

Advanced Software Engineering

COURSEWORK-1

Group-01

IslandLink Sales Distribution Network (ISDN)

Course Submission Cover Sheet

Module: Advanced Software Engineering

Assignment no: 001

Weighting: 100% of the module mark

Deadline: 3PM Wednesday the xxth of January 2025

Module Leader: Mr. Akila Udara Akalanka Student ID: 25026764

Please note that there are specific regulations concerning **the use of AI and Academic Misconduct**. Below are extracts from these regulations. By signing, you acknowledge that you have read and understood these extracts.

(signature:) yasaspasindufernando@gmail.com Date: 02/15/2026

This header sheet should be attached to the work you submit.

Academic Integrity means being honest in your academic work and your studies and making sure that you acknowledge the work of others and giving credit where you have used other people's ideas as part of presenting your arguments. Your assessment submissions must therefore always be entirely your own work, based on your own learning and appropriately referenced including how you have used Generative AI. The University regards the use of Generative AI applications by students to deceive to gain unfair advantage as **academic misconduct**. This usage includes:

Plagiarism, where AI tools are used to generate output and ideas that are presented or submitted as if they were the student's own work, without proper citation or references.

Where a complete assignment is created using Generative AI and represented as a student's own work, this will be regarded as contract cheating in the same way as commissioning an 'Essay Mill' or other third party to complete your work. Further information can be found on : [Guidance on the use of Artificial Intelligence](#).

Academic misconduct: The University takes academic misconduct very seriously and seeks at all times to rigorously protect its academic standards. Plagiarism, collusion and other forms of cheating constitute academic misconduct, for which there is an explicit range of graduated penalties depending on the particular type of academic misconduct. The penalties that can be applied if academic misconduct is substantiated range from a reprimand to expulsion in very serious cases and for repeated instances of misconduct. You are also responsible for ensuring that all work submitted is your own and that it is appropriately referenced. The University does not tolerate cheating of any kind. You are strongly advised

Acknowledgement

We would like to express my sincere appreciation to all the members of our project team for their valuable contributions, dedication, and cooperation throughout the completion of this project. This project would not have been possible without the collective effort, commitment, and teamwork shown by each member.

Every team member played an important role by sharing ideas, supporting one another, and completing assigned tasks responsibly. The open communication, mutual respect, and willingness to help each other greatly contributed to the successful completion of this project.

We would also like to thank our lecturer/supervisor for the guidance, encouragement, and constructive feedback provided during the project period. Their support helped us stay focused and improve the quality of our work.

And, We grateful to everyone who directly or indirectly supported us in completing this project successfully.

Executive summary

The IslandLink Sales Distribution Network (ISDN) project focuses on the design and development of a centralized, island-wide Sales Distribution Management System to address operational inefficiencies caused by manual and fragmented processes. ISDN is one of the island's largest wholesale and retail distribution companies, serving over 5,000 active retail outlets through five Regional Distribution Centres (RDCs) located in the North, South, East, West, and Central provinces. The existing system relied heavily on phone calls, emails, paper documentation, and Excel spreadsheets, resulting in frequent order errors, inaccurate inventory records, delayed reporting, and limited delivery visibility for customers.

To overcome these challenges, the proposed solution introduces a web-based, centralized platform accessible to Head Office, RDC staff, logistics teams, and retail customers. The system enables online order placement, real-time inventory synchronization across all RDCs, automated delivery scheduling with GPS tracking, digital billing and online payment integration, and real-time management dashboards for informed decision-making. Role-based access control ensures secure and controlled system usage, while centralized data storage improves accuracy, consistency, and transparency across the organization.

This implementation of a centralized, web-based sales and distribution management system aimed at improving the operational efficiency of IslandLink's nationwide distribution network. Prior to this project, since ISDN relied heavily on manual processes such as Excel spreadsheets, paper-based documentation, and phone communications across its Head Office and Regional Distribution Centres (RDCs), These practices resulted in data inconsistency, delayed decision-making, limited visibility of inventory, and increased operational errors.

The proposed ISDN system addresses these challenges by introducing an integrated digital platform developed using PHP and MySQL, following a three-tier architectural model. The system supports core business functions including user authentication, order management, inventory control, delivery tracking, invoicing, and role-based access control. Retailers, RDC staff, logistics personnel, and management users are provided with secure and role-specific access to real-time data, enabling accurate coordination and improved transparency across the distribution process.

The project adopts an Agile development methodology, allowing incremental development and early stakeholder feedback to reduce implementation risks and ensure alignment with real operational requirements. Modern software engineering practices such as version control, modular coding, automated testing, and database normalization were applied to enhance system reliability, scalability, and maintainability.

A comprehensive feasibility study confirms that the system is technically achievable using open-source technologies, economically viable with minimal development costs, operationally practical for end users, legally compliant with applicable regulations, and achievable within the proposed development schedule. The system design is supported by detailed UML diagrams, database schemas, architectural diagrams, and user interface designs to ensure clarity and consistency throughout development.

The implementation phase successfully transformed ISDN's manual operations into an automated, centralized solution that improves data accuracy, enhances security, and supports nationwide scalability. Rigorous testing processes, including unit, integration, system, and user acceptance testing, were conducted to validate system functionality and performance.

Overall, the ISDN system delivers a sustainable, secure, and efficient digital solution that strengthens IslandLink's sales and distribution operations. The project demonstrates how structured system design, appropriate methodology selection, and modern development tools can effectively replace manual workflows and support long-term organizational growth.

Table of Contents

Course Submission Cover Sheet.....	2
Executive summary.....	4
List of figures.....	10
List of table	12
Abbreviations.....	14
1. Introduction.....	1
1.1 Background Study.....	1
1.2 Motivation of the Project	2
1.3 Scope of the Project	5
1.4 Software Requirements.....	5
1.4.1 Functional Requirements	6
1.4.2 Non-Functional Requirements	9
1.4.3 Domain Requirements	11
1.5 Team Collaboration and Task Allocation	11
Chapter 02.....	14
2. Software Designing.....	14
2.1 System Design Overview.....	14
2.2 Development Methodology	14
2.2.1 Selected Methodology	14
2.2.2 Justification for Methodology Selection.....	15
2.3 Tools and Technologies	16
2.4 Development Requirements.....	19
2.4.1 Hardware Requirements.....	19
2.4.2 Software Requirements.....	20
2.5 Running Requirements (End-User Environment).....	21
2.5.1 Hardware Requirements.....	21
2.5.2 Software Requirements.....	22
2.6 Server Requirements.....	24
2.6.1 Hardware Requirements.....	24
2.6.2 Server Software Requirements	25
2.5 Feasibility Study	26
2.5.1 Introduction.....	26

2.5.2 Technical Feasibility	26
2.5.3 Economic Feasibility	28
2.5.6 Legal and Regulatory Feasibility	30
2.5.7 Schedule Feasibility	32
2.6 Use Case Diagram and Explanation	34
2.6.1 Use Case Diagram.....	34
2.6.2 Diagram Explanation	35
2.6.3 Real day scenario for the RDC Sales Representative and the RDC Clerk	38
2.7. Class Diagram.....	40
2.7.1. Class Diagram Overview	40
2.7.2. Entity Attributes and Relationships	46
2.7.3. Relationship Specifications and Quantity Constraints.....	48
2.7.4. Relationship Implementation Patterns	50
2.8 Database Schema	51
2.9 High-Level Architecture Diagram	70
2.9.1 Analysis of System Architecture	70
2.10. Gantt chart.....	75
2.10.1 Project phases and task breakdown.....	76
2.10.2 Timeline and Scheduling Approach.....	77
2.10.3 Task Sequence and Phase Relationships.....	77
2.10.4 Progress Tracking and Color Coding.....	78
2.10.5. Milestones and Dependencies.....	79
Chapter 3 – Implementation.....	130
3.1 Implementation Overview	130
3.2 Implemented Features	131
3.3 System Workflow	132
3.4 Implementation Challenges	133
4. Testing.....	134
4.1 Testing Overview.....	134
4.3 Testing Strategy	134
4.3.1 Selected Approach – Manual Black Box Testing.....	134
4.4 Unit Testing	135
4.4.1 The Objective of Unit Testing	135

4.4.2 The Unit Testing Process	135
4.4.3 Key Areas Tested.....	136
4.4.4 Significant Benefits.....	136
4.5 Integration Testing	137
4.5.1 The Objective of Integration Testing.....	137
4.5.2 Integration Strategy.....	137
4.5.3 Key Modules Integrated.....	138
4.5.4 Expected Outcomes	138
4.6 System Testing.....	139
4.6.1 Scope of System Testing.....	139
4.6.2 Non-Functional Aspects.....	140
4.6.3 Expected Outcomes	140
4.7 Test Case Design Techniques	142
4.7.1 Boundary Value Analysis	142
4.7.2 Equivalent Partitioning	142
4.8 User Acceptance Testing	143
4.8.1 Methodology	143
4.8.2 User Feedback Survey	144
4.8.3 Results and Analysis	146
4.9 Test Documentation and Traceability	146
4.10 Defect Management	146
4.11 UAT Conclusion	149
5 Conclusion	150
6 Individual Contribution.....	153
6.1 Introduction.....	153
6.2 Role and Responsibilities.....	153
• Officially Assigned Responsibilities (Yasas - 25026764).....	153
6.2.1 System Architecture & Project Structure Creation.....	154
6.2.2 Database Initialization & Migration Strategy	156
6.2.3 Authentication, Authorization & Security Engineering.....	156
6.2.4 Secure Password Reset System.....	160
6.2.5 Google OAuth Integration	161
6.2.6 Role-Based Dashboards & Analytics.....	162

6.2.7 Reporting & Export Features	167
6.2.8 Edit Profile Function for All Users	171
6.2.9 Progressive Web App (PWA) Enhancement	172
6.2.10 Version Control Strategy (Git & Collaboration)	173
6.2.11 Server management and hosting the site.....	175
6.2.12 Evaluation of Development Approach.....	177
6.3 Unit Testing and Quality Assurance	177
6.3.1 Testing Scope and Coverage.....	177
6.3.2 Mocking and Isolation Strategy	182
6.3.3 Code Coverage Implementation	182
6.3.4 Challenges Encountered while unit testing.....	182
6.3.4 Unit Test Repository Reference.....	183
6.3.5 Manual Black-Box Testing	183
6.3.6 Evaluation of unit test approach	186
6.4 Team Support & Leadership	186
6.5 Contribution to Documentation	188
6.5 Individual Conclusion.....	189
7. Appendix.....	190
References.....	191

List of figures

Figure 1 Task among team members	13
Figure 2 feasibility Gantt chart	34
Figure 3 User Case Diagram	35
Figure 4 Class Diagram	41
Figure 5 Database Schema Diagram	52
Figure 6 High-Level Architecture Diagram	71
Figure 7 Gantt Diagram	76
Figure 8 Login	82
Figure 9 Login	83
Figure 10 Register	84
Figure 11 Customer Registration screen.	85
Figure 12 customer Dashboard	86
Figure 13 Mobile view of customer dashboard	87
Figure 14 Browse products interface	88
Figure 15 Browse products mobile view	89
Figure 16 Mobile view of Products page	90
Figure 17 Shopping cart	91
Figure 18 Shopping cart mobile view	92
Figure 19 Shopping cart mobile view	93
Figure 20 Checkout	94
Figure 21 Payment Page	95
Figure 22 My Orders page	96
Figure 23 Order information page	97
Figure 24 Sales Ref dashboard	98
Figure 25 Sales ref order page	99
Figure 26 Sales ref order page with filter option	100
Figure 27 Logistic officer overview	101
Figure 28 Create Schedule Interface	101
Figure 29 Order schedule History interface	102
Figure 30 Stock Report Interface	103
Figure 31 Order Tracking Interface	104
Figure 32 RDC Driver overview interface	106

Figure 33 Update Delivery Staters	107
Figure 34 Routing Map Interface	108
Figure 35 Delivery History Interface	109
Figure 36 Delivery Rider Profile Interface	110
Figure 37 Admin dashboard	111
Figure 38 Users Manage	112
Figure 39 Manage Products	112
Figure 40 Audit Log	113
Figure 41 Head office manager dashboard	114
Figure 42 Head office manager dashboard	116
Figure 43 Stock report	117
Figure 44 Stock report	118
Figure 45 delivery report	118
Figure 46 delivery report	119
Figure 47 RDC Manager dashboard	121
Figure 48 Request transfer units view	123
Figure 49 Transfer units approval view	124
Figure 50 Stock movements view	126
Figure 51 Stock reports dashboard	127
Figure 52 RDC Clerk dashboard	128
Figure 53 Promotions view page	129
Figure 54 Request transfer units view of RDC Clerk	130
Figure 55 Stock reports dashboard view of RDC Clerk	131
Figure 56 Test cases	144
Figure 57 Test cases	145
Figure 58 google form for feedback	147
Figure 59 google form for feedback	149
Figure 60 google form for feedback analysis	151
Figure 61 google form for feedback analysis	152
Figure 62 google form for feedback analysis	153
Figure 63 google form for feedback analysis	154

List of table

Table 1 Tools and Technologies	17
Table 2 Hardware Requirements	20
Table 3 Software Requirements	21
Table 4 (End-User Environment) Hardware Requirements	22
Table 5(End-User Environment) Software Requirements	23
Table 6 Server Requirements Hardware Requirements	25
Table 7 Server Software Requirements	26
Table 8 Technical Feasibility	27
Table 9 Estimated Budget for ISDN System Development	29
Table 10 Classes, Properties, and Methods	37
Table 11 rdc	53
Table 12 rdc_districts	53
Table 13 users	54
Table 14 retail customers	55
Table 15 rdc_sales_refs	55
Table 16 rdc clercks	56
Table 17 rdc managers	56
Table 18 rdc drivers	57
Table 19 rdc logistic officers	58
Table 20 head office mangers	58
Table 21 system admins	59
Table 22 product catergories	59
Table 23 products	60
Table 24 promotions	61
Table 25 ordes	61
Table 26 order items	62
Table 27 odder delivers	63
Table 28 payments	63
Table 29 product stocks	64
Table 30 stock_transfe	65

Table 31 stock_transfer_items	66
Table 32 transfer status logs	66
Table 33 stock movement logs	67
Table 34 shopping carts	68

Abbreviations

1. ISDN – IslandLink Sales Distribution Network
2. FMCG – Fast-Moving Consumer Goods
3. RDC – Regional Distribution Centre
4. HO – Head Office
5. KPI – Key Performance Indicator
6. UI – User Interface
7. UX – User Experience
8. API – Application Programming Interface
9. DBMS – Database Management System
10. GPS – Global Positioning System
11. CRUD – Create, Read, Update, Delete

1. Introduction

IslandLink Sales Distribution Network (ISDN) is one of the largest Fast-Moving Consumer Goods (FMCG) distributors on the island, serving more than 5,000 active retail outlets, supermarkets, and small resellers. The company distributes a wide range of products including packaged foods, beverages, home cleaning items, and personal care products. ISDN operates through five Regional Distribution Centres (RDCs) located in the North, South, East, West, and Central provinces, ensuring nationwide coverage and timely deliveries within 24 to 48 hours. The Head Office (HO), situated in the Central region, is responsible for strategic planning, procurement, and overall corporate management.

Despite its wide operational reach and strong market presence, ISDN currently depends heavily on manual and semi-manual processes for order handling, inventory management, delivery scheduling, billing, and reporting. These traditional practices have resulted in inefficiencies, inaccuracies, and delays across the distribution chain. To address these challenges and remain competitive in a rapidly digitising distribution industry, ISDN plans to implement a Centralised Island-wide Sales Distribution Management System that integrates all RDCs, the Head Office, and retail customers onto a single web-based, centralized digital platform.

1.1 Background Study

At present, ISDN's operational workflow is fragmented and largely manual. Retailers place orders through phone calls, emails, or visits by sales representatives, after which orders are manually communicated to the respective RDCs. Clerks at RDCs enter order details into separate Excel spreadsheets, increasing the risk of data entry errors and inconsistencies. Inventory checks are performed manually using locally maintained stock records, which are often outdated and do not reflect real-time stock movements such as sales, returns, or damaged goods.

Each RDC manages its inventory independently, with only weekly summary reports shared with the Head Office. This lack of real-time synchronisation leads to frequent discrepancies

between physical and recorded stock levels, making it difficult to accurately plan replenishments or inter-branch transfers. Delivery scheduling is also handled manually, without route optimisation tools or GPS tracking, resulting in inefficient routes and no shipment visibility for customers.

Billing and payment collection occur at the time of delivery using manually prepared invoices, with payments typically collected in cash or cheque. Monthly performance reports on sales, inventory turnover, and delivery efficiency are compiled manually, delaying access to critical business insights. These limitations negatively affect operational efficiency, customer satisfaction, and management decision-making, especially when competitors are increasingly adopting digital distribution platforms.

1.2 Motivation of the Project

The primary motivation for this project arises from the significant operational inefficiencies and business risks faced by IslandLink Sales Distribution Network (ISDN) due to its continued reliance on manual and semi-manual processes. Although ISDN is one of the island's largest wholesale and retail distribution companies, supplying over 5,000 active retail outlets through five Regional Distribution Centres (RDCs), its internal operational model has not evolved at the same pace as its business growth. This mismatch between scale and operational capability has created systemic weaknesses that directly affect accuracy, efficiency, customer satisfaction, and strategic decision-making.

At present, order placement is handled through phone calls, emails, or physical visits by sales representatives. These orders are then manually communicated to RDCs, where clerks re-enter the same information into Excel spreadsheets. This multi-step manual process introduces a high risk of data duplication, missing details, and human error. For example, a retailer may place an order for multiple FMCG items over the phone, but due to miscommunication or incorrect data entry at the RDC, quantities or product codes may be recorded inaccurately. Such errors often go unnoticed until the delivery stage, resulting in partial deliveries, incorrect invoices, or order cancellations. These mistakes directly harm retailer trust and damage ISDN's service reputation.

Inventory management presents an even more critical challenge. Each RDC maintains its own stock records independently using spreadsheets that are updated manually. These records are

only summarised and sent to the Head Office on a weekly basis. As a result, inventory information is often outdated and does not reflect real-time stock movements such as sales, returns, damages, or inter-RDC transfers. For instance, a sales representative may promise a retailer that a product is available based on an outdated stock report, only for the RDC to later discover that the item is out of stock. This leads to order delays, backorders, or customer dissatisfaction. The lack of real-time visibility across RDCs also prevents efficient inventory balancing, where excess stock in one region could otherwise be transferred to meet demand in another.

Delivery operations further highlight the urgent need for digital transformation. Delivery scheduling at ISDN is currently performed manually by logistics officers who prepare daily routes without access to automated route optimisation tools or GPS tracking. Drivers are assigned based on experience rather than real-time data, and customers are not notified of delivery progress. From the retailer's perspective, this creates uncertainty, as they have no visibility into when their goods will arrive. If a delivery is delayed due to traffic, vehicle breakdowns, or route inefficiencies, retailers are left uninformed. In today's competitive FMCG distribution environment, this lack of transparency is unacceptable and significantly reduces customer confidence.

Billing and payment collection processes also suffer from inefficiencies. Invoices are generated manually at the point of delivery, and payments are often collected in cash or by cheque. These payments must then be physically handed over to the RDC accounts department, increasing the risk of delays, loss, or misreporting. For example, reconciling daily collections with issued invoices becomes time-consuming and error-prone, making it difficult to accurately track outstanding payments. This directly affects ISDN's cash flow management and financial planning. Additionally, customers increasingly expect digital invoices and online payment options, which the current system fails to provide.

Reporting and management decision-making are heavily impacted by the manual nature of operations. Key performance indicators such as sales trends, stock turnover, delivery efficiency, and staff performance are compiled manually at the end of each month. By the time reports reach management, the data is already outdated. This prevents proactive decision-making and limits the organisation's ability to respond quickly to demand changes, supply chain disruptions, or underperforming regions. For example, without real-time analytics,

management cannot easily identify slow-moving stock or optimise distribution routes to reduce fuel costs and delivery times.

Another strong motivation for this project comes from evolving customer expectations. Retailers today expect self-service ordering platforms, instant order confirmation, real-time order tracking, digital billing, and faster communication. ISDN's existing manual processes cannot meet these expectations, placing the company at a competitive disadvantage. Rival distributors have already adopted modern online platforms that offer seamless ordering experiences and delivery visibility. As a result, ISDN risks losing customers not because of product quality or pricing, but due to inferior service delivery and outdated operational practices.

From a strategic perspective, the lack of integration between RDCs and the Head Office significantly restricts effective planning and control. Without a centralized system, Head Office management cannot obtain an accurate, real-time view of nationwide operations. Decisions related to procurement, promotions, inventory redistribution, and capacity planning are made based on incomplete or delayed information. This increases operational costs and reduces overall efficiency. A centralized system would enable management to monitor performance across all regions, identify bottlenecks, and make data-driven decisions that align with business objectives.

The motivation for this project is therefore rooted in the urgent need to modernise ISDN's distribution operations. By introducing a centralized, web-based Sales Distribution Management System, ISDN can eliminate redundant manual work, reduce errors, improve delivery efficiency, and enhance transparency across all operational levels. Real-time inventory synchronization, automated order processing, GPS-based delivery tracking, digital invoicing, and management dashboards will collectively transform ISDN's operational capability.

this project is motivated not only by the desire to improve efficiency, but by the necessity to ensure ISDN's long-term sustainability and competitiveness. Without digital transformation, ISDN risks operational stagnation, declining customer satisfaction, and loss of market share. The proposed system provides a scalable, reliable, and future-ready solution that supports data-driven management, meets modern customer expectations, and enables ISDN to continue expanding its island-wide distribution network with confidence.

1.3 Scope of the Project

The scope of this project covers the design and implementation of a web-based Centralised Island-wide Sales Distribution Management System that integrates all operational units of ISDN. The system will provide a unified order management portal accessible via web and mobile devices, allowing retail customers to browse products, place orders, receive instant confirmations, and view estimated delivery dates.

The project includes real-time inventory synchronisation across all RDCs, ensuring accurate stock visibility and supporting inter-branch stock transfers. Delivery scheduling and tracking will be enhanced through route optimisation and GPS integration, enabling real-time tracking for both customers and internal staff. Automated billing and payment features will generate digital invoices and support online payment gateway integration for faster settlements.

The system will also provide a management dashboard with real-time analytics, reporting, and KPI tracking for sales performance, inventory turnover, delivery efficiency, and staff productivity. Role-based access control will be implemented to ensure secure system usage by different user groups such as customers, RDC staff, logistics teams, and Head Office managers. The project scope focuses on improving operational efficiency, accuracy, transparency, and customer experience while ensuring data security and compliance standards are met.

1.4 Software Requirements

Software requirements define what the proposed Centralised web-based Island-wide Sales Distribution Management System for ISDN must achieve to support business operations efficiently and reliably. These requirements are categorised into functional, non-functional, and domain-specific requirements.

1.4.1 Functional Requirements

The system must provide the following core functionalities,

1. Common System Functions (All Users)

1. The system shall allow all users to log in using a unique username and password.
2. The system shall allow users to view and update their personal profile information.
3. The system shall display a role-based dashboard after successful login.

2. Customer / Retailer Functional Requirements

4. The system shall allow customers to create a new account and register as a retail outlet.
5. The system shall allow customers to browse available products with prices and descriptions.
6. The system shall allow customers to browse promotional and discounted items.
7. The system shall allow customers to add products to a shopping cart.
8. The system shall allow customers to update quantities or remove items from the shopping cart.
9. The system shall allow customers to place orders using the shopping cart.
10. The system shall send order confirmation notifications to customers via email.
11. The system shall automatically generate a digital invoice for each confirmed order.
12. The system shall allow customers to make online payments for placed orders.
13. The system shall allow customers to track their order delivery status in real time.
14. The system shall allow customers to view order history.
15. The system shall allow customers to view detailed information of past and current orders.

3. Sales Representative Functional Requirements

16. The system shall allow sales representatives to log in securely.
17. The system shall allow sales representatives to browse products and promotional items.
18. The system shall allow sales representatives to place orders on behalf of customers.
19. The system shall generate invoices and send order confirmation emails for sales-rep-placed orders.
20. The system shall allow sales representatives to view customer order history and order details.
21. The system shall allow sales representatives to track order delivery status.

4. RDC Clerk Functional Requirements

22. The system shall allow RDC clerks to log in to the system.
23. The system shall allow RDC clerks to view all orders assigned to their RDC.
24. The system shall display available stock levels for the RDC in real time.
25. The system shall allow RDC clerks to create new promotional items.
26. The system shall allow RDC clerks to view existing promotions.
27. The system shall allow RDC clerks to view and manage customer details related to their RDC.

5. RDC Manager Functional Requirements

28. The system shall allow RDC managers to log in securely.
29. The system shall allow RDC managers to view all available products and stock within their RDC.
30. The system shall allow RDC managers to request stock transfers from other RDCs.
31. The system shall allow RDC managers to approve or reject incoming stock transfer requests.

6. Logistics Officer Functional Requirements

- 32. The system shall allow logistics officers to log in to the system.
- 33. The system shall allow logistics officers to view all orders that are ready for delivery.
- 34. The system shall allow logistics officers to create delivery schedules.
- 35. The system shall allow logistics officers to assign drivers to delivery routes.

7. RDC Driver Functional Requirements

- 36. The system shall allow drivers to view their assigned delivery schedules.
- 37. The system shall allow drivers to update delivery status (dispatched, in transit, delivered).
- 38. The system shall allow drivers to view their delivery history.

8. Head Office Manager Functional Requirements

- 39. The system shall allow Head Office managers to log in securely.
- 40. The system shall allow Head Office managers to view RDC-wise stock availability.
- 41. The system shall generate RDC-wise sales performance reports.
- 42. The system shall generate RDC-wise delivery efficiency reports.

9. System Administrator Functional Requirements

43. The system shall allow system administrators to create, update, and deactivate system users.
44. The system shall allow system administrators to register new products in the system.
45. The system shall allow system administrators to manage user roles and access permissions.

1.4.2 Non-Functional Requirements

The system must meet the following quality and performance standards,

1. The system shall reliably support all critical operations, including orders, inventory updates, delivery tracking, invoicing, and payments, ensuring accurate data without loss or duplication even during high usage or failures.
2. The system shall ensure high availability and reliability to support daily business operations.
3. The system shall support simultaneous access by users from all RDCs and Head Office without significant performance delays.
4. The system shall operate reliably across all Regional Distribution Centres, Head Office, and retail customers without unexpected system failures.
5. The system shall ensure that all critical transactions, including order placement, inventory updates, delivery scheduling, invoicing, and payments, are processed accurately and completely.
6. The system shall prevent data loss or duplication in the event of system errors, network failures, or power interruptions.
7. The system shall maintain data consistency across all RDCs through reliable real-time synchronization mechanisms.
8. The system shall automatically save and recover in-progress transactions to ensure business continuity.

9. The system shall perform automatic data backups at regular intervals to protect critical operational data.
10. The system shall support reliable data restoration to resume normal operations within an acceptable recovery time after a failure.
11. The system shall ensure consistent system performance during peak usage periods, such as high-volume order placements.
12. The system shall log system errors and failures to support timely detection and corrective actions.
13. The system shall provide stable and dependable service to support uninterrupted island-wide distribution operations.
14. The system shall secure all sensitive data using authentication, authorisation, and encryption mechanisms.
15. The system shall be user-friendly and accessible on desktop and mobile devices.
16. The system shall be scalable to accommodate future growth in customers, products, and RDCs.
17. The system shall be maintainable, allowing easy updates, enhancements, and error handling.
18. The system shall include data backup and recovery features to prevent data loss.

1.4.3 Domain Requirements

The system must comply with ISDN's operational and business environment,

1. The system must support FMCG distribution processes such as high-volume orders and fast inventory turnover.
2. The system must operate across multiple geographically distributed RDCs under centralised management.
3. The system must support delivery timelines of 24–48 hours based on regional logistics practices.
4. The system must maintain audit trails for orders, inventory movements, and payments.
5. The system must align with standard wholesale and retail distribution practices followed by ISDN.

1.5 Team Collaboration and Task Allocation

The development of the IslandLink Sales Distribution Network (ISDN) system was carried out by a multi-disciplinary team, with each member assigned specific responsibilities aligned to their expertise. Efficient collaboration and clear task allocation were essential to ensure timely delivery, maintain quality, and avoid duplication of work. The team followed an Agile methodology, which allowed incremental development, regular progress reviews, and effective communication between members.

The table below illustrates the distribution of tasks among team members, detailing each member's assigned responsibilities to ensure clear role allocation and efficient project progress.

Team Member	LMU ID	Document Tasks	Development Tasks
E280041 – Mr E.K.Y.P Fernando	25026764	Chapter3 - implementation	Architectural & UI Template Design and development
			User creation/Management
			User Login and role based access
			Mobile app support enable via PWA support (APK)
			Delivery efficiency report
			Application Hosting
			Git Organization/Git creation
E286082 – Ms W.G.M Gayara	E286082	Class diagram	Retail Customer Dashboards and Profiles
		Gantt chart	RDC Sales Ref Dashboards and Profiles
		User Manual	
E236222 – Mr J.K.I Hasaranga	25026647	2.4. feasibility study	Delivery Scheduling and driver assign
		User Manual	Delivery Tracking
			UI Design
E283798 – Mr K.N Madanayaka	25026634	Chapter 2.3. tools and technologies list: all the tools	Stock Transfer Process
		Chapter 2.4. development environment	Stock Reports
		Chapter 2.5 Running Requirements (End-User Environment)	Stock Transfer Tracking
		Installation/Deployment steps	Stock Movements
			RDC Manager Dashboard
			Database Migration
E284699 – Mr S.H.D Madushanka	25026826	Chapter 2.1. methodology and justification	Customer and Sales ref -
		System Architecture diagram	Add products to cart
		Schema Diagram	Order checkout
			Order Payment
			Order History and order information0
			Invoice Generation and Email
			RDC wise Sales performance report
E281862 – Ms L.H.A Nimsara	25026797	Chapter 4 - Testing	RDC Manager Dashboards and Profile
		Usecase diagram	Logistics Officer Dashboards and Profile
			Rdc Driver Dashboards and Profile
E231306 – Ms Kosma Senuri Upadhy	25026667	Document Structure creation and assembling final document	Browse/Manage products
		Cover to end of Chapter 1	Browse/Manage Promotions
		Requirement analysing	RDC Clerk Dashboards and Profiles
		Chapter 2.1. methodology and justification	
		Chapter 2.6 Designing	

Figure 1 Task among team members

Collaboration Tools

In addition to Excel, the team leveraged several digital tools to facilitate collaboration

- Git and GitHub – For version control, branch management, and code reviews to prevent conflicts between developers.
- Trello – To manage sprint backlogs, track development progress, and assign user stories to team members.
- Google Meet and Slack – For synchronous and asynchronous communication, ensuring instant resolution of issues.
- Draw.io – For collaborative creation of system diagrams, and flowcharts.

Benefits of Structured Task Allocation

This approach ensured,

- Efficient Work Distribution – Each member focused on specific functional areas, reducing overlap and confusion.
- Clear Accountability – Each task was traceable to a responsible team member, allowing easier performance evaluation.
- Transparency – Progress tracking through Excel and Trello provided visibility to all stakeholders.
- Smooth Integration – Coordinated development of frontend, backend, and database components reduced integration issues.
- Enhanced Quality Assurance – Structured responsibilities allowed dedicated attention to testing, security, and usability.

Chapter 02

2. Software Designing

2.1 System Design Overview

The proposed Centralised Island-wide Sales Distribution Management System for IslandLink Sales Distribution Network (ISDN) is designed as an integrated web-based platform that connects retail customers, Regional Distribution Centres, and the Head Office through a single system. The system follows a centralised architecture with a shared database to enable real-time access to orders, inventory, deliveries, and financial data across all regions. It is built using modular components such as Order Management, Inventory Management, Delivery Management, Billing, User Management, and Reporting, which work together to support end-to-end distribution operations. Role-based access control ensures secure and appropriate system usage, while the scalable and flexible design allows the organisation to expand operations and improve efficiency, transparency, and decision-making across the entire distribution network.

2.2 Development Methodology

2.2.1 Selected Methodology

For the successful development and delivery of any software system, it's an industry standard to employ a suitable development methodology. Although a number of traditional and modern development models are currently available, the proposed development methodology for the ISDN Distribution Management System is Agile combined with the Scrum framework.

ISDN operations are currently based on manual processes and require a transition to a centralized software solution to streamline business activities. Therefore, the ISDN system can be identified as a new system developed using a new technology stack. During the system development process, there is a high possibility of changing requirements, new feature additions, or modification of implemented functionality. Consequently, the selected

methodology must be adaptable to these dynamic conditions and capable of accommodating changes in application scope and product features.

2.2.2 Justification for Methodology Selection

Considering the above factors, Agile has been selected as the most suitable software development methodology to be adopted for this system development. Agile is one of the most widely used project management methodologies across different domains, not only in software development but also in industries such as manufacturing, product development, finance and banking, construction and engineering management, and marketing etc.

Agile Scrum can be considered as a highly suitable software development methodology for the proposed IslandLink Sales Distribution Management System because the project operates within a complex, dynamic, and highly distributed business environment. ISDN's transition from manual, traditional processes to a centralized, island-wide digital distribution platform introduces significant uncertainty in system requirements, business workflows, and user expectations. Agile Scrum provides the flexibility required to manage this uncertainty by providing incremental development and continuous requirements refinement based on stakeholder feedback.

The proposed system involves multiple stakeholder groups, including retail customers, regional distribution Centre (RDC) staff, logistics teams, and head office management. Each of these user groups with set of distinct functional needs. Scrum supports active involvement of project stakeholders through the Product Owner role and regular review sessions by ensuring that evolving key business requirements such as self-service ordering, real-time inventory visibility, and delivery tracking and stock transfer are accurately translated into system functionality. Frequent Sprint Reviews allow stakeholders to validate features early, reducing the risk of misalignment between operational expectations and technical implementation.

The modular nature of the proposed solution aligns well with Scrum's iterative structure. Core components such as order management, inventory synchronization, delivery tracking, billing, and reporting can be developed and delivered incrementally by following Sprint based process. This enables ISDN to realize early operational benefits, such as centralized order processing or

real-time stock updates, without waiting for fully completion of the system. Incremental delivery also reduces implementation risk and supports phased deployment across RDCs.

ISDN's operational challenging features such as unclear inventory synchronization flow, delayed reporting, role base user access, and limited delivery visibility, require continuous inspection and refinement of the requirements. Scrum ceremonies such as Daily Scrums, Sprint Reviews, and Sprint Retrospectives support transparency, fast issue identification, and ongoing process improvement. This is especially valuable in coordinating management across geographically distributed RDCs and ensuring consistent system behavior across the island.

Furthermore, the system's reliance on real-time data, integrations with GPS services, and secure payment gateway introduces technical complexity that takes the advantage from iterative development and frequent testing. Scrum facilitates early validation of integrations and performance under real operational conditions by reducing last stage failures.

Overall, Agile Scrum offers the adaptability, stakeholder engagement, risk control, and incremental value delivery that required to support ISDN's digital transformation. It's a structured and flexible framework makes it well suitable for developing a centralized distribution management system in a competitive and rapidly evolving business environment.

2.3 Tools and Technologies

Table 1 Tools and Technologies

Category	Tool / Technology	Purpose in ISDN Project
Backend Development	PHP 8.3 (Pure)	Business logic, session management and API endpoints.

Frontend (Web & Mobile)	HTML5, Tailwind CSS & JavaScript	The creation of the responsive UI. The web app can act as a mobile app by using features of PWA (Service Workers).
Mobile Strategy	PWA (Progressive Web App)	Browses with Google Chrome as a way of installing the web app to mobile devices so that they can access it navigates like an app.
Database Management	MySQL	Saving relationship information such as rdc's, orders, customer information, and RDC staff information.
Database Design	MySQL Workbench	Visual application to develop the ER diagram and administer the MySQL database.
IDE / Code Editor	Visual Studio Code	The main programming language used in writing the code of PHP, HTML, CSS, and JavaScript.
Version Control	Git & GitHub	Code management to manage cooperation of group members and submission.
Dependency Manager	Composer	The dependencies to PHP, such as the reporting library and payment SDKs.
Diagramming	Draw.io	Development of the UML diagrams and DFDs (Use Case, Class, 3-Tier Architecture).

Project Management	Trello	Monitoring activities, bugs, and progress of the development within the group.
Testing	Selenium	The web application is going to be tested automatically to make sure that it complies with functional requirements.
Reporting Library	Dompdf or TCPDF (<i>via Composer</i>)	Third-party library to generate PDF report on Sales and Inventory summary.
Email Service	Mailtrap (SMTP)	The email delivery is to be simulated to test the order confirmation and notification without spamming actual email addresses.
Maps Integration	Google Maps Platform	Added to the PWA to track the deliveries and see the address.
Payment Gateway	PayHere	Acquiring online payment to retail orders securely.
Server environment	XAMPP (Apache)	Local server application to test PHP and MySQL.

2.4 Development Requirements

2.4.1 Hardware Requirements

Table 2 Hardware Requirements

Component	Minimum Specification	Recommended Specification
Processor (CPU)	Intel Core i3 / AMD Ryzen 3	Intel Core i5/ AMD Ryzen 5 or above.
Memory (RAM)	4 GB	8 GB or above (Enough to run VS Code, Chrome, and XAMPP at the same time)
Storage	128 GB SSD	256 GB SSD (Web development project Standard)
Display	13-inch (1366 x 768)	15 inch(1920x 1080) to enhance viewing of the code.
Mobile Device	Any Smartphone with Chrome	Android/iOS phone to run the PWA installation and responsiveness.
Connectivity	Stable Internet Connection	High-speed Broadband (Needed to update Google maps API and Composer)

2.4.2 Software Requirements

Table 3 Software Requirements

Software Category	Requirement	Description
Operating System	Windows 10/11, macOS, or Linux	The main place of growth.
Local Web Server	XAMPP or WAMP	The Apache server and mysql database are required to run the execution of php.
Runtime Environment	PHP 8.3	Basic language runtime needed to decode the backend code.
Web Browser	Google Chrome	Critical to testing PWA (Manifest/Service Workers) and Lighthouse auditing.
Testing Tools	Selenium WebDriver	Needed to run the automated test scripts on the browser.
Code Editor	Visual Studio Code	Lightweight code editor with PHP, Tailwind and Git extensions.
Documentation	Microsoft Office / Google Docs	Writing the reports of the project and developing the presentations.

2.5 Running Requirements (End-User Environment)

The ISDN system is made to be lightweight and compatible with a variety of devices because it is a web-based platform with Progressive Web App (PWA) features. The minimal requirements for users to effectively operate the system are outlined in the specifications that follow.

2.5.1 Hardware Requirements

Table 4 (End-User Environment) Hardware Requirements

Device Type	Component	Minimum Specification	Recommended Specification
Desktop / Laptop (For RDC Managers & Head Office)	Processor	Intel Core i3 / AMD Ryzen 3 or equivalent	Intel Core i5 or higher
	Memory (RAM)	4 GB	8 GB (For smooth multitasking with Excel/Reports)
	Display	1366 x 768 Resolution	1920 x 1080 (Full HD)
	Network	Stable Broadband Connection (2 Mbps)	High-Speed Fiber/4G (10 Mbps+)
Mobile / Tablet	Processor	Quad-core 1.4 GHz	Octa-core 2.0 GHz or higher

<i>(For Drivers & Retail Customers)</i>			
	Memory (RAM)	2 GB	4 GB or higher
	Sensors	GPS Module (Required for Delivery Tracking)	GPS + Accelerometer
	Connectivity	3G / 4G Data Connection	4G LTE / Wi-Fi

2.5.2 Software Requirements

Table 5(End-User Environment) Software Requirements

Component	Requirement	Notes for PWA Compatibility
Web Browser (Desktop)	Google Chrome (v80+) Microsoft Edge (v80+) Mozilla Firefox (v75+)	For the best desktop PWA installation experience, Google Chrome or Edge are strongly advised.

	Safari (v13+)	
Mobile Operating System	Android 7.0 (Nougat) or higher iOS 13.0 or higher	Although Android 5.0+ is technically supported, stable Service Worker performance is best achieved with Android 7.0+.
Mobile Browser	Google Chrome for Android Safari for iOS	To install the app to the home screen on Android using the "Add to Home Screen" prompt, Chrome is necessary. Installing iOS PWA requires Safari.
System Permissions	Location Services (GPS)	"Location Access" is required in order to track deliveries in real time.
Internet Protocol	HTTPS (SSL)	The application must be served over a secure HTTPS connection in order to use PWA features (Service Workers).

2.6 Server Requirements

The system will be installed on a Linux-based shared hosting environment that is controlled by cPanel in order to guarantee that the ISDN Centralised Web Portal and Progressive Web App (PWA) are available to all RDCs and retail clients around-the-clock. Because of its affordability, ease of upkeep, and integrated support for crucial security features like SSL encryption, this strategy was chosen.

2.6.1 Hardware Requirements

For the shared hosting environment to effectively manage concurrent user requests and database transactions, the following hardware resources must be assigned.

Table 6 Server Requirements Hardware Requirements

Component	Specification	Description / Justification
Processor (CPU)	2 – 8 vCPU Cores	During peak hours, virtual CPU allocation ensures quick execution of database queries and stock reports.
Memory (RAM)	1 GB – 4 GB (Shared)	enough memory allotted to manage both the mobile PWA's API calls and the concurrent user requests from RDC Managers.
Storage Space	10 GB (SSD Recommended)	necessary to store the growing database transaction logs, uploaded product images, and the application code.

2.6.2 Server Software Requirements

To run the PHP 8.3 backend and secure the PWA connection, the server environment needs to support the following software stack.

Table 7 Server Software Requirements

Component	Requirement	Description / Justification
Hosting Control Panel	cPanel	allows for the easy management of files, databases, and PHP versions without the need for extensive knowledge of Linux administration.
Web Server Language	PHP 8.3	For the application's core logic and Composer dependencies to function, the server needs to support the most recent version of PHP.
Database Engine	MySQL (v7.2+)	It is necessary to have a local or remote MySQL instance. Note- For improved performance with PHP 8.3, v8.0 is advised.
Security Protocol	SSL Certificate (Free)	PWA functionality is required (HTTPS is required for Service Workers). We make use of free Auto-SSL (like Google Trust Services).
Operating System	Linux (CloudLinux/CentOS)	designed to run PHP applications in a shared environment with maximum security and stability.

2.5 Feasibility Study

2.5.1 Introduction

This feasibility report evaluates the practicality of implementing a Centralised Island-wide Sales Distribution Management System for IslandLink Sales Distribution Network (ISDN). The purpose of this report is to assess whether the proposed system is viable from technical, economic, operational, legal, and schedule perspectives. The findings of this feasibility study will support management in deciding whether to proceed with the system development project.

2.5.2 Technical Feasibility

Technical feasibility examines whether the required technology, infrastructure, and technical expertise are available to develop and operate the proposed system successfully.

The proposed system will be a web-based and mobile-compatible platform supported by a centralised database. Such technologies are widely available and mature, including cloud hosting, web application frameworks, and online payment gateways. ISDN already uses computers, internet connectivity, and basic digital tools such as spreadsheets across its RDCs and Head Office, which provides a foundation for system adoption.

Table 8 Technical Feasibility

Category	Tool / Technology
Backend Development	PHP 8.3 (Pure)

Frontend (Web & Mobile)	HTML5, Tailwind CSS & JavaScript
Mobile Strategy	PWA (Progressive Web App)
Database Management	MySQL
Database Design	MySQL Workbench
IDE / Code Editor	Visual Studio Code
Version Control	Git & GitHub
Dependency Manager	Composer
Diagramming	Draw.io
Project Management	Trello

The integration of real-time inventory updates, delivery tracking, And the modular system design allows individual components such as order management, inventory, and reporting to be developed and integrated gradually.

Therefore, from a technical perspective, the proposed system is feasible and can be implemented using proven and reliable technologies without excessive technical risk.

2.5.3 Economic Feasibility

Economic feasibility evaluates whether the benefits of the proposed system outweigh its development and operational costs.

The initial costs of the project include system development, hardware upgrades if necessary, cloud hosting, staff training, and ongoing maintenance. However, these costs are justified by the significant long-term benefits the system will provide. Automation of order processing, inventory updates, billing, and reporting will reduce manual labour, minimise errors, and lower operational costs.

Improved inventory accuracy will reduce losses caused by overstocking, stockouts, and expired products. Faster order processing and delivery visibility will enhance customer satisfaction, helping ISDN retain existing customers and attract new ones. Real-time analytics will support better decision-making, leading to improved sales performance and route efficiency.

Overall, the long-term financial and operational benefits of the system outweigh the initial investment, making the project economically feasible.

Project Budget Estimate

Estimated Budget for ISDN System Development

Table 9 Estimated Budget for ISDN System Development

Category	Item	Description	Estimated Cost (LKR)
Software Tools	Open-source Tools	PHP, MySQL, VS Code, GitHub, Selenium, Draw.io	0
	Composer Libraries	Dompdf / TCPDF, payment SDKs	0
	Project Management Tools	Trello (Free plan)	0
Server & Hosting	Local Server Setup	(Free plan)	0
Payment Integration	PayHere Setup	Sandbox and test integration	0

Hardware	Development Laptops	Already available (student / team-owned)	0
	Mobile Devices	Personal smartphones for PWA testing	0
Internet & Utilities	Internet Charges	Development and testing usage	3,000
Training & Documentation	Documentation Tools	Google Docs / MS Office	0
Contingency	Miscellaneous Costs	Unexpected minor expenses	2,000
Payment Integration	PayHere Setup	Sandbox and test integration	0
Hardware	Development Laptops	Already available (student / team-owned)	0
	Mobile Devices	Personal smartphones for PWA testing	0
Internet & Utilities	Internet Charges	Development and testing usage	3,000
Training & Documentation	Documentation Tools	Google Docs / MS Office	0
Contingency	Miscellaneous Costs	Unexpected minor expenses	2,000
		Total Estimated Budget	5,000 LKR

2.5.4 Operational Feasibility

Operational feasibility assesses how well the proposed system will fit into ISDN's existing business processes and whether users will be able to adopt it effectively.

The system is designed to align closely with ISDN's current distribution workflow while replacing manual tasks with automated processes. Retail customers will benefit from self-service ordering and delivery tracking, reducing reliance on phone calls and manual

communication. RDC staff and logistics teams will be able to perform their duties more efficiently using real-time data and automated tools.

Although staff training will be required, the system's user-friendly interface and role-based access will simplify daily operations. Management will gain timely access to accurate reports and dashboards, improving planning and coordination between RDCs and the Head Office.

As the system improves efficiency, transparency, and communication without disrupting core business operations, it is considered operationally feasible.

2.5 6 Legal and Regulatory Feasibility

Legal feasibility evaluates whether the proposed system complies with applicable laws, regulations, and industry standards.

The system will store customer data, transaction records, and financial information. Therefore, it must comply with relevant data protection, electronic transaction, and consumer protection regulations applicable within the country. Secure authentication, encrypted data transmission, and access control mechanisms will be implemented to protect sensitive information.

Digital invoicing and online payment integration will be designed to comply with standard accounting and taxation practices. Audit trails for orders, inventory movements, and payments will support regulatory compliance and internal controls.

