	2022-02-05	2022-05-30	sumsilk thick and long(300ml)	9 - Out Of Stock	150	1350
G020	tea bag pack	2022-02-05	2022-05-29	flora	30	30	900
G012	Shampoo	2022-02-05	2022-10-04	Dove (250ml)	30	300	9000
G002	Soya	2022-02-05	2023-02-05	Chiken flavor	8 - Out Of Stock	150	1200

Generate Report- Drivers

[Download PDF](#)

#	Name	Assign	
1	Amal sampath	Assign	<b>2022-05-21 12:51:06</b>
2	Kamal Peries	not	<b>2022-05-22 06:22:38</b>
3	Nimal Jayasoorya	not	<b>2022-05-22 18:31:08</b>
4	W.S.Perera	New	<b>2022-05-22 18:34:53</b>
5	Kamal Sampath	Assign	<b>2022-05-22 18:40:37</b>
6	shashini Sankalpana	not	<b>2022-05-23 11:16:33</b>

Fig: 6.k

```
public function export_report_pdf() {
    $incomes = Income::all();
    $tasks = Task::all();

    $pdf = PDF::loadView('report_pdf', ['incomes' => $incomes, 'tasks' => $tasks]);

    return $pdf->download('report_pdf.pdf');
}
```

```
public function index_debt_report(){
    //get debt amounts
    $wtotal = Whbuyer::whereMonth('created_at', Carbon::now()->month)->sum('wh_D_Amount');
    $ctotal = DebtRCustomers::whereMonth('created_at', Carbon::now()->month)->sum('rcus_D_Amount');
    return view('DebtorsReport')->with('ctotal',$ctotal)->with('wtotal',$wtotal);
}
```

```
<script>
    function generate_debt_report_PDF() {
        const element = document.getElementById('invoice');
        html2pdf().from(element).save();
    }
</script>
```

Fig: 6.l

# THANK YOU!