1. Local Acts

a) Electronic Transactions Act (ETA)

The ETA is relevant to the ISDN system because it supports online order placement, digital invoicing, and electronic payments. This act legally recognizes electronic records, digital signatures, and transactions, making them valid and enforceable. For ISDN, this means that all online orders, invoices, and digital payments must be securely recorded, and proper authentication must be implemented to ensure that the system's electronic transactions are legally valid and protected against fraud.

b) Data Protection Act (DPA)

The DPA governs the collection, storage, and processing of customer data, RDC inventory information, and payment details. Compliance with the DPA ensures privacy and security of personal and corporate data. For ISDN, this requires implementing strong access controls, data encryption, and secure storage mechanisms to prevent unauthorized access, misuse, or breaches of sensitive information, ensuring that all customer and operational data remain confidential and secure.

c) Consumer Protection Act (CPA)

The CPA ensures that retail customers receive accurate product information, correct billing, and timely deliveries while protecting them from unfair practices such as overcharging or misrepresentation. For ISDN, this means the system must maintain accurate order records, provide correct invoices, offer reliable delivery tracking, and handle customer complaints effectively to comply with consumer rights and maintain customer trust across all RDCs and retail outlets.

2. Standards and Regulations

a) IEEE Standards

IEEE standards, such as IEEE 829 and IEEE 12207, provide guidelines for software development, testing, and maintenance. Applying these standards to ISDN ensures that the system is developed systematically, tested thoroughly, and documented properly. This reduces legal risks arising from software failures, transaction errors, or operational disputes, while ensuring the system functions reliably across all RDCs and for all customers.

b) ISO Standards

ISO standards, including ISO/IEC 27001 for information security management and ISO 9001 for quality management, are important for the ISDN system because it handles sensitive data and operational processes. Compliance ensures robust security, standardized operational procedures, and consistent quality in service delivery. This reduces the risk of data breaches, fraud, or operational errors while maintaining high-quality distribution and customer service across the island.

c) GDPR (General Data Protection Regulation)

Although GDPR is an EU regulation, it is applicable to ISDN because it may process international customer data or sensitive personal information. GDPR mandates lawful data collection, explicit consent, secure storage, and the rights of individuals to access or delete their data. For ISDN, implementing GDPR principles means using encryption, audit logging, strict access controls, and consent management to protect customer privacy, enhance trust, and ensure compliance with international data protection standards.

As the system can be designed to meet all legal and regulatory requirements, it is legally feasible.

2.5.7 Schedule Feasibility

Schedule feasibility considers whether the system can be developed and deployed within a reasonable and achievable timeframe.

The project can be implemented in phases, starting with core modules such as order management and inventory control, followed by delivery tracking, billing, and reporting. Using an iterative development approach allows early deployment of essential features while reducing project risk

Given the availability of standard development tools, experienced developers, and modular design, the system can be completed within the planned project timeline without major delays. Therefore, the project is considered schedule feasible.

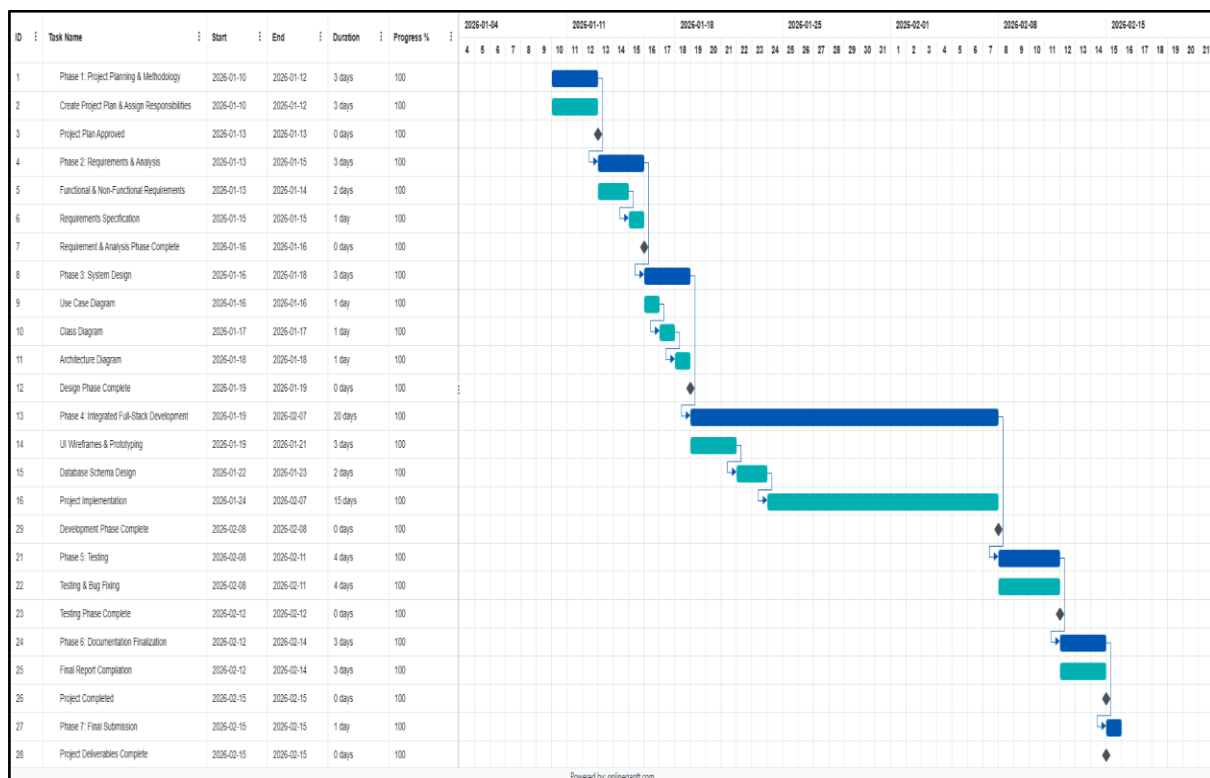


Figure 2 feasibility Gantt chart

Based on the technical, economic, operational, legal, and schedule evaluations, the proposed Centralised Island-wide Sales Distribution Management System for ISDN is feasible. The system addresses existing operational challenges, supports future growth, improves customer satisfaction, and enhances management decision-making. It is recommended that ISDN proceed with the development and implementation of the proposed system.

2.6 Use Case Diagram and Explanation

2.6.1 Use Case Diagram

The following diagram illustrates the functional requirements of the proposed IslandLink Sales Distribution Network (ISDN) system and the interactions between different actors and the system.

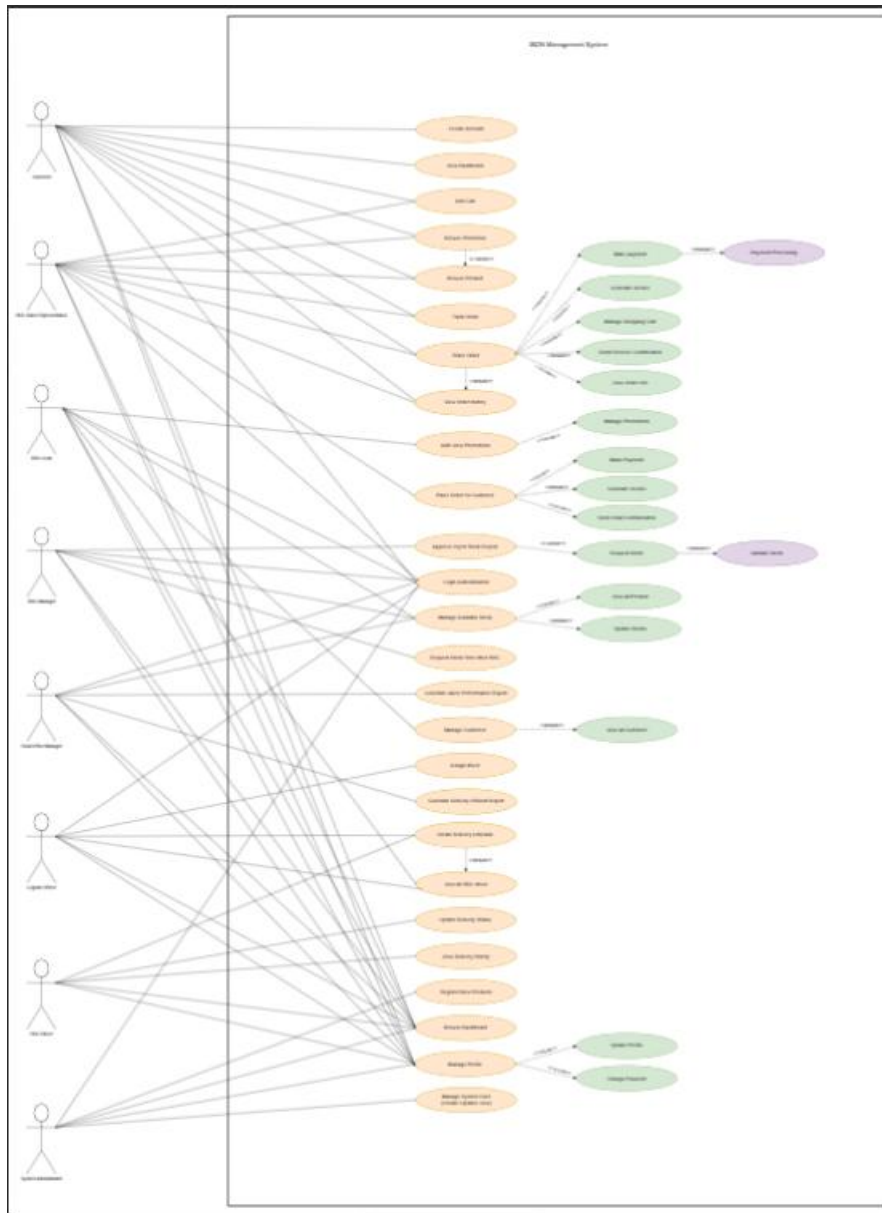


Figure 3 User Case Diagram

Figure 1 Use Case Diagram - reference

https://drive.google.com/file/d/1DytgUtJEIiw54K2emcH0XFGV1lBz1_Si/view?usp=sharing

2.6.2 Diagram Explanation

The Use Case Diagram shows how different users interact with the new digital system. By centralizing all operations, the system replaces manual paperwork and phone calls with a streamlined digital process.

Actors of the ISDN Distribution Management System

1. Customers and Front-line Team

- . **Retail Customer-** This is the regular shopper. They can create an account, browse for organic products or deals, add items to a cart, and pay for their order. They also use the system to track where their package is and see what they bought in the past.

RDC Sales Representative- These are the field staff who help customers. They can place orders on behalf of a customer, handle the payment and make sure the customer gets their invoice and email confirmation.

2. Branch Team (Regional Distribution Center - RDC)

- . **RDC Clerk-** Think of them as the branch organizer. They keep the customer list updated, manage product stock at their location and set up new promotional offers. They are also responsible for checking and approving orders coming into their RDC.

RDC Manager- The person in charge of the branch's inventory. If they are running low on a product they use the system to request stock from another branch. They also decide whether to approve or reject stock requests from other managers.

3. Delivery Team

- . **Logistics Officer-** The coordinator. They look at all the orders that are ready to go, create a delivery schedule and assign a driver to each trip.

- **RDC Driver-** The person on the road. They log in to see their daily schedule and map. After they drop off a package, they update the system so everyone knows the delivery is finished.

4. Tech Team

- **Head Office Manager-** The high-level boss. They don't handle daily orders but look at the "Big Picture" by checking reports on sales performance and how efficient the deliveries are across all branches.

System Administrator- The tech expert. They keep the system running by creating accounts for staff members and adding new products to the database.

Regardless of their role, every person mentioned above can Login, Manage their own profile (like changing a password) and view a Dashboard that shows the most important information for their job.

5. Classes, Properties, and Methods

Table 10 Classes, Properties, and Methods

Class	Properties	Methods
User	UserID FullName Email Phone Address, Role Password	Login() Logout() UpdateProfile()
Customer	CustomerID FullName Email	Register() BrowseProduct() AddToCart()

	Phone Address	PlaceOrder() ViewOrderHistory()
RDC Clerk	ClerkID RDC_Location Designation	ViewRDCOrders() ManageStock() AddPromotion() ManageCustomer()
Product	ProductID CategoryID Name Price StockQuantity Description	AddProduct() UpdateProduct() DeleteProduct() GetProductDetails()
Order	OrderID CustomerID OrderDate TotalAmount Status	CreateOrder() UpdateStatus() CancelOrder() GetOrderInfo()
Promotion	PromoID Name DiscountRate StartDate EndDate	CreatePromo() UpdatePromo() DeletePromo() ViewActivePromos()
Stock	StockID ProductID RDC_ID AvailableQuantity	UpdateStockLevel() CheckAvailability() RequestStock()
Delivery	DeliveryID OrderID DriverID	CreateSchedule() AssignDriver() UpdateDeliveryStatus()

	ScheduleDate	
	Status	

2.6.3 Real day scenario for the RDC Sales Representative and the RDC Clerk

1. Sales Representative (Making a Sale)

"As a Sales Rep, I want to place an order for a customer so they can get their products quickly."

The Scenario- You are visiting a small shop. The shop owner wants to order 50 bottles of organic juice.

The Flow-

1. You Login to the system on your tablet.
2. You Browse Products and find the juice.
3. You Add to Cart 50 bottles.
4. You Place Order on behalf of the shop owner.
5. The system automatically Generates an Invoice and Sends a Confirmation Email to the customer.
6. You confirm the Payment is received.

2. RDC Clerk (Managing Stock)

"As a Clerk, I want to update our stock levels so we don't run out of products."

The Scenario- A new shipment of organic honey just arrived at your warehouse.

The Flow-

0. You can Login to the dashboard.
1. You can click on View RDC Available Stock.
2. You can find "Organic Honey" and select Update Stock.
3. You can add the 100 new jars to the system.

4. Now, when the Customer or Sales Rep looks at the app, they can see the honey is back in stock.

3. Logistics Officer (Scheduling)

"As a Logistics Officer, I want to assign drivers to orders so customers get their items on time."

The Scenario- There are 10 orders ready to be delivered this morning.

The Flow-

1. You can Login and go to View Delivery Ready Orders.
2. You can see the list and click Create Delivery Schedule.
3. You can Assign a Driver (like "Saman") to the route.
4. The can Driver gets a notification on their phone to see their Daily Route Map.

2.7. Class Diagram

2.7.1. Class Diagram Overview

Reference--

<https://drive.google.com/file/d/1oI5-L0Cq5PGbzFM33ac9sio9QteIRdr2/view?usp=sharing>

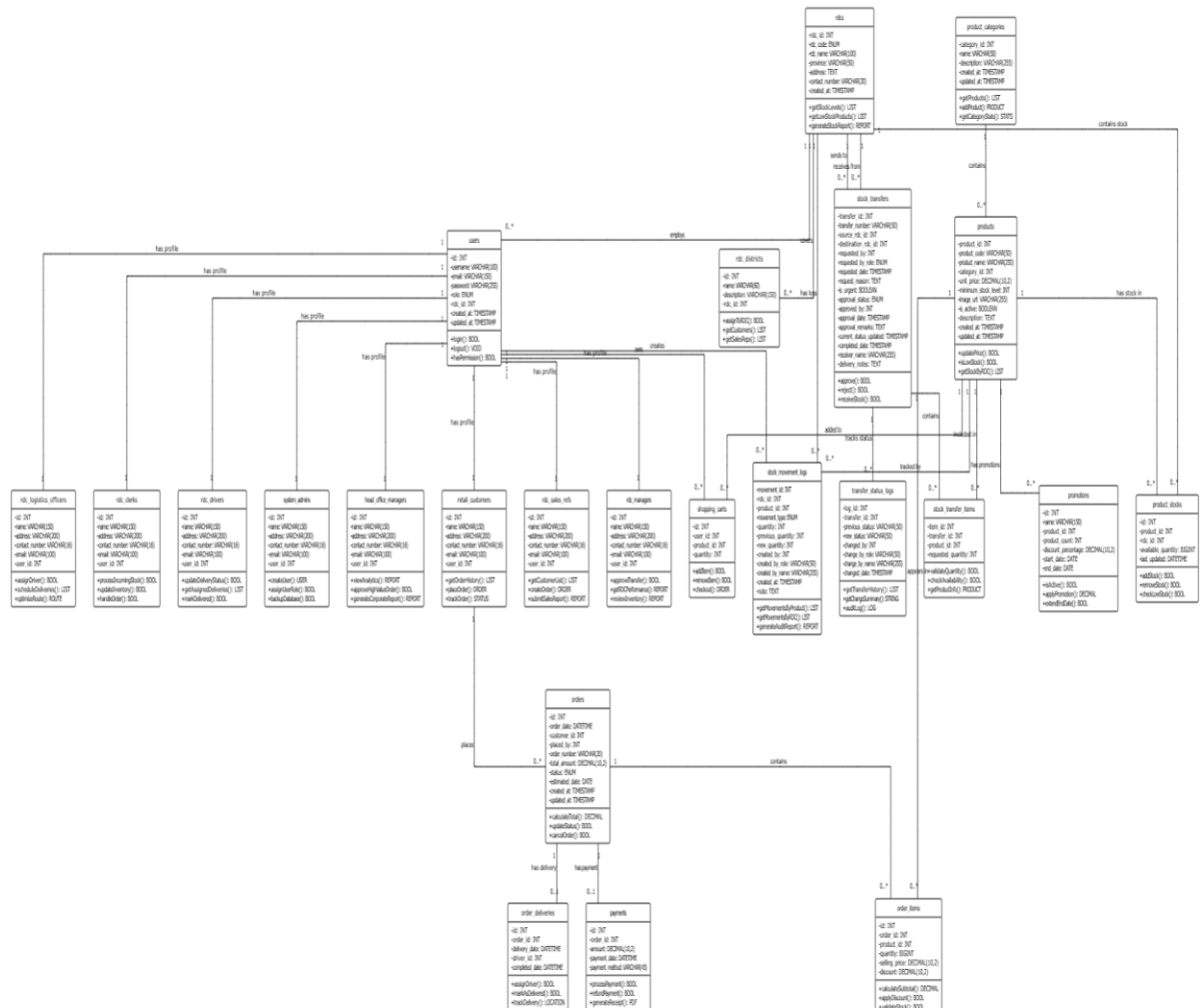


Figure 4 Class Diagram

Class Structure and Specifications

Class	Attributes	Methods
users	id:INT username:VARCHAR(100) email:VARCHAR(150) password:VARCHAR(255) role: ENUM rdc_id: INT created_at:TIMESTAMP updated_at: TIMESTAMP	login(): BOOL logout():VOID hasPermission(): BOOL
rdcs	rdc_id: INT rdc_code:ENUM rdc_name:VARCHAR(10) province:VARCHAR(50) address:TEXT contact_number:VARCHR(20) created_at: TIMESTAMP	getStockLevels():LIST getLowStockProducts(): LIST generateStockReport(): REPORT
product_categories	category_id: INT name:VARCHAR(50) description:VARCHAR(25) created_at:TIMESTAMP updated_at: TIMESTAMP	getProducts():LIST addProduct():PRODUCT getCategoryStats(): STATS

products	product_id:INT product_code:VARCHAR(50) product_name:VARCHAR(255) category_id:INT unit_price:DECIMAL(10,) minimum_stock_level: INT image_url: VARCHAR(255) is_active:BOOLEAN description:TEXT created_at:TIMESTAMP updated_at: TIMESTAMP	updatePrice():BOOL isLowStock():BOOL getStockByRDC(): LIST
retail_customers	id: INT -name: VARCHAR(150) address:VARCHAR(200) contact_number: VARCHAR(16) email: VARCHAR(100) user_id: INT	getOrderHistory():LIST placeOrder():ORDER trackOrder(): STATUS
orders	id: INT order_date:DATETIME customer_id: INT placed_by: INT order_number:VARCHAR(20) total_amount:DECIMAL(10,2) status: ENUM estimated_date:DATE created_at:TIMESTAMP updated_at: TIMESTAMP	calculateTotal(): DECIMAL updateStatus():BOOL cancelOrder(): BOOL

stock_transfers	transfer_id:INT transfer_number: VARCHAR(50) source_rdc_id:INT destination_rdc_id:INT requested_by:INT requested_by_role:ENUM requested_date:TIMESTAMP request_reason:TEXT is_urgent:BOOLEAN approval_status:ENUM approved_by:INT approval_date:TIMESTAMP approval_remarks:TEXT current_status_updated: TIMESTAMP completed_date:TIMESTAMP receiver_name:VARCHAR(255) delivery_notes: TEXT	approve():BOOL reject():BOOL receiveStock(): BOOL
stock_transfer_items	item_id: INT transfer_id: INT product_id:INT requested_quantity: INT	validateQuantity():BOOL checkAvailability():BOOL getProductInfo(): PRODUCT
transfer_status_logs	log_id: INT transfer_id:INT previous_status:VARCHAR(50) new_status:VARCHAR(50) changed_by:INT change_by_role: VARCHAR(50) change_by_name: VARCHAR(255)	getTransferHistory(): LIST getChangeSummary(): STRING auditLog(): LOG

	changed_date: TIMESTAMP	
order_deliveries	id: INT order_id: INT delivery_date: DATETIME driver_id: INT completed_date: DATETIME	assignDriver(): BOOL markAsDelivered(): BOOL trackDelivery(): LOCATION
order_items	id: INT order_id: INT product_id: INT quantity: BIGINT selling_price: DECIMAL(10,2) discount: DECIMAL(10,2)	calculateSubtotal(): DECIMAL applyDiscount(): BOOL validateStock(): BOOL
payments	id: INT order_id: INT amount: DECIMAL(10,2) payment_date: DATETIME payment_method: VARCHAR(45)	processPayment(): BOOL refundPayment(): BOOL generateReceipt(): PDF
product_stocks	id: INT product_id: INT	addStock(): BOOL removeStock(): BOOL checkLowStock(): BOOL

	rdc_id: INT available_quantity: BIGINT last_updated: DATETIME	
promotions	id: INT name: VARCHAR(150) product_id: INT product_count: INT discount_percentage: DECIMAL(10,2) start_date: DATE end_date: DATE	isActive(): BOOL applyPromotion(): DECIMAL extendEndDate(): BOOL
rdc_districts	id: INT name: VARCHAR(60) description: VARCHAR(150) rdc_id: INT	assignToRDC(): BOOL getCustomers(): LIST getSalesReps(): LIST
rdc_sales_refs	id: INT name: VARCHAR(150) address: VARCHAR(200) contact_number: VARCHAR(16) email: VARCHAR(100) user_id: INT	getCustomerList(): LIST createOrder(): ORDER submitSalesReport(): REPORT
rdc_managers	id: INT name: VARCHAR(150)	approveTransfer(): BOOL getRDCPerformance(): REPORT reviewInventory(): REPORT

	address: VARCHAR(200) contact_number: VARCHAR(16) email: VARCHAR(100) user_id: INT	
rdc_logistics_officers	id: INT -name: VARCHAR(150) address: VARCHAR(200) contact_number: VARCHAR(16) email: VARCHAR(100) -user_id: INT	assignDriver(): BOOL scheduleDeliveries(): LIST optimizeRoute(): ROUTE

Table 1: System Class Specifications

This class diagram represents the core entity-relationship model for a comprehensive retail and distribution management system. The system is designed to manage users, products, inventory, orders, logistics, and payments through a structured, object-oriented architecture. It supports eight specialized staff roles (sales representatives, RDC managers, logistics officers, clerks, drivers, head office managers, and system administrators) together with retail customers, tracks products across Regional Distribution Centers (RDCs), handles the complete order lifecycle from cart to delivery, and integrates promotional and payment processing. The architecture follows a modular design where each subsystem (user management, inventory, ordering, logistics, stock transfers, and role-based profiles) maintains clear boundaries while interacting through well-defined relationships, enabling maintainability and scalability for enterprise operations.

2.7.2. Entity Attributes and Relationships

1. User Management Core

The system's foundation is the User entity, which contains universal login credentials (id, username, email, password, created_at, updated_at) along with role selection as an ENUM field

and a direct association to an RDC through `rdc_id`. Each User is employed at one RDC in a many-to-one relationship, meaning multiple users can work at the same distribution center. The role ENUM defines whether a user is a retail customer or one of eight staff types. Each User can optionally be linked via a "has profile" one-to-one relationship to a specific staff profile such as RDC Manager, RDC Sales Ref, RDC Logistics Officer, RDC Clerk, RDC Driver, Head Office Manager, or System Admin or to a Retail Customer profile. This generalization structure allows a user to exist without a specialized profile during registration, and once specialized, can only be one specific profile type. All staff profiles and retail customers share common contact attributes (name, address, `contact_number`, email) and maintain strong identity linkage through the `user_id` foreign key. Additionally, each User can own multiple shopping cart entries and create multiple stock movement logs, establishing one-to-many relationships with these entities.

2. Product and Inventory Core

The Product entity, defined by attributes like name, code, category, `unit_price`, and status, is central to the inventory system. Each Product "has stock" in one or more ProductStock records through a one-to-many relationship, which tracks `available_quantity` at specific RDCCenter locations (via `rdc_id`). This many-to-many relationship between Product and RDCCenter (connected through the ProductStock) allows products to be stored across multiple distribution centers while maintaining separate inventory counts. Products can also be associated with multiple Promotion entities (with `discount_percentage`, `start_date`, `end_date`) through a one-to-many "has" relationship, enabling seasonal or targeted marketing campaigns.

3. Order Management and Shopping

A RetailCustomer (or a User generally) "places" one or more Order entities in a one-to-many relationship, allowing customers to have comprehensive purchase histories. An Order, characterized by `order_number`, amount, status, and `estimated_date`, "contains" one or more OrderItem lines detailing quantity and `selling_price` in a one-to-many relationship. Each OrderItem links back to a single Product that "appears in" the order through a many-to-one

relationship, maintaining a fixed record of the purchase price at purchase time separate from current catalog pricing. Pre-purchase, a User "owns" an optional ShoppingCart in a one-to-zero-or-one relationship, which "contains" multiple Product entries with a temporary quantity through a many-to-many relationship (as the same product can be in multiple carts and carts contain multiple products).

4. Logistics, Delivery, and Payment

For delivery execution, an Order "has" an optional OrderDelivery schedule (delivery_date, driver_id) in a one-to-zero-or-one relationship, supporting both delivery and in-store pickup options. Logistics are managed through RDCCenter entities (name, description), which "serve" multiple RDCDistrict areas in a one-to-many relationship and "maintain" various ProductStock records through another one-to-many relationship. Users are also "assigned to" an RDC via the rdc_id attribute in a many-to-one relationship, enabling organizational structure and localized operations. Following order completion, an Order "has" an optional Payment record storing amount, payment_method, and payment_date in a one-to-zero-or-one relationship, supporting various payment timing scenarios including cash-on-delivery.

2.7.3. Relationship Specifications and Quantity Constraints

1. One-to-One Relationships-

- User → UserRole- Each user has exactly one role (1-1)
- User → StaffUser/RetailCustomer- Each user specializes to at most one profile type (1-0..1)
- Order → Payment- Each order has zero or one payment record (1-0..1)
- Order → OrderDelivery- Each order has zero or one delivery schedule (1-0..1)
-

2. One-to-Many Relationships-

- User → Order- A user can place multiple orders (1-0..*)
- User → ShoppingCart- A user may own zero or many shopping cart items (1-0..*)
- RetailCustomer → Order- A customer can have multiple orders (1-0..*)
- User → StockMovementLog- A user can create multiple stock movement logs (1-0..*)
- StaffUser → Order- Staff can process multiple orders (1-0..*)
- Order → OrderItem- An order contains multiple line items (1-1..*)
- RDCCenter → RDCDistrict- A center serves multiple districts (1-1..*)
- RDCCenter → ProductStock- A center maintains multiple stock records (1-0..*)
- RDCCenter → User- A center has multiple assigned users (1-0..*)
- RDCCenter → StockMovementLog- A center has multiple stock movement logs (1-0..*)
- Product → ProductStock- A product can have multiple stock records across distribution centers (1-0..*)
- Product → StockMovementLog- A product has multiple stock movement logs (1-0..*)
- Product → Promotion- A product can have multiple promotions (1-0..*)
- Product → OrderItem- A product appears in multiple order lines (1-0..*)
- StockTransfer → StockTransferItem- A transfer contains multiple items (1-1..*)
- StockTransfer → TransferStatusLog- A transfer has multiple status logs (1-0..*)

4.Many-to-Many Relationships (through association classes)-

- Product ↔ RDCCenter- Mediated by ProductStock, enabling distributed inventory
- Product ↔ ShoppingCart- Products can be in multiple carts, carts contain multiple products
- Product ↔ Order- Mediated by OrderItem, as products are ordered multiple times
- RDCCenter ↔ RDCCenter- Mediated by StockTransfer for inter-warehouse transfers

2.7.4. Relationship Implementation Patterns

The diagram implements several key relationship patterns-

1. Generalization/Specialization- User to StaffUser/RetailCustomer for role-based behavior
2. Composition- Order to OrderItem for strong lifecycle coupling
3. Aggregation- ShoppingCart to Product for temporary associations
4. Association Classes- ProductStock mediating Product and RDCCenter many-to-many

This model ensures a clear separation of concerns, enabling scalable management of identity, inventory, sales transactions, and distribution logistics within a unified system. The relationship constraints implement business rules while maintaining flexibility, and the relationship patterns support efficient data access and business process flows across all system operations.

rdcs

Stores ISDN Regional Distribution Centers information.

Table 11 rdc

Column Name	Data Type	Key
rdc_id	int	PK
rdc_code	enum	
rdc_name	varchar(100)	
province	varchar(50)	
address	text	
contact_number	varchar(20)	
created_at	timestamp	

rdc_districts

Stores districts information under each RDC.

Table 12 rdc_districts

Column Name	Data Type	Key
id	int	PK
name	varchar(60)	

description	varchar(150)	
rdc_id	int	FK

users

Stores user authentication and role information of the system users.

Table 13 users

Column Name	Data Type	Key
id	int	PK
username	varchar(100)	
email	varchar(150)	UK
password	varchar(255)	
role	enum	
rdc_id	int	
created_at	timestamp	
updated_at	timestamp	

retail_customers

Stores profile information for retail customers of the system.

Table 14 retail customers

Column Name	Data Type	Key
id	int	PK
name	varchar(150)	
address	varchar(200)	
contact_number	varchar(16)	
email	varchar(100)	
user_id	int	FK

rdc_sales_refs

Stores RDC sales representatives profile information.

Table 15 rdc_sales_refs

Column Name	Data Type	Key
id	int	PK
name	varchar(150)	
address	varchar(200)	
contact_number	varchar(16)	
email	varchar(100)	

user_id	int	FK
---------	-----	----

rdc_clerks

Stores RDC clerks profile information.

Table 16 rdc clercks

Column Name	Data Type	Key
id	int	PK
name	varchar(150)	
address	varchar(200)	
contact_number	varchar(16)	
email	varchar(100)	
user_id	int	FK

rdc_managers

Stores RDC managerx profile information.

Table 17 rdc managers

Column Name	Data Type	Key
id	int	PK
name	varchar(150)	

address	varchar(200)	
contact_number	varchar(16)	
email	varchar(100)	
user_id	int	FK

rdc_drivers

Stores RDC drivers profile information.

Table 18 rdc drivers

Column Name	Data Type	Key
id	int	PK
name	varchar(150)	
address	varchar(200)	
contact_number	varchar(16)	
email	varchar(100)	
user_id	int	FK

rdc_logistics_officers

Stores RDC logistics officers profile information.

Table 19 rdc logistic officers

Column Name	Data Type	Key
id	int	PK
name	varchar(150)	
address	varchar(200)	
contact_number	varchar(16)	
email	varchar(100)	
user_id	int	FK

head_office_managers

Stores head office managers profile information.

Table 20 head office mangers

Column Name	Data Type	Key
id	int	PK
name	varchar(150)	
address	varchar(200)	
contact_number	varchar(16)	
email	varchar(100)	

user_id	int	FK
---------	-----	----

system_admins

Stores system administrators profile information.

Table 21 system admins

Column Name	Data Type	Key
id	int	PK
name	varchar(150)	
address	varchar(200)	
contact_number	varchar(16)	
email	varchar(100)	
user_id	int	FK

product_categories

Stores product category information.

Table 22 product categories

Column Name	Data Type	Key
category_id	int	PK
name	varchar(50)	UK
description	varchar(255)	

created_at	timestamp	
updated_at	timestamp	

products

Stores all the Product information with pricing and inventory information.

Table 23 products

Column Name	Data Type	Key
product_id	int	PK
product_code	varchar(50)	UK
product_name	varchar(255)	
category_id	int	FK
unit_price	decimal(10,2)	
minimum_stock_level	int	
image_url	varchar(255)	
is_active	tinyint(1)	
description	text	
created_at	timestamp	
updated_at	timestamp	

promotions

Stores product promotion and discount information.

Table 24 promotions

Column Name	Data Type	Key
id	int	PK
name	varchar(150)	
product_id	int	FK
product_count	int	
discount_percentage	decimal(10,2)	
start_date	date	
end_date	date	

orders

Stores customer sales order information.

Table 25 ordes

Column Name	Data Type	Key
-------------	-----------	-----

id	int	PK
order_date	datetime	
customer_id	int	FK
placed_by	int	
order_number	varchar(20)	UK
total_amount	decimal(10,2)	
status	enum	
estimated_date	date	
created_at	timestamp	
updated_at	timestamp	

order_items

Stores order line items with product details and pricing.

Table 26 order items

Column Name	Data Type	Key
id	int	PK
order_id	int	FK
product_id	int	FK

quantity	bigint	
selling_price	decimal(10,2)	
discount	decimal(10,2)	

order_deliveries

Stores customer order delivery tracking information.

Table 27 order deliveries

Column Name	Data Type	Key
id	int	PK
order_id	int	
delivery_date	datetime	
driver_id	int	
completed_date	datetime	

payments

Stores payment transaction records for orders.

Table 28 payments

Column Name	Data Type	Key
-------------	-----------	-----

id	int	PK, FK
order_id	int	
amount	decimal(10,2)	
payment_date	datetime	
payment_method	varchar(45)	

product_stocks

Stores product inventory levels by RDC.

Table 29 product stocks

Column Name	Data Type	Key
id	int	PK
product_id	int	FK
rdc_id	int	FK
available_quantity	bigint	
last_updated	datetime	

stock_transfers

Stores stock transfer requests between RDCs.

Table 30 stock_transfe

Column Name	Data Type	Key
transfer_id	int	PK
transfer_number	varchar(50)	UK
source_rdc_id	int	FK
destination_rdc_id	int	FK
requested_by	int	
requested_by_role	enum	
requested_date	timestamp	
request_reason	text	
is_urgent	tinyint(1)	
approval_status	enum	
approved_by	int	
approval_date	timestamp	
approval_remarks	text	
current_status_updated	timestamp	

completed_date	timestamp	
receiver_name	varchar(255)	
delivery_notes	text	

stock_transfer_items

Stores line items for stock transfer requests.

Table 31 stock_transfer_items

Column Name	Data Type	Key
item_id	int	PK
transfer_id	int	FK
product_id	int	FK
requested_quantity	int	

transfer_status_logs

Stores audit log for stock transfer status changes.

Table 32 transfer status logs

Column Name	Data Type	Key
log_id	int	PK

transfer_id	int	FK
previous_status	varchar(50)	
new_status	varchar(50)	
changed_by	int	
change_by_role	varchar(50)	
change_by_name	varchar(255)	
changed_date	timestamp	

stock_movement_logs

Stores detailed audit log for all stock movements.

Table 33 stock movement logs

Column Name	Data Type	Key
movement_id	int	PK
rdc_id	int	FK
product_id	int	FK
movement_type	enum	
quantity	int	
previous_quantity	int	

new_quantity	int	
created_by	int	
created_by_role	varchar(50)	
created_by_name	varchar(255)	
created_at	timestamp	
note	text	

shopping_carts

Temporary stores selected products for customers and sales refs to be used for placing the orders.

Table 34 shopping carts

Column Name	Data Type	Key
id	int	PK
user_id	int	FK
product_id	int	FK
quantity	int	

The database schema diagram of ISDN database follows a modular and normalized structure that supports for the product and inventory management, stock transfers, user and role administration, customer handling, order processing and auditing. The database structure has been implemented adhering to relational database principles such as entity separation, referential integrity, and auditability.

At the inventory management, the main related tables are product_categories and products tables. The product_categories table provides the categorization of products and it includes attributes such as category name, description, and timestamps for records creation and update purposes. Each product record in the products table is linked to a category in the category table through a foreign key relationship. Which means each of the product belong to a particular product category. The products table stores important product attributes such as product_code, product_name, unit_price, minimum_stock_level, active status and timestamps for updates. This product and category separation ensures data normalization and enables efficient category-based filtering and reporting.

Promotional products are stored in promotions table where any product can be given a discount percentage for a certain number of items purchase within a specific period of time. Promotional and non-promotional items are temporary stored in shopping_carts table before placing the order, where user and the selected products stored along with the quantity.

This database schema has followed a structured approach to managing product stocks across multiple locations using the rdcs and product_stocks tables, rdc table represents Regional Distribution Centers which operates in 5 main regions of the island. Each RDC maintains attributes such as rdc code, name, province, address, and contact number. Stock availability at each RDS is tracked through the product_stocks table, which establishes a many to one relationship between products and RDCs. This design allows the system to maintain accurate per RDC inventory levels.

Stock movement and logistics are handled using a set of interrelated tables centered around the stock_transfers table. The stock_transfers table records requests to move stock between RDC locations and captures information such as source RDC, destination RDC, transfer status, requested product quantities, approval workflow, and timestamps. Detailed product level quantities related to each transfer are stored in the stock_transfer_items table. The transfer_status_logs table provides an audit history of status changes, including who performed

the change, when it was happened, and what are the previous and new statuses etc. This logging approach enhances the traceability and accountability within the stock movement process.

User management is handled through the users table, which stores user information related to authentication and authorization process such as username, email, password, role, and associated RDC. Role based access control is supported by an Enum based user type column in the user table, which allow system users to separately identify by their roles and assign privileges accordingly. The schema further divides into 8 different tables which classify system users through specialized user roles such as retail_customers, rdc_clerks, rdc_drivers, rdc_logistics_officers, rdc_managers, rdc_sales_refs, system_admins, head_office_managers. Each of these tables extends the users table via foreign key relationships and stores set of role specific attributes such as personal contact information. This approach supports both extensibility and clear separation of concerns of the database.

All system customers are separately handled by retail_customers table, mainly for the order processing purposes. Retail customers around the country who registered with this distribution system and places orders, their related data is managed using this table, which links customers to user accounts in the user table where applicable. This table stores customer names, addresses, and contact information, enabling relationship between sales operations and the user management subsystem.

The order processing module is mainly handled by the combination of orders, order_items, payments, and order_deliveries tables. The orders table stores important order information including order number, customer, total amount, order status, estimated delivery date, and audit timestamps. Each placed order can contain one or multiple products, which are further detailed in the order_items table along with purchase quantities, unit prices, and discounts given. Orders and the order_items table maintain a one-to-many relationship. Order Payment details such as payment amount, payment method and payment date are stored in the payments table, while delivery specific information such as delivery dates and assigned drivers are maintained in the order_deliveries table. This separation ensures transactional clarity and supports future expansion of payment or delivery mechanisms.

Overall, this ISDN database schema has been designed as a well-structured relational design suitable for a MySQL based database environment. It enforces data integrity through primary and foreign keys. It supports user authentication and authorization, operational workflows such as stock transfers and order management, and incorporates very important audit logging for

critical processes. The modular organization of entities allows the system to scale while maintaining its consistency, reliability, and maintainability in an enterprise level inventory and distribution management system context

2.9 High-Level Architecture Diagram

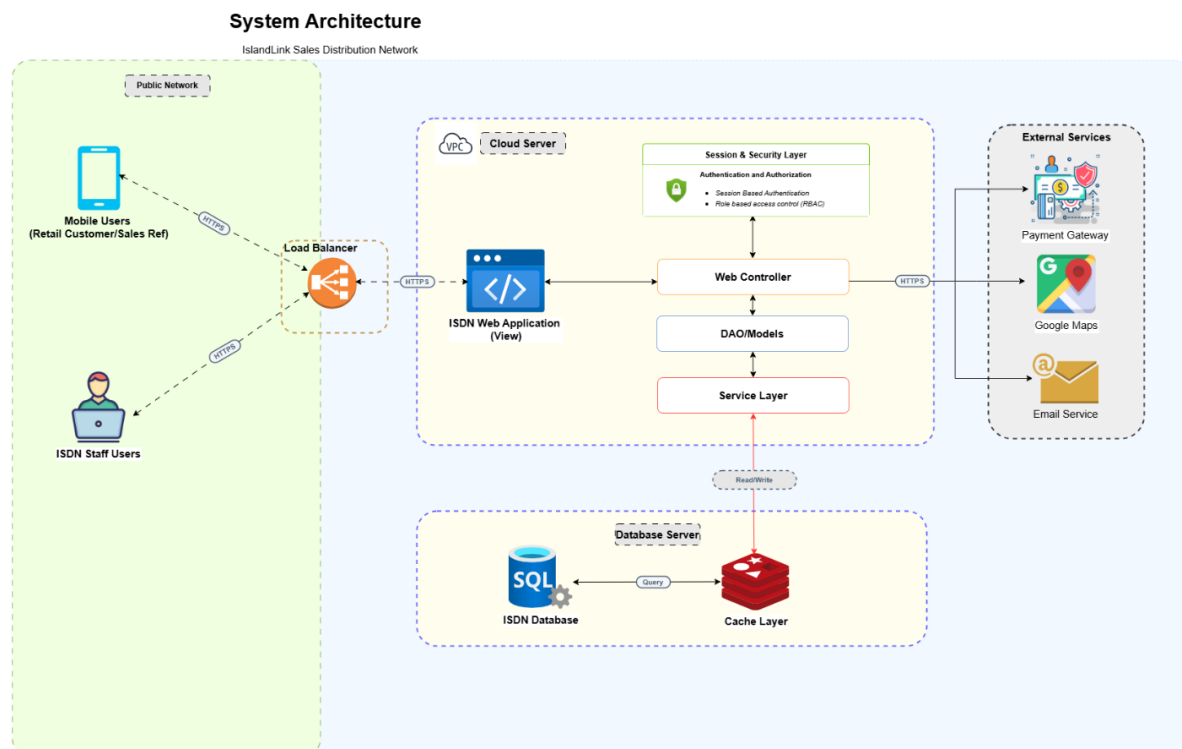


Figure 6 High-Level Architecture Diagram

2.9.1 Analysis of System Architecture

Overview

The IslandLink Sales Distribution Network (ISDN) presents as a comprehensive web-based system architecture designed to facilitate their island wide sales distribution operations. The

system follows a combination of traditional three-tier architecture pattern along with the Model-View-Controller (MVC) design pattern to create a scalable, maintainable, and secure enterprise application. This hybrid approach provides a clear separation of concerns while maintaining its modularity across presentation, business logic, and data management layers.

Architectural Framework

The proposed system architecture adopts a layered approach that separate the system functionality into distinct tiers, each provides a specific responsibility of application processing. The three-tier architecture divides the system into 3 main layers which are presentation, application logic, and data management layers. Along with that, the MVC design pattern introduces separation within the application tier by separating the View (presentation logic), Controller (request and response handling), and Model (business logic and data representation). By mixing these 2 architectural strategies ensure the application scalability and improves its maintainability while promoting code reusability and testability.

User Access Layer

This system is primarily accessed by two types of user categories, they are mobile users (retail customers and sales representatives) and ISDN staff users. Both user groups interact with the system over a secure HTTPS connection through the public network, that ensuring the encrypted communication channels. The mobile users represent the customer facing component of the system where a responsive web UI is dedicated for customers and sales representatives to interact with the order placement UIs through the mobile application. ISDN staff users access administrative and operational workflow functionalities through secure and authenticated sessions. All incoming requests are handled and forwarded through a load balancer, which evenly distributes incoming network traffic across multiple instances of the application to ensure high availability, optimal resource utilization, and fault tolerance of the overall system.

Presentation Tier- View Component

The presentation tier begins through the ISDN Web Application (View), which serves as the primary interface for user interaction. This component is hosted on cloud infrastructure (VPC) and renders the user interface and manages client-side interactions. The View layer implements the presentation layer of MVC pattern which translate user actions into controller requests received from the application tier. The view component communicates with the Web Controller through HTTPS protocol, maintaining secure data transmission channel in the request-response flow. By deploying the application in a cloud infrastructure provides elasticity, allowing the system to scale its resources dynamically based on demand changes.

Application Tier- Controller and Business Logic

The application tier made up with multiple components that handles business operations and coordinate data flow between the presentation and data tiers. The application core contains Web Controller, which functions as the primary request handler and router in the MVC architecture. Its main responsibility is receiving requests from the View layer, processes them according to the business rules, and coordinates with relevant service components to complete the requests and generate the response to send back to the view.

The Session and Security Layer provide a robust user authentication and authorization mechanisms. This critical component provides session-based authentication protocols and role-based access control (RBAC), ensuring that users can only access system resources and perform operations limited with their assigned privileges. The security layer acts as a gatekeeper, that validating user login credentials and maintaining session state throughout user interact with the system.

The DAO/Models component represents the Model component of the MVC architecture, containing data access objects and business entities. This responsible for handling the database operations, by providing a dedicated interface for data manipulation while separating business logic from underlying data storage mechanisms. The DAO pattern facilitates application maintainability by separating its data access logic and enabling to modify of data persistence strategies whenever required without affecting the business logic.

The Service Layer implements core business logic and performs complex operations that may affect across multiple data entities or require communication with external services. This layer processes business rules, performs data validations, and process business transactions. By

centralizing business logic in the service layer, the architecture promotes code reusability and simplifies testing of business operations independently from presentation and data access processes.

Data Tier- Persistence and Caching

The data tier constitutes two primary components that manage data persistence and performance optimization. MySQL server has been used as the database of the ISDN system, that serves as the primary data repository for the system. This relational database management system stores all the persistent application data, such as user information, product catalogs, transaction records, and operational metadata. The database server supports for standard query operations, enabling the DAO layer to perform CRUD (create, read, update, and delete) operations on structured data in the database tables.

A Cache Layer has been used to enhance the system performance and database load. The cache stores frequently accessed application data in its memory, that significantly helping to reduce query response times and database server load. The architecture indicates a read/write relationship between the service layer and the cache. Query operations can be served directly from the cache when data is available. If any database query unable to find result in the cache, it can fetch the data from the database and store it in the cache to be used in future requests.

External Services Integration

The architecture integrates with different kind of external services to extend system functionalities. The Payment Gateway provides facilities to perform payment processing functionality, by enabling the system to handle financial transactions for sales operations. This integration follows industry-standard payment protocols adhering to the PCI DSS compliances and secure handling of sensitive financial information.

Google Maps integration enables the system with geolocation capabilities, mainly supporting features such as location-based services, delivery route optimization, and driver navigation functionalities. The Email Service integration enables the system to send transactional emails, notifications, and other business emails to customers and other users. These external service

integrations communicate by the respective Web Controller mainly via HTTPS by maintaining a secure data exchange while utilizing specific third-party service features.

Security Architecture

Security mechanisms have been followed in each component of this architecture. All external communication channels utilize HTTPS protocol, ensuring encryption of exchange data. The Session and Security Layer enable multi-factor authentication approaches, combining session-based authentication with role-based access control. This layered security approach ensures that authentication and authorization mechanisms are centralized and consistently utilized across all system operations.

Scalability and Performance Considerations

The ISDN system architecture demonstrates several architecture level design decisions oriented toward to enhance system scalability and performance optimization. The load balancer facilitates the horizontal scaling of the web application tier, that allowing to create additional server instances as demand increases. Separating the cache layer from the main database promotes a performance optimization approach that reduces latency for frequently accessed data. The cloud-based deployment of the web application offers infrastructure elasticity, allowing automatic scaling based on traffic patterns. The modular architecture, with its distinct separation between presentation, application, and data tiers, facilitates the opportunity to scale independent components based on specific performance challenges.

Conclusion

The IslandLink Sales Distribution Network system architecture represents a well-structured enterprise application design that handles scalability, security, and maintainability requirements of the system. By combining three-tier and MVC architectural patterns, the system achieves both horizontal and vertical separation of concerns, creating a robust foundation for a complex sales distribution platform. Integrating external services by implementing comprehensive security measures, and incorporation of performance optimization strategies enables architectural planning supporting towards a production-grade business application.

2.10. Gantt chart

Reference--

<https://www.onlinegantt.com/#!/public/0e72443c-6fbe-421f-969b-f11730d5b910>

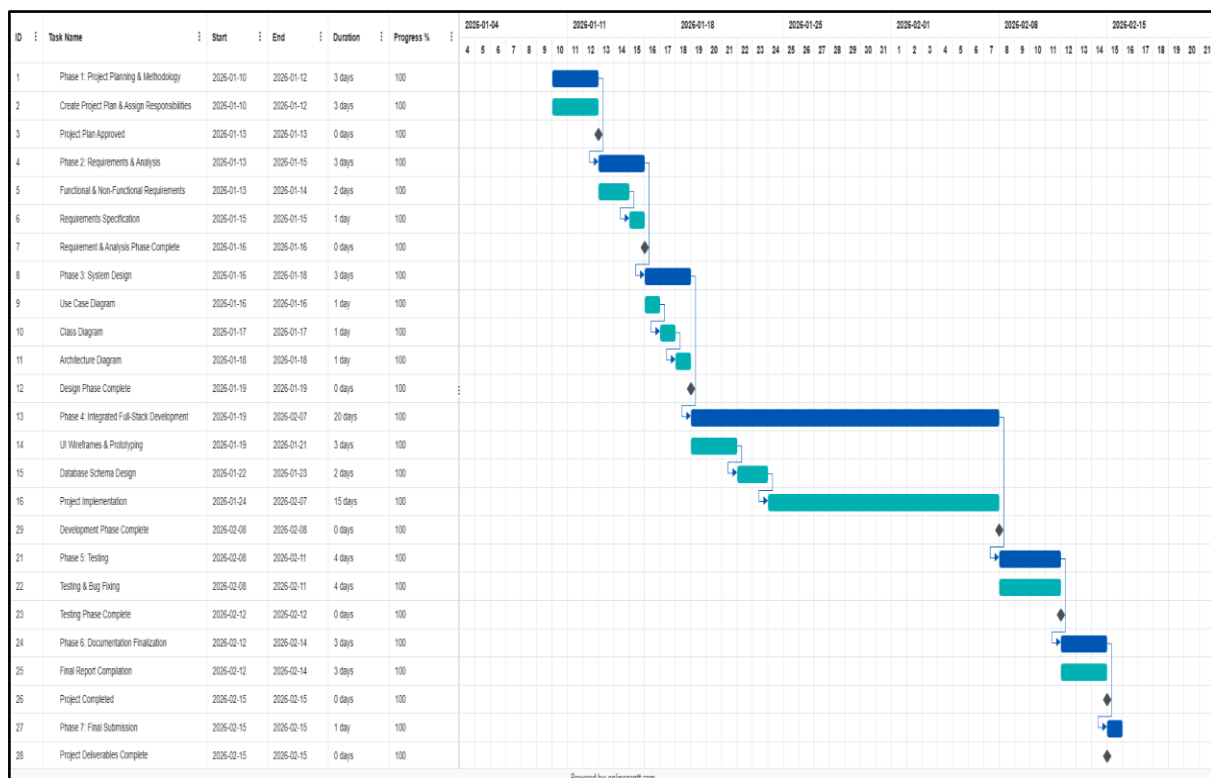


Figure 7 Gantt Diagram

Reference- https://drive.google.com/file/d/1kzoAf_zm9_C_Fi5lFp8-WWuG1O_K5CmN/view?usp=drive_link

2.10.1 Project phases and task breakdown

The Gantt chart divides the project into seven distinct phases. The first phase is Project Planning and Methodology, scheduled from 10 January 2026 to 12 January 2026. In this phase, the project plan is developed and responsibilities are assigned to each group member, with approval completed by 13 January 2026. The second phase is Requirements and Analysis, which runs from 13 January to 15 January 2026. During this phase, functional and non-functional requirements are identified, and the Software Requirements Specification document is prepared. The phase is marked complete by 16 January 2026. The third phase is System Design, from 16 January to 18 January 2026, which includes the creation of use case diagrams, class diagrams, and architecture diagrams. The design phase is completed by 19 January 2026. The fourth phase is Integrated Full-Stack Development, scheduled from 19 January to 7 February 2026. This phase begins with UI wireframes and prototyping (19–21 January), followed by database schema design (22–23 January), and project implementation (24 January–7 February). Development is completed by 8 February 2026. The fifth phase is Testing, from 8 February to 11 February 2026, focusing on testing and bug fixing, with completion on 12 February 2026. The sixth phase is Documentation Finalization, from 12 February to 14 February 2026, where the final report is completed. The seventh and final phase is Final Submission, scheduled on 15 February 2026, marking the completion of the project deliverables. A summary of the seven phases is provided below:

- Phase 1: Project Planning – 10 January to 12 January 2026
- Phase 2: Requirements and Analysis – 13 January to 15 January 2026
- Phase 3: System Design – 16 January to 18 January 2026
- Phase 4: Integrated Full-Stack Development – 19 January to 7 February 2026
- Phase 5: Testing – 8 February to 11 February 2026
- Phase 6: Documentation Finalization – 12 February to 14 February 2026
- Phase 7: Final Submission – 15 February 2026

By dividing the project into smaller phases and tasks, the group can manage the work more effectively and reduce the risk of overlooking important activities.

2.10.2 Timeline and Scheduling Approach

The project timeline in the Gantt chart begins on 10 January 2026 and ends on 15 February 2026, which is the final deadline. The horizontal axis at the top of the chart presents the timeline in weekly intervals. This allows the group to clearly identify the start and end dates for each task. To complete the project within a short duration, overlapping of tasks has been applied in the schedule. The main characteristics of the scheduling approach are as follows:

1. Frontend and backend development are conducted in parallel, enabling both teams to work simultaneously.
2. Unit testing is carried out during the implementation phase, while final testing activities begin only after both the frontend and backend development are completed.
3. Each phase follows the previous phase in a logical order.

This scheduling approach allows the project to be completed within approximately five weeks while maintaining quality standards throughout each phase.

2.10.3 Task Sequence and Phase Relationships

The Gantt chart clearly shows the sequence of tasks from the beginning to the end of the project. Each phase is connected to the next phase in a logical order, and the chart helps the group understand which tasks must come first and which tasks can follow. The project begins with Phase 1: Project Planning and Methodology. This phase must be completed first, because without a project plan and clear responsibilities, the group cannot move forward. Once the project plan is approved, Phase 2: Requirements and Analysis begins immediately on the same day. This shows a direct handover from planning to requirements work.

The completion of the Requirements Specification allows Phase 3: System Design to start on 16 January. The design phase includes three important diagrams, arranged in a logical sequence:

6. Use Case Diagram on 16 January – helps identify what the system should do
7. Class Diagram on 17 January – shows the structure of the system

8. Architecture Diagram on 18 January – shows how all parts fit together

After the design phase is completed, Phase 4: Integrated Full-Stack Development begins on 19 January 2026. The phase is broken into three main tasks:

- 1 UI Wireframes and Prototyping from 19 January to 21 January
- 2 Database Schema Design from 22 January to 23 January
- 3 Project Implementation from 24 January to 7 February

This shows efficient scheduling with no gaps between related tasks, allowing design, database planning, and development to flow continuously.

Phase 5: Final testing runs from 8 February to 11 February, immediately after Development completion. Testing and Bug Fixing is a single continuous task during this period. After testing is complete on 11 February, the Testing Phase Complete milestone is placed on 11 February, followed immediately by Final Report Compilation from 12 February to 14 February.

The final milestones are:

1. Project Completed – 15 February 2026
2. Final Submission – 15 February 2026

The structured progression of phases and tasks presented in the Gantt chart ensures the project is completed efficiently and within the scheduled timeframe.

2.10.4 Progress Tracking and Color Coding

To monitor project progress effectively, colour coding was used throughout the project timeline. In the Gantt chart, green and dark blue rectangles indicated tasks that had been fully completed, while grey rectangles represented tasks that were in progress and light blue rectangles showed tasks that had started but were not yet finished. This colour coding system allowed the group to quickly identify which areas required more attention and effort. The meaning of each colour is summarised below:

- Green = Task is 100 percent complete
- Grey = Task is currently in progress

- Light blue = Task has started but is not yet complete

As of the current project status, all tasks have been successfully completed. This includes Project Planning, Requirements Analysis, System Design, Frontend Development, Backend Development, Testing, Documentation, and Final Submission. Therefore, the entire Gantt chart appears in green, indicating that every phase and task has reached 100 percent completion. During the project, the group regularly reviewed the Gantt chart in meetings and updated the colours based on work completed. This visual approach helped the group stay aware of deadlines, distribute responsibilities, and maintain motivation throughout the project lifecycle. The final chart confirms that the project was delivered on time, with the Project Completed milestone achieved on 15 February 2026 and all deliverables submitted by 15 February 2026.

2.10.5. Milestones and Dependencies

Several milestones have been placed along the project timeline to mark important achievements. A milestone differs from a regular task in that it has zero duration. It represents a key event or checkpoint in the project. These milestones serve as indicators of progress and provide a sense of accomplishment for the group. The key milestones identified in the Gantt chart are:

1. Project Plan Approved – 13 January 2026
2. Requirements Phase Complete – 16 January 2026
3. Design Phase Complete – 19 January 2026
4. Development Phase Complete – 8 February 2026
5. Test Phase Complete – 12 February 2026
6. Project Completed – 15 February 2026
7. Final Submission – 15 February 2026

In addition to milestones, the Gantt chart also reflects dependencies between tasks. Certain tasks cannot begin until previous tasks have been completed. For example, frontend development cannot start until the design phase is finished. Similarly, final testing cannot begin until both frontend and backend development are complete. These dependencies ensure that the project follows a logical sequence and help prevent confusion regarding task order. By

following the milestones and dependencies outlined in the Gantt chart, the group can maintain organization and complete the project within the scheduled time.

2.11 User Interface Design

1. For Customer and sales ref

Login

ISDN
DISTRIBUTION NETWORK

Login Register

Welcome Back!
Login to your ISDN account

Enter your email

Enter your password

☐ Remember me [Forgot Password?](#)

Login Now

OR CONTINUE WITH

[Continue with Google](#)

Don't have an account? [Register here](#)

SECURE **FAST** **SUPPORT**

ISDN
IslandLink Sales Distribution Network - Your trusted partner for modern wholesale and retail distribution.

Quick Links

- > Products
- > About Us
- > Contact
- > FAQ

RDC Locations

- > Central Region - Kandy
- > Northern Region - Jaffna
- > Southern Region - Galle
- > Eastern Region - Batticaloa
- > Western Region - Colombo

Contact Us

+94 11 234 5678

info@isdn.lk

f t i

© 2026 ISDN - IslandLink Sales Distribution Network. All rights reserved. [Privacy Policy](#) [Terms](#)

Figure 8 Login

This is the login page of ISDN system. This page is common to all the types of users interact with the system. User can enter their registered credentials and click on login button to complete the login. If any user doesn't have an account yet they can do below actions.

1. If retail customer, he can click on Register button and register as a new Retail customer.
2. If user is a RDC sales ref or staff member, he can request new account creation from the System Administrator.

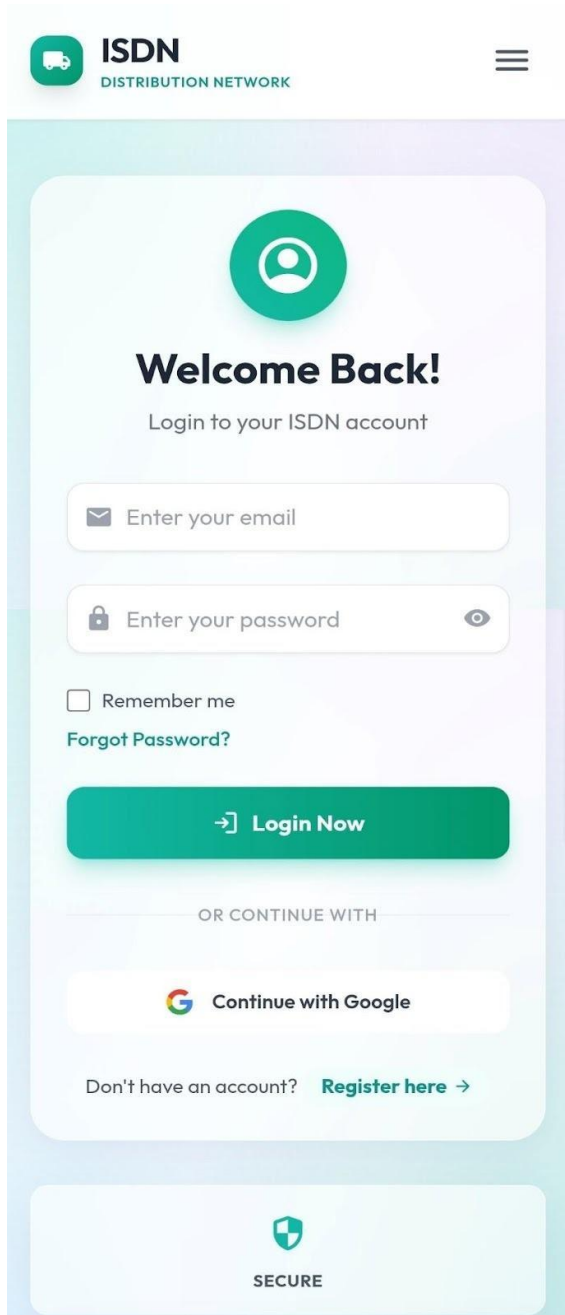
The image shows a mobile application interface for the ISDN Distribution Network. At the top, there is a header with the ISDN logo (a green circle with a white person icon) and the text "ISDN DISTRIBUTION NETWORK". To the right of the header is a hamburger menu icon. The main content area has a light blue background. It starts with a green circular icon containing a white person silhouette. Below this is the text "Welcome Back!" in bold, followed by "Login to your ISDN account". There are two input fields: "Enter your email" with an envelope icon and "Enter your password" with a lock icon and a toggle eye icon. Below the password field is a checkbox labeled "Remember me" and a link "Forgot Password?". A large green button with a white arrow icon and the text "Login Now" is positioned below these fields. Underneath the button is the text "OR CONTINUE WITH". Below this is a button with the Google logo and the text "Continue with Google". At the bottom of the main content area, there is a link "Don't have an account? Register here" with a right-pointing arrow. At the very bottom of the screen, there is a light blue bar with a green shield icon and the word "SECURE".

Figure 9 Login

Register

ISDN
DISTRIBUTION NETWORK

Login Register

Create Account

Register as a Customer on ISDN

Customer Registration Only
Other user roles (RDC Staff, Managers, etc.) are created by the System Administrator.

Full Name

Email Address

Create Password (min 6 chars)

Confirm Password

☐ I agree to the [Terms & Conditions](#) and [Privacy Policy](#)

Create Customer Account

OR REGISTER WITH

Continue with Google

Already have an account? [Login here](#)

FAST DELIVERY BEST PRICES VERIFIED

ISDN
IslandLink Sales Distribution Network - Your trusted partner for modern wholesale and retail distribution.

Quick Links

- > Products
- > About Us
- > Contact
- > FAQ

RDC Locations

- Central Region - Kandy
- Northern Region - Jaffna
- Southern Region - Galle
- Eastern Region - Batticaloa
- Western Region - Colombo

Contact Us

+94 11 234 5678

info@isdn.lk

f t i

© 2026 ISDN - IslandLink Sales Distribution Network. All rights reserved. Privacy Policy Terms

Figure 10 Register

This is the Customer Registration page of ISDN system. Only the new Retail customers can create their account here. To register successfully as a new customer, Customer need to provide his Full name, email address, password and password confirmation information to create an account. The system validates the entered data to ensure accuracy, completeness and agreement with security requirements during registration.

**ISDN**
DISTRIBUTION NETWORK





Create Account

Register as a Customer on ISDN

**Customer Registration Only**
Other user roles (RDC Staff, Managers, etc.) are created by the System Administrator.

 Full Name

 Email Address

 Create Password (min 6 chars) 

 Confirm Password 

☐ I agree to the [Terms & Conditions](#) and [Privacy Policy](#)

 **Create Customer Account**

OR REGISTER WITH

 Continue with Google

Already have an account? [Login here](#) → 

Figure 11 Customer Registration screen.

Mobile view of new Customer Registration screen.

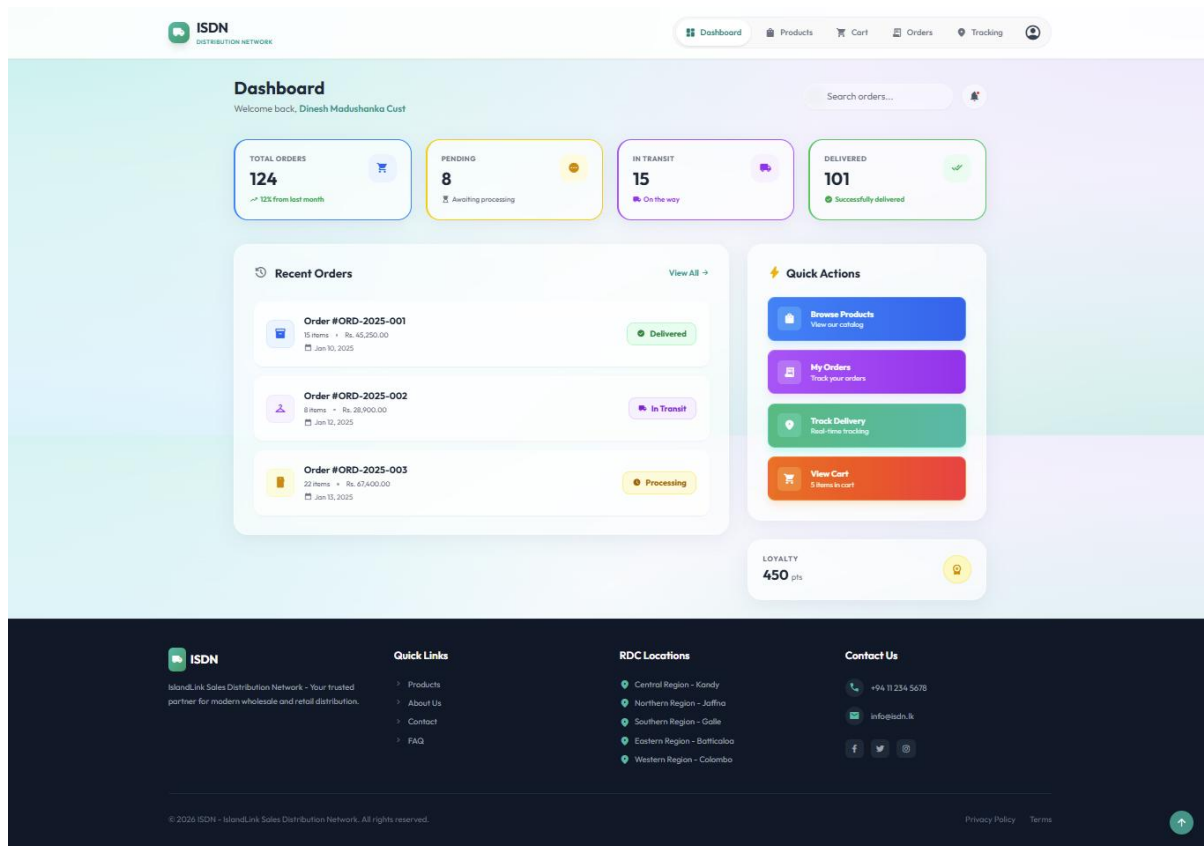


Figure 12 customer Dashboard

This dashboard serves as the central overview page, providing structured access to key functionalities and personalized information. The interface includes a top navigation menu for primary system navigation, a summarized overview related to customer's order activity displayed in tiled format, and a list of recent orders for quick reference. Additionally, the dashboard offers quick access links that enable users to quickly navigate to main features such

as product browsing, order history, shopping cart management, and delivery tracking.

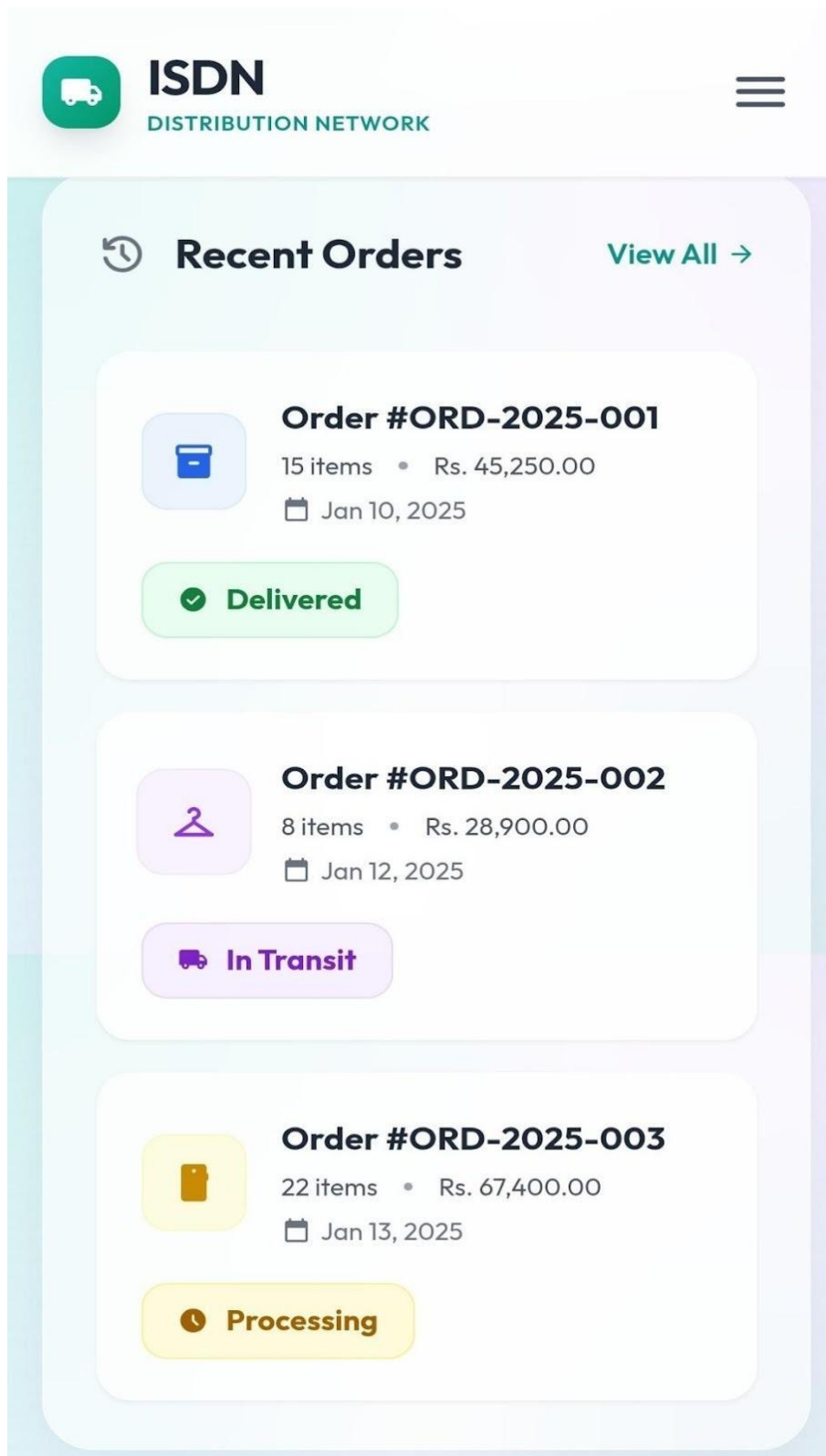


Figure 13 Mobile view of customer dashboard

Mobile view of customer dashboard

Browse products

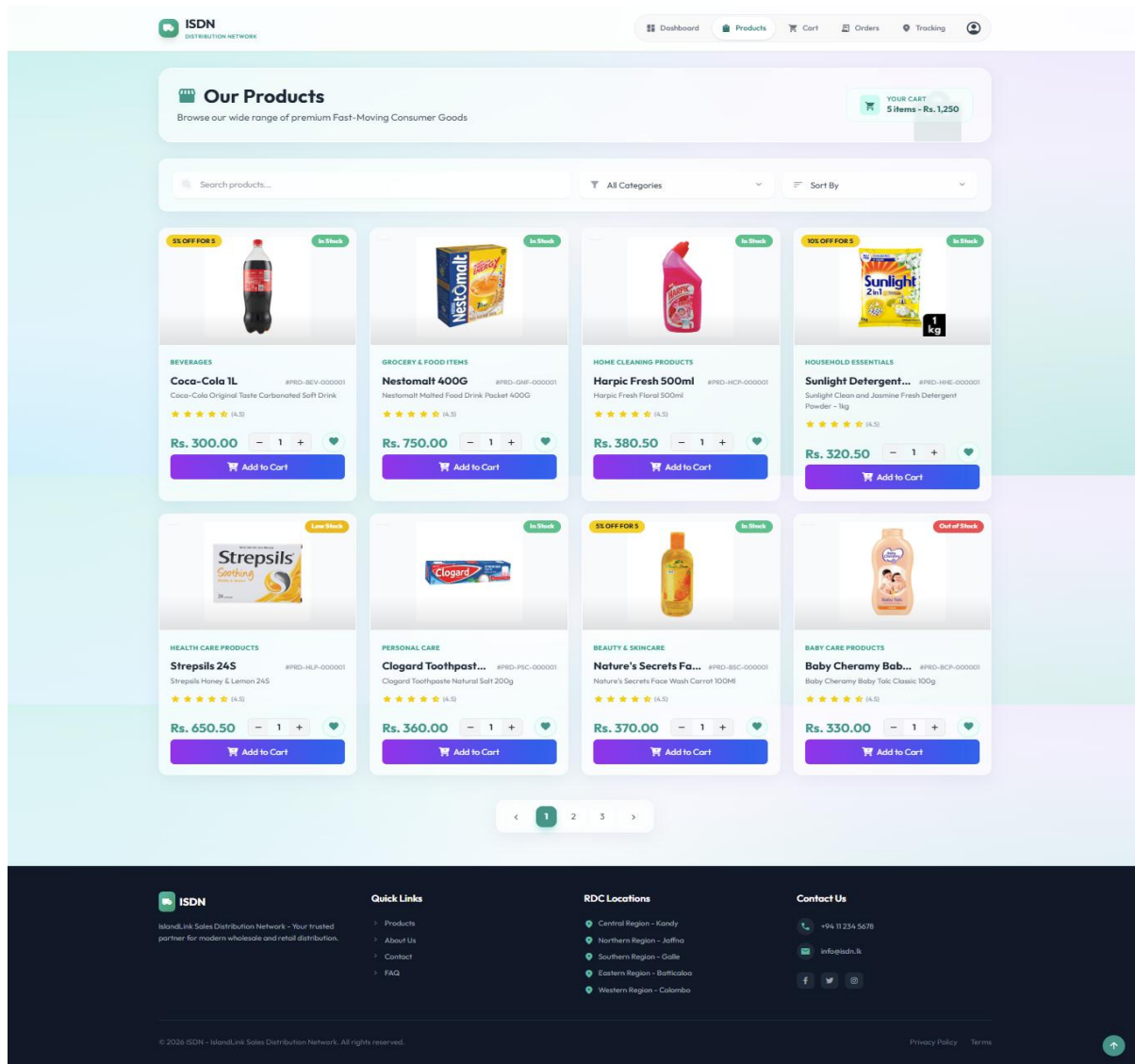


Figure 14 Browse products interface

When the customer selects the Products menu, the system navigates to the Product Listing interface. This page displays the complete catalog of ISDN products along with relevant details such as product name, category, and pricing information. The interface provides dynamic search functionality, allowing users to locate products by name, as well as filtering options based on category and sorting actions based on price or other attributes. Furthermore, customers are able to adjust the desired quantity of a selected item and add it to the shopping

cart for future checkout. This interface is designed to support efficient product discovery and streamlined purchasing interactions.

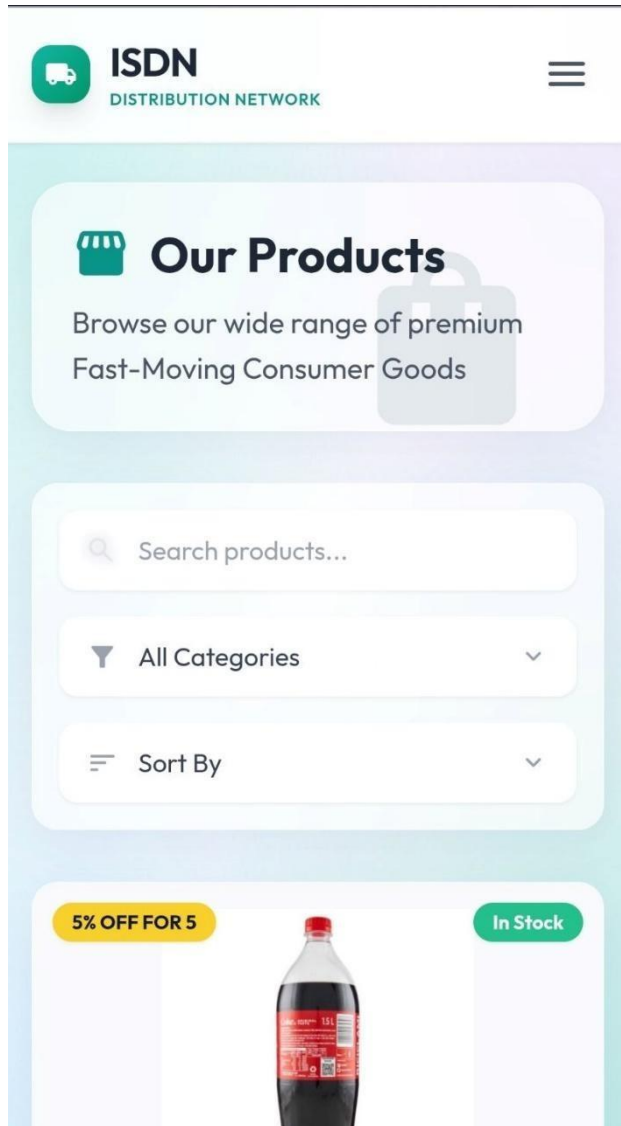


Figure 15 Browse products mobile view

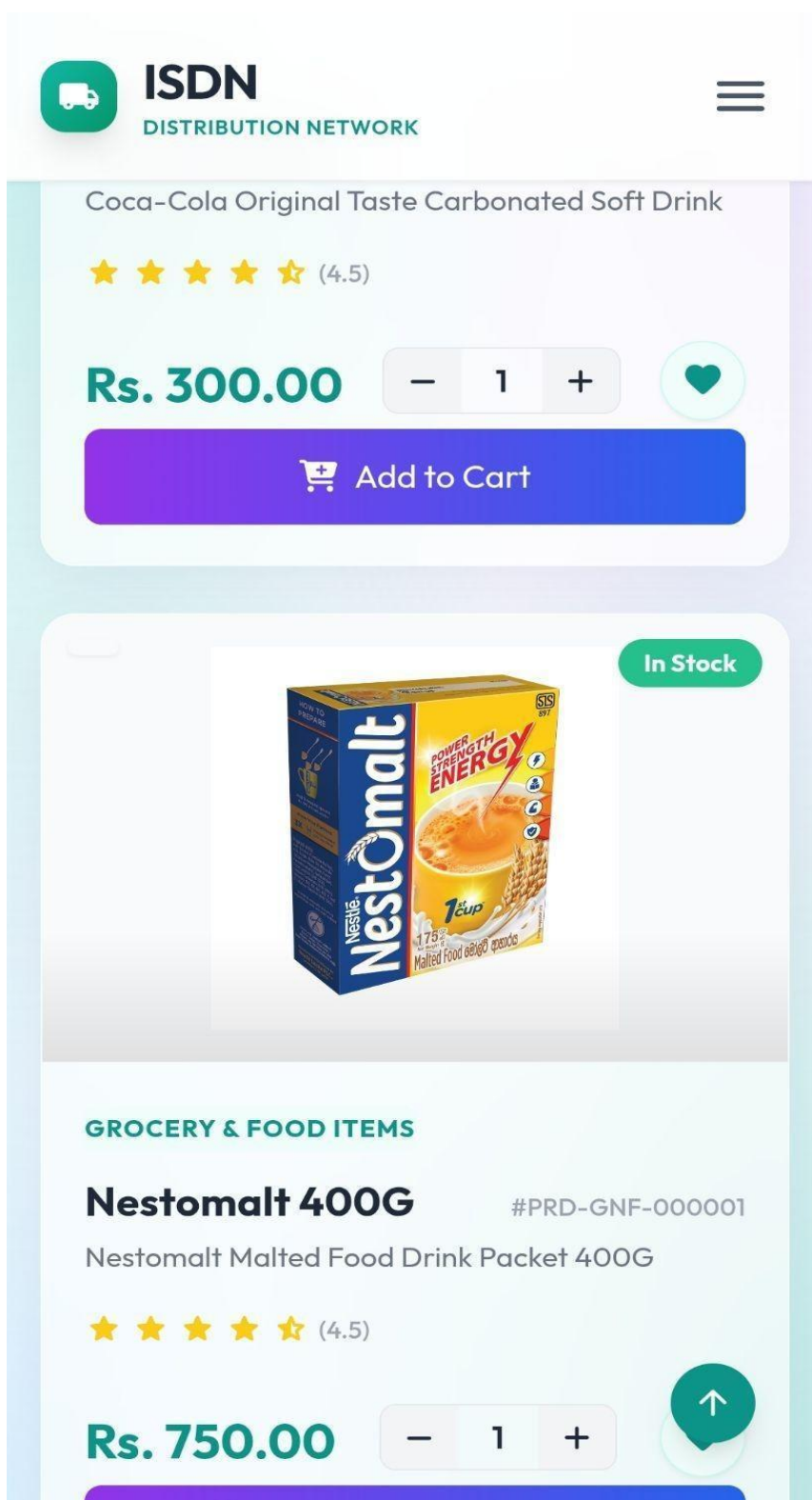


Figure 16 Mobile view of Products page

Mobile view of Products page

Shopping cart

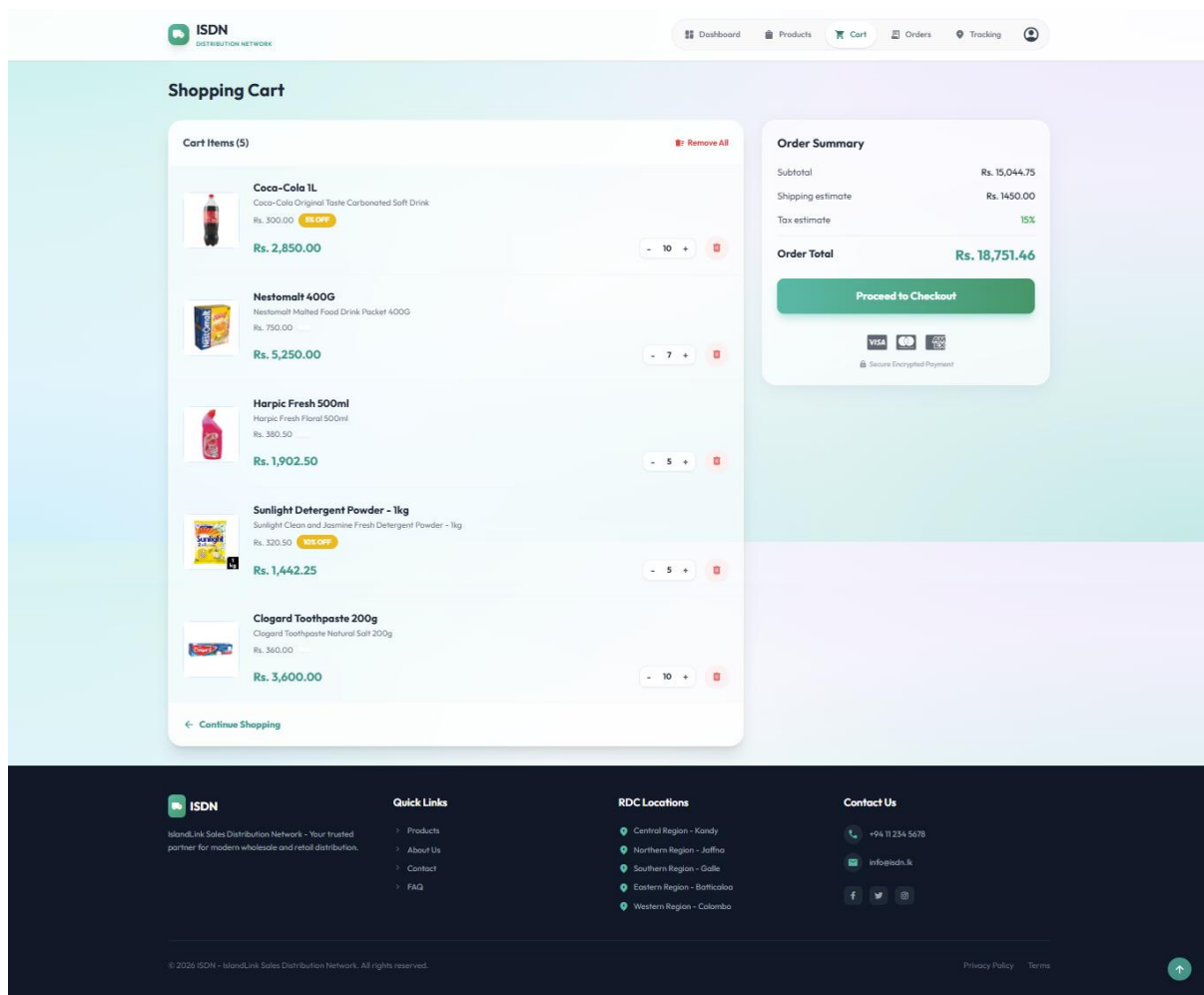


Figure 17 Shopping cart

This interface represents the Customer Shopping Cart page, which is accessible through the Cart option in the top navigation menu. The shopping cart screen enables customers to manage selected items prior to purchase by updating product quantities, adding new items, or removing existing items from the cart. The system dynamically calculates the order summary in real time, providing a detailed cost breakdown that includes item-level totals and aggregate pricing information. Upon completing cart management activities, the customer may proceed to the order checkout process by selecting the “Proceed to Checkout” button, which redirects the user to the Order Checkout page for final order confirmation and payment processing.



ISDN

DISTRIBUTION NETWORK



Shopping Cart

Cart Items (5)

 **Remove All**



Coca-Cola 1L

Coca-Cola Original Taste Carbon Soft Drink

Rs. 300.00

5% OFF

**Rs.
2,850.00**

- 10 +



Nestomalt 400G

Nestomalt Malted Food Drink Pack 400G

Rs. 750.00

**Rs.
5,250.00**

- 7 +



Harpic Fresh 500ml

Harpic Fresh Floral 500ml

Figure 18 Shopping cart mobile view

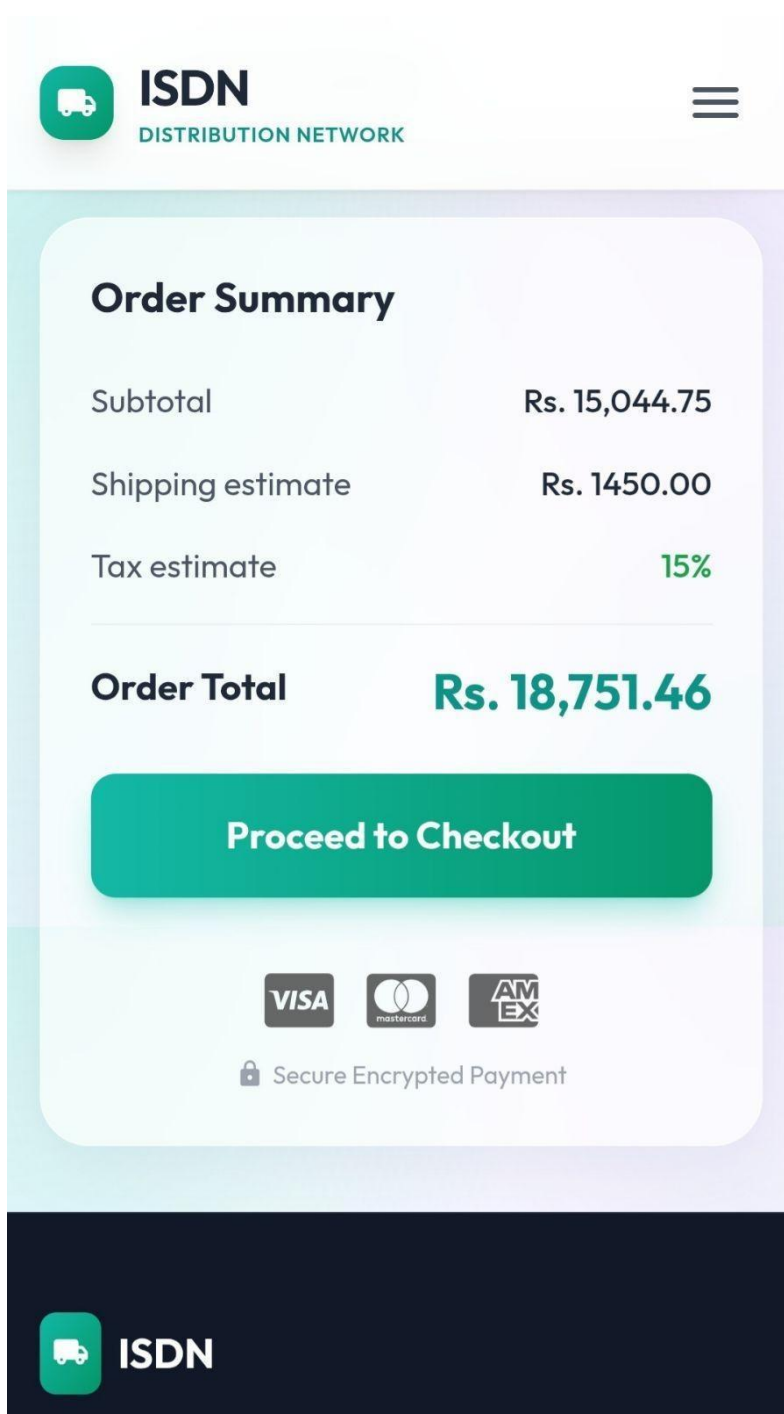
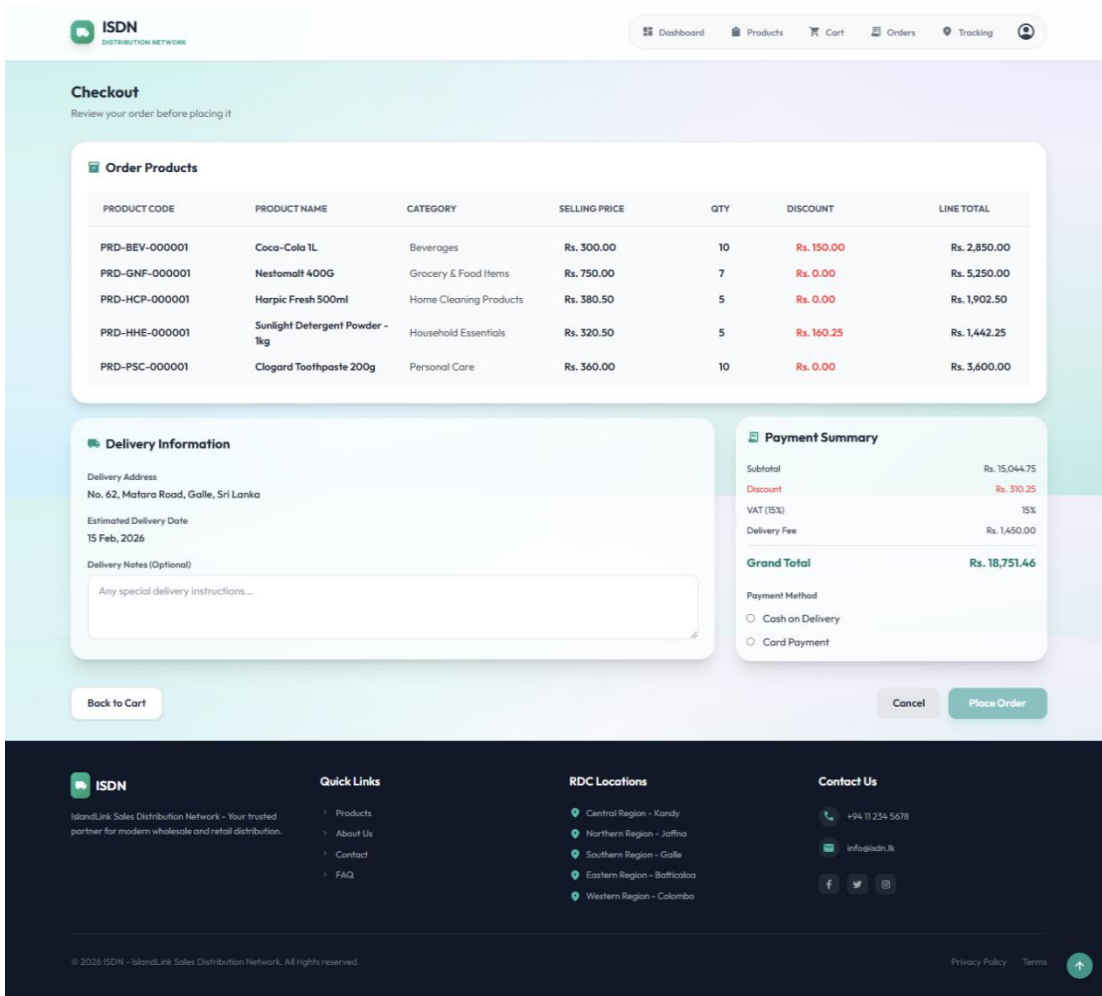


Figure 19 Shopping cart mobile view

Shopping cart mobile view

Checkout



The screenshot shows the ISDN Checkout page. At the top, there's a navigation bar with links to Dashboard, Products, Cart, Orders, and Tracking. The main heading is "Checkout" with a subtext "Review your order before placing it".

Order Products

PRODUCT CODE	PRODUCT NAME	CATEGORY	SELLING PRICE	QTY	DISCOUNT	LINE TOTAL
PRD-BEV-000001	Coca-Cola 1L	Beverages	Rs. 300.00	10	Rs. 150.00	Rs. 2,850.00
PRD-GNF-000001	Nestomalt 400G	Grocery & Food Items	Rs. 750.00	7	Rs. 0.00	Rs. 5,250.00
PRD-HCP-000001	Harpic Fresh 500ml	Home Cleaning Products	Rs. 380.50	5	Rs. 0.00	Rs. 1,902.50
PRD-HHE-000001	Sunlight Detergent Powder - 1kg	Household Essentials	Rs. 320.50	5	Rs. 160.25	Rs. 1,442.25
PRD-PSC-000001	Clogard Toothpaste 200g	Personal Care	Rs. 360.00	10	Rs. 0.00	Rs. 3,600.00

Delivery Information

Delivery Address
No. 62, Matara Road, Galle, Sri Lanka

Estimated Delivery Date
15 Feb, 2026

Delivery Notes (Optional)
Any special delivery instructions...

Payment Summary

Subtotal Rs. 15,044.75
Discount Rs. 310.25
VAT (15%) 15%
Delivery Fee Rs. 1,450.00
Grand Total Rs. 18,751.46

Payment Method
☐ Cash on Delivery
☐ Card Payment

Buttons: Back to Cart, Cancel, Place Order

Footer:

- ISDN**
IslandLink Sales Distribution Network - Your trusted partner for modern wholesale and retail distribution.
- Quick Links**
 - Products
 - About Us
 - Contact
 - FAQ
- RDC Locations**
 - Central Region - Kandy
 - Northern Region - Jaffna
 - Southern Region - Galle
 - Eastern Region - Batticaloa
 - Western Region - Colombo
- Contact Us**
 - +94 11 234 5678
 - info@isdn.lk

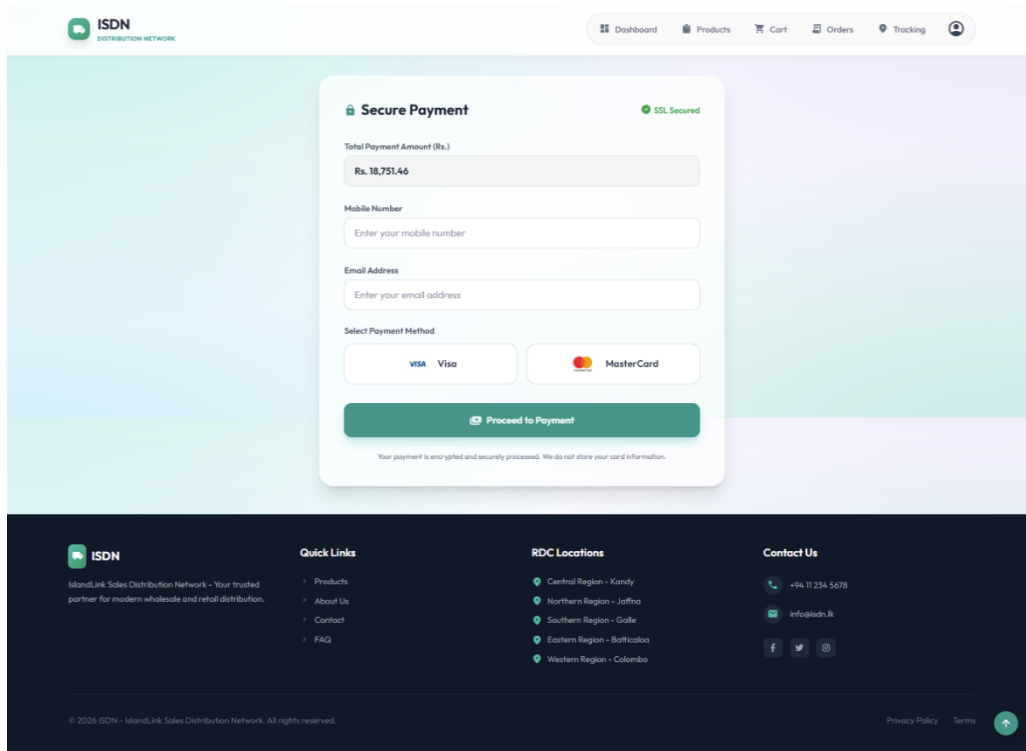
© 2026 ISDN - IslandLink Sales Distribution Network. All rights reserved. Privacy Policy Terms

Figure 20 Checkout

This interface represents the Order Checkout page of the ISDN customers. It provides a full and final summary of the products selected for purchase, including quantities and a detailed breakdown of the total order amount. The page also displays delivery related information, including the specified delivery address and the estimated delivery date. An input field is available for customers to provide additional delivery instructions to facilitate accurate order fulfillment.

Furthermore, the customer is required to select a preferred payment method, such as Cash on Delivery or Card payment. Upon selecting a valid payment option, the “Place Order” button becomes active. When triggered, the system proceeds to the payment processing stage by redirecting the user to the corresponding payment interface for transaction completion.

Payment Page



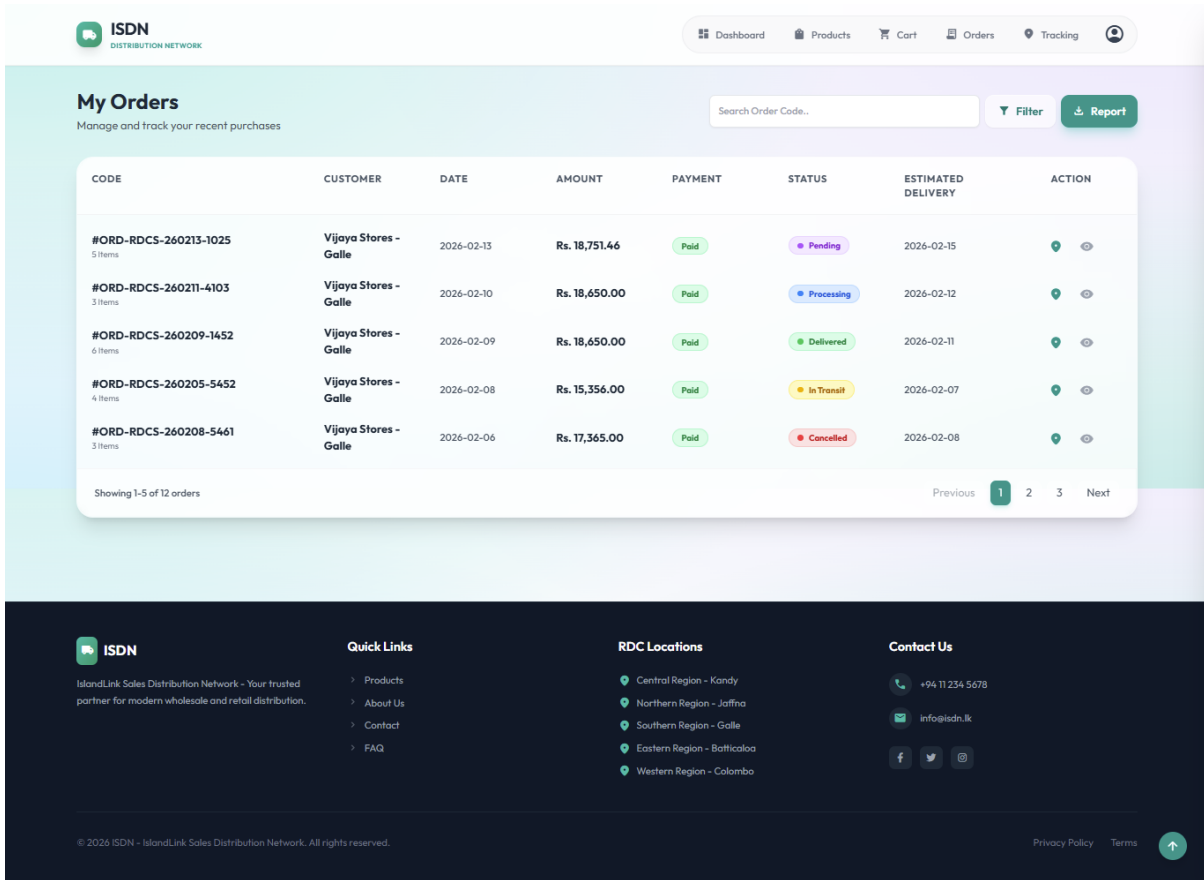
The screenshot shows the 'Secure Payment' interface of the ISDN (IslandLink Sales Distribution Network) website. The page has a light blue and purple background. At the top, there is a navigation bar with links: Dashboard, Products, Cart, Orders, Tracking, and a user profile icon. The main content area is a white card titled 'Secure Payment' with a green 'SSL Secured' indicator. It displays the 'Total Payment Amount (Rs.)' as 'Rs. 18,751.46'. Below this are input fields for 'Mobile Number' (with placeholder 'Enter your mobile number') and 'Email Address' (with placeholder 'Enter your email address'). There is a section for 'Select Payment Method' with two buttons: 'VISA Visa' and 'MasterCard'. A large green button at the bottom of the card says 'Proceed to Payment'. A small disclaimer at the bottom of the card states: 'Your payment is encrypted and securely processed. We do not store your card information.' The footer of the page is dark blue and contains four sections: 'ISDN' (IslandLink Sales Distribution Network - Your trusted partner for modern wholesale and retail distribution), 'Quick Links' (Products, About Us, Contact, FAQ), 'RDC Locations' (Central Region - Kandy, Northern Region - Jaffna, Southern Region - Galle, Eastern Region - Batticaloa, Western Region - Colombo), and 'Contact Us' (+94 11 234 5678, info@isdn.lk, and social media icons). At the very bottom, there is a copyright notice: '© 2020 ISDN - IslandLink Sales Distribution Network. All rights reserved.' and links to 'Privacy Policy' and 'Terms'.

Figure 21 Payment Page

If the customer selects card payment as the preferred payment method on the Order Checkout page, the system redirects the user to the Payment interface upon activation of the “Place Order” function. This page displays the total payable amount and provides input fields for capturing essential order transaction details, including the customer’s mobile number and email address. The customer is also required to select the appropriate card type, such as Visa or MasterCard, before to proceeding.

After entering the required information and selecting the “Proceed to Payment” option, the system securely redirects the user to the external payment gateway for transaction processing. Upon successful completion of the payment, the system generates an order confirmation and automatically sends a confirmation email to the customer’s registered email address, which includes the corresponding order invoice for record-keeping and verification purposes.

My Orders page



ISDN
DISTRIBUTION NETWORK

Dashboard Products Cart Orders Tracking

My Orders

Manage and track your recent purchases

Search Order Code.. Filter Report

CODE	CUSTOMER	DATE	AMOUNT	PAYMENT	STATUS	ESTIMATED DELIVERY	ACTION
#ORD-RDCS-260213-1025 5 Items	Vijaya Stores - Galle	2026-02-13	Rs. 18,751.46	Paid	Pending	2026-02-15	View Details
#ORD-RDCS-260211-4103 3 Items	Vijaya Stores - Galle	2026-02-10	Rs. 18,650.00	Paid	Processing	2026-02-12	View Details
#ORD-RDCS-260209-1452 6 Items	Vijaya Stores - Galle	2026-02-09	Rs. 18,650.00	Paid	Delivered	2026-02-11	View Details
#ORD-RDCS-260205-5452 4 Items	Vijaya Stores - Galle	2026-02-08	Rs. 15,356.00	Paid	In Transit	2026-02-07	View Details
#ORD-RDCS-260208-5461 3 Items	Vijaya Stores - Galle	2026-02-06	Rs. 17,365.00	Paid	Cancelled	2026-02-08	View Details

Showing 1-5 of 12 orders

Previous 1 2 3 Next

ISDN
IslandLink Sales Distribution Network - Your trusted partner for modern wholesale and retail distribution.

Quick Links

- Products
- About Us
- Contact
- FAQ

RDC Locations

- Central Region - Kandy
- Northern Region - Jaffna
- Southern Region - Galle
- Eastern Region - Batticaloa
- Western Region - Colombo

Contact Us

+94 11 234 5678
info@isdn.lk

f t i

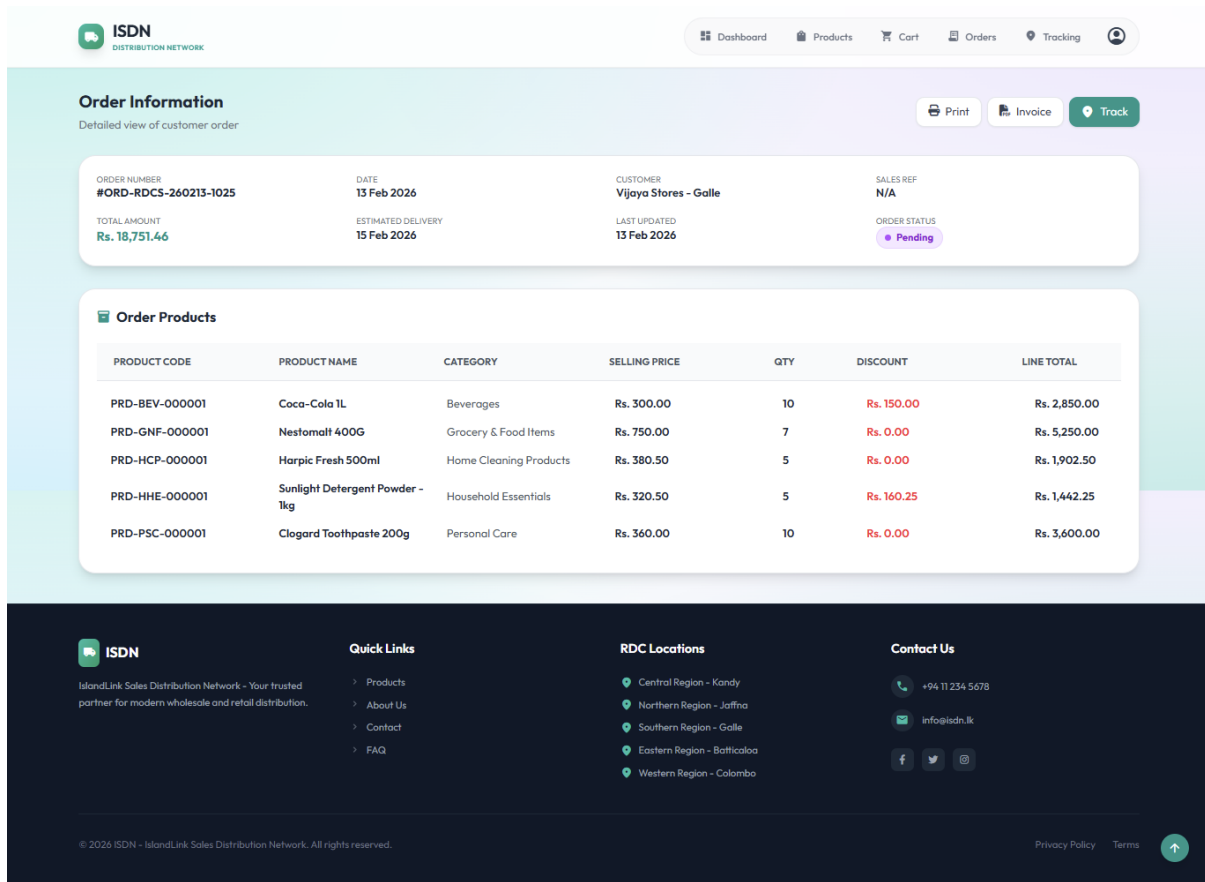
© 2026 ISDN - IslandLink Sales Distribution Network. All rights reserved. Privacy Policy Terms

Figure 22 My Orders page

When the customer selects the Orders option from the top navigation bar, the system redirects the user to the Order History interface. This page presents a comprehensive record of the customer's previously placed orders. The interface provides search and filtering options, enabling users to efficiently locate specific orders based on defined criteria.

By selecting an individual order record, the customer can view detailed order information, including purchased items, pricing breakdowns, delivery details, and transaction information. Additionally, the system clearly displays the delivery status (e.g., Pending, Processing, Delivered) and payment status (e.g., Pending, Paid, Failed) for each order, ensuring transparency and real-time tracking of order progress.

Order information page



The screenshot shows the 'Order Information' page for the ISDN Distribution Network. The page header includes navigation links: Dashboard, Products, Cart, Orders, Tracking, and a user profile icon. The main content area is titled 'Order Information' with a subtitle 'Detailed view of customer order'. It features a summary card with the following details:

- ORDER NUMBER:** #ORD-RDCS-260213-1025
- DATE:** 13 Feb 2026
- CUSTOMER:** Vijaya Stores - Galle
- SALES REF:** N/A
- TOTAL AMOUNT:** Rs. 18,751.46
- ESTIMATED DELIVERY:** 15 Feb 2026
- LAST UPDATED:** 13 Feb 2026
- ORDER STATUS:** Pending

Below the summary card is a table titled 'Order Products' with the following columns: PRODUCT CODE, PRODUCT NAME, CATEGORY, SELLING PRICE, QTY, DISCOUNT, and LINE TOTAL.

PRODUCT CODE	PRODUCT NAME	CATEGORY	SELLING PRICE	QTY	DISCOUNT	LINE TOTAL
PRD-BEV-000001	Coca-Cola 1L	Beverages	Rs. 300.00	10	Rs. 150.00	Rs. 2,850.00
PRD-GNF-000001	Nestomalt 400G	Grocery & Food Items	Rs. 750.00	7	Rs. 0.00	Rs. 5,250.00
PRD-HCP-000001	Harpic Fresh 500ml	Home Cleaning Products	Rs. 380.50	5	Rs. 0.00	Rs. 1,902.50
PRD-HHE-000001	Sunlight Detergent Powder - 1kg	Household Essentials	Rs. 320.50	5	Rs. 160.25	Rs. 1,442.25
PRD-PSC-000001	Clogard Toothpaste 200g	Personal Care	Rs. 360.00	10	Rs. 0.00	Rs. 3,600.00

The footer section contains the ISDN logo, a description of the network, quick links (Products, About Us, Contact, FAQ), RDC Locations (Central Region - Kandy, Northern Region - Jaffna, Southern Region - Galle, Eastern Region - Batticaloa, Western Region - Colombo), contact information (+94 11 234 5678, info@isdn.lk), and social media icons. It also includes a copyright notice for 2026 ISDN and links to Privacy Policy and Terms.

Figure 23 Order information page

Upon selecting a specific order from the My Orders interface, the system navigates the customer to the Order Information page. This interface provides comprehensive information related to the selected order, including the list of purchased items, corresponding quantities, and individual pricing details. Additionally, key order metadata such as the Order ID, order date, delivery date, and current order status are clearly displayed. The interface ensures transparency by presenting both delivery progress and payment status, thereby enabling customers to effectively monitor the lifecycle of their orders.

Sales Ref

The RDC Sales Representative user interfaces are functionally similar to those of retail customers, providing access to similar system features and workflows. Sales representatives are able to authenticate into the system and perform core operations such as browsing products, managing the shopping cart, and completing the order checkout process. Also, he can access the system through his mobile app.

The primary difference is in the order placement functionality. Unlike retail customers, RDC Sales Representatives are provided with the capability to select a registered customer during the checkout process. This feature enables them to place orders on behalf of retail customers, facilitating assistance sales and field-based order processing. Apart from this additional functionality, the navigation structure and overall interface design remain consistent with the retail customer module, ensuring usability and reducing the learning curve for system users.

Dashboard

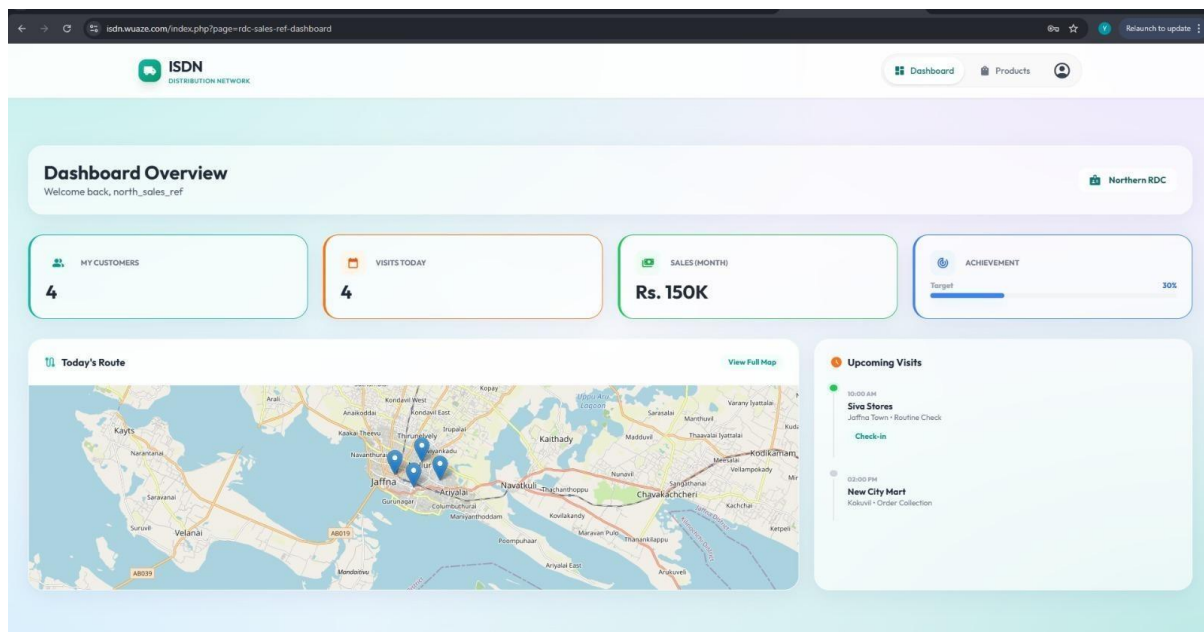


Figure 24 Sales Ref dashboard

The Sales Representative Dashboard provides a consolidated overview of individual sales performance metrics, presented in a structured tile-based layout.

Orders page

[Dashboard](#)
[Products](#)
[Cart](#)
[Orders](#)
[Tracking](#)

My Orders

Manage and track your recent orders

[Filter](#)
[Report](#)

CODE	CUSTOMER	DATE	AMOUNT	PAYMENT	STATUS	ESTIMATED DELIVERY	ACTION
#ORD-RDCS-260211-6542 5 Items	Wijitha Stores	2026-02-11	Rs. 25,300.00	Paid	Pending	2026-02-13	View Details
#ORD-RDCS-260211-2546 3 Items	Amal Stores	2026-02-10	Rs. 18,650.00	Paid	Processing	2026-02-12	View Details
#ORD-RDCS-260209-2545 6 Items	Vijaya Stores	2026-02-09	Rs. 18,650.00	Paid	Delivered	2026-02-11	View Details
#ORD-RDCS-260205-7854 4 Items	Shanthi Stores	2026-02-08	Rs. 15,356.00	Paid	Delivered	2026-02-07	View Details
#ORD-RDCS-260208-6984 3 Items	LGJ Super Store	2026-02-06	Rs. 17,365.00	Paid	Cancelled	2026-02-08	View Details

Showing 1-5 of 12 orders

[Previous](#)
[1](#)
[2](#)
[3](#)
[Next](#)

IslandLink Sales Distribution Network - Your trusted partner for modern wholesale and retail distribution.

Quick Links

- Products
- About Us
- Contact
- FAQ

RDC Locations

- Central Region - Kandy
- Northern Region - Jaffna
- Southern Region - Galle
- Eastern Region - Batticaloa
- Western Region - Colombo

Contact Us

+94 11 234 5678

info@isdn.lk

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

© 2026 ISDN - IslandLink Sales Distribution Network. All rights reserved.

[Privacy Policy](#) [Terms](#) [Up](#)

Figure 25 Sales ref order page

Sales ref order page

[Dashboard](#)

My Orders

Manage and track your recent orders

CODE	CUSTOMER	DATE	AMOUNT	PAYMENT	STATUS	ESTIMATED DELIVERY	ACTION
#ORD-RDCS-260211-6542 5 Items	Wijitha Stores	2026-02-11	Rs. 25,300.00	Paid	Pending	2026-02-13	View Details
#ORD-RDCS-260211-2546 3 Items	Amal Stores	2026-02-10	Rs. 18,650.00	Paid	Processing	2026-02-12	View Details
#ORD-RDCS-260209-2545 6 Items	Vijaya Stores	2026-02-09	Rs. 18,650.00	Paid	Delivered	2026-02-11	View Details
#ORD-RDCS-260205-7854 4 Items	Shanthi Stores	2026-02-08	Rs. 15,356.00	Paid	Delivered	2026-02-07	View Details
#ORD-RDCS-260208-6984 3 Items	LGJ Super Store	2026-02-06	Rs. 17,365.00	Paid	Cancelled	2026-02-08	View Details

Showing 1-5 of 12 orders

[Previous](#)
[1](#)
[2](#)
[3](#)
[Next](#)

IslandLink Sales Distribution Network - Your trusted partner for modern wholesale and retail distribution.

Quick Links

- Products
- About Us
- Contact
- FAQ

RDC Locations

- Central Region - Kandy
- Northern Region - Jaffna
- Southern Region - Galle
- Eastern Region - Batticaloa
- Western Region - Colombo

Contact Us

+94 11 234 5678

info@isdn.lk

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

© 2026 ISDN - IslandLink Sales Distribution Network. All rights reserved.

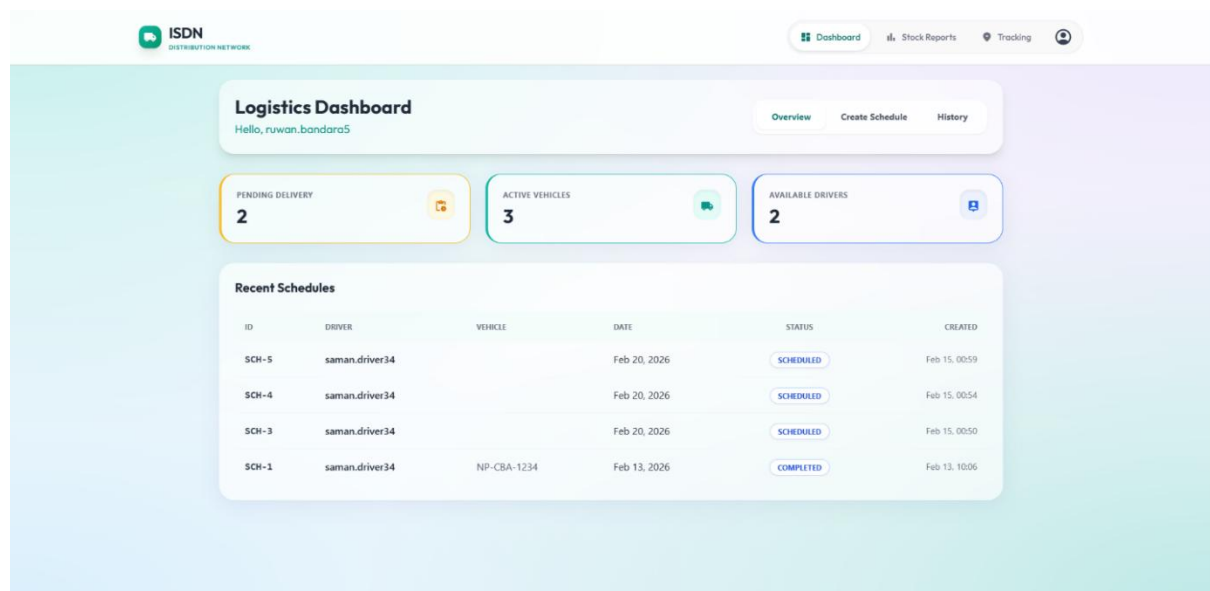
[Privacy Policy](#) [Terms](#) [Up](#)

Figure 26 Sales ref order page with filter option

Sales ref order page with filter option

2. logistic and driver interfaces

Logistic Officer



The Overview screen gives the logistics officer dashboard for navigation and decision-making by showing only the key wayfinding points, Pending Delivery, Active Vehicles, and Available Drivers as instant status indicators, plus Recent Schedules as the running log to confirm what has already been arranged and what needs attention next.

ISDN
DISTRIBUTION NETWORK

Dashboard Stock Reports Tracking

Logistics Dashboard
Hello, ruwan.bandara5

Overview **Create Schedule** History

Select Orders for Delivery

ORDER DETAILS	CUSTOMER	AMOUNT
☐ ORD-2026-27611 ID: 15041	City Super City 128 123 Main St, Jaffna	Rs. 26,126
☐ ORD-2026-71194 ID: 15042	City Super City 128 123 Main St, Jaffna	Rs. 49,167

Trip Details

SELECT DRIVER
-- Choose Driver --

SCHEDULE DATE
mm/dd/yyyy

Create Schedule

Figure 28 Create Schedule Interface

The Create Schedule screen supports simple, guided navigation by letting the officer select the orders for delivery, choose a driver, and pick a delivery date, and finally confirm the action with Create Schedule, which schedules the delivery and moves the order forward in the workflow.

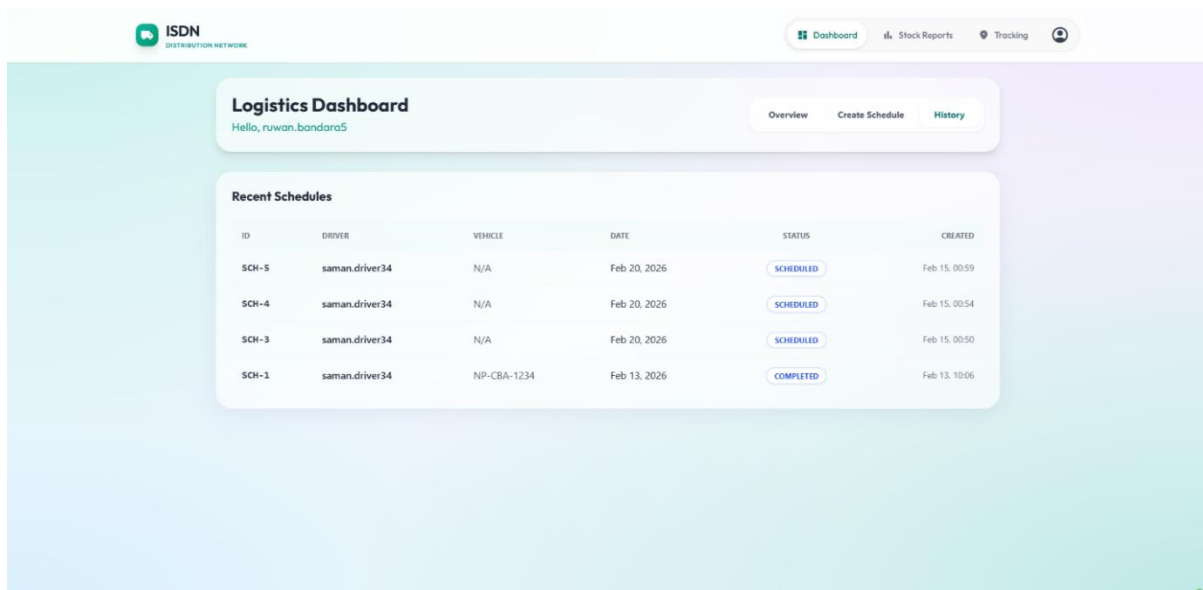


Figure SEQ Figure * ARABIC 29 Order schedule History interface

The History screen supports navigation through past work by showing a clear list of recent schedules (assigned orders) with their key identifiers and status, so the logistics officer can quickly review what was scheduled or completed and trace previous assignments.

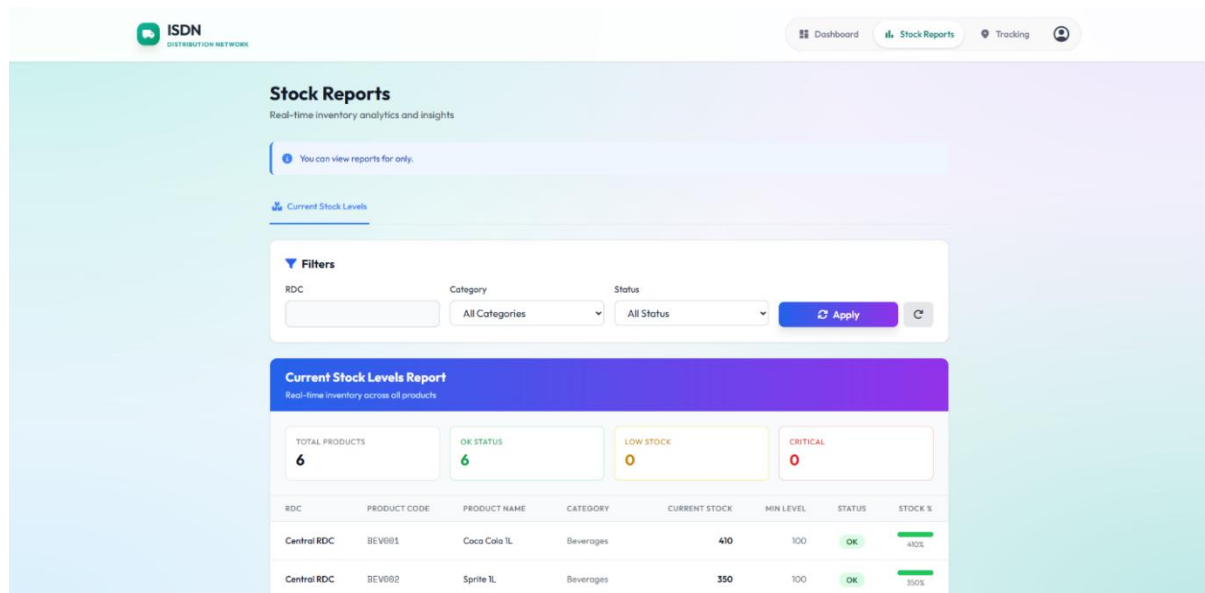


Figure SEQ Figure * ARABIC 30 Stock Report Interface

The Stock Reports screen supports clear navigation through inventory by showing the current stock levels and providing simple filters (RDC, category, and status) so the officer can quickly narrow the view, apply the filter, and check stock conditions without searching through unnecessary data.

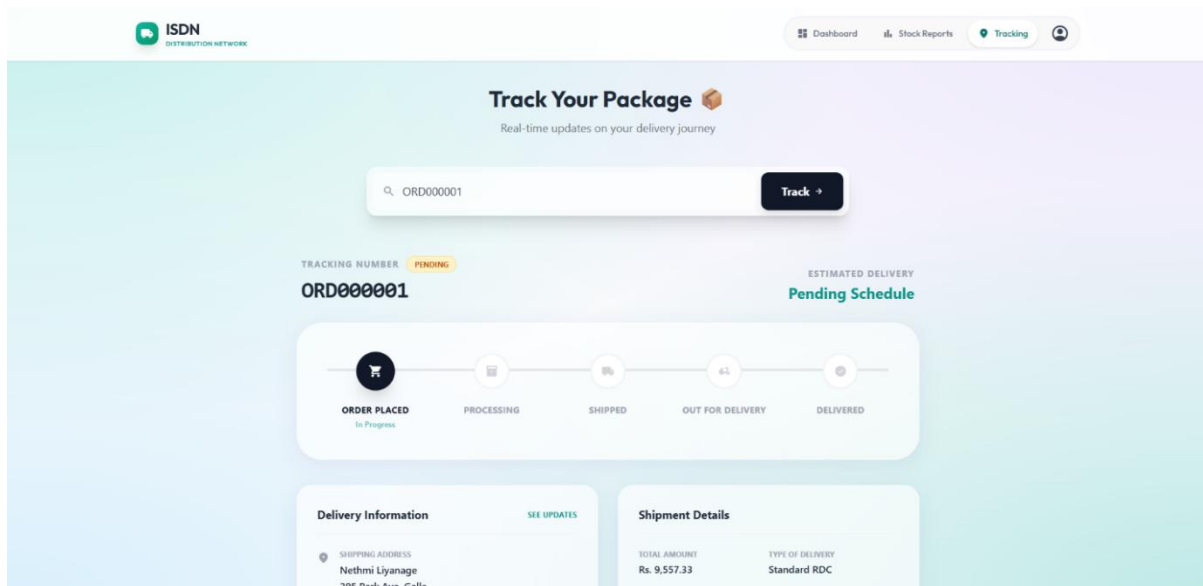


Figure SEQ Figure * ARABIC 31 Order Tracking Interface

The Tracking screen supports navigation through the delivery journey by letting the user search an order number and then view a clear status timeline with supporting delivery details, so the officer can quickly understand where the order is right now and follow its movement until completion.

RDC Driver

The interface displays the following information:

- ISDN DISTRIBUTION NETWORK** logo and navigation tabs: Today's Route, Route Map, Delivery History, and a user profile icon.
- Delivery Status Summary:**
 - ASSIGNED: 13
 - REMAINING: 8
 - COMPLETED: 5
- Current Stop Details:**
 - Ruwan Perera** (Uduvil)
 - Order #ORD-30500 • 1 items
 - Buttons: Navigate, Call, Details, Update
- Map:** A map showing the current location and route, with labels for Uduvil, Chunnakam, Atchelu, and Siruppiddy.
- Today's Route (8 orders):**

Order Number	Customer Name	Address	Status	Order ID
1	Ruwan Perera	Uduvil	Out for delivery	#ORD-30500
2	Nimal Perera	Chunnakam	Out for delivery	#ORD-10528
3	Chamara Samaraweera	481 Hill St, Negombo	Processing	#ORD000006
4	Kamal Perera	Nallur	Processing	#ORD-60193
5	Maresh Perera	Manipay	Processing	#ORD-24371
6	Saman Perera	Jaffna Town	Confirmed	#ORD-37793
7	Saman Perera	Jaffna Town	Confirmed	#ORD-90848
8	Wanindu Perera	Chavakachcheri	Pending	#ORD-36508

Figure 32 RDC Driver overview interface

The RDC Driver dashboard supports navigation during delivery execution by showing a quick route overview (Assigned, Remaining, Completed), highlighting the current stop, and providing a map plus a sequenced stop list so the driver can orient instantly, move step by step to the next destination, and use simple actions (navigate, call, view details, update) to stay aligned with the planned route.

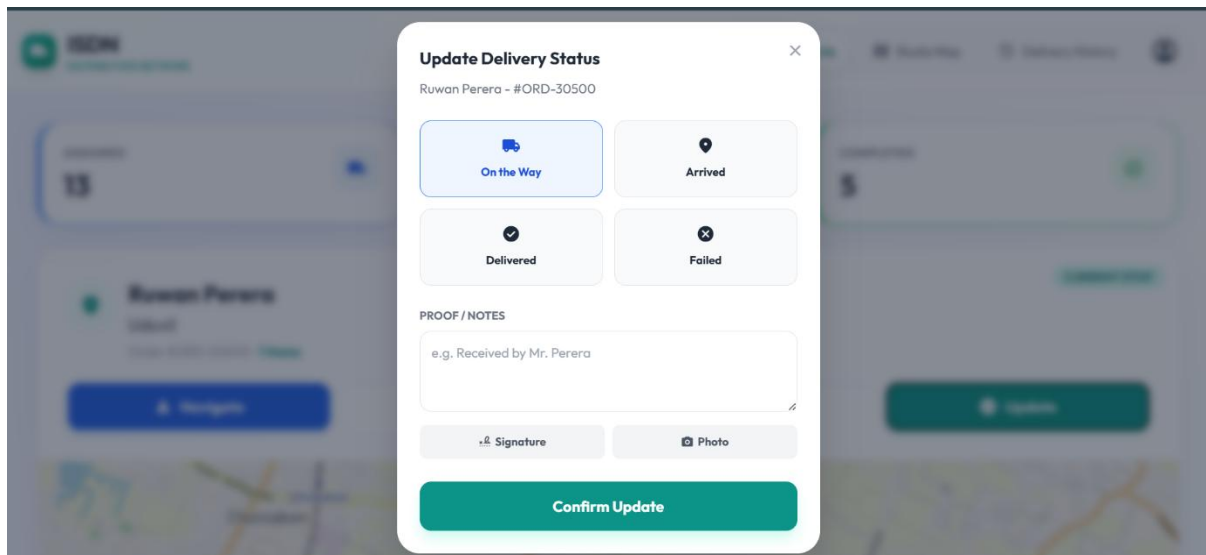


Figure 33 Update Delivery Staters

This Update Delivery Status popup in the RDC Driver dashboard supports fast, error-reducing navigation by letting the driver choose the current checkpoint (On the Way, Arrived, Delivered, Failed), add brief proof / notes then confirm the update. so, the delivery journey stays accurately tracked with clear step by step state changes.

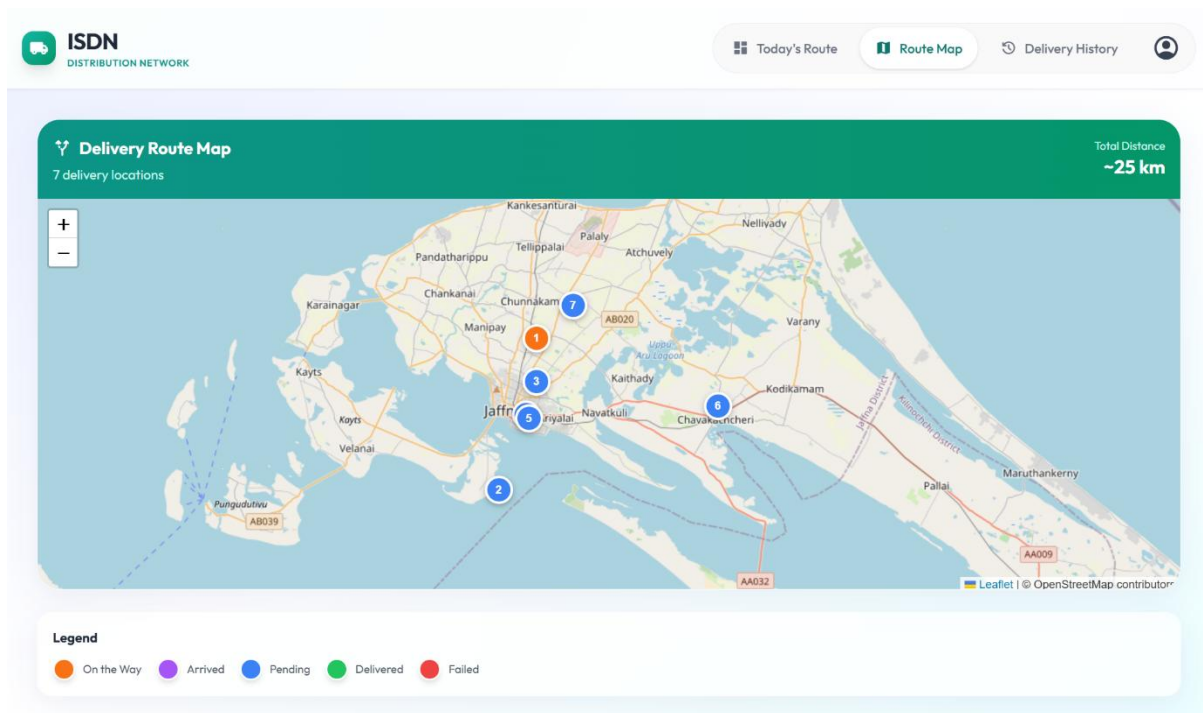


Figure 34 Routing Map Interface

The Route Map screen in the RDC Driver dashboard supports navigation by visualizing the driver's full delivery path on a map with numbered stops and color coded statuses, so the driver can quickly understand the route structure, see progress at a glance, and use the map as a real time guide for moving efficiently from one location to the next.

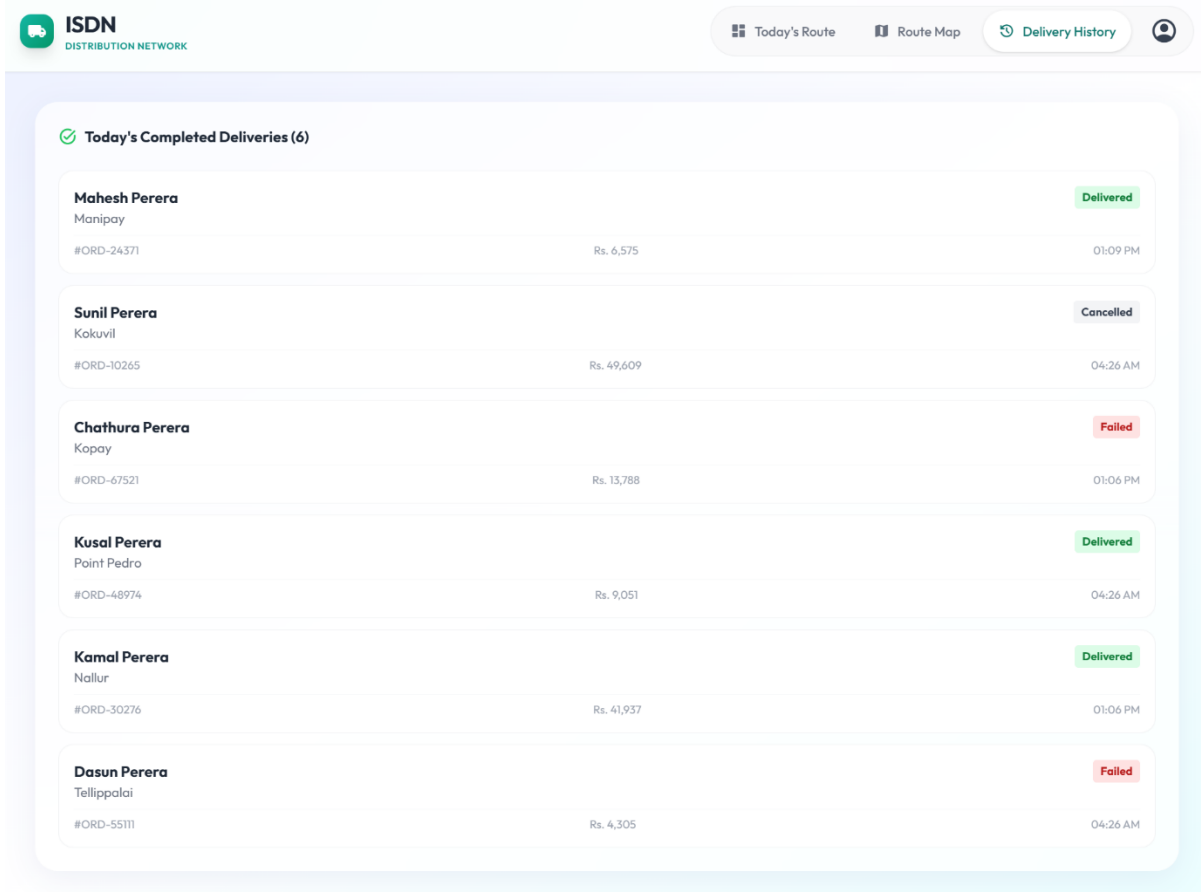


Figure 35 Delivery History Interface

The Delivery History screen in the RDC Driver dashboard supports navigation by presenting a clear timeline style list of completed stops with their final status (delivered, cancelled, failed), so the driver can quickly review what has been finished, confirm outcomes, and maintain an accurate record of the day's route progress.

Today's Route
Route Map
Delivery History

Login successful!

←
My Profile
Manage your account settings

Profile Information

USERNAME

EMAIL

FULL NAME

CONTACT NUMBER

ADDRESS

Role: **Rdc Driver**

Change Password

CURRENT PASSWORD

NEW PASSWORD

Account Details

User ID	#22
Joined	Feb 12, 2026
Role	Rdc Driver

Figure 36 Delivery Rider Profile Interface

The My Profile screen in the RDC Driver dashboard supports navigation by giving the user a clear, manage account details, view and update profile information and change the password

3. For Admin interfaces

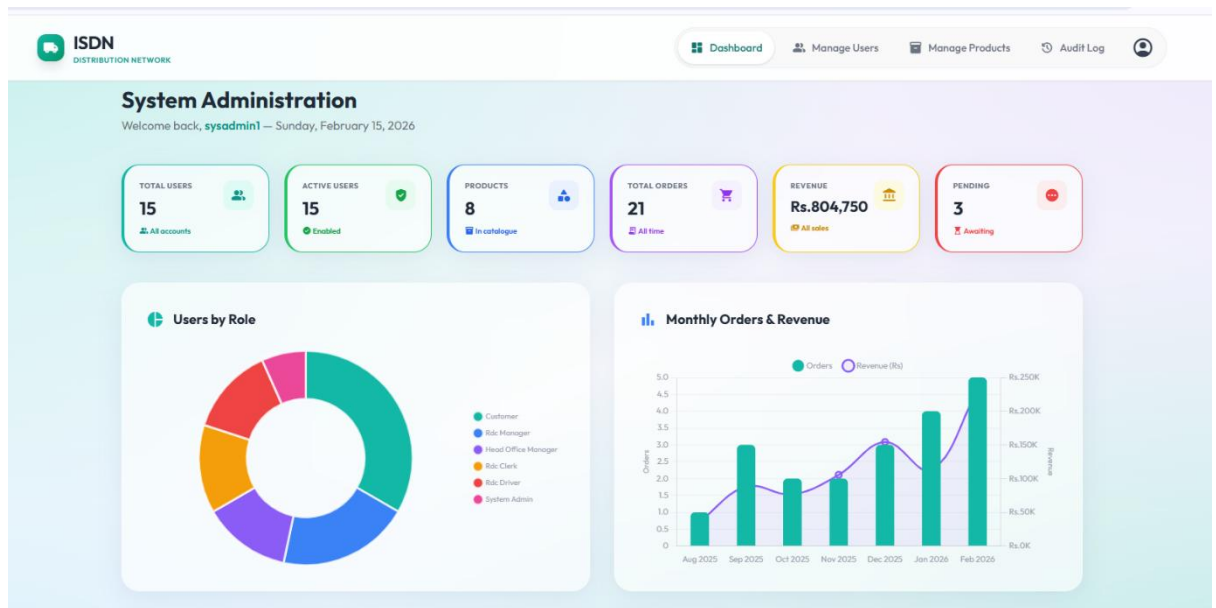
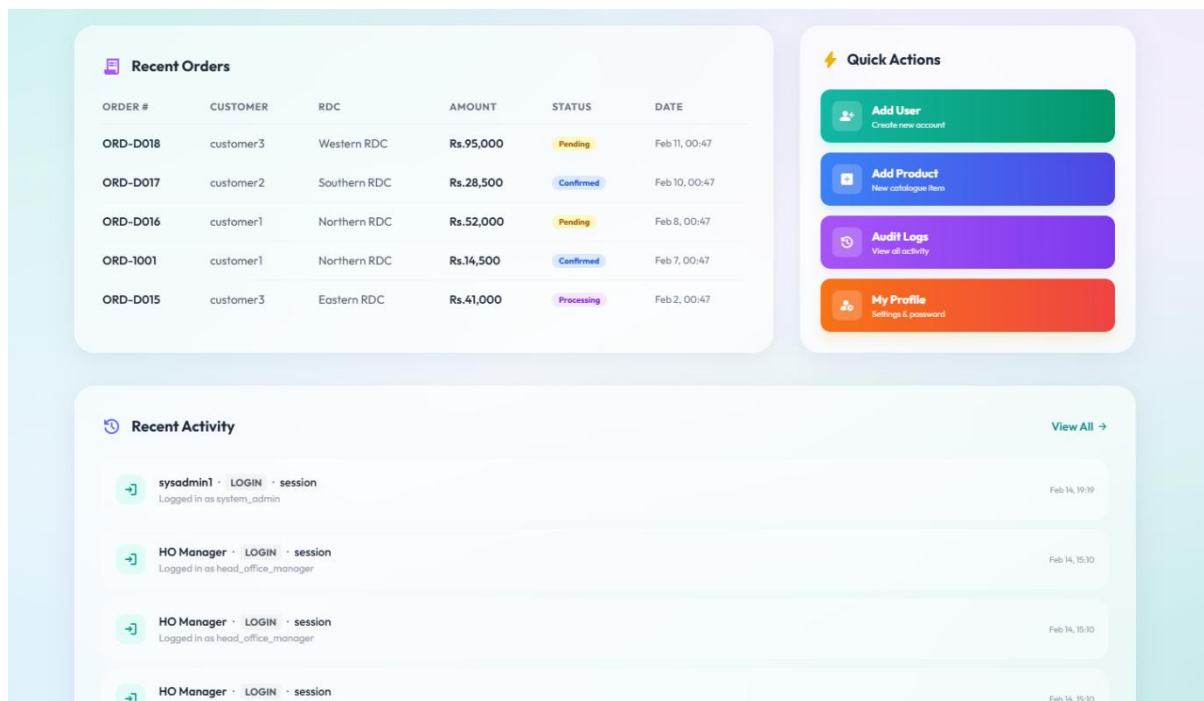


Figure 37 Admin dashboard



The design phase created a dashboard layout which enables users to navigate between system modules. The System Administration interface features a top navigation bar which contains access links that lead to Dashboard, Manage Users, Manage Products, and Audit Log.

User Management
Manage accounts and roles

15 TOTAL USERS

15 IN LIST

8 roles
Customer, RDC, Admin...

SEARCH
Username or email...

ROLE
All Roles

Search

#	USER	EMAIL	ROLE	RDC	STATUS	JOINED	ACTIONS
15	yasas pasindu	yasaspasindufernando@gmail.com	Customer	-	Active	Feb 12, 2026	
14	nayana kumari	nayanakumarimama@gmail.com	Customer	-	Active	Feb 12, 2026	
13	HO Manager	hom@isdn.com	Head Office Manager	-	Active	Feb 12, 2026	
12	yasas	yasasnew@gmail.com	Rdc Manager	-	Active	Feb 9, 2026	
1	sysadmin1	sysadmin1@mail.com	System Admin	-	Active	Feb 6, 2026	
2	headoffice1	headoffice1@mail.com	Head Office Manager	-	Active	Feb 6, 2026	

Figure 38 • Users Manage

- Manage Products

Product Management
Catalogue and inventory

8 TOTAL PRODUCTS

8 IN LIST

3 LOW STOCK (THIS PAGE)

SEARCH
Name or code...

CATEGORY
All

Search

CODE	PRODUCT	CATEGORY	PRICE	STOCK	STATUS	ACTIONS
P001	Cement 50kg	Construction	Rs.1,450.00	445 / 100	Active	
P002	Steel Rod 12mm	Construction	Rs.2,200.00	230 / 200	Active	
P003	Sand 1 Cube	Raw Material	Rs.7,500.00	208 / 50	Active	
P004	Paint 5L White	Finishing	Rs.3,200.00	80 / 75	Active	
P005	PVC Pipe 2inch	Plumbing	Rs.1,200.00	40 / 150	Active	
P006	Bricks Pack 100	Construction	Rs.1,800.00	100 / 300	Active	

Figure 39 • Manage Products

- Audit Log

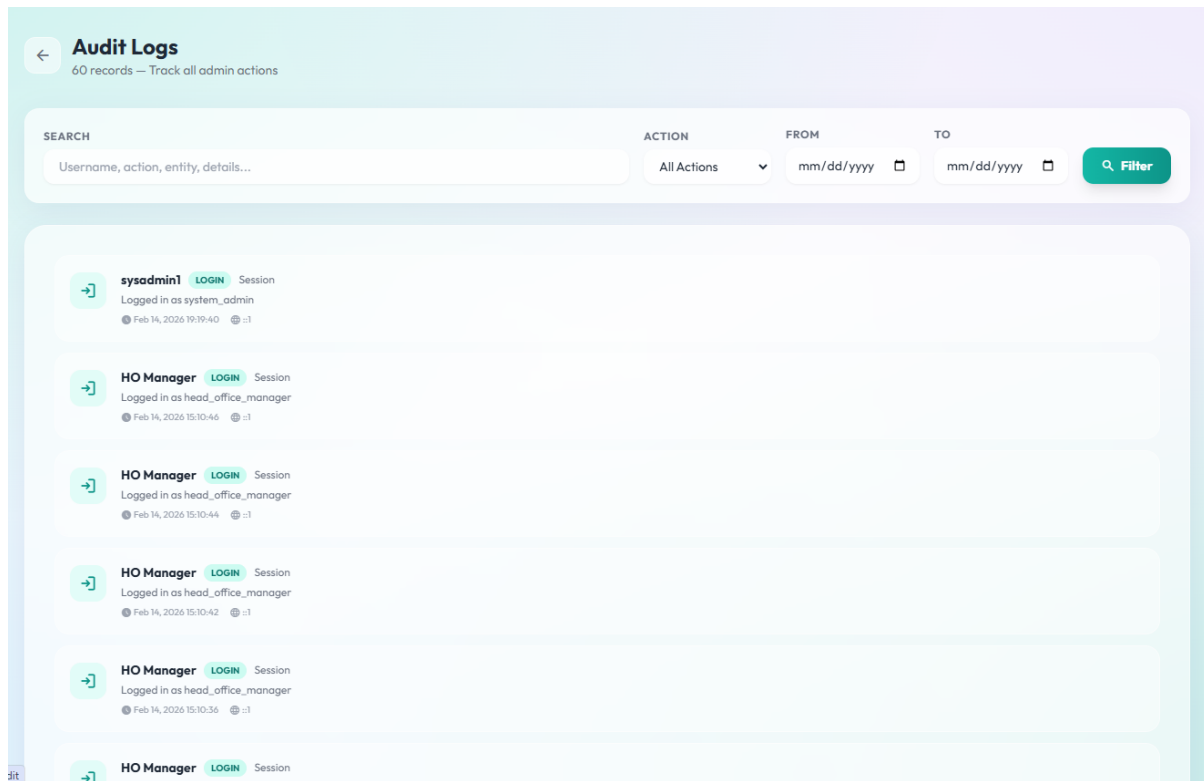


Figure 40 • Audit Log

The navigation system enables users to travel between website sections without experiencing any confusion. The system uses navigation items to represent distinct system modules which create distinct operational boundaries while linking associated functions.

The dashboard homepage was designed to act as a centralized control panel displaying:

- The dashboard displays KPI summary cards which show Total Users Active Users Products Orders Revenue and Pending Orders.
- Role distribution visualization (Users by Role chart)
- Monthly performance analytics (Orders Revenue chart)

The design emphasizes:

- Visual clarity through color-coded information card

- Scalable content blocks which use modular design elements
- Role-based navigation visibility using RBAC controls
- Multi-device accessibility through responsive layout design

The UI design uses reusable layout components together with specified navigation routes to create a system which is user-friendly and easy to maintain and ready for future growth.

Head office manager dashboard

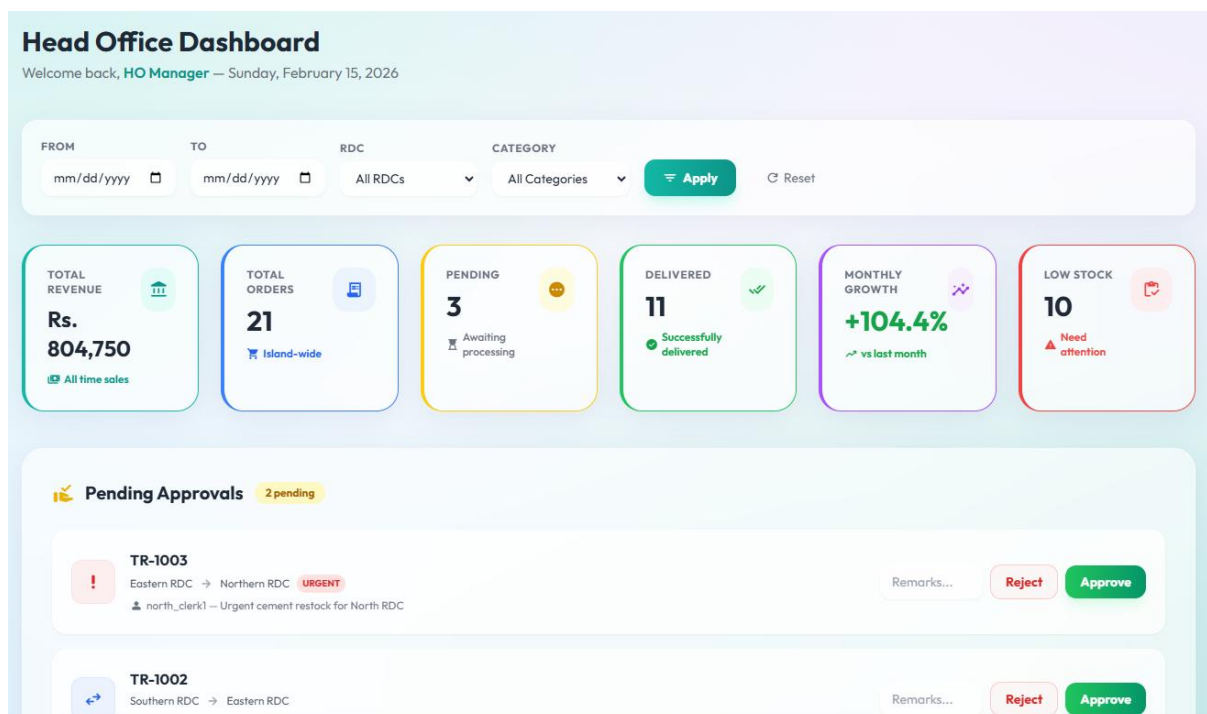
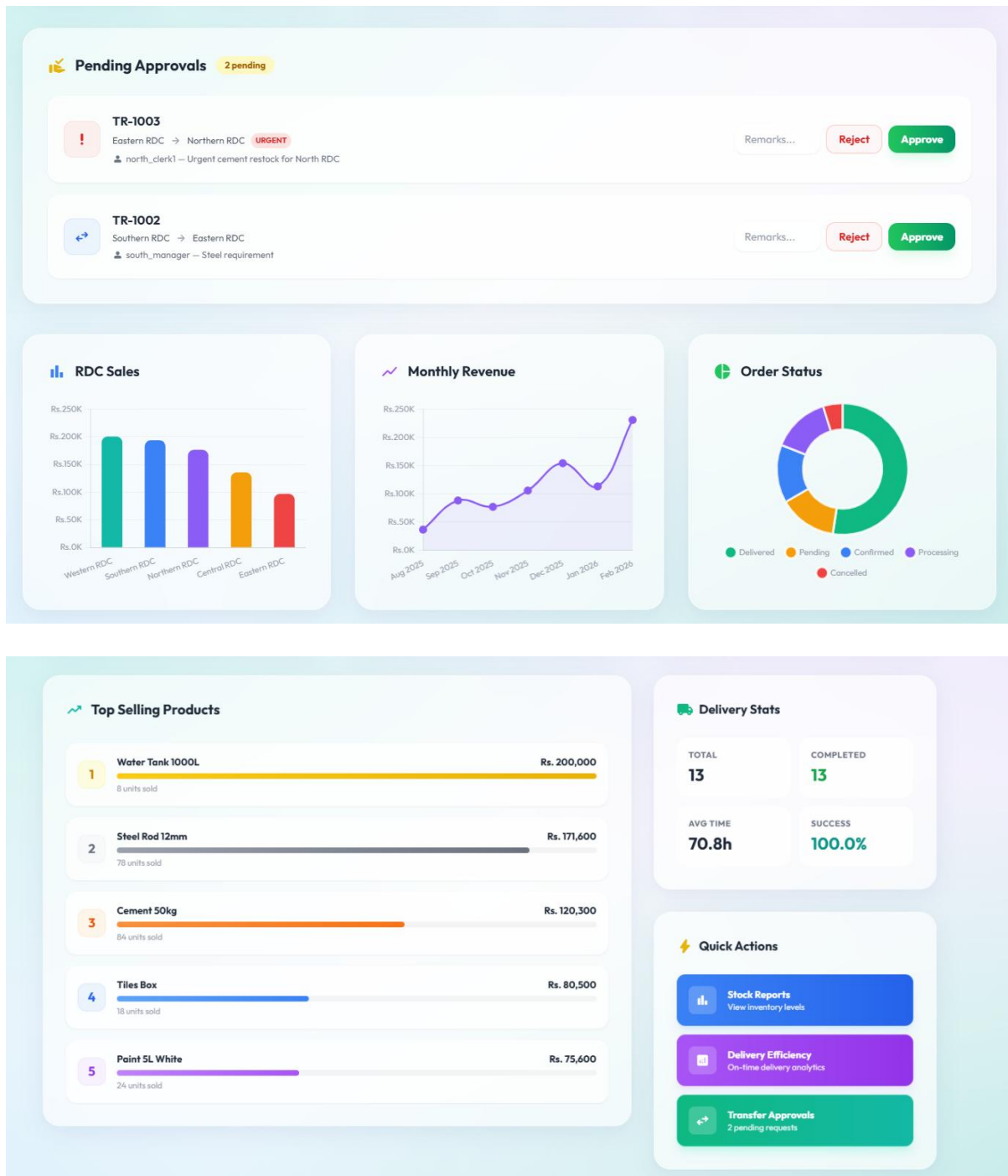


Figure 41 Head office manager dashboard



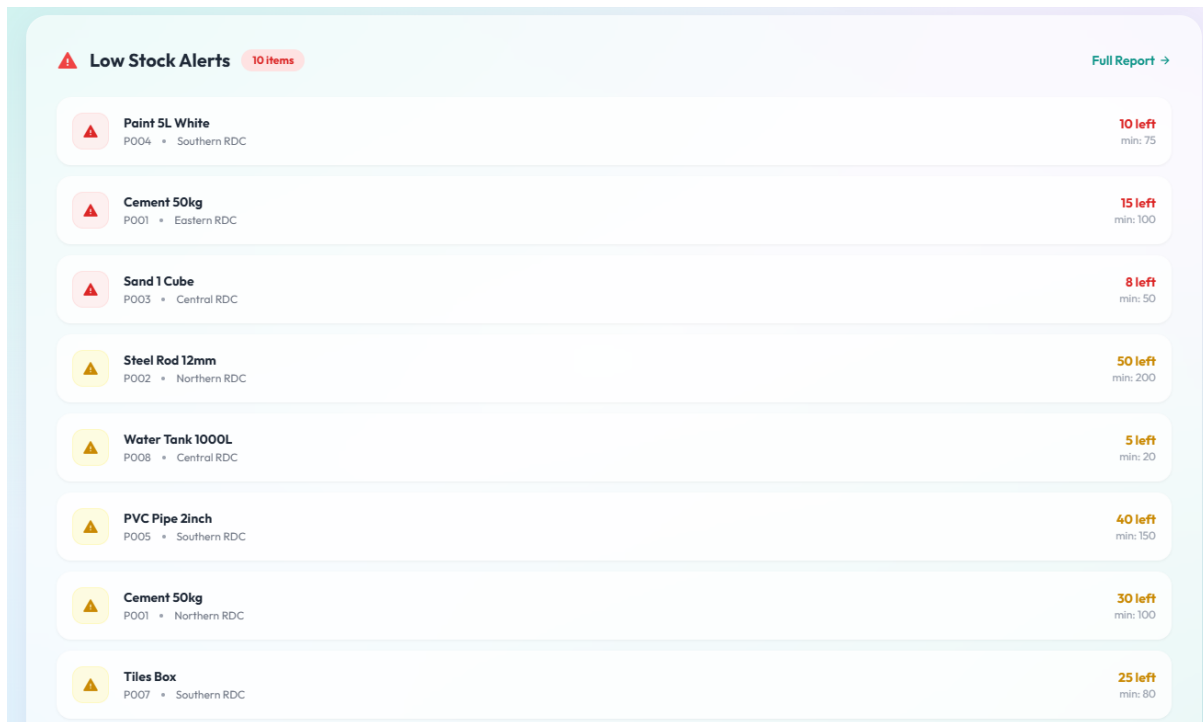


Figure 42 Head office manager dashboard

The Head Office Manager dashboard was designed as a centralized decision-support interface that provides high-level operational visibility across all Regional Distribution Centers (RDCs). The layout uses a modular dashboard design which combines reusable UI elements with a grid system that adapts to different screen sizes to maintain design readability and system performance.

The top section presents key performance indicator (KPI) cards including total revenue, total orders, pending orders, delivered orders, monthly growth percentage, and low stock alerts. The system calculates KPI components through aggregated SQL queries which display results through reusable dashboard card templates. Each card highlights performance status using color-coded indicators to support quick managerial decision-making.

Users can dynamically filter data through three options which include date range, RDC selection, and category filtering. This system improvement facilitates better usability while helping users analyze operational trend data.

The dashboard includes different visualization modules which provide multiple ways to display information:

1. Revenue and order trend analysis using line and bar charts
2. RDC-wise sales comparison charts
3. Order status distribution through donut chart visualization
4. Delivery performance statistics (success rate, average delivery time)
5. Top-selling product rankings with revenue indicators
6. Pending transfer approvals with action controls (approve/reject)
7. Low stock alerts with threshold-based visual warnings

access links that lead to Dashboard, Manage Users, Manage Products, and Audit Log.

- Stock report

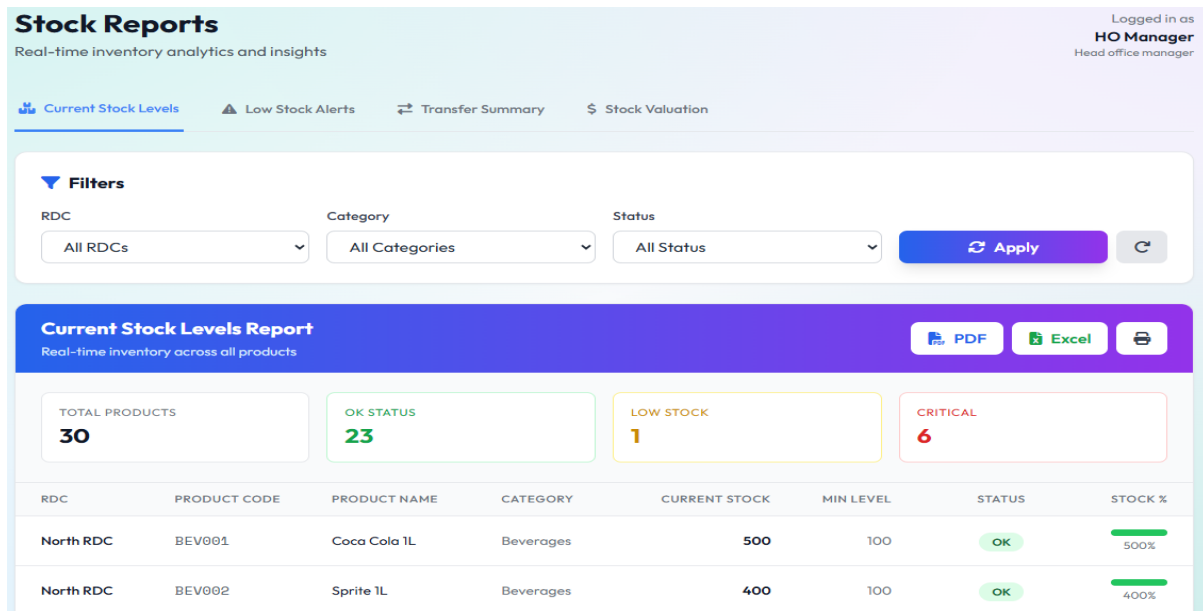


Figure 43 • Stock report

North RDC	BEV002	Sprite 1L	Beverages	400	100	OK	400%
North RDC	FOOD001	Rice 5kg	Packaged Foods	200	50	OK	400%
North RDC	FOOD002	Bread Loaf	Packaged Foods	300	200	OK	150%
North RDC	CLEAN001	Detergent 500g	Home Cleaning	280	80	OK	350%
North RDC	CARE001	Toothpaste 100ml	Personal Care	350	150	OK	233%
South RDC	BEV001	Coca Cola 1L	Beverages	20	100	CRITICAL	20%
South RDC	BEV002	Sprite 1L	Beverages	15	100	CRITICAL	15%
South RDC	FOOD001	Rice 5kg	Packaged Foods	5	50	CRITICAL	10%
South RDC	FOOD002	Bread Loaf	Packaged Foods	30	200	CRITICAL	15%
South RDC	CLEAN001	Detergent 500g	Home Cleaning	150	80	OK	188%
South RDC	CARE001	Toothpaste 100ml	Personal Care	80	150	LOW	53%

Figure 44 Stock report

- delivery report

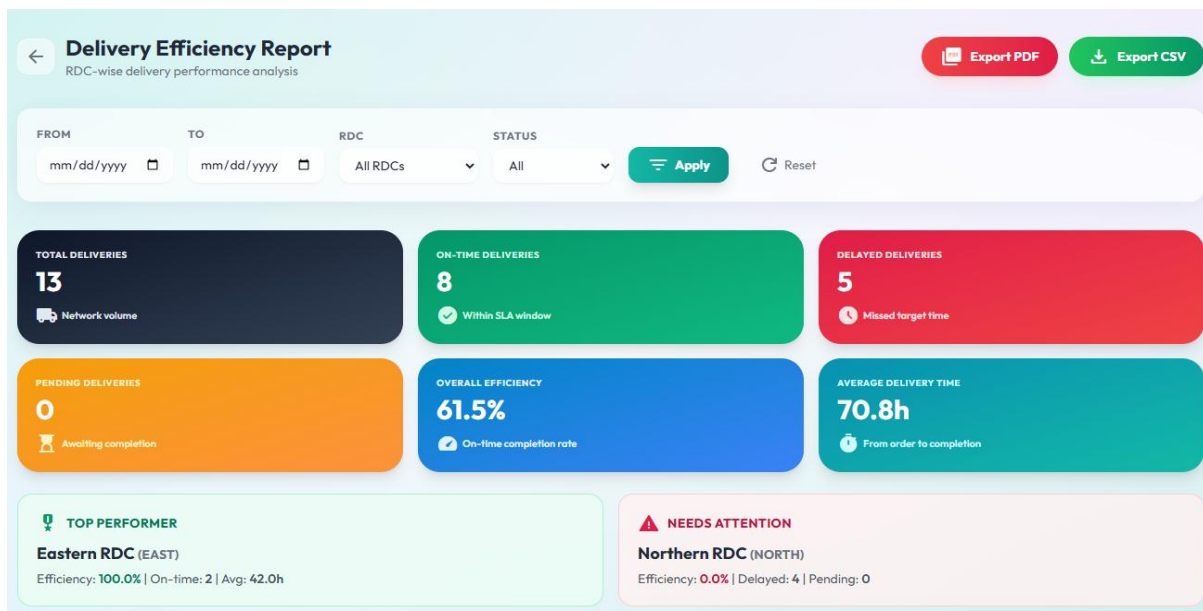


Figure 45 • delivery report

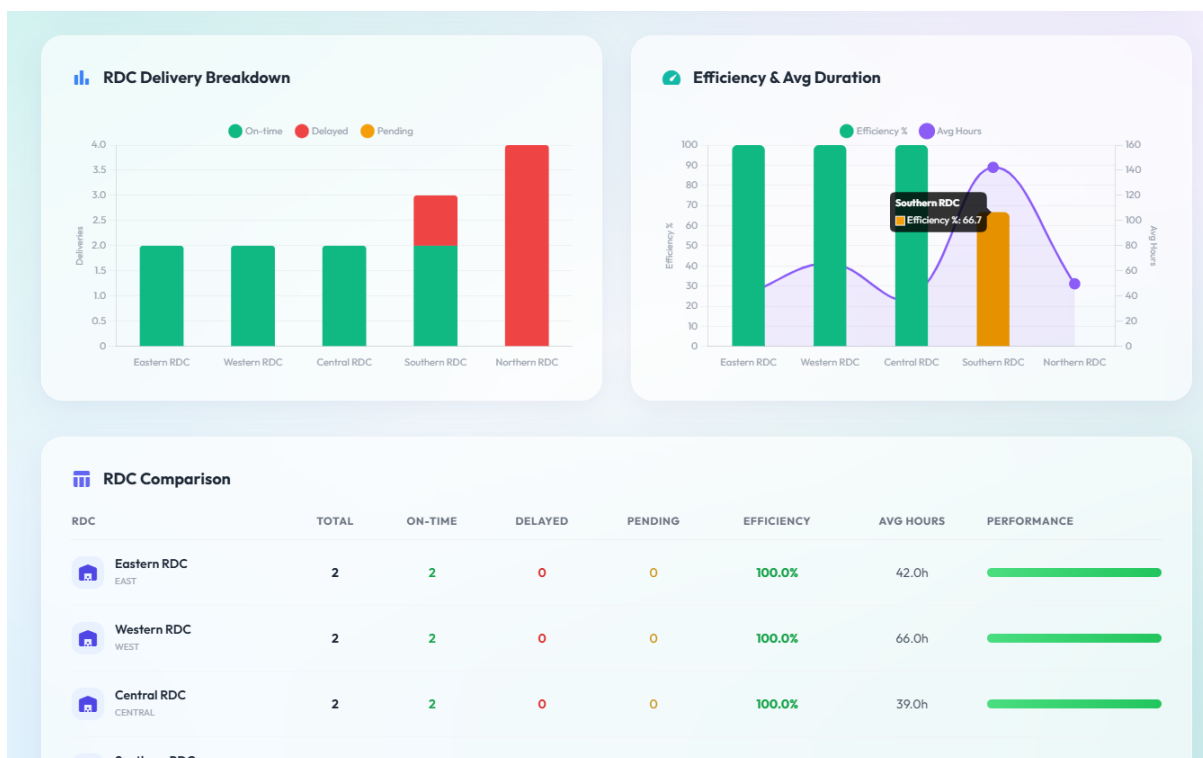
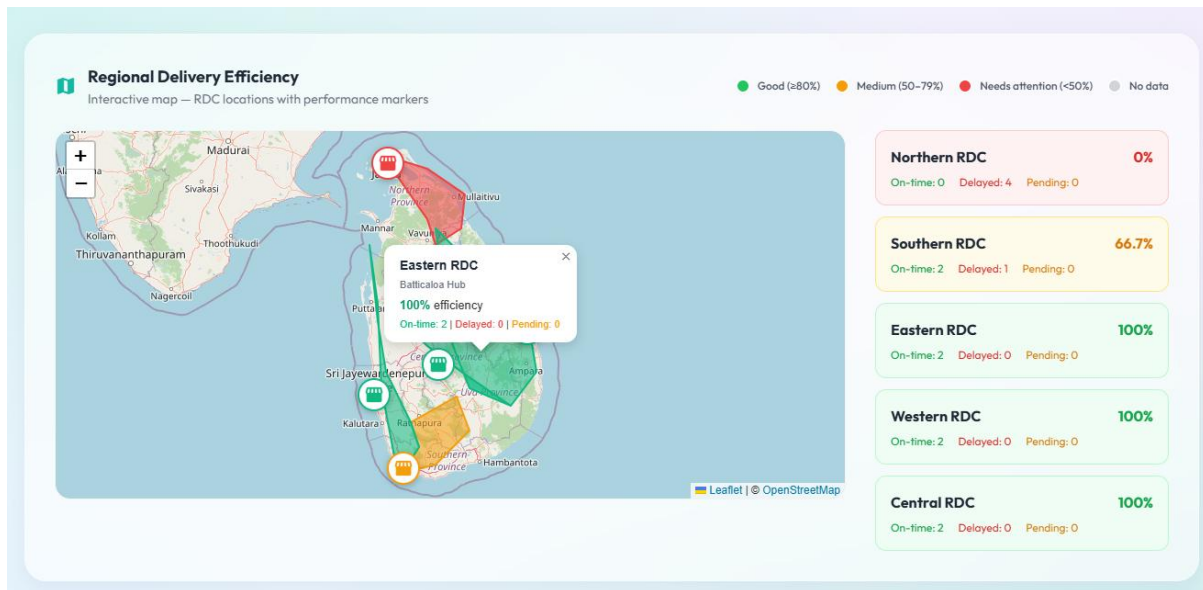


Figure 46 delivery report

- Sales report

All visual components are implemented using Chartjs and structured UI cards, ensuring consistency across different managerial dashboards. The dashboard design prioritizes data readability, action-oriented insights, and role-based access control to ensure that only authorized managerial users can access strategic business data.

The dashboard shows all visual elements through Chart.js and UI card components, which maintain design uniformity across different managerial dashboard displays. The dashboard design prioritizes data readability, action-oriented insights, and role-based access control to ensure that only authorized managerial users can access strategic business data.

The enterprise web application shows the implementation of modular UI design together with real-time data aggregation and RBAC-based interface control and analytical visualization techniques.

RDC Manager

Dashboard

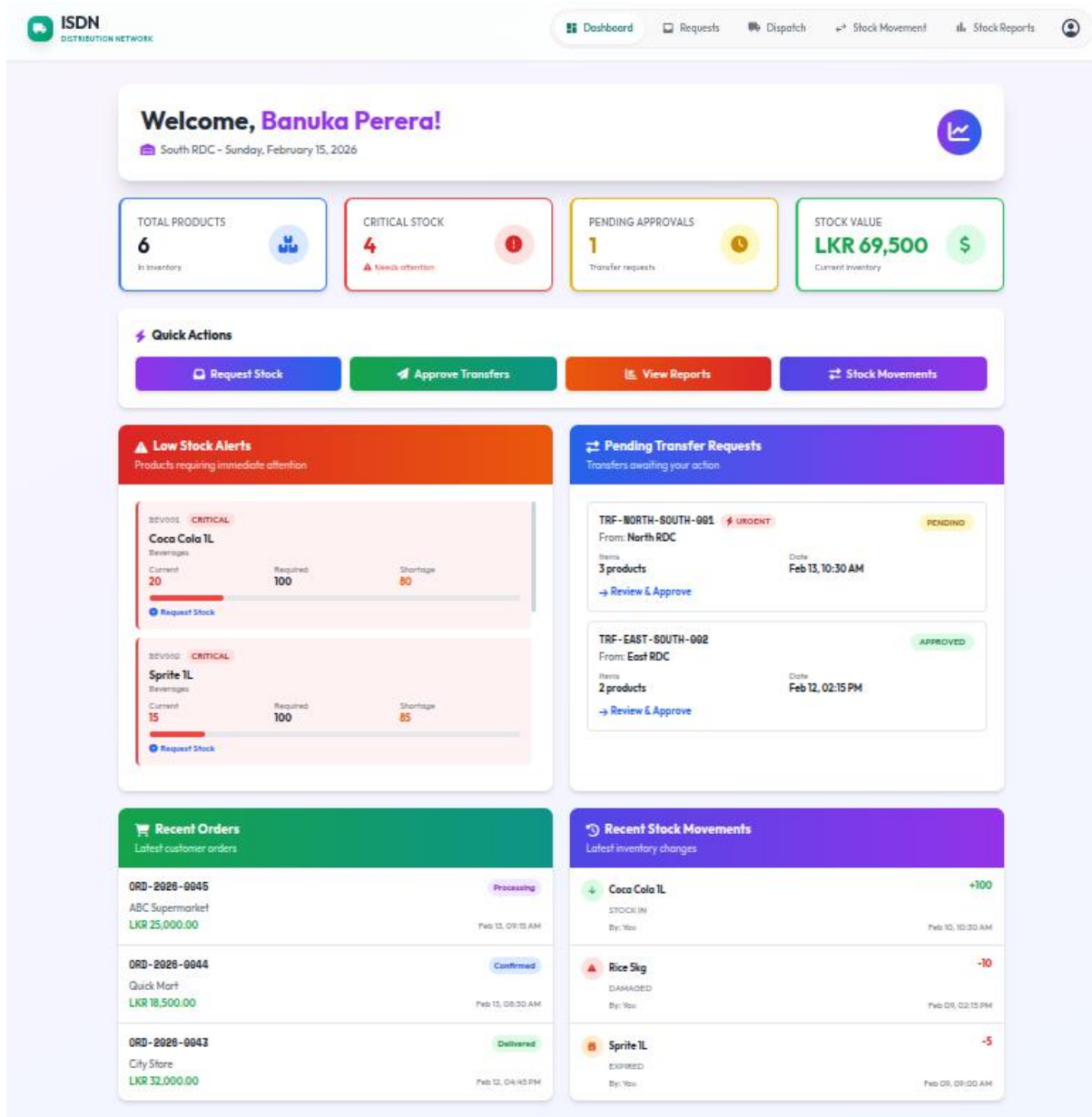


Figure 47 RDC Manager dashboard

The RDC Manager Dashboard is the main operational centre that gives a brief account of the main performance indicators (KPIs) of a distribution centre and allows responding quickly to urgent requirements. The dashboard has four main metrics: Total Product Count, which provides an inventory breadth overview; Critical Stock Count which shows the items with a

high level of danger; Pending Approvals which represent the number of transfer requests pending approval; and Stock Value which gives the monetary value of the inventory at the moment. The KPIs are displayed below, with a convenient Quick Action Menu that provides access to key functions, such as Request Stock, Approve Transfer, View Reports and Stock Movements. The bottom part of the dashboard is devoted to operational vigilance and history and presents the actionable information, e.g. Low Stock Alerts, a summary of Recent Pending Transfer Requests, a record of Recent Orders, and a timeline of Recent Stock Movements.

Request Transfer

ISDN

CONTRACT ORATION

Dashboard

Requests

Dispatch

Stock Movement

Stock Reports

Active Transfer Requests

TSP-82879-82879-888

From North RDC

2 products (333 units)

URGENT

CLASS REQUESTED

View Details

TSP-8287-82879-888

From East RDC

2 products (150 units)

PENDING

View Details

TSP-8887-82879-888

From West RDC

1 products (50 units)

URGENT

APPROVED

View Details

Request Product Units

Select products with low stock and request units from other RDCs

LOW STOCK ITEMS

5

CRITICAL STOCK

1

PENDING REQUESTS

3

YOUR RDC

SOUTH RDC

Check Stock at Other RDCs

Select an RDC to view their available stock for comparison

Default RDC to Check

Select RDC

Check Stock

BEVERAGE

Coca Cola 1L

Beverages

LOW STOCK

20

Not available

Minimum

500 Bottles

Request Quantity

Enter quantity needed

BEVERAGE

Sprite 1L

Beverages

LOW STOCK

15

Not available

Minimum

500 Bottles

Request Quantity

Enter quantity needed

FOODSTUFF

Rice 5kg

Perishable Foods

CRITICAL

5

Not available

Minimum

50 Bags

Request Quantity

Enter quantity needed

FOODSTUFF

Bread Loaf

Perishable Foods

LOW STOCK

30

Not available

Minimum

200 Pieces

Request Quantity

Enter quantity needed

PERSONAL CARE

Toothpaste 100ml

Personal Care

LOW STOCK

80

Not available

Minimum

500 Tubes

Request Quantity

Enter quantity needed

Submit Transfer Request

Request From RDC

Select Source RDC

Priority

Normal

URGENT

Reason for Request

Explain why this transfer is needed...

Estimated Delivery Time

2-3 Business Days

Request Period

Feb 02-04, 2024

Selected Products

We products selected yet. Check the boxes on product cards above.

Submit Transfer Request

Cancel

Figure 48 Request transfer units view

Request transfer process helps in enabling the RDC Clerks to request stock effectively to other Distribution Centers. The upper section of the interface provides the user with a list of recent transfer requests that can be easily checked and its status. Key performance indicators (KPIs) are evidently presented with immediate operational background: Low Stock Items Count, Critical Stock Count, Pending Request Count, and the Current User RDC. The RDC Manager can take the initiative of enquiring the availability of products and the current stock levels of all other RDCs before deciding to place a request. Its key feature allows requesting products in large quantities in a chosen RDC, and this request should be assigned to the Urgent or the Normal type, and a Remark should be omitted to complete and send the transfer request.

Transfer Approval

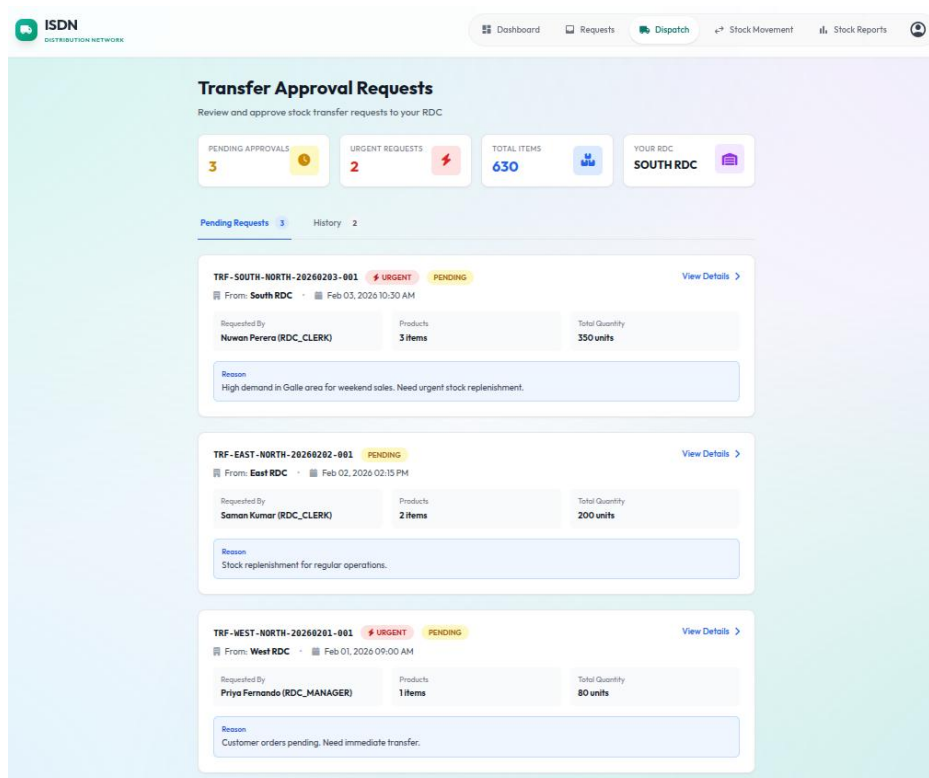


Figure 49 Transfer units approval view

Transfer Approval interface is the assigned system of the RDC Manager in reviewing and responding effectively to all the inter-Distribution Center stock transfers requests. The main view gives a complete list of all the transfer requests, logically divided into Pending Transfers where an action can be taken now and Transfer History (accessible by a separate tab) where the previous approvals, rejections and transferred items can be reviewed. Important information that can be seen in the list view just to do some quick screening is the Transfer ID, urgency (Urgent/Normal status), the Current Status (e.g., Pending, Approved) and the Requesting RDC. The manager is able to move into each of the requests to get to Transfer Details View. Such detailed view will give a full picture in high level header information, a more specific Product Breakdown (with items requested, quantities, and stock available) and the Remark of a Requester. Importantly, the manager will be assigned the power to approve or reject the request. Both choices are required to have Remark by the RDC Manager that will be recorded as a justification and will forever remain in the history of the request.

Stock Movements

Figure 50 Stock movements view

Stock Movement Management page is an important tool that RDC Managers use to manage inventory exceptions that do not qualify as part of the normal sales or transfer processes. The interface is a tabbed design with a simple layout which differentiates between instant activities and past auditing. The stock levels on the Create Movement tab can be changed manually during certain events like the acceptance of new supplies (stock In), registered damages (Damaged) or a customer returns (Returns). This is made easy by the system which includes different, color-coded icons which are used to represent each type of movement and new stock balance is calculated automatically before confirmation of submission. To avoid mistakes, a very rigid validation is provided in the interface where it is impossible to enter the quantities

which would lead to negative stock. Moreover, each change must have a mandatory "Reason" or note, and this is to ensure that a clear audit trail is maintained of the change. The "Movement History" tab can be used to filter previous records based on date, product or type of movement to know when and why the inventory levels have changed.

Stock Reports

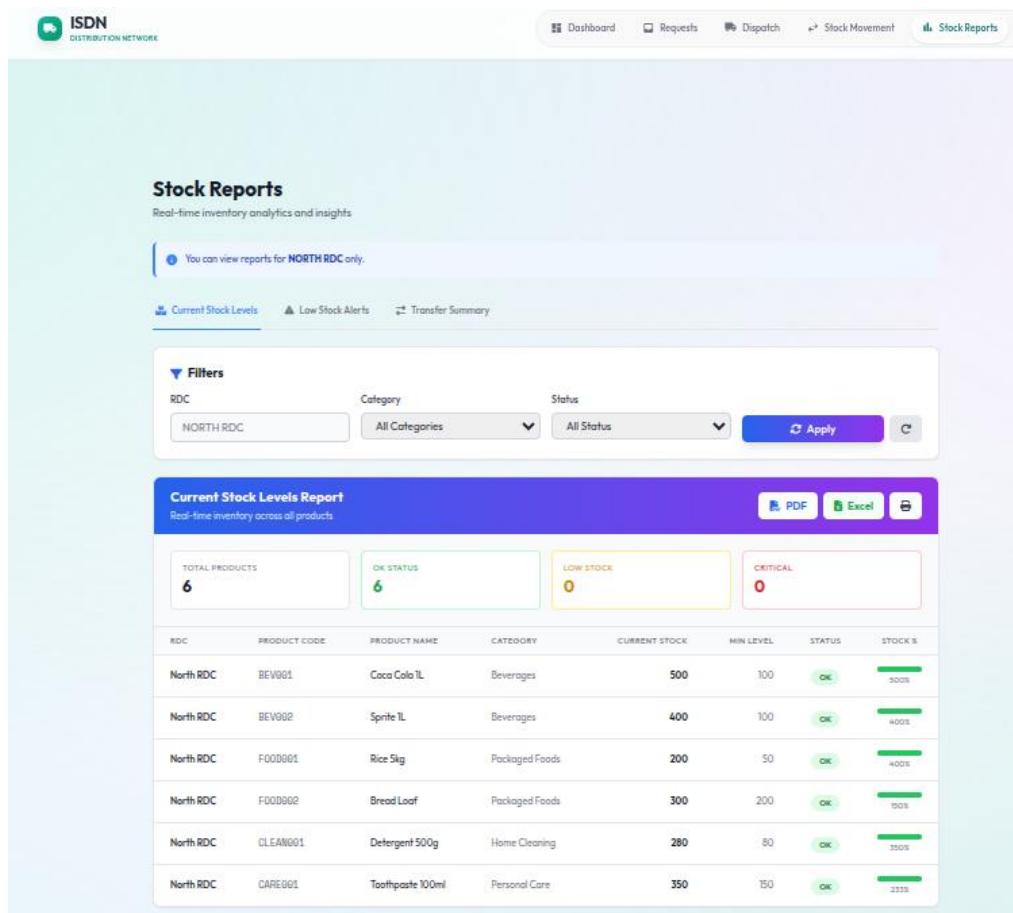


Figure 51 Stock reports dashboard

The Stock Reports page is a specialized analytics dashboard of the RDC Manager, which shows important information with the strict scope of the inventory of the local Regional Distribution

Centre. It is easy to access three main operational reports namely Current Stock Levels, Low Stock Alerts and Transfer Summary. The Current Stock Levels report provides an immediate view of all the products in the definite RDC in the form of visual status indicators, which help to see the availability at a glance. The Low Stock Alerts report is used to offer instant focus on the items that are below the reorder point, allowing items to be replenished before they become out of stock. The Transfer Summary report monitors the quantity and condition of incoming and outgoing goods flow, thus, the transparency of inter-branch logistics is guaranteed. To enable more detailed analysis, the interface contains strong category and status based filtering as well as the ability to produce any report in PDF or Excel format to read elsewhere or hold meetings.

RDC Clerk

Dashboard

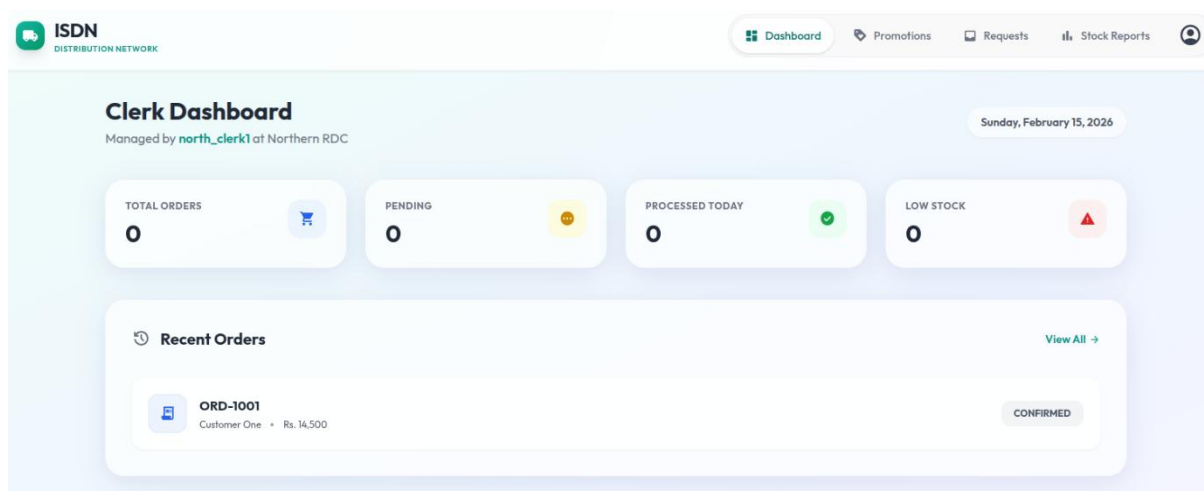
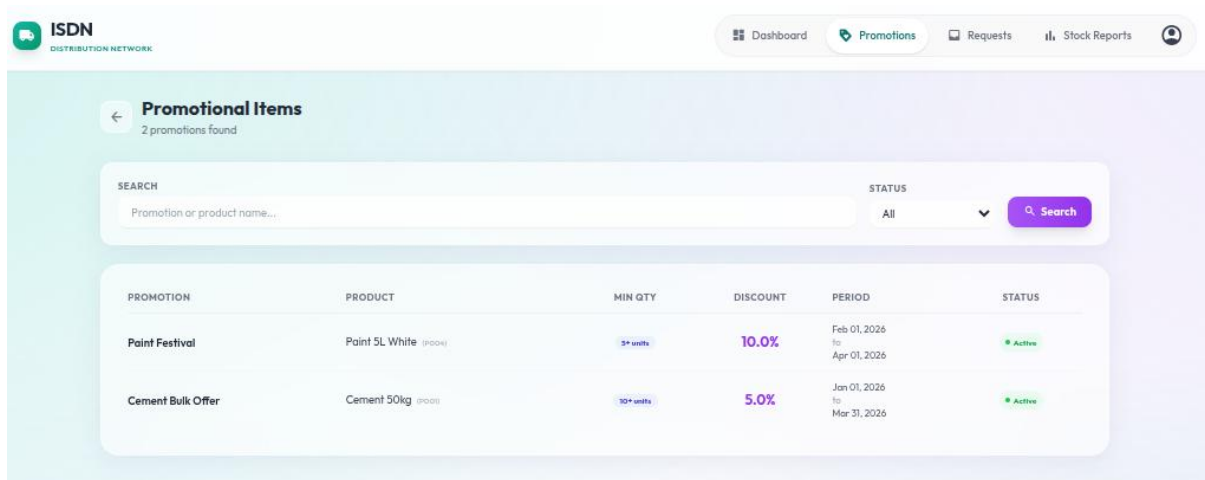


Figure 52 RDC Clerk dashboard

RDC Clerk Dashboard is the main working area that provides the direct, immediate picture of the operational priorities as well as the performance indicators of the Clerk. It is characterized by four crucial Key Performance Indicators (KPIs) that help the clerk get through the day: Total Order Count, which indicates the total number of requests a clerk deals with on a daily basis;

Pending Order Count, which shows how many requests the clerk works with at a given day; Processed Today Order Count, which shows the number of orders done by the clerk in a day; and Low Stock Count, which serves as a warning signifying a near-death state of a part of inventory. Under these metrics, the dashboard gives an actionable list of Recent Orders, which the Clerk can use to check the progress of the recently requested or processed orders. This list gives the instant access to the detailed information because the Clerk can move to the particular order, and see the complete picture of the items, quantities, and the status in which the stock replenishment request is.

Promotions



ISDN
DISTRIBUTION NETWORK

Dashboard Promotions Requests Stock Reports

Promotional Items
2 promotions found

SEARCH
Promotion or product name...

STATUS
All

Search

PROMOTION	PRODUCT	MIN QTY	DISCOUNT	PERIOD	STATUS
Paint Festival	Paint 5L White (POOL)	5+ units	10.0%	Feb 01, 2026 to Apr 01, 2026	Active
Cement Bulk Offer	Cement 50kg (POOL)	10+ units	5.0%	Jan 01, 2026 to Mar 31, 2026	Active

Figure 53 Promotions view page

Promotions interface is where RDC Clerk can keep the awareness of all current and future product promotions. This attribute is essential in predicting demand fluctuations of stock with regards to the events of sales. The main page will present the full list of the ongoing and planned promotions. The system has robust search and filtering functionality to enable fast access to information. The search bar gives clerks the ability to search by product name and the status filter allows them to reduce the list to individual categories, such as Active now (promotions that are currently running), Upcoming (promotions that will run in the future) and Expired.

This specific sight assists the RDC Clerk to anticipate the rise in picking and packing quantity and they are made aware of the products which might need closer attention to the inventory.

Request Transfer

The screenshot displays the 'Request Transfer Request' interface within the ISDN system. At the top, there's a navigation bar with 'Dashboard', 'Promotions', 'Requests', and 'Stock Reports'. The main section is titled 'Request Transfer Request' and shows a list of requests with columns for ID, From RDC, To RDC, Status, and Actions. Below this, there's a 'Request Product Units' section with a summary of stock levels (Low Stock: 5, Critical Stock: 1, Pending Requests: 3) and a 'Check Stock at Other RDCs' section with a dropdown to select an RDC and a 'Check Stock' button. The 'Request Product Units' section also lists products like Coca Cola 1L, Sprite 1L, Rice Slog, Bread Loaf, and Toothpaste 100ml, each with a 'Request Quantity' input field. At the bottom, there's a 'Submit Transfer Request' form with a 'Request From RDC' dropdown, a 'Priority' selector (Normal or Urgent), a 'Reason for Request' text area, an 'Estimated Delivery Time' of 2-3 Business Days, and a 'Submit Transfer Request' button.

Figure 54 Request transfer units view of RDC Clerk

Request Transfer interface gives the authority to RDC Clerks, to actively detect the gaps in inventory and formulate stock replenishment requests. The dashboard shows a tracking list of the latest activities at the top of the dashboard and is supported by such visibility icons as Low Stock and Critical Stock counts, which help the Clerk make the ordering choices. Before initiation, the system enables product availability can be thoroughly checked throughout the network to identify the most optimal source RDC. The main functionality makes the bulk selection of products easier, as it needs assigning an Urgent or Normal priority level and comments. In this position, unlike any other, the form is not submitted and the order is sent at once to the external branch but rather the request is stored with a pending status, which reminds the local RDC Manager that the order must be reviewed. It is not until the Manager confirms the order that they change status to Pending, and can be seen by the supplying RDC to be filled.

Stock Reports

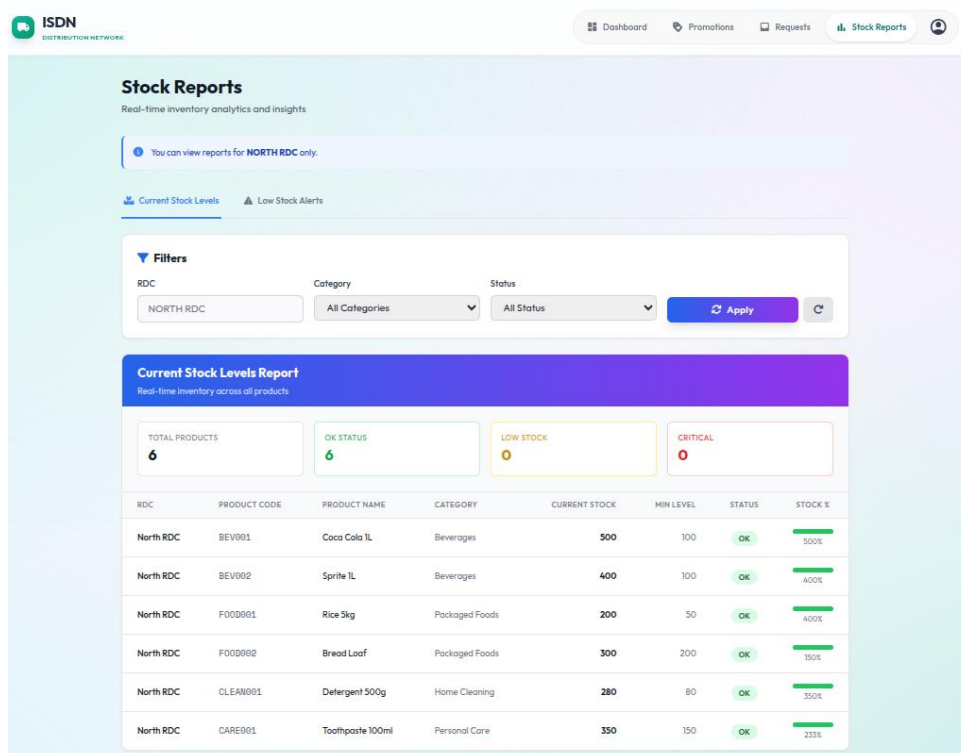


Figure 55 Stock reports dashboard view of RDC Clerk

The RDC Clerk interface of the Stock Reports is simplified to implement only the daily operational inventory requirements with a strict access control to two key report categories Current Stock Levels and Low Stock Alerts. The visibility of data is restricted to its local Regional Distribution Centre only and as such, the clerks only see and deal with only inventory which is pertinent to the picking and packing tasks at hand. The Current Stock Levels report will give an up-to-date inventory of all items on the floor which will help in making instant stock checks and also in filling orders. At the same time, the Low Stock Alerts report is an essential monitoring solution that will automatically mark those items that have already dropped below the minimum level to ensure that it receives a higher priority in the reorder requests. The system limits access to more comprehensive transfer summaries or financial values data to ensure that the RDC Clerk is not exposed to the sensitive network-wide logistics data.

Chapter 3 – Implementation

3. Implementation

This chapter describes the implementation of the IslandLink Sales Distribution Network (ISDN) system. It covers the system overview, the features implemented, the workflow of the system, and the challenges faced during development. The system replaces manual operations with a centralized, web-based solution that enhances operational efficiency, data security, and process automation.

3.1 Implementation Overview

Reference to the Project Implementation

The IslandLink Sales Distribution Network (ISDN) system was implemented as a centralized, web-based sales distribution management platform using PHP as the server-side scripting language and MySQL as the relational database management system. The implementation replaces ISDN's legacy manual operations which used to depend on Excel spreadsheets and phone calls and paper documentation with a software solution that provides structured and scalable and transaction-based operations.

The system uses a three-tier architectural model which creates distinct functional areas that make maintenance easier. The presentation layer consists of HTML, CSS, and JavaScript interfaces rendered dynamically through PHP, enabling responsive access for retailers, RDC staff, logistics personnel, and management users. The business logic layer uses PHP scripts which define validation rules and workflow operations and business policies for order processing and stock management and delivery status updates and billing calculations. The data layer operates through a central MySQL database which maintains all transactional data and master records of users and products and inventory and orders and deliveries and payments.

The application functions through an HTTP request-response cycle which enables client browsers to send requests to PHP endpoints that run on the server. Business logic components handle these requests by executing structured SQL queries to access the MySQL database. Users receive results through dynamically generated HTML pages which render information

back to them. PHP session management system maintains user state between requests while providing secure authentication and role-based access control to users.

The Head Office and all five Regional Distribution Centres (RDCs) depend on a centralized MySQL database which serves as their primary source of accurate information. The system uses ACID-compliant database transactions for its essential functions which include order placement and inventory updates and payment recording to maintain data consistency and system reliability. The system uses foreign key constraints to enforce referential integrity which prevents orphaned records and maintains correct relationships between system entities.

Development used local environments for XAMPP and WAMP while Git and GitHub provided tools for version control and collaboration and change tracking. The project used a structured file organization system which followed the Model-View-Controller (MVC) design pattern to enhance code readability and create modular components and establish long-term maintenance capabilities.

3.2 Implemented Features

New Development Approaches Adopted

The ISDN system implementation marks a major transition from existing manual operations toward contemporary software engineering practices. The new system uses established development methodologies together with architectural patterns and quality assurance techniques while the previous system depended on uncontrolled Excel files and informal workflows.

The main advancement brings better data protection together with complete data validation processes. The manual system faced several problems because Excel files could be deleted by mistake and data formats became inconsistent and duplicate records appeared. The PHP/MySQL system enforces validation requirements through two mechanisms which include server-side validation and database-level validation. MySQL data types ensure correct storage of quantities, prices, and dates, while foreign key and unique constraints prevent invalid or duplicate records. The system uses prepared statements which protect against SQL injection attacks to achieve better security measures.

The system creates centralized data management control which replaces existing RDC-specific Excel files with a single database system. The same platform provides real-time inventory and order data access to all RDCs which enables them to achieve complete stock visibility across the entire country. The system uses transactional control to handle stock deduction and order creation processes as single units which prevents errors from occurring due to incomplete system updates.

The field of process automation has achieved significant progress through new technological advancements. The system now performs business rules handling through direct coding into PHP business logic rather than executing manual processes which used to handle invoice calculation and stock availability verification and order status tracking. The system generates invoices automatically to maintain consistent billing operations while PHP scripts run on a schedule through cron jobs to perform routine functions which include low-stock alerts and daily reporting activities. Automated alerts use status-based triggers to notify users about order dispatch and delivery events.

Role-Based Access Control (RBAC) improves both security and access control protection to a significant degree. System access rights for each user are determined by their assigned role which includes retailer, RDC staff member, logistics officer and manager. The system uses hashed passwords for authentication protection while audit logs track user activities to create a system of traceability and accountability.

The project used Agile and incremental development methods which were suited for its development lifecycle needs. The project team delivered system functionality through multiple sprints which started with basic user authentication and order placement functions. The method allowed stakeholders to provide feedback at an early stage which helped to decrease implementation risks while system features could be continuously improved through actual operational requirements.

3.3 System Workflow

The system demonstrates its distinctiveness from traditional methods through its implementation of version control and its use of modular coding techniques and its application of organized testing methods. The system achieves reduced regression risks and enhanced stability through its process which tracks and evaluates changes before deployment. The system

maintains acceptable performance during data growth through database normalization indexing techniques and efficient SQL query execution.

The workflow begins with user authentication, followed by role-based access that directs users to the functionalities relevant to their role. Retailers place orders, which are validated and stored in the centralized database. Inventory is automatically updated, RDC staff review and approve orders, logistics personnel dispatch items, and automated invoices and notifications are generated. Throughout the workflow, business rules, data validation, and transactional control ensure accurate, reliable, and consistent operations across all RDCs.

3.4 Implementation Challenges

Overall, the ISDN system demonstrates a new development method through its transition from informal manual work to its implementation of formal engineering processes. The combination of PHP and MySQL in a three-tier framework together with Agile methods and centralized data management creates a platform that ISDN can use to expand its operations across the country while maintaining security and operational efficiency.

During implementation, several challenges were encountered, including transitioning from manual Excel-based processes to a centralized database, ensuring data accuracy across multiple RDCs, managing role-based access securely, and maintaining system performance during data growth. These challenges were mitigated using validation mechanisms, transactional control, modular development, and incremental Agile sprints.

4. Testing

4.1 Testing Overview

The Testing Phase of IslandLink Sales Distribution Network System Software Development Life Cycle for the was conducted to ensure that the implemented solution satisfies functional and non-functional requirements defined during requirement analysis phases. The system required multi role operations including Head Office Administration, Regional Distribution Center staff management, Sales Representatives, and Retail Customers.

The testing activities were structured including System Testing, Integration Testing, and User Acceptance Testing methods. The primary testing strategy adopted for this phase was **Manual Black Box Testing**, which supported by structured test case documentation and a formal UAT questionnaire.

4.2 Testing Objectives

The primary objectives of the testing phase were to validate that all functional requirements were implemented correctly and to verify proper interaction among system modules. Particular attention was given to ensuring data integrity across transactional processes such as order placement, stock updates, and invoice generation. The testing phase also aimed to evaluate the reliability, usability, and responsiveness of the system while confirming that stakeholder expectations were satisfied prior to deployment.

4.3 Testing Strategy

4.3.1 Selected Approach – Manual Black Box Testing

The ISDN system was primarily tested using the Black Box Testing approach. Under this method, the system is evaluated based on inputs and expected outputs without examining internal code structures or implementation logic. This approach was considered appropriate because the system is role-driven and contains multiple medium and complex functionalities. Validation of business workflows, including login, product search, stock transfer, order

placement, and report generation, is largely output-oriented and focused on user-level interactions.

System reliability, user experience, and usability were identified as critical success factors; therefore, validation was required at the business functionality level rather than through code-level inspection. Black Box Testing does not require knowledge of programming languages such as PHP or database queries, as the focus remains on requirement verification and functional validation from an end-user perspective.

The applied black box methodology emphasized requirement-based test case development, input validation testing, comparison of outputs against expected results, validation of error handling mechanisms, and verification of role-based access controls.

4.4 Unit Testing

Unit testing is the foundation of our testing process. In this stage, the isolate each individual component or "unit" of the **IslandLink Sales Distribution Network (ISDN)** to verify that the internal logic is correct. By testing units in isolation, ensure that if a bug occurs later, it isn't coming from these basic building blocks.

4.4.1 The Objective of Unit Testing

The primary goal is to validate that each small piece of code performs exactly as expected according to the functional requirements. For example, when a clerk enters order details at an RDC, the unit test ensures the data entry logic works before it even touches the main database.

4.4.2 The Unit Testing Process

The following structured approach was applied during this phase:

1. **Isolation of Components:** Each function, such as `calculateTax()` or `validateLogin()`, was tested independently to ensure that the logic within that specific block of code is accurate.

2. **Edge Case Identification:** Testing scenarios were designed to handle not only normal data but also "edge cases," such as empty input fields, invalid characters, or out-of-range numerical values (ex: negative stock levels).
3. **Error Handling:** The system's ability to catch and manage errors gracefully was verified, ensuring that incorrect inputs do not cause the entire application to crash.

4.4.3 Key Areas Tested

Within the implemented modules, the focus remained on:

1. **Authentication Logic:** Verifying that the login function correctly validates user credentials and redirects users based on their specific roles, such as Retail Customer or RDC Staff.
2. **Product Management:** Ensuring that the product catalog correctly displays data like prices and stock levels retrieved from the centralized inventory.
3. **Data Integrity:** Validating that input forms (ex: the order placement form) strictly follow the required data formats before the information is processed.

4.4.4 Significant Benefits

1. **Early Bug Detection:** Identifying logical errors at the unit level is more cost-effective and faster than fixing them during later stages of development.
2. **Simplified Debugging:** When a unit test fails, the exact location of the error is immediately clear, making the debugging process much more efficient.
3. **Foundation for Integration:** A solid suite of passed unit tests provides the necessary stability required to proceed to **Integration Testing**.

4.5 Integration Testing

Integration testing was conducted to verify interactions among interconnected system components. These included the authentication module and role-based dashboards, the order module and inventory management, the promotion module and pricing calculation mechanisms, and the stock transfer process integrated with inventory updates. Additionally, invoice and email generation processes were tested in conjunction with order confirmation, while the mobile responsive interface was evaluated against backend processing logic.

The integration test cases validated correct data flow between modules and ensured proper updates of stock levels following both order confirmation and stock transfer processes. The accurate application of promotional discounts and correct status transitions from Pending to Delivered were also verified. Furthermore, consistency of order records across the database and generated reports was examined. This process ensured that individual modules functioned cohesively during system-wide execution.

4.5.1 The Objective of Integration Testing

While unit testing ensures each function works independently, integration testing focuses on the flow of data across the entire system. It verifies that the interfaces between modules are functioning correctly and that the centralized database remains consistent when multiple operations occur simultaneously.

4.5.2 Integration Strategy

A structured approach was followed to ensure seamless communication between system components:

1. **Interface Testing:** Verification of data exchange between the User Management and Role-Based Access Control (RBAC) modules to ensure secure access to the system.
2. **Database Connectivity:** Ensuring that when a customer places an order, the Inventory Synchronisation module automatically updates stock levels across all Regional Distribution Centres (RDCs).

3. **End-to-End Workflow:** Testing the complete process from order placement to delivery scheduling and digital invoice generation.

4.5.3 Key Modules Integrated

The integration phase focused on the following critical connections:

1. **Orders & Inventory:** Ensuring that product availability is checked in real-time before an order is confirmed, replacing the previous manual spreadsheet process.
2. **Distribution & GPS Tracking:** Verifying that delivery routes assigned by logistics officers are correctly visible on the real-time tracking dashboard for both staff and customers.
3. **Sales & Reporting:** Ensuring that all transaction data flows correctly into the Management Dashboard to provide real-time analytics and KPI tracking.

4.5.4 Expected Outcomes

1. **Data Consistency:** Information remains accurate as it moves from one module to another (ex:- from the order portal to the payment gateway).
2. **System Reliability:** The centralized system can handle multiple RDC operations without delays or data conflicts.
3. **Improved Visibility:** Ensuring that retailers and managers have full transparency over shipments and stock movements.

4.6 System Testing

System testing evaluated the complete and fully integrated ISDN application in an environment that simulated real operational conditions. The prepared system test cases covered user authentication and authorization, product management, search and filtering functionality, cart management, order processing workflow, stock transfer workflow, payment and invoice generation, order delivery tracking, report generation, error handling, and mobile responsiveness.

Both positive and negative test scenarios were executed. These included validation of correct login credentials, handling of invalid password attempts, testing order quantities exceeding maximum limits, empty input field validation, behavior under simulated network delays, and scenarios involving a high number of user interactions. System testing confirmed that the fully integrated application operated in accordance with the Software Requirement Specification established during the requirement analysis phase.

4.6.1 Scope of System Testing

This phase covers both the visible features and the underlying performance of the system:

1. **Functional Testing:** Verifying that all core features, such as real-time inventory synchronisation, GPS tracking, and automated billing, work according to the design specifications.
2. **End-to-End Testing:** Simulating a complete transaction where a customer browses products, places an order, receives an instant confirmation, and tracks the delivery in real-time.
3. **User Interface (UI) Testing:** Ensuring the web and mobile-compatible interfaces are user-friendly and consistent for all roles, including RDC staff and logistics teams.

4. **Security & Access Control:** Validating the Role-Based Access Control (RBAC) to ensure that only authorized personnel can access sensitive financial reports or strategic procurement data.

4.6.2 Non-Functional Aspects

System testing also evaluates the system's performance under various conditions:

1. **Performance:** Ensuring the management dashboard can load consolidated real-time analytics for all five regions without significant delays.
2. **Reliability:** Verifying that the system remains stable during peak ordering times when multiple retail outlets are accessing the portal simultaneously.
3. **Data Integrity:** Confirming that automated updates across RDCs are accurate and that no discrepancies occur between recorded and actual inventory.

4.6.3 Expected Outcomes

1. **Requirement Validation:** Confirming that the proposed solution effectively addresses the operational challenges of manual data entry and limited delivery visibility.
2. **Final Readiness:** Providing a clear indication that the software is stable, secure, and ready for deployment across the island-wide network.

TC_ID	Test Case	Test Scenario	Pre-Condition	Test Steps	Expected Result	Actual Result
001	Full Order Cycle	Verify Full Order-to-Delivery Process	System is live.	1. Login as Retailer. 2. Place order. 3. RDC dispatches. 4. Confirm delivery.	Complete process works without errors.	Pass
002	Multi-User Load	Verify that the system does not crash when many users are logged in.	50+ users active.	1. Many users login and browse at the same time.	System stays fast and stable.	Pass
003	Data Privacy	Verify the system blocks users from seeing private data of others.	User is logged in.	1. Try to view another user's private info via URL.	Access is blocked.	Pass
004	Data recovery after a sudden interruption	Verify that data is not lost if the browser is closed during a transaction.	During payment.	1. Refresh page while payment is processing.	There is no data loss or double billing.	Fail
005	Global Search	Verify that searching for a product returns results quickly.	Large database.	1. Search for a specific product name.	Results appear in less than 2 seconds.	Pass
006	Mobile Functionality Check on Admin	Verify if the system is easy to use on a small screen.	Using a smartphone.	1. Open the system on a mobile browser.	The layout adjusts correctly for mobile.	Pass
007	Admin Real-time Notification Test	Check if the Admin gets an alert immediately after an order.	New high-priority order.	1. Retailer submits a bulk order.	Admin receives an instant alert.	Fail
008	Multi-Browser Compatibility Check	Verify that the system works on Chrome, Firefox, and Edge.	Multiple browsers.	1. Open system in Chrome, Firefox, and Edge.	Works perfectly in all browsers.	Fail
009	Handling of Invalid Page URLs	Check if the system shows a 404 error for wrong links.	Invalid URL typed.	1. Type a wrong URL, like /wrong-page.	Shows a friendly "Page Not Found" message.	Fail
010	Yearly Sales Report	Verify that the system calculates annual sales correctly.	Data for 12 months.	1. Generate a Yearly Sales Report.	Total sales match monthly sums.	Pass
011	Out of Stock Block	Verify that users cannot order items that have zero stock.	Zero inventory.	1. Try to order an out-of-stock item.	"Add to Cart" button is disabled.	Pass
012	Database Integrity	Verify that deleting a product does not break old sales reports.	Product deletion.	1. Delete a product. 2. Check old order reports.	Old reports still show the product name.	Pass
013	Multiple Sessions	Verify how the system handles a user logging in from two different devices.	Same account.	1. Login from two different PCs.	System manages sessions securely.	Pass
014	Large List Scrolling	Verify that the "Order History" page loads quickly when there are many records.	500+ order history.	1. Scroll through the "All Orders" page.	Page loads smoothly using pagination.	Pass
015	Tax Calculation	Verify that the system adds the correct tax amount to the final bill.	Items with VAT.	1. Add items to cart. 2. Check final bill.	VAT/Tax is calculated accurately.	Fail
016	Help & Support	Verify that the "Contact Support" feature works correctly.	User needs help.	1. Click "Contact Support" link.	Support page/form opens correctly.	Pass
017	Password Strength	Verify that the system rejects very weak passwords during registration.	New registration.	1. Try a weak password like "abc".	System asks for a stronger password.	Pass
018	Regional Security	Verify that an RDC Manager can only see data for their specific region.	RDC Manager access.	1. Try to view orders from a different RDC.	Access is denied for other regions.	Pass
019	Email Formatting	Verify that the automated emails sent by the system are readable.	Registration email.	1. Sign up as a new user. 2. Open email.	Email looks professional with logo/text.	Pass
020	Logout Safety	Verify that a user cannot go back to private pages after logging out.	Active session.	1. Click Logout. 2. Click browser "Back" button.	User cannot go back to private pages.	Pass
021	Product Image Load	Verify that the system shows placeholders if images load slowly.	Slow internet.	1. Load the product catalog.	Small "placeholder" shows until image loads.	Pass
022	System Date/Time	Verify that orders placed at midnight are recorded on the correct day.	Daily sales.	1. Place an order at 11:59 PM.	Order is recorded on the correct date.	Pass
023	Bulk Edit (Admin)	Verify that the Admin can change prices for many items at once.	Multiple products.	1. Select 10 items. 2. Change price for all.	All 10 items are updated successfully.	Fail

Figure 56 Test cases

024	Auto-Refresh	Verify that the Admin dashboard updates without a manual page refresh.	Dashboard data.	1. Keep Admin dashboard open. 2. Place order.	Dashboard updates without a manual refresh.	Pass
025	User Onboarding	Verify that new users see a "Welcome Guide" upon their first login.	First-time login.	1. Login as a new user.	System shows a "Welcome" or setup guide.	Pass
026	Inventory Audit	Verify that the system stock matches the actual physical count.	Stock mismatch.	1. Compare physical stock with system stock.	System data matches physical stock.	Pass
027	Export to PDF	Verify that the "Export to PDF" button creates a readable file.	Sales report.	1. Click "Export as PDF" on any report.	A clean PDF file is downloaded.	Pass
028	Session Expiry	Verify that the system logs out a user if they are away for 1 hour.	Idle for 1 hour.	1. Stay idle for 60 minutes. 2. Click any link.	Forced logout for security.	Pass
029	Duplicate Email	Verify that two people cannot sign up with the same email address.	Registration.	1. Try to register with an existing email.	Error: "Email already in use."	Fail
030	Discount Expiry	Verify that expired promo codes are rejected by the system.	Seasonal promo.	1. Use a promo code that expired yesterday.	Error: "Promo code has expired."	Pass
031	Map Navigation	Verify that the delivery map shows the correct RDC location.	Delivery tracking.	1. Click "Track Delivery" map.	Map shows the correct RDC location.	Pass
032	Admin Audit Log	Verify that the system records when an Admin deletes a user.	Critical changes.	1. Admin deletes a user. 2. Check Audit Logs.	Action is recorded with date and name.	Pass
033	Font Size/Scaling	Verify that the system layout doesn't break when zoomed in.	Browser zoom.	1. Zoom browser to 150%.	UI components stay readable and clean.	Fail
034	Database Search	Verify that the search engine finds items with incomplete names.	Partial name search.	1. Search for "Orange" instead of "Orange".	System finds all orange-related products.	Pass
035	Order Return Flow	Verify the full process of returning a damaged item.	Damaged goods.	1. Request return. 2. RDC confirms. 3. Stock updates.	The whole return flow works correctly.	Pass
036	Notification Sound	Verify that a sound alert plays for the Admin during a new sale.	Desktop alerts.	1. Keep system in background. 2. Place order.	Admin hears an alert sound for new orders.	Pass
037	Data Backup	Verify that data can be restored if the system crashes.	System failure test.	1. Restore data from yesterday's backup.	All data is recovered successfully.	Pass
038	API Integration	Verify that the system connects to the bank's payment page correctly.	External payment.	1. Use a test credit card for payment.	Payment gateway communicates correctly.	Pass
039	Session Cleanup After Browser Close	Verify that the server clears a session after the user closes the tabs.	Browser close.	1. Close browser without logging out.	Session is cleared safely by the server.	Pass

Status	Count
Pass	31
Fail	3

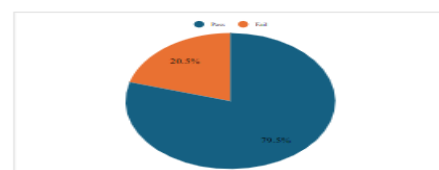


Figure 57 Test cases

Reference-

<https://docs.google.com/spreadsheets/d/163Wdr-kH0Ls37qFQc4BPJM7aCkonwuo6/edit?usp=sharing&ouid=106901725103800812807&rtpo f=true&sd=true>

4.7 Test Case Design Techniques

To ensure systematic and comprehensive test coverage, formal test case design techniques were employed.

4.7.1 Boundary Value Analysis

Boundary Value Analysis, a black box testing technique, focuses on testing input values at their extreme limits. Within the ISDN system, boundary testing was applied to functionalities such as minimum stock levels, order quantity limits, discount percentage ranges, stock transfer limits, login password length requirements, and various numeric input fields.

For example, if the permissible order quantity for a product ranged between 1 and 1000 units, test cases were designed to include values below, at, and above these boundaries. Such values included 0, 1, 2, 999, 1000, and 1001. This technique is particularly effective because many defects are likely to occur at boundary conditions rather than within middle-range values.

4.7.2 Equivalent Partitioning

Equivalent Partitioning divides input data into valid and invalid partitions based on the assumption that system behavior remains consistent within each partition. For instance, if a discount percentage is permitted between 0 percent and 50 percent, this range represents the valid partition, while values less than 0, greater than 50, and non-numeric inputs represent invalid partitions.

Rather than testing every possible value, representative samples such as 25, -5, 60, and non-numeric strings were selected to evaluate system behavior. This approach reduces the total number of test cases while maintaining effective coverage. Equivalent Partitioning was applied across user input forms, numeric validation fields, date range validations, and role-based access inputs.

4.8 User Acceptance Testing

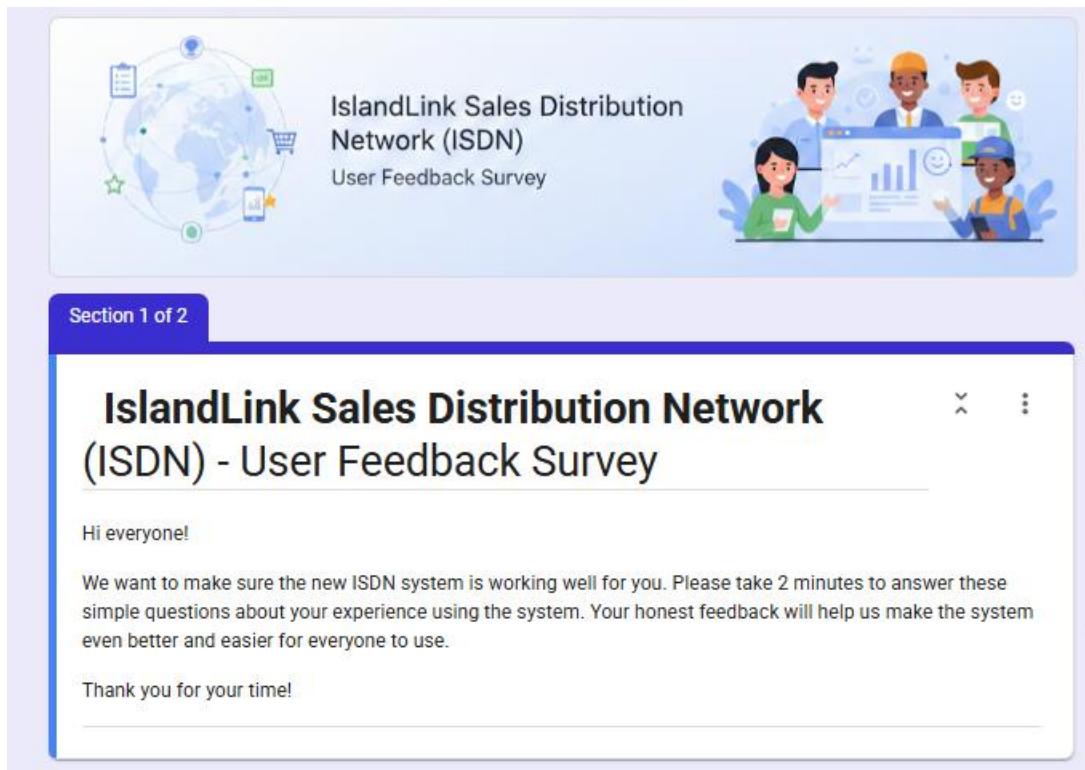
User Acceptance Testing was conducted using a structured questionnaire distributed among multiple user roles, including RDC Managers, Head Office Administrators, Sales Representatives, Retail Customers, Drivers, and Logistics Managers. The questionnaire assessed system usability, login performance, product search efficiency, order placement experience, mobile responsiveness, clarity of order status updates, readability of invoices and reports, reliability, stability, overall satisfaction, likelihood of recommendation, and potential improvement suggestions.

The use of scale-based evaluation enabled quantitative measurement of user satisfaction, while open-ended questions provided qualitative insights for future enhancements. The UAT process ensured that the system aligned with ISDN's business requirements and reflected practical usability within a real operational environment.

4.8.1 Methodology

To get feedback, the Project is created by a Google Form and sent to different users like Admins, RDC Managers, and Retailers. The questionnaire included about how easy the system is to use and if they faced any problems.

A combination of multiple-choice questions and a rating scale (1 to 5) to understand their experience clearly.



IslandLink Sales Distribution Network (ISDN)
User Feedback Survey

Section 1 of 2

IslandLink Sales Distribution Network (ISDN) - User Feedback Survey

Hi everyone!

We want to make sure the new ISDN system is working well for you. Please take 2 minutes to answer these simple questions about your experience using the system. Your honest feedback will help us make the system even better and easier for everyone to use.

Thank you for your time!

Figure 58 google form for feedback

Figure 1 Methodology

4.8.2 User Feedback Survey

The survey included questions about the login process, ordering, mobile experience, and report accuracy.

1. What is your role? *

☐ HO Admin

☐ RDC Manager

☐ Sales Representative

☐ Retailer / Shop Owner

2. Is the system easy to use? *

Very Hard 1 2 3 4 5 Very Easy

☐ ☐ ☐ ☐ ☐

3. How do you feel about the Login process? *

☐ It is very fast and easy.

☐ It is okay.

☐ It is a bit difficult/confusing.

4. Can you find the products you need quickly? *

☐ Yes, the search works great.

☐ Sometimes it takes time.

☐ No, it is hard to find items.

5. How was your experience placing an order? *

☐ Very smooth and simple.

☐ It's fine, but could be better.

☐ It takes too many steps.

Figure 59 google form for feedback

[https-
//docs.google.com/forms/d/e/1FAIpQLSdP0TUEc6IOt7gA1_14QUWUMrlw1_P2icW3juEvF
aXDwZQJTA/viewform?usp=header](https://docs.google.com/forms/d/e/1FAIpQLSdP0TUEc6IOt7gA1_14QUWUMrlw1_P2icW3juEvFaXDwZQJTA/viewform?usp=header)

4.8.3 Results and Analysis

After the users completed the form, the results were reviewed. Most of the feedback received was very positive. Below are the summarized responses.

- **Ease of Use-** 90% of the users gave a 4 or 5 rating, saying the system is very easy to navigate.
- **Mobile Experience-** Almost all users said the system works well on their phones.
- **Order Process-** Users like that they can place an order with just a few clicks.

4.9 Test Documentation and Traceability

Comprehensive testing artifacts were prepared, including system test case documentation, integration test case documentation, the UAT questionnaire, test execution results, and defect logs. Traceability between requirements and test cases was maintained to ensure complete coverage. Each functional requirement was mapped to one or more corresponding test cases to verify systematic validation.

4.10 Defect Management

Identified defects were classified according to severity levels, including critical defects affecting core workflows, major defects impacting functional performance, minor defects related to user interface or usability concerns, and enhancement suggestions. All critical and major defects were resolved prior to final User Acceptance Testing approval, ensuring system readiness for deployment.

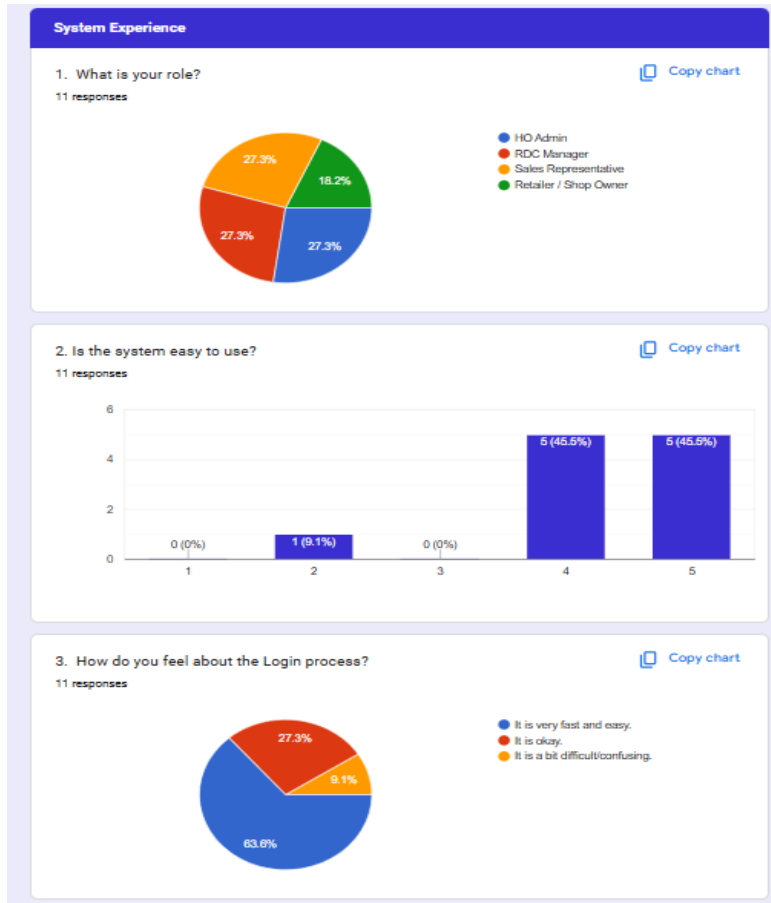


Figure 60 google form for feedback analysis

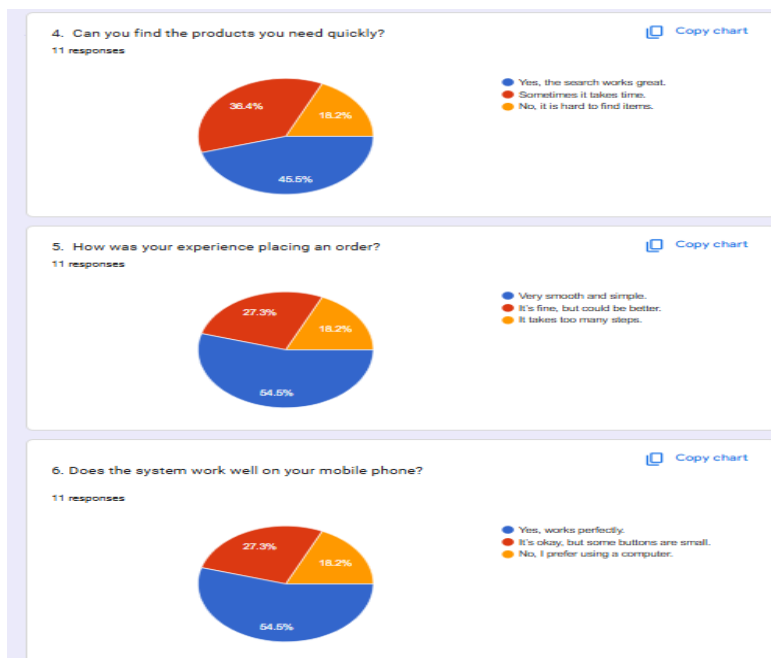


Figure 61 google form for feedback analysis

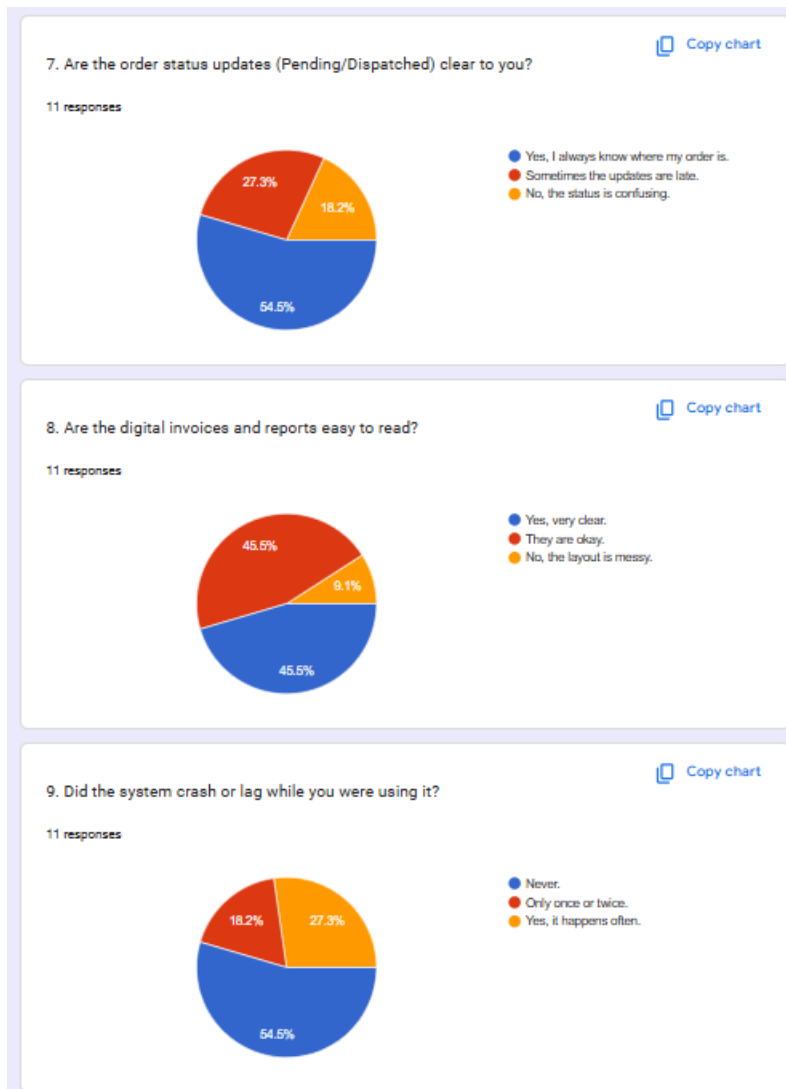


Figure 62 google form for feedback analysis

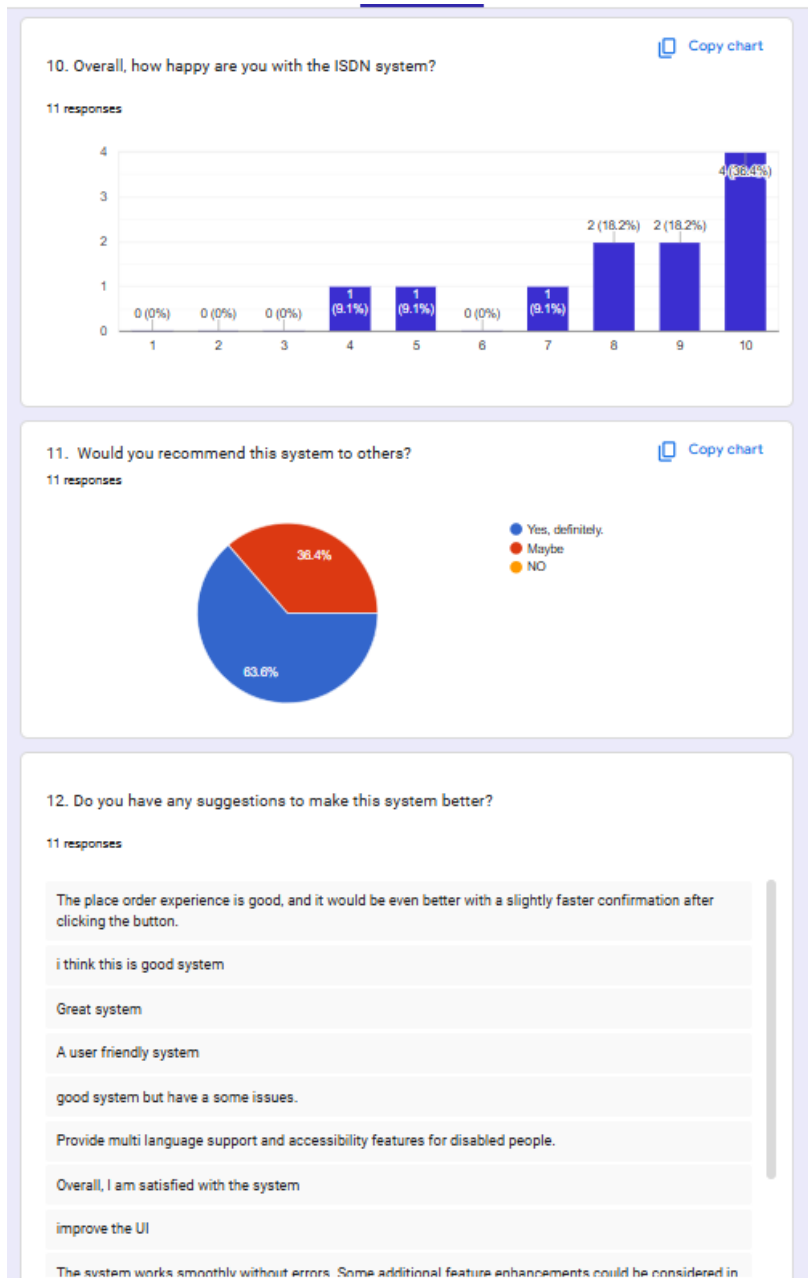


Figure 63 google form for feedback analysis

4.11 UAT Conclusion

Based on the survey results, the ISDN system is ready for real-world implementation. Users expressed satisfaction with the features and found the system highly helpful. The minor suggestions provided will be considered for future updates and improvements.

5 Conclusion

The evolution of the ISDN system was able to resolve the flaws of the current manual distribution processes with a centralized web-based platform that facilitated operations and enhanced performance in general. The system was aimed at integrating all of the major functionalities, such as order management, customer and sales officer interface, and real-time order status monitoring, into a single integrated platform. This has not only minimized the amount of redundancy and errors that occurred in the process of handling the data manually but also increased the accessibility where the user could also communicate with the system anytime and anywhere. The ISDN platform achieved a significant level of transparency, operational responsiveness and accuracy of data across the distribution network by replacing the manual system.

This project took advantage of the fact that the requirements of the system and database were well understood to design a powerful solution that fulfilled all the functional requirements. In the development process, an elaborate database schema diagram was developed through Draw.io which provided a visual representation of tables, columns and relationships to assist in clear interpretation of data flows. The user manual and system design chapter included screenshots of built modules and a detailed description of how it works so that even those who were not technical could easily navigate and use the system. These documentation activities offered a good source of training, maintenance and subsequent system upgrades.

Strict testing methodology was used to make sure that the system is reliable and performs. There was Manual testing, such as the Black Box testing, which was conducted to ensure that all the functionalities worked as anticipated by an end-user. Individual component unit testing was used to test that single components were behaving correctly when isolated, whilst automated testing was used to speed up regression testing and provided consistency between repeated test runs. Such a mix of testing plans enabled the team to detect and fix bugs early in the process and ensure that the system was stable as well as provide a high quality platform that satisfied all requirements.

The use of the Agile Scrum approach played a significant role in realizing such results. With the process of constantly developing and the team discussions and constant feedback, the project could modify itself to keep up with the changing requirements and guarantee that it was oriented to the expectations of the stakeholders. The applicability of agile practices in other

projects in the industry further confirmed its applicability in this system. The teamwork and systematic process allowed it to come up with a system that fulfilled the operational criteria and offered a scalable and user-friendly platform. Finally, the ISDN system illustrates how the use of modern software practices can be used to remove the need to use manual procedures in order to have a centralized, efficient, and reliable platform of distribution management..

6 Individual Contribution

6.1 Introduction

The author of this reflection describes his complete work for the IslandLink Sales Distribution Network (ISDN) system. The **technical implementation** of the project together with the **teamwork process** needs to be evaluated according to the coursework. The project of ISDN aimed to develop an enterprise web application which would replace manual distribution operations that used spreadsheets. The primary responsibility of the team member involved creating the **main technical components** needed for system development while providing development support to all team members.

6.2 Role and Responsibilities

- **Officially Assigned Responsibilities (Yasas - 25026764)**

The team members received their specific task responsibilities during the **first Agile sprint planning meeting**. The agreed task distribution resulted in the official assignment of core components which were completed through their implementation.:

1. **MVC Architecture & Base UI Structure Creation**

The application development process used the foundation of **MVC architecture** was structured to organize the application into models, controllers, and views. A centralized front controller, reusable UI templates (**header, footer, layout**), and **base dashboard template** were **established** to ensure modularity, clean separation of concerns, and consistent interface design across all modules.

2. **User Registration and Login – User Authentication Process**

User Authentication Process The authentication process was developed through its complete sequence which included the **secure registration** process, **bcrypt password hashing** for password protection, **login system verification**, **session management**, **role-based access control** implementation, **password reset** procedure, **login alert email system**, and **Google OAuth** authentication system.

3. **Mobile App Development & APK (PWA Implementation)**

The system was developed with Progressive Web App (**PWA**) capabilities through the implementation of **manifest file** configuration, **standalone mode** display, application icon creation, and service worker framework development which allowed **PHP** systems to operate **as mobile app installations**.

4. **HEAD_OFFICE_MANAGER Dashboard**

The Head Office Manager dashboard enables complete operational management through **revenue analysis**, **order processing** overview, **advanced search**

capabilities, performance metric tracking, and built-in visual chart display functions.

5. SYSTEM_ADMINISTRATOR Dashboard and Profile

The System Administrator dashboard was developed to show **system performance** data while allowing users to **manage accounts** and **track product status** and **audit records** while enabling them to control **administrator** account settings.

6. RDC-Wise Delivery Efficiency Report

The delivery efficiency reporting system was developed through the creation of optimized **SQL aggregation** queries which included **percentage calculations** and enabled users to **export data** in **CSV** and **PDF** formats.

7. Deployment & Hosting Setup

After the development, the system was deployed to a live hosting environment using a free shared hosting platform with a **remote MySQL database**. **Domain configuration** and **SSL(HTTPS)** were enabled to ensure secure public accessibility. This deployment simulated a real-world production setup and validated system readiness beyond local development.

8. Chapter 3 Implementation

Chapter 3 of the report was prepared to formally document the system architecture, technical design decisions, development methodologies, and implementation strategies adopted during the ISDN project.

The project team members **accomplished extra work** during the project's development phase which extended beyond their **designated tasks**. The team worked on multiple tasks which included building the system architecture and improving **security features** and creating reusable dashboard templates and establishing **email systems** for authentication processes and **developing methods** to start **databases** and building **unit testing frameworks** and managing **Git workflows** and setting up **deployment processes** and delivering **technical assistance** to team members who encountered difficulties during their work.

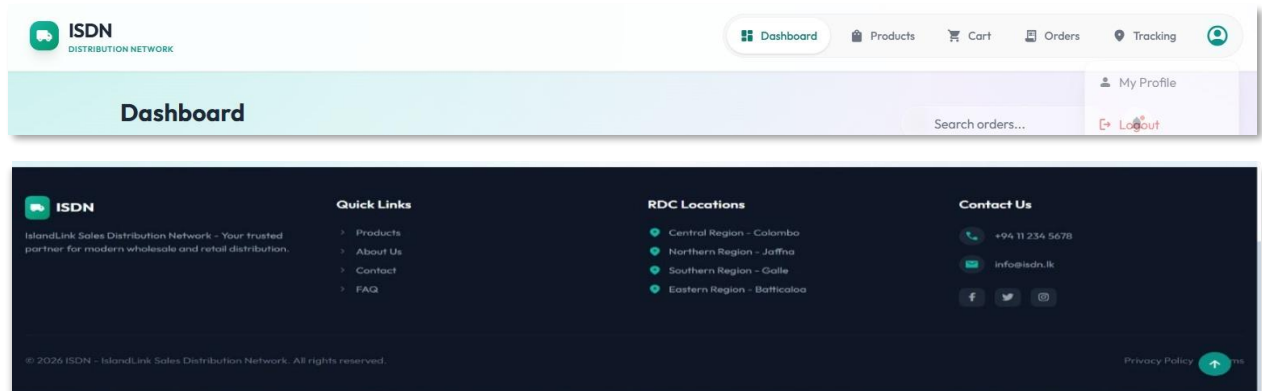
The additional work shows that the person worked on more than their **assigned project tasks** while demonstrating **active efforts** to maintain system stability and system performance and project achievement.

6.2.1 System Architecture & Project Structure Creation

The base **MVC-inspired** architecture of the system was designed and implemented using PHP during the early stages of development. The application established its structure through the use of **models**, **controllers**, **views**, and **configuration** layers which helped maintain project **transparency** and **uniformity**:

- Models (database interaction & business logic)
- Controllers (request handling & flow control)

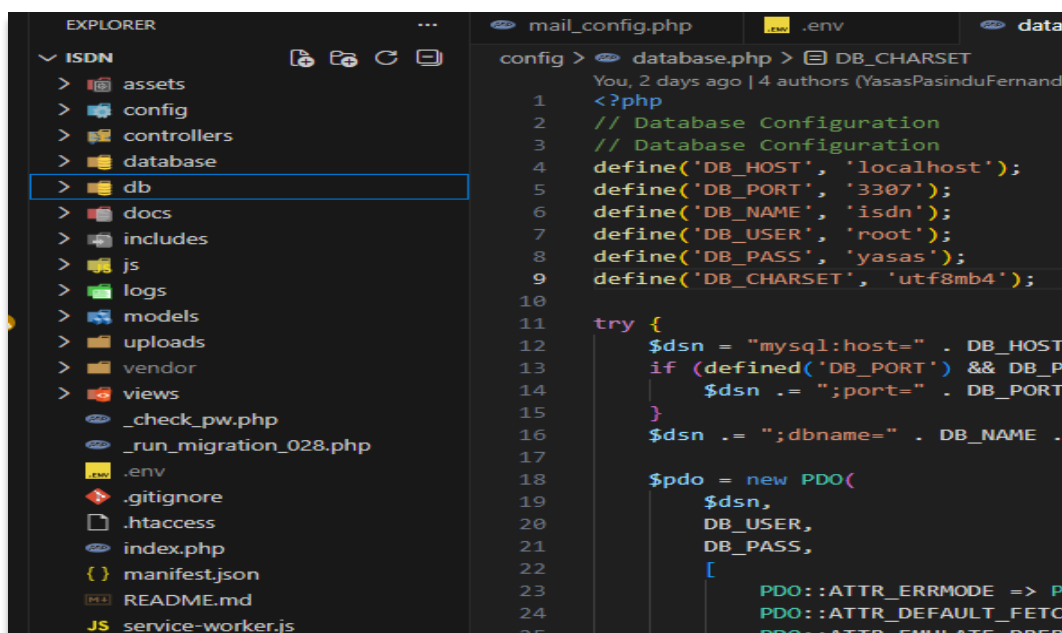
- Views (presentation layer)
- A centralized front controller (*index.php*)
- **Reusable Responsive UI templates** (header, footer, navbar, base dashboard layout)



- Configuration layer (*config.php, database.php, mail_config.functions.php, AuthController*)

The developer **created a lightweight custom framework** system which used native PHP to build a reusable MVC architecture that included structured routing and modular controller components and Tailwind CSS to create **standardized UI** components which worked on multiple devices. The system developed through this method established a clean architectural base which developers could reuse and maintain without needing **external PHP frameworks**.

The separation of concerns in this project created three advantages because it enabled teams to work **independently** while maintaining their **individual development** processes.



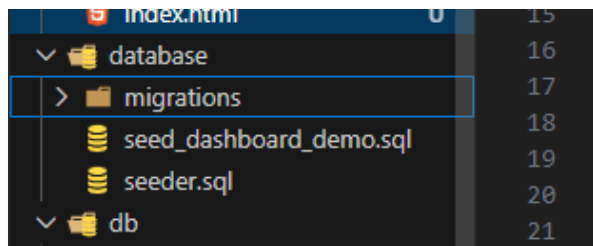
The implementation of a **three-tier** architectural model enables better maintenance and expansion capabilities because it allows separate updates to presentation components and business logic components and data storage components without disrupting system operations.

6.2.2 Database Initialization & Migration Strategy

The development process commenced immediately after MySQL schema design was completed through the **establishment of initial database structures** which required **full database design** from **another team member**. The task required **more than table creation** because of that had to establish a structured relational foundation which included primary keys and foreign key constraints and indexing and data integrity rules.

The system needed to establish a reliable and **expandable data framework** which could handle authentication needs and order processing tasks and dashboard functionalities and reporting requirements from the beginning of development work. The future database extensions could achieve seamless integration through normalization principles because they would maintain both scalability and consistency with existing database structure.

The **migration-style SQL initialization script** was **developed** to create the complete development database from one execution. The system ensured that all contributors maintained identical working conditions while it reduced potential configuration differences and it used **industry-standard database deployment** methods together with **version control database management** techniques.



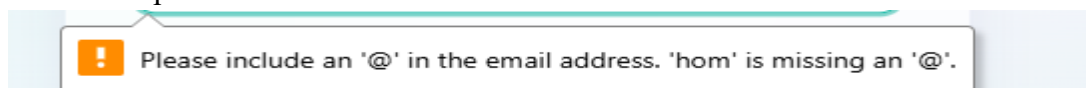
6.2.3 Authentication, Authorization & Security Engineering

A significant portion of the assigned responsibilities involved designing and implementing the complete authentication and security framework of the system. This component served as the core protection layer ensuring controlled access and data security.

Secure Registration & Login

Implemented:

- Customer-only **self-registration**
- Login with **bcrypt** hashing (*password_hash, password_verify*)
- Input validation added to prevent **Sql Attacks** and **validate inputs** while entering emails and phone numbers



- Session-based authentication
- HttpOnly **cookie** configuration
- Role-Based Access Control (**RBAC**)
- Protected **route validation** inside the **front controller**

Unauthorized users cannot access restricted dashboards, and direct URL manipulation is prevented through role validation.

The image shows two mobile app screens for the ISDN (IslandLink Sales Distribution Network) application.

Left Screen: Welcome Back!
 - Header: "Welcome Back!" with a green circular icon containing a white person silhouette.
 - Sub-header: "Login to your ISDN account".
 - Form fields: Email (sysadmin1@mail.com) and Password (masked with dots).
 - Options: "Remember me" checkbox and "Forgot Password?" link.
 - Button: "Login Now" in a green box.
 - Separator: "OR CONTINUE WITH".
 - Button: "Continue with Google" with the Google logo.
 - Footer: "Don't have an account? Register here →".
 - Bottom bar: Three icons labeled "SECURE", "FAST", and "SUPPORT".

Right Screen: Create Account
 - Header: "Create Account" with a green circular icon containing a white person silhouette and a plus sign.
 - Sub-header: "Register as a Customer on ISDN".
 - Note: "Customer Registration Only. Other user roles (RDC Staff, Managers, etc.) are created by the System Administrator."
 - Form fields: Username (yasas pasindu), Email (yasaspasindufernando@gmail.com), Password (masked with dots), and Confirm Password (masked with dots).
 - Option: "I agree to the Terms & Conditions and Privacy Policy" checkbox.
 - Button: "Create Customer Account" in a green box.
 - Separator: "OR REGISTER WITH".
 - Button: "Continue with Google" with the Google logo.

6.2.3 Email Infrastructure & SMTP Implementation for Authentication Services

Responsibility for designing and implementing the authentication-related email infrastructure was fully undertaken. This covered **automated** system-generated **emails** triggered during:

- User registration (**Welcome email**)

The image shows a mobile app screen displaying a "Welcome to ISDN" email template. The screen has a green header with the text "ISDN" and "IslandLink Sales Distribution Network".

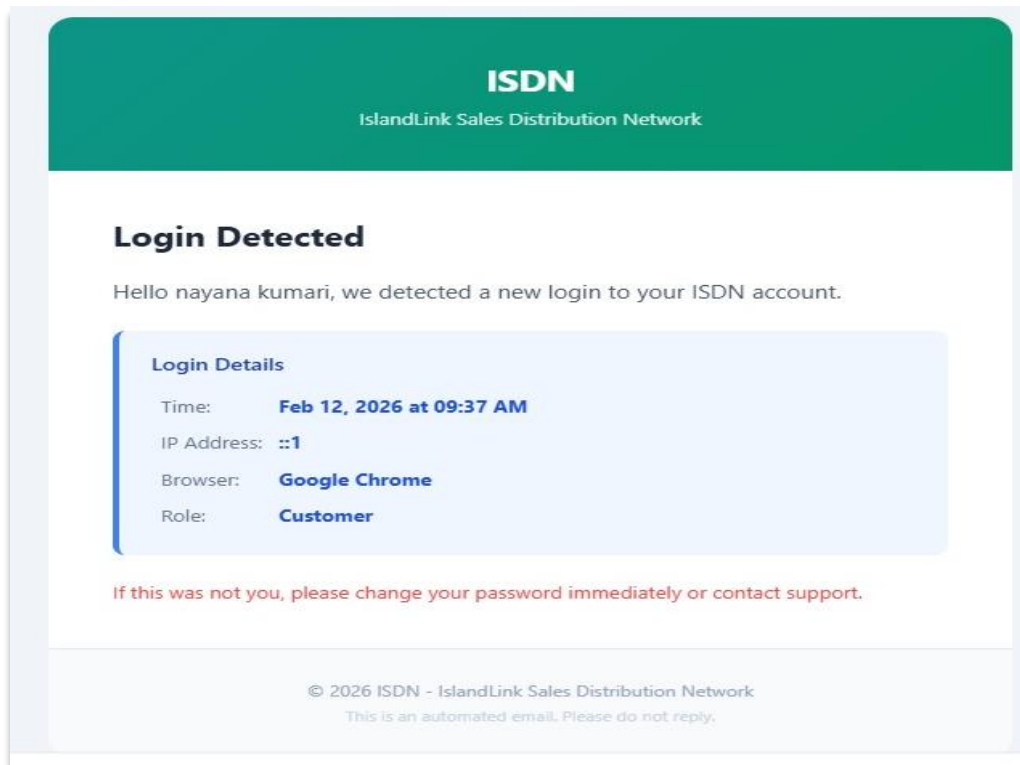
Welcome to ISDN, yasas pasindu!
 Thank you for creating your account with IslandLink Sales Distribution Network. You can now browse products, place orders, and track deliveries.

Your Account Details
 Email: yasaspasindufernando@gmail.com
 Username: yasas pasindu
 Role: Customer

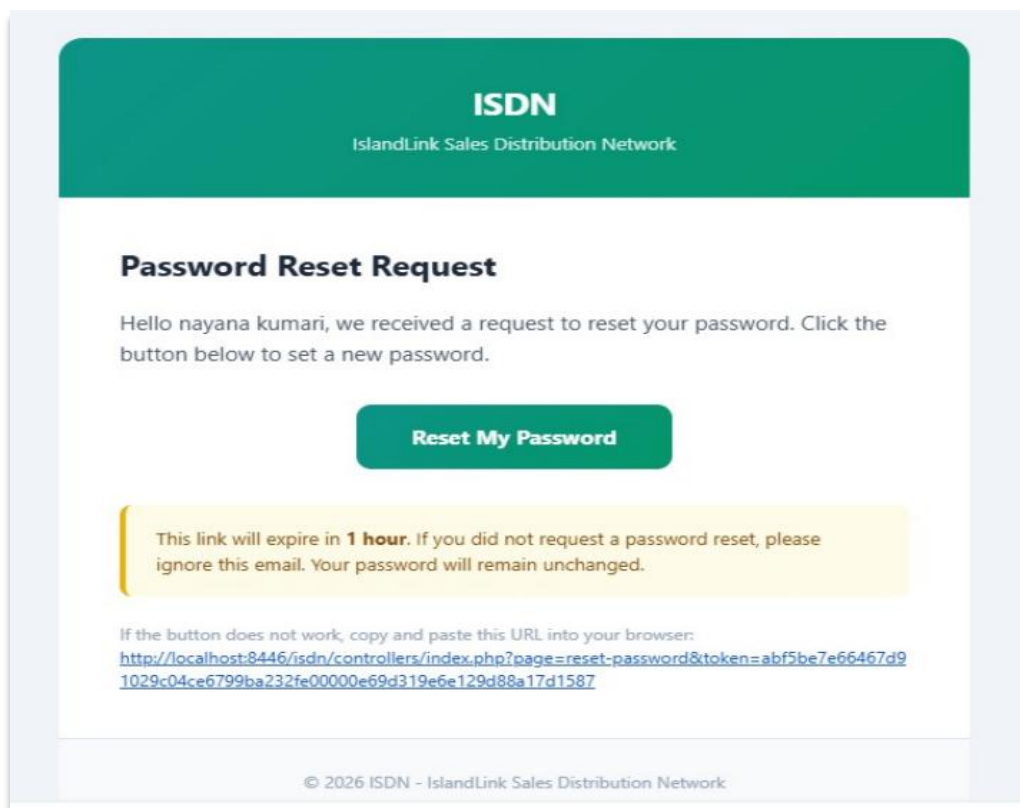
Login to Your Account

If you did not create this account, please ignore this email.

- User login (**Login notification email**)



- Password reset requests



While **SMTP** functionality for **order-related notifications** was implemented by **another team member**, the complete **SMTP-based email service** dedicated to **authentication and security workflows** was **independently** engineered.

Instead of relying solely on PHP's basic **mail()** function, a custom SMTP-based email service compatible with **Google SMTP** was implemented to ensure **secure and reliable** communication.

created:

- ***mail_config.php*** for authentication email configuration
- A custom ***SmtMailer*** class for **secure SMTP handling**
- ***mail_helper.php*** containing reusable authentication email functions
- Database-based email logging (***email_logs*** table)

SMTP Technical Implementation (Authentication Layer)

The SMTP client performs:

- Secure **SSL** connection to **Gmail SMTP** server
- **EHLO** handshake negotiation
- **AUTH LOGIN** authentication
- **Base64** credential encoding
- MAIL FROM / RCPT TO command execution
- Proper **MIME HTML** email formatting

This authentication email service operates in three modes:

- SMTP (production)
- PHP mail() fallback
- Log-only mode (development testing)

All authentication-related outgoing emails are logged with:

- Recipient email
- Email type (**welcome, login_notification, password_reset**)
- Delivery status (**sent / failed / logged**)
- Timestamp

This structured approach ensures security traceability, auditability, and reliable system-generated communication for authentication processes only.

6.2.4 Secure Password Reset System

A token-based password reset mechanism was implemented with the following safeguards:

Forgot Password?
Enter your email and we'll send you a reset link

If an account with that email exists, a password reset link has been sent.

Enter your registered email

Send Reset Link

A password reset link will be sent to your email. The link expires in **1 hour**.
Check your spam folder if you don't see the email.

Remember your password? [Back to Login →](#)

Reset Password
Enter your new password below

New Password (min 6 chars)

Confirm New Password

Reset Password

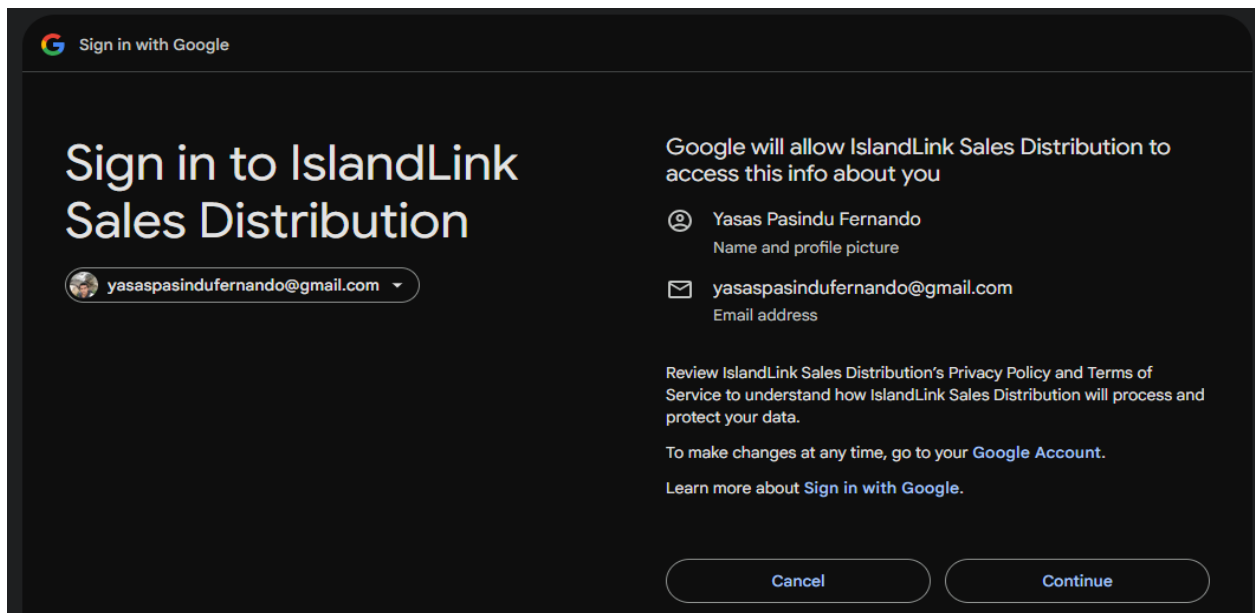
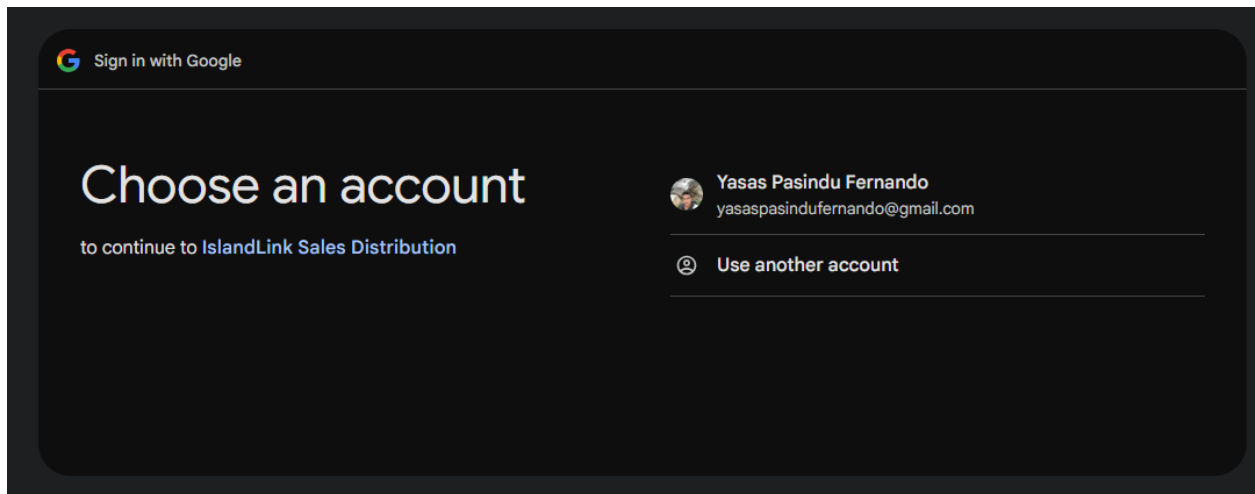
[← Back to Login](#)

- Secure token generation using *bin2hex (random bytes (32))*
- Expiry timestamp stored in database
- Validation query with **expiry check**
- **Token invalidation** after successful reset
- Password **re-hashing using bcrypt**

To prevent email enumeration attacks, the system displays the same response whether or not an email exists in the database.

6.2.5 Google OAuth Integration

Google OAuth 2.0 login functionality was **fully implemented** to modernize the authentication system and reduce dependency on password-based login.



This included:

- **Redirecting** users to **Google authentication**
- Handling secure **callback validation**
- Retrieving verified user profile data
- **Linking existing accounts** using email
- **Automatically creating new customer** accounts
- **Storing Google ID** in the database

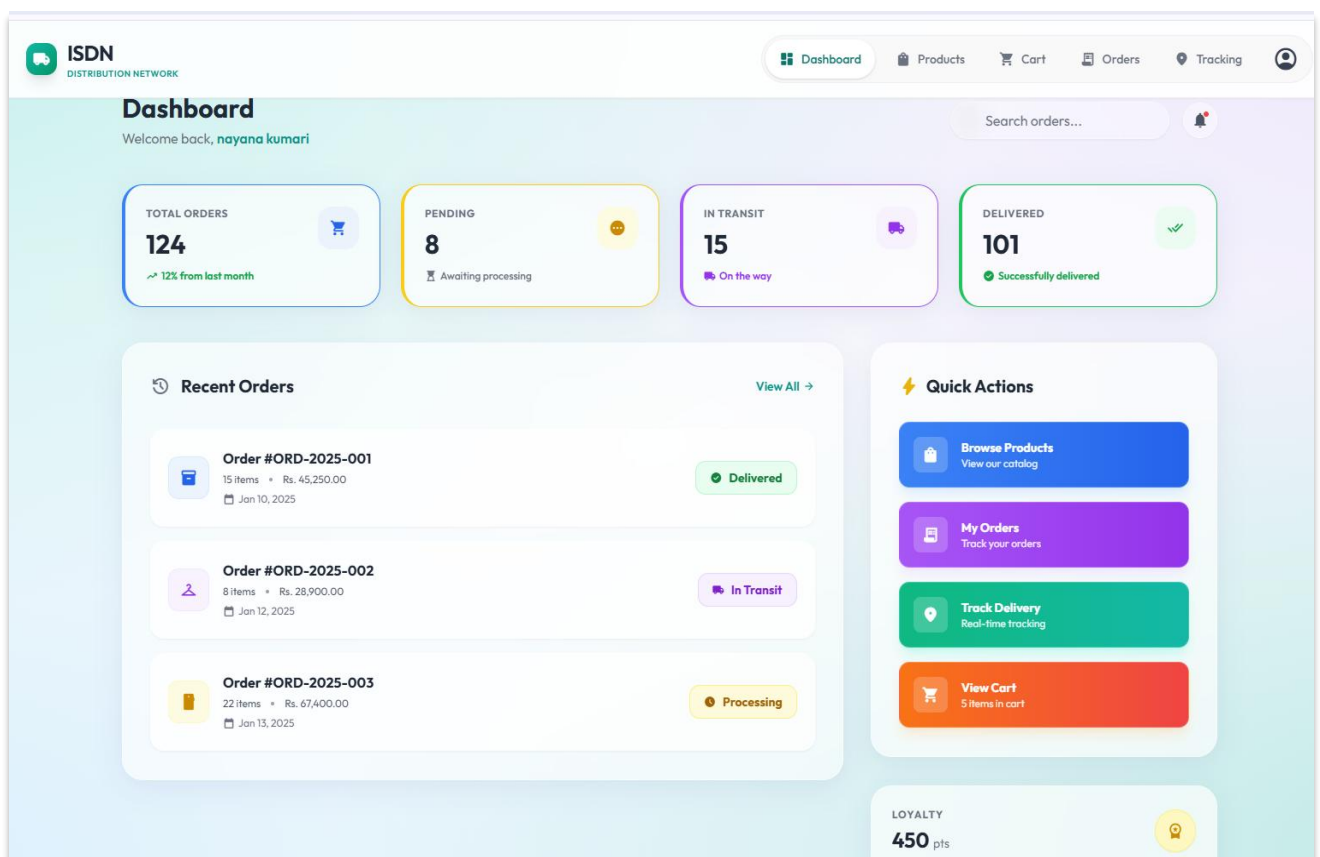
This reduces password dependency and aligns the system with modern authentication standards.

6.2.6 Role-Based Dashboards & Analytics

A **reusable dashboard** template architecture was designed and implemented to support multiple user roles within the system. **Rather than developing separate dashboards** independently, a structured base dashboard layout was established, **including a shared header, sidebar navigation, KPI card components, modular chart containers, and a responsive layout grid**. This architecture allows reuse and extension for various roles without duplication of structural code.

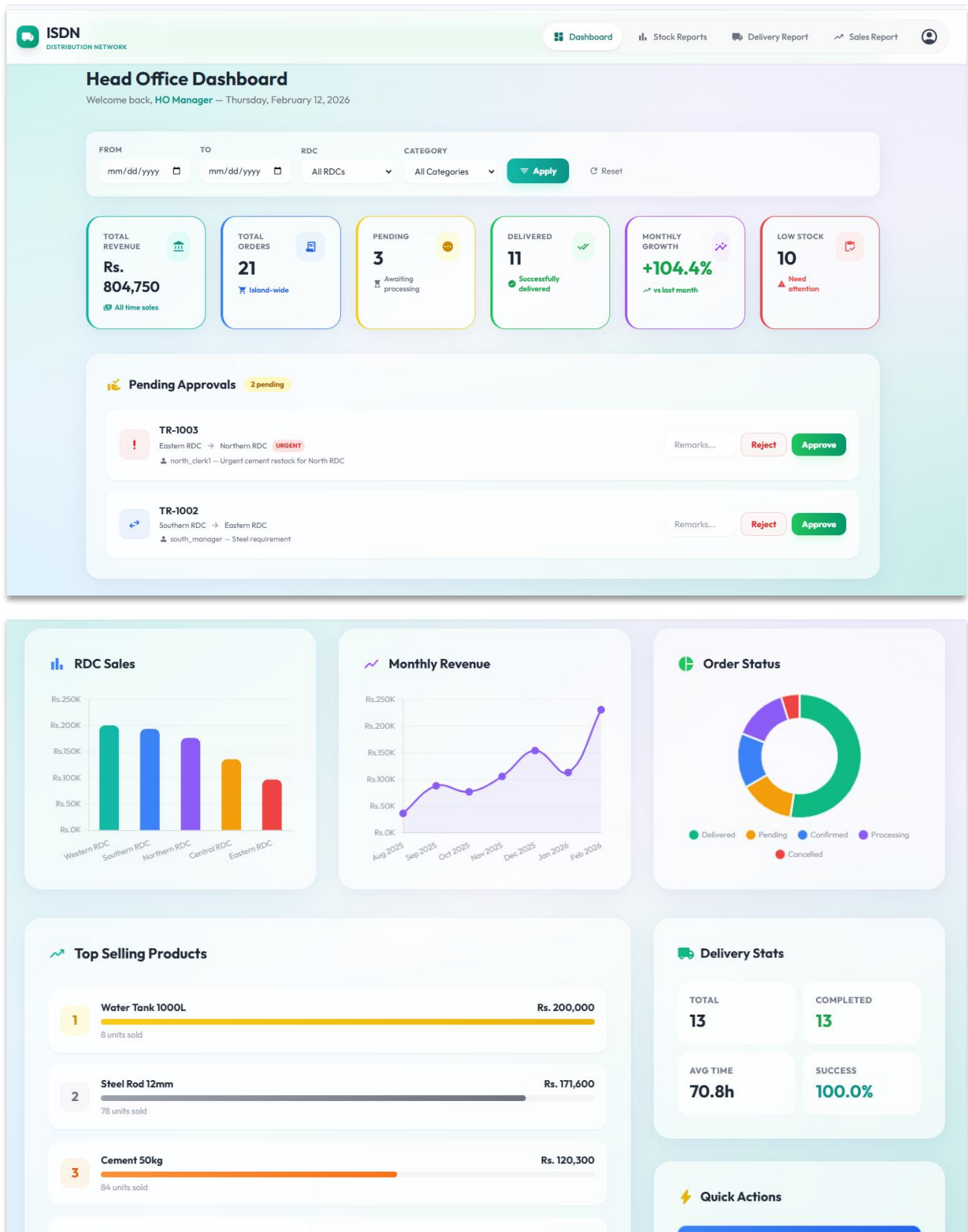
Using this reusable dashboard framework, I developed dashboards for:

- Customer dashboard-The **initial design** created the **first dashboard of the system**.

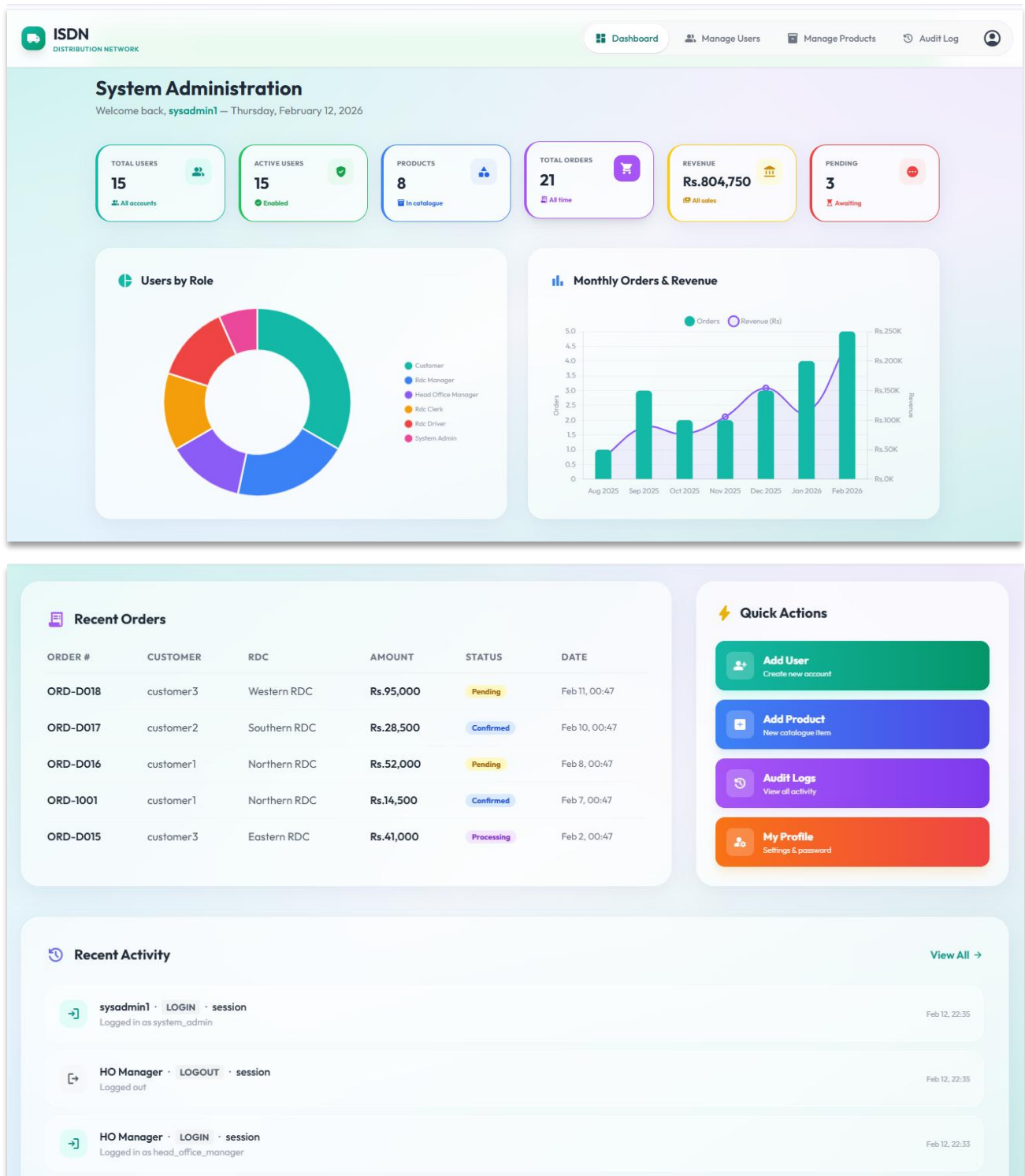


The project structure and **base user interface development** work **began** with the **creation** of the **Customer Dashboard interface**. The Customer Dashboard interface functioned as the main layout system which designers used to develop the complete dashboard design pattern with its navigation system and reusable user interface elements. The first version of the system established standard design elements for headers sidebars content **grids** and **KPI cards**. The system structure developed into a **primary template** which designers **used to create dashboards for various user roles**.

- **Head_Office Manager dashboard**



- **System_Admin dashboard**



The **core dashboard** template includes:

- Reusable **KPI summary** card components
- Modular chart containers (integrated with **Chart.js**)

- Standardized **layout grid system**
- **Role-based navigation** rendering
- Flash **messaging** and **notification** components

This design ensured consistency in **UI/UX** while reducing **code duplication**. Each role-specific dashboard inherits the base template structure but loads different datasets and modules depending on user permissions.

For example:

- The **Head Office Manager dashboard** extends the **template with revenue analytics, delivery efficiency charts, filter-based reporting**, and performance summaries.

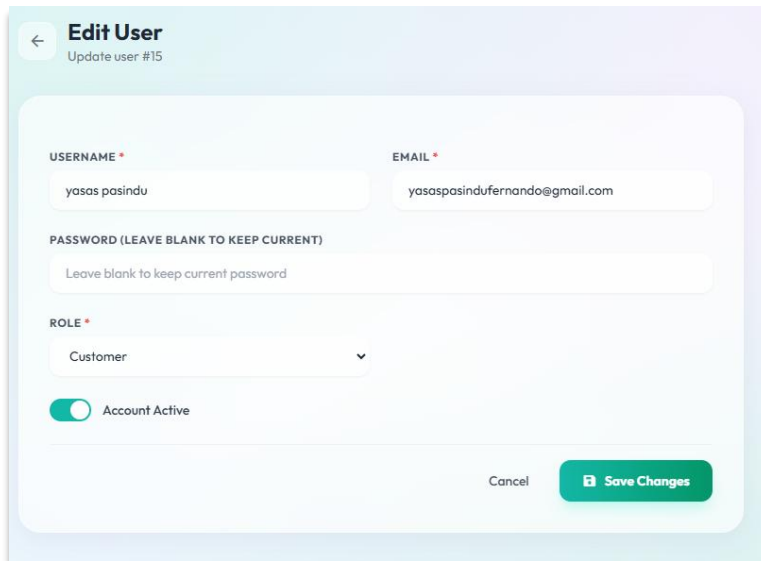


- The **System Administrator dashboard** extends the same template with **user management statistics, product management metrics**, and **audit log previews**.

The screenshot shows the 'User Management' dashboard with the following components:

- Summary Cards:**
 - 15 TOTAL USERS
 - 15 IN LIST
 - 8 roles (Customer, RDC, Admin...)
- Search and Filter:**
 - SEARCH: Username or email...
 - ROLE: All Roles (dropdown)
 - Search button
- User List Table:**

#	USER	EMAIL	ROLE	RDC	STATUS	JOINED	ACTIONS
15	yasas pasindu	yasaspasindufernando@gmail.com	Customer	-	Active	Feb 12, 2026	[Edit] [Delete] [More]
14	nayana kumari	nayanakumarimama@gmail.com	Customer	-	Active	Feb 12, 2026	[Edit] [Delete] [More]
13	HO Manager	hom@isdn.com	Head Office Manager	-	Active	Feb 12, 2026	[Edit] [Delete] [More]
12	yasas	yasasnew@gmail.com	Rdc Manager	-	Active	Feb 9, 2026	[Edit] [Delete] [More]
1	sysadmin1	sysadmin1@mail.com	System Admin	-	Active	Feb 6, 2026	[Edit] [Delete] [More]
2	headoffice1	headoffice1@mail.com	Head Office Manager	-	Active	Feb 6, 2026	[Edit] [Delete] [More]
3	north_manager	north_manager@mail.com	Rdc Manager	Northern RDC	Active	Feb 6, 2026	[Edit] [Delete] [More]
4	south_manager	south_manager@mail.com	Rdc Manager	Southern RDC	Active	Feb 6, 2026	[Edit] [Delete] [More]



Edit User
Update user #15

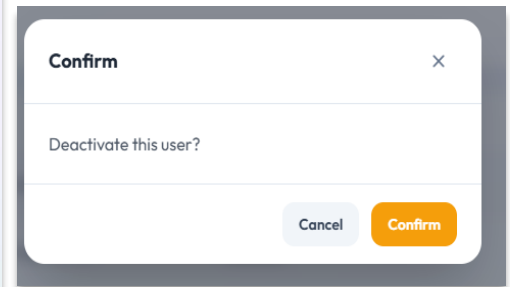
USERNAME * yasas pasindu EMAIL * yasaspasindufernando@gmail.com

PASSWORD (LEAVE BLANK TO KEEP CURRENT)
Leave blank to keep current password

ROLE * Customer

☒ Account Active

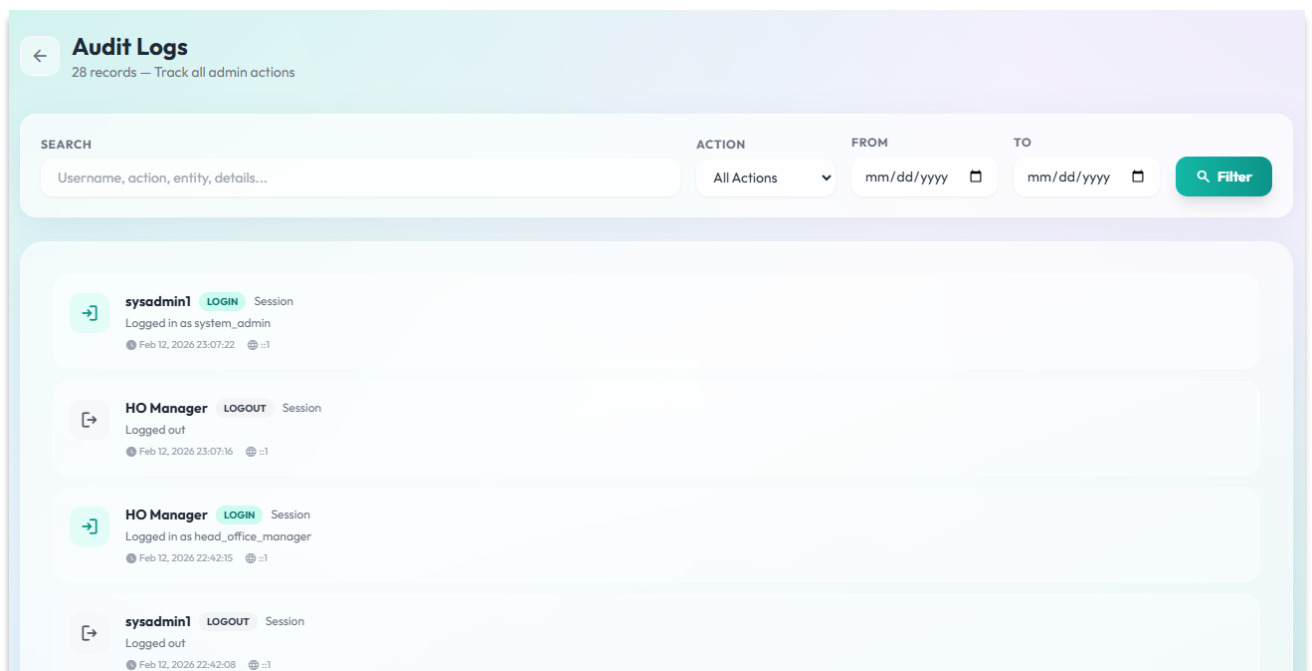
Cancel Save Changes



Confirm X

Deactivate this user?

Cancel Confirm



Audit Logs
28 records — Track all admin actions

SEARCH: Username, action, entity, details... ACTION: All Actions FROM: mm/dd/yyyy TO: mm/dd/yyyy Filter

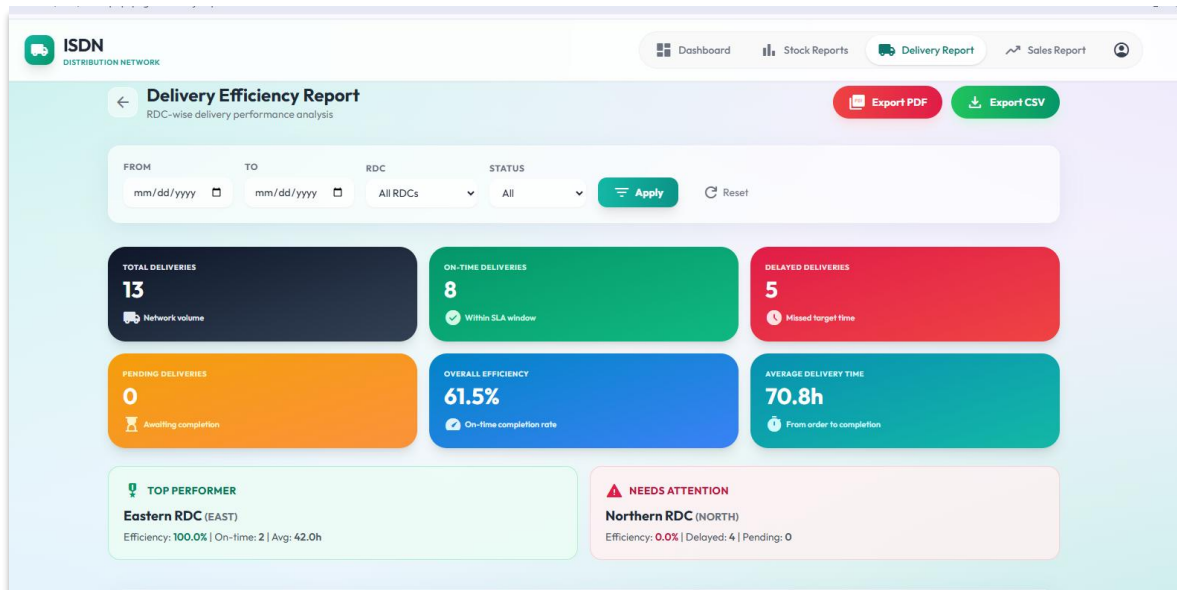
→	sysadmin1 LOGIN Session	Logged in as system_admin	Feb 12, 2026 23:07:22 1
→	HO Manager LOGOUT Session	Logged out	Feb 12, 2026 23:07:16 1
→	HO Manager LOGIN Session	Logged in as head_office_manager	Feb 12, 2026 22:42:15 1
→	sysadmin1 LOGOUT Session	Logged out	Feb 12, 2026 22:42:08 1

The **reusable dashboard architecture** system provides **better scalability and maintenance** options for **development work**. The system allows new roles to be added because existing structural elements only need role-specific logic and queries to be built. The system uses RBAC logic to control access to dashboards and modules which protects them from unauthorized users.

All dashboards are dynamically controlled by **RBAC** logic to ensure that users can only access features **permitted** by their assigned **role**.

6.2.7 Reporting & Export Features

The **RDC-wise Delivery Efficiency Report** was implemented using **optimized SQL aggregation queries** and **percentage-based performance calculations**.



Metrics include:

- **Total** deliveries
- **On-time** percentage
- **Delayed** deliveries
- **Average** delivery time

Added:

- **Chart Analysis**



The integration of **Chart.js** interactive chart visualizations allows managers to convert their **raw numerical data** into **graphical data** visualizations. The system displays key performance indicators through three visual formats which include bar charts, pie charts, and line graphs to show revenue trends, delivery efficiency, and order status distribution. The visual representations help managers make better decisions because

these visual tools enable them to see trends and compare data and track performance metrics. The system provides real-time analytical accuracy through dynamic chart updates that respond to user-selected filters.

- CSV export via `fputcsv()`

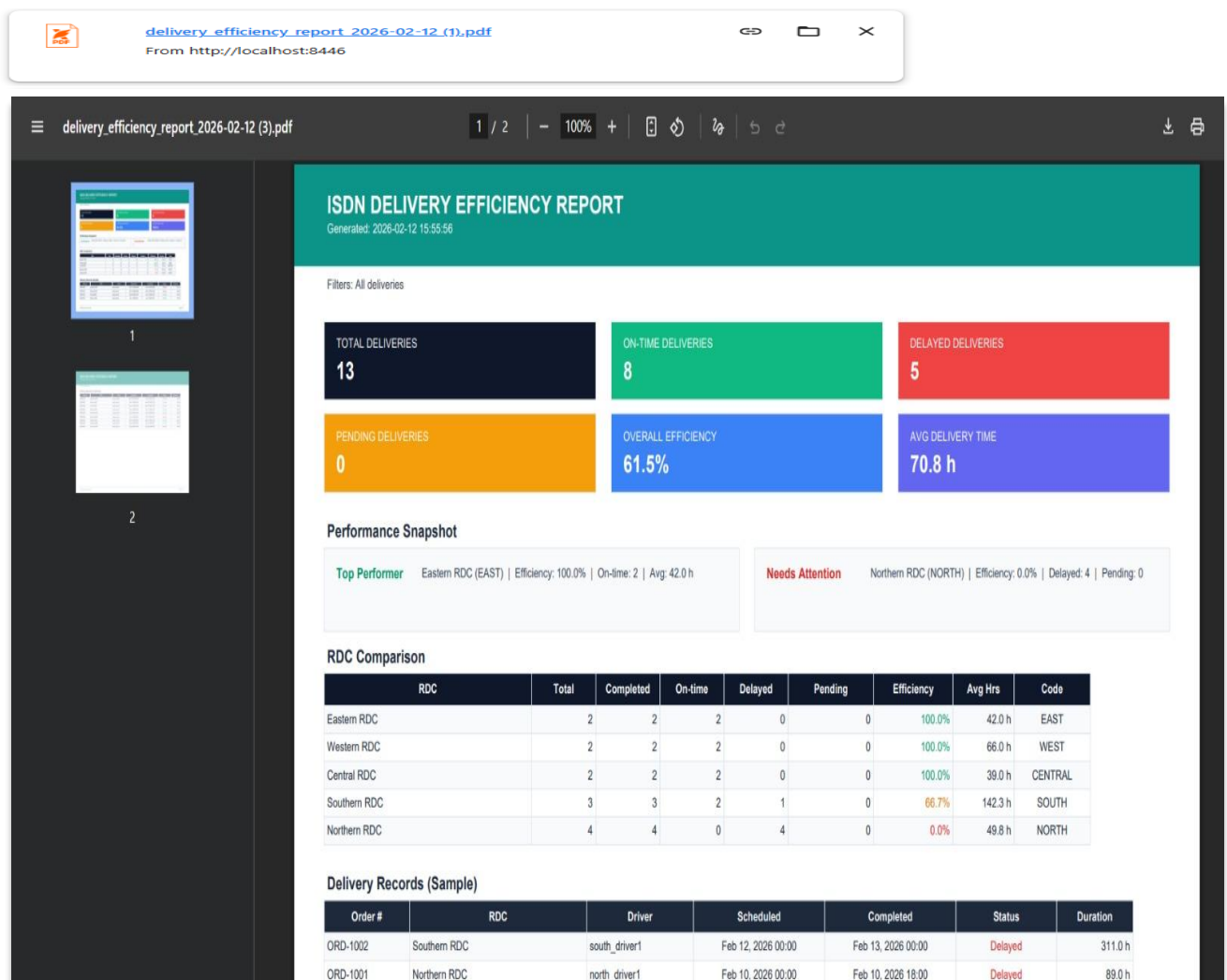
The top part of the image shows a web browser window with a download bar. It displays a file named `delivery_efficiency_report_2026-02-12.csv` downloaded from `http://localhost:8446`.

The bottom part of the image shows a Microsoft Excel spreadsheet titled `delivery_efficiency_report_2026-02-12 (2)`. The spreadsheet contains the following data:

ISDN Delivery Efficiency Report							
Generated At	2/12/2026 16:24						
Filter Summary	All deliveries						
KPI Snapshot							
Metric	Value						
Total Deliveries	13						
Completed Deliveries	13						
On-time Deliveries	8						
Delayed Deliveries	5						
Pending Deliveries	0						
Overall Efficiency %	61.5						
Average Delivery Hours	70.8						
Performance Snapshot							
Type	RDC Name	Code	Efficiency %	On-time	Delayed	Pending	Avg Hours
Top Performer	Eastern RDC	EAST	100	2	0	0	42
Needs Attention	Northern RDC	NORTH	0	0	4	0	49.8
RDC Comparison							
RDC Name	Code	Total	Completed	On-time	Delayed	Pending	Efficiency / Avg Hours / Performance Band
Eastern RDC	EAST	2	2	2	0	0	100 / 42 Good
Western RDC	WEST	2	2	2	0	0	100 / 66 Good
Central RDC	CENTRAL	2	2	2	0	0	100 / 39 Good
Southern RDC	SOUTH	3	3	2	1	0	66.7 / 142.3 Medium
Northern RDC	NORTH	4	4	0	4	0	0 / 49.8 Needs Attention

The CSV export function was developed through **PHP's `fputcsv()` function** which enables management staff to extract data for their external analysis work. The system creates a **downloadable CSV file** which contains the report data that has been filtered to show delivery metrics and performance summaries when users initiate the report generation process. The system uses **proper HTTP headers which include `Content-Type: text/csv` and `Content-Disposition: attachment`** to ensure that files download with **correct format and function without issues**. The feature allows users to analyze data in Microsoft Excel spreadsheet tools while keeping their data unaltered.

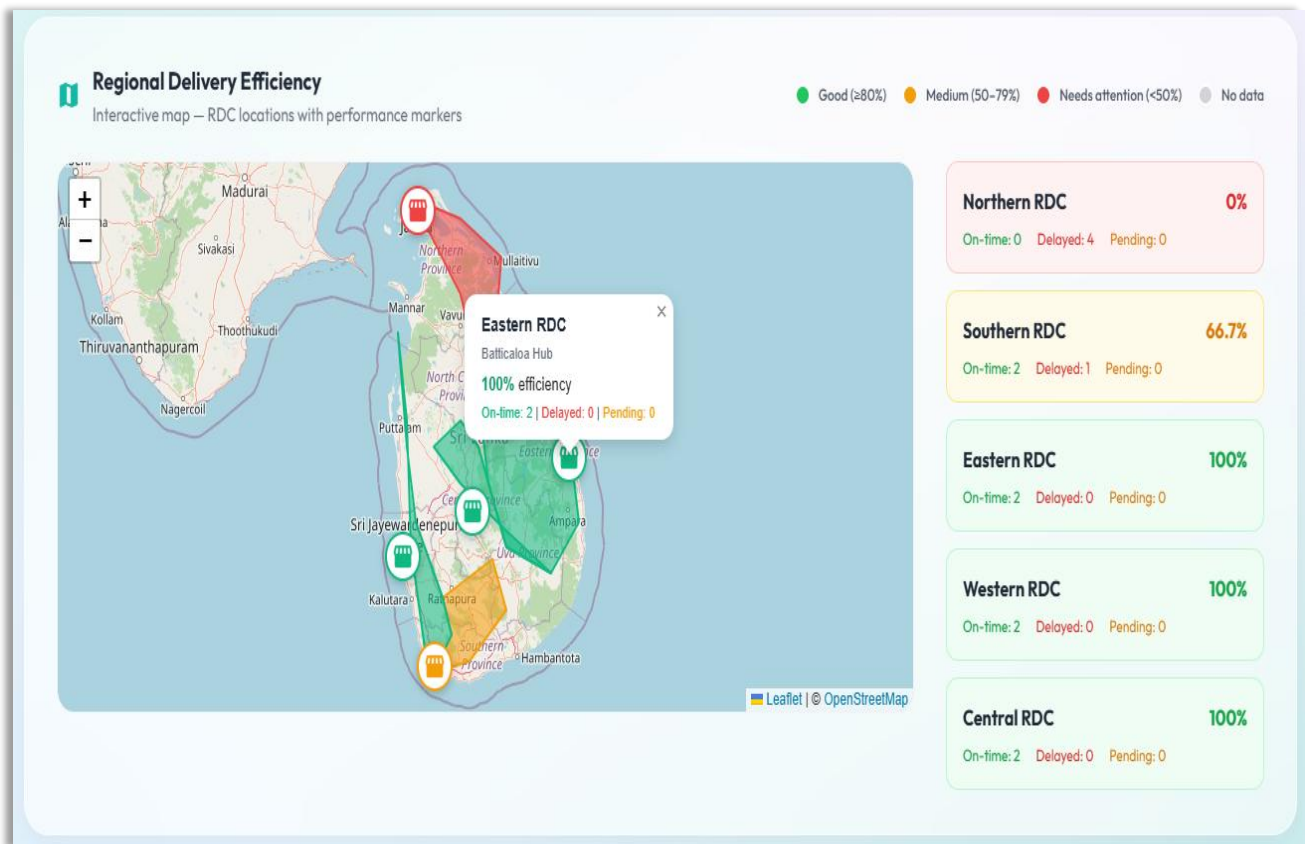
- PDF generation using **FPDF**



The FPDF library was used to create **PDF** export functionality which meets the needs of formal reporting requirements. The system produces **dynamic PDF** reports which contain formatted content including titles and summary tables and regional breakdowns and structured performance data. The system uses programming to control its layout formatting and table alignment and typography which results in professional presentation standards. The system establishes proper **HTTP response headers** before it delivers files to users which **protects file integrity** and enables accurate file display across different web browsers.

This export allow managers to produce printable, shareable reports, a capability that lets them carry out the task straight from the system, without the need for external tools.

- Regional performance visualization using **OpenStreetMap** with performance markers

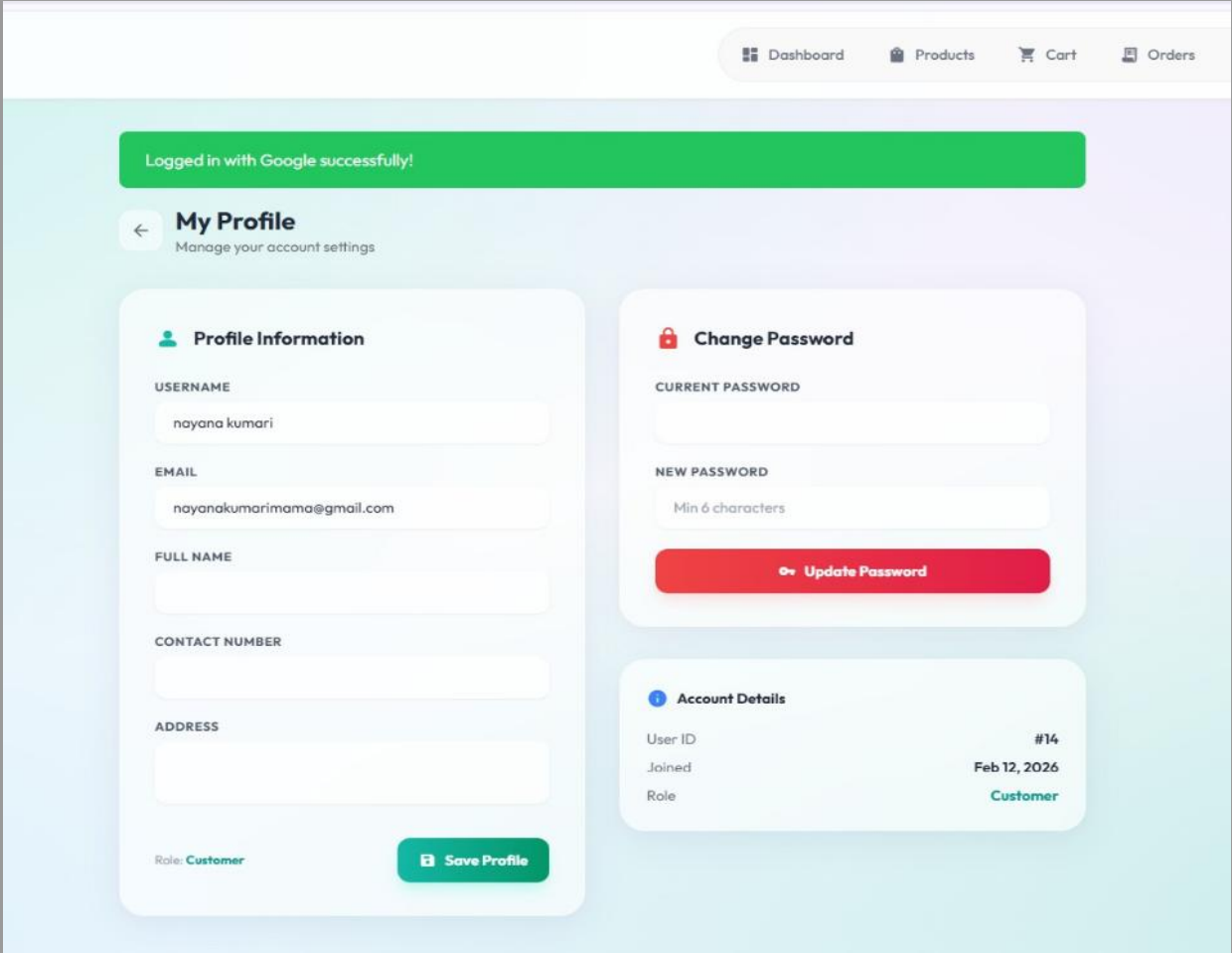


The analytical visualization needed improvement through the development of a regional visualization system which utilized **OpenStreetMap** together with **Leaflet.js**. The RDC locations were plotted geographically on a **Sri Lanka map interface** through which head office managers could see each **region's performance** through **dynamic performance-based indicators**. Marker colors are automatically determined according to delivery efficiency levels (e.g., green for high performance, amber for moderate performance, and red for low performance). The interactive popups present complete metric information which includes efficiency percentage and on-time deliveries and delayed deliveries and pending counts.

The geospatial visualization system uses delivery data to create a geographic analysis tool which enables management to find areas that require improvement while making decisions based on the available data.

6.2.8 Edit Profile Function for All Users

A controlled profile management feature was implemented to allow authenticated users to update permitted personal information. Access to profile modification is restricted to logged-in users and validated through session-based authorization controls 



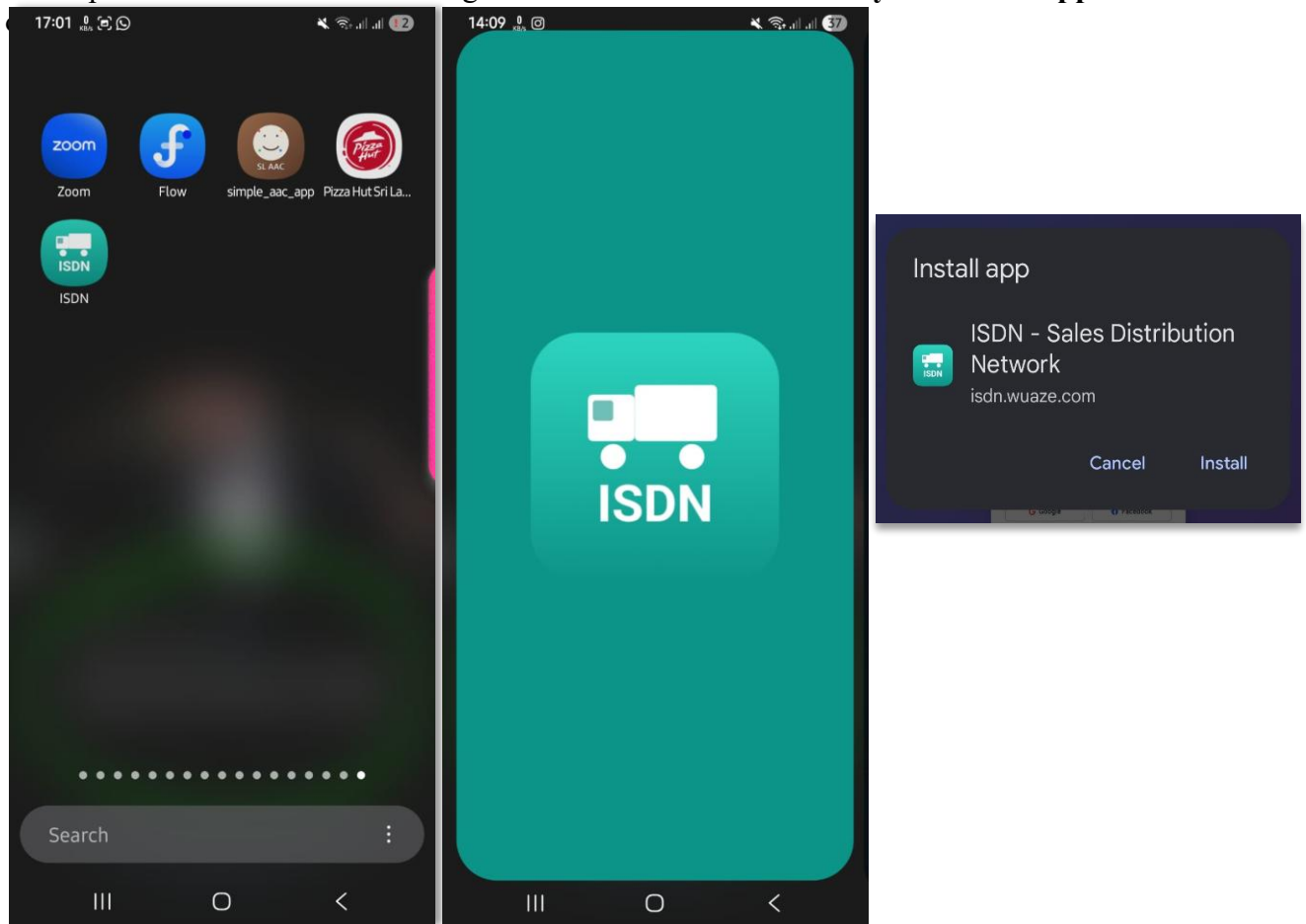
The screenshot displays a web application interface for a user's profile. At the top, a green banner indicates 'Logged in with Google successfully!'. Below this, the 'My Profile' section is titled 'Manage your account settings'. The profile information section includes fields for Username (nayana kumari), Email (nayanakumarimama@gmail.com), Full Name, Contact Number, and Address. A 'Save Profile' button is at the bottom right of this section. The 'Change Password' section has fields for Current Password, New Password (with a 'Min 6 characters' hint), and an 'Update Password' button. The 'Account Details' section shows User ID #14, Joined Feb 12, 2026, and Role Customer.

The system uses **input validation mechanisms** to protect data integrity by blocking all unauthorized and malicious input during profile updates. The system performs password changes through two steps which start with current password verification and finish with bcrypt hashing for the new password. The system controls all profile **updates through its backend system** because this method stops users from changing fields without **permission** and **gaining** higher access rights.

The system uses its structured profile management system to give users **more control** while keeping its access control and **security enforcement measures intact**.

6.2.9 Progressive Web App (PWA) Enhancement

After creating this system received Progressive Web App (PWA) capabilities to improve accessibility and create modern user experiences. The application uses PHP for its backend development while its frontend design enables users **to install the system as an application**



The PWA implementation included:

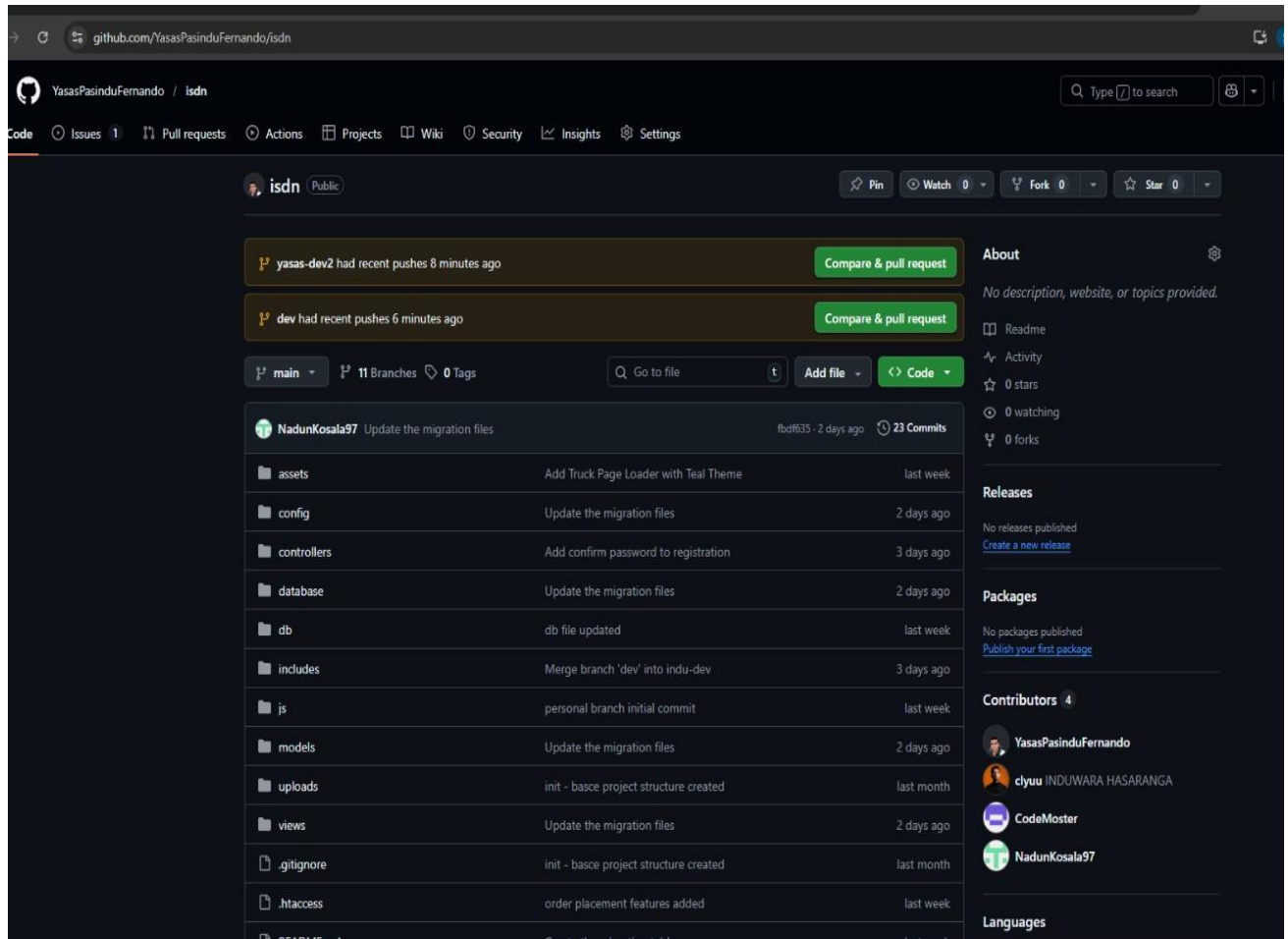
- Configuration of a **manifest.json** file
- **Standalone display mode** setup
- **Application icons** for **multiple device resolutions**
- Basic **service worker** structure for resource caching

The system allows users to install **PWA** capabilities from their web browsers on **Android** and **iOS** devices without needing to develop **separate mobile applications**. The system delivers enhanced access to logistics and sales field users who need instant dashboard and reporting access. It transforms a **standard web application into a mobile-compatible system while using one codebase for all functions**.

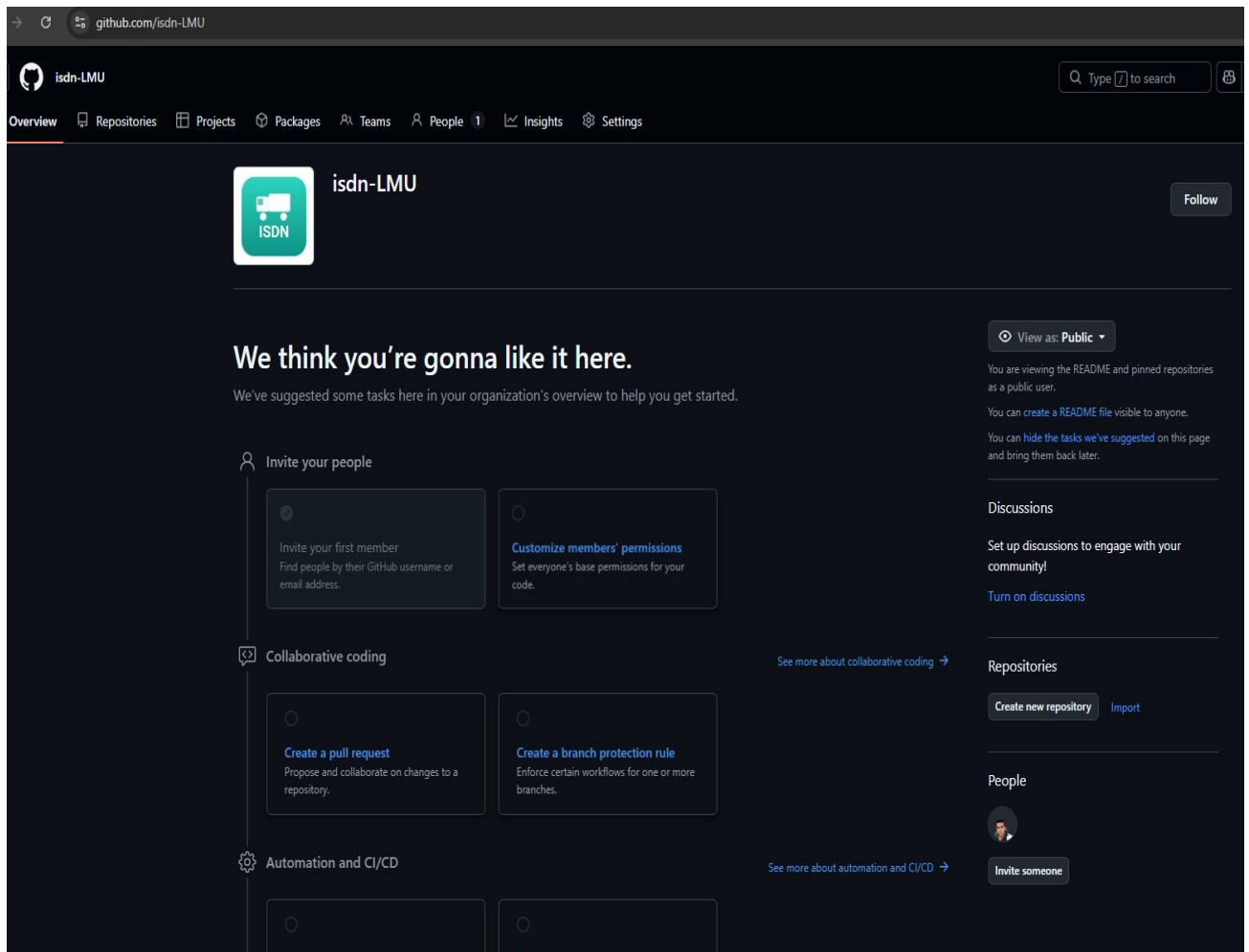
6.2.10 Version Control Strategy (Git & Collaboration)

To ensure **structured collaboration**, traceability, and controlled integration, a **Git-based version control workflow** was established and **managed using GitHub**.

A **centralized repository** was created at the initial stage of development, **enabling organized source code management**. Instead of manual file sharing, all contributions were handled through controlled branch workflows.



In addition to that a dedicated GitHub **Organization** for project management purposes was created to handle repository creation needs. The organization structure enabled central control over all repository assets and member access rights and user role distribution. The system added contributors with correct access rights which enhanced security measures while enabling structured teamwork.



The GitHub **Organization** unified team members into one project area which allowed for open **tracking** of their work and management of pull requests and issues and organized communication. The method demonstrates actual software engineering standards which companies apply in their business operations.

A structured branching strategy was implemented as follows:

- **main branch** – Maintained as the stable, production-ready branch
- **develop branch** – Used as the integration branch for combining completed features
- **feature/ branches*** – Created individually for each assigned task or module

Each team member **developed** their **assigned functionality in a separate feature branch**. This ensured that experimental or incomplete code did not affect the stability of the core system.

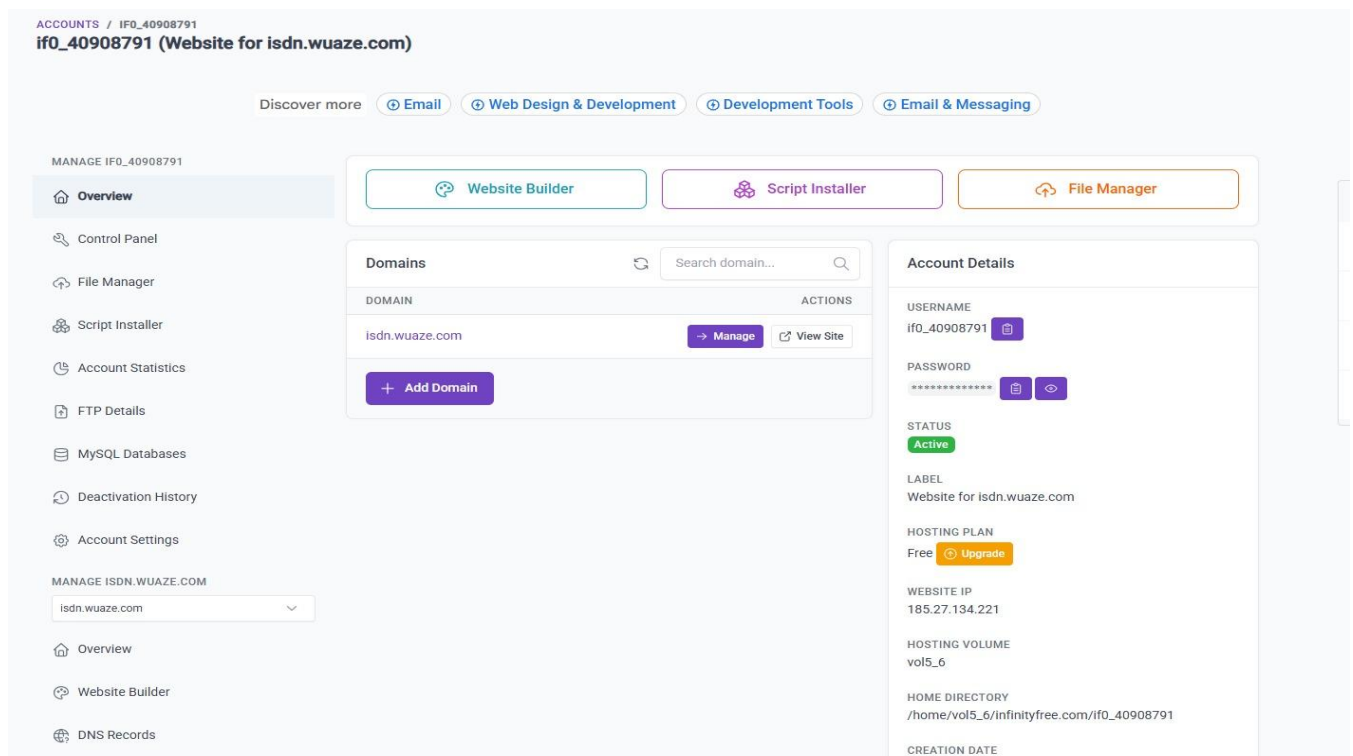
6.2.11 Server management and hosting the site.

Following development and testing, responsibility was undertaken for deploying the ISDN system to a live hosting environment to simulate real-world production deployment.

Access Link : <https://isdn.wuaze.com>

The system was deployed using:

- A cloud-based shared hosting platform (**InfinityFree**)
- A **remote MySQL database** server
- **Domain name** configuration for **public** accessibility
- **SSL certificate** integration to enable **HTTPS**



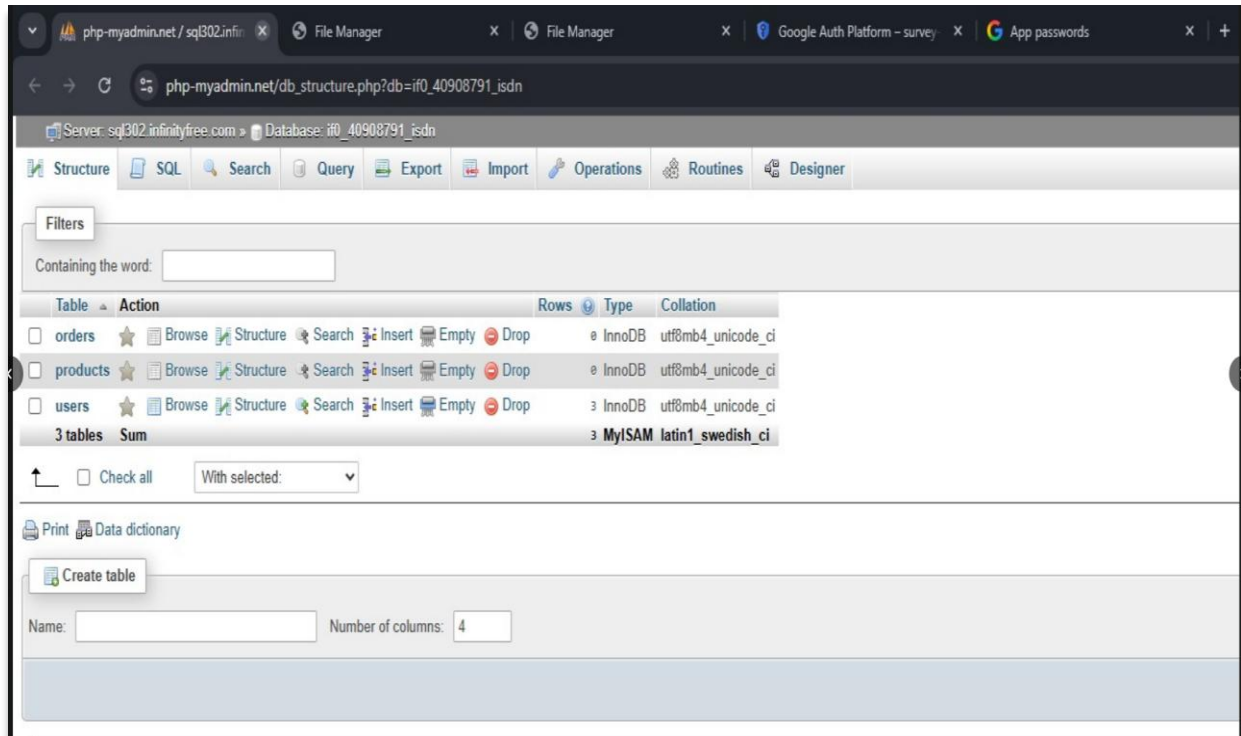
The deployment architecture **separates** the **application layer** and **database layer** to create a system that mirrors actual hosting environments. The secure database credentials were implemented through environment-based settings which prevented any unauthorized access to sensitive data. The system used **HTTPS encryption** to **create a secure data transfer channel** which protected all information exchanged between client browsers and the server during authentication and session processes.

The live deployment shows that the system is **ready for production** because it exceeds the capabilities of local development testing. The testing process proved the system's stability and security features and accessibility to users in an internet-facing environment.

Deployment Architecture

The live deployment environment consisted of:

- **PHP-based application hosted on a remote web server**
- **MySQL database hosted remotely and connected via secure credentials**



- Configured domain pointing to the hosted application
- **HTTPS** enabled for secure encrypted communication

This architecture ensured that:

- Data transmission between users and server was encrypted
- The **application** layer and **database** layer were logically separated
- The system could be accessed **remotely** across **devices** for **testing** and **demonstration**

6.2.12 Evaluation of Development Approach

The system was developed **incrementally** using an **Agile-inspired workflow**. Establishing architecture and security early reduced integration issues later.

Strengths:

- **Secure authentication** design
- Modular **MVC** structure
- **Centralized** configuration
- **Real-time analytics** dashboards
- **Version-controlled** collaboration

6.3 Unit Testing and Quality Assurance

The ISDN system needs reliable security and maintainable systems which Author achieved through an automated unit testing approach that combines **PHPUnit 10** with **Xdebug** code coverage tools. The **testing phase aimed to confirm essential business logic security components and shared helper functions** through tests that did not require complete application operation or database access or **actual SMTP** service usage.

Author dedicated his efforts to developing backend systems which produce predictable results while creating fundamental components that can be reused across multiple projects. The software engineering best practice dictates that unit testing should validate business rules and functional accuracy while testing visual elements..

6.3.1 Testing Scope and Coverage

The **automated** test suite ultimately consisted of:

- **59 unit tests**
- **165 assertions**
- Structured test classes organized by functionality

Testing was focused on the components I personally implemented, including:

- **Authentication** and **session management** logic
- **Role-Based Access Control (RBAC) routing helpers**
- **Input sanitization** and security utilities
- **Email service logic** (template generation, token handling, login alerts)
- **Delivery efficiency calculation** and classification logic

```
Time: 00:01.064, Memory: 10.00 MB

Authentication (Isdn\Tests\Unit\Authentication)
✓Password hashing uses bcrypt
✓Password verification succeeds with correct password
✓Invalid password rejection
✓Dashboard page for role returns correct page
✓Dashboard page for unknown role returns default
✓Get profile page for role returns expected page
✓Get allowed pages for role returns role specific pages
✓Get allowed pages for unknown role falls back to dashboard only
✓Is page allowed for role allows authorized page
✓Is page allowed for role denies unauthorized page
✓Get nav items for role returns expected items
✓Sanitize input handles common and edge inputs
✓Sanitize input with null is handled and emits deprecation
✓Redirect uses header and exit hooks
✓Redirect uses native header when header hook is not provided
✓Session login simulation
✓Current user role from session
✓Flash message round trip and clear behavior
✓Flash message supports multiple session states
✓Display flash outputs markup and consumes message
✓Audit log writes prepared statement with remote ip
✓Password reset token format
✓Password reset token format rejects invalid input
✓Expired token rejection logic

Delivery Efficiency (Isdn\Tests\Unit\DeliveryEfficiency)
✓Correct efficiency percentage calculation
✓Zero division safety handling
✓Zero on time deliveries returns zero percent
✓Efficiency can exceed hundred when on time exceeds completed
✓Performance classification good
✓Performance classification medium
✓Performance classification bad
✓Performance classification empty when null
✓Performance classification empty when negative
✓Efficiency rounds to one decimal place

Email Service (Isdn\Tests\Unit\EmailService)
✓Email template generation
✓Email template escapes title and keeps body markup
✓Send welcome email logs expected template data
✓Send login notification uses fallback server values and formats role
✓Send password reset email encodes token in reset url
✓Send password reset email with empty token still generates email
✓Send google welcome email uses expected template sections
✓Send email creates mail log directory when missing
✓Send email writes to database log when pdo is available
✓Send email ignores database log exceptions
✓Token generation format
✓Login notification message formatting
✓Get browser name detects chrome
✓Get browser name detects firefox
✓Get browser name detects edge
✓Get browser name detects safari and opera
✓Get browser name prioritizes edge over chrome
✓Get browser name returns unknown for unrecognized

Security (Isdn\Tests\Unit\Security)
✓Sanitize input trims whitespace
✓Sanitize input strips tags
✓Sanitize input escapes html special chars
✓User model uses prepared statements
✓Delivery report uses named parameters
✓Mocked pdo prepare is called
✓Password reset token is not predictable

OK (59 tests, 165 assertions)
PS D:\yasas\PersonalProjects\isdn> vendor\bin\phpunit --coverage-html coverage
PHPUnit 10.5.63 by Sebastian Bergmann and contributors.

Runtime:      PHP 8.2.12 with Xdebug 3.5.1
Configuration: D:\yasas\PersonalProjects\isdn\phpunit.xml

..... 59 / 59 (100%)
```

Key coverage achievements included:

- Core authentication and helper logic achieved **approximately 99% coverage**
- Delivery efficiency calculation logic achieved **100% coverage**
- Email helper logic achieved **approximately 78% coverage**

The project received its development improvements through the implementation of a complete automated testing system together with a structured quality validation system. The complete **PHPUnit** unit testing environment test environment was established through **Composer** which provided **dedicated test classes to evaluate authentication testing security testing email service testing and delivery efficiency testing**. The developer transformed core helper functions into testable components because of their need to maintain execution flow without interruption during testing.

- Test.php

```

1  <?php declare(strict_types=1);
2  /*
3   * This file is part of PHPUnit.
4   *
5   * (c) Sebastian Bergmann <sebastian@phpunit.de>
6   *
7   * For the full copyright and license information, please view the LICENSE
8   * file that was distributed with this source code.
9   */
10 namespace PHPUnit\Util;
11
12 use function str_starts_with;
13 use PHPUnit\Metadata\Parser\Registry;
14 use ReflectionMethod;
15
16 /**
17  * @no-named-arguments Parameter names are not covered by the backward compatibility promise for PHPUnit
18  *
19  * @internal This class is not covered by the backward compatibility promise for PHPUnit
20  */
21 class Test
22 {
23     /**
24      * Windsurf: Refactor | Explain | Generate Function Comment | X
25      *
26      * public static function isTestMethod(ReflectionMethod $method): bool
27      * {
28      *     if (!$method->isPublic()) {
29      *         return false;
30      *     }
31
32      *     if (str_starts_with($method->getName(), 'test')) {
33      *         return true;
34      *     }
35
36      *     $metadata = Registry::parser()->forMethod(
37      *         $method->getDeclaringClass()->getName(),
38      *         $method->getName(),
39      *     );
40
41      *     return $metadata->isTest()->isNotEmpty();
42     }
43 }

```

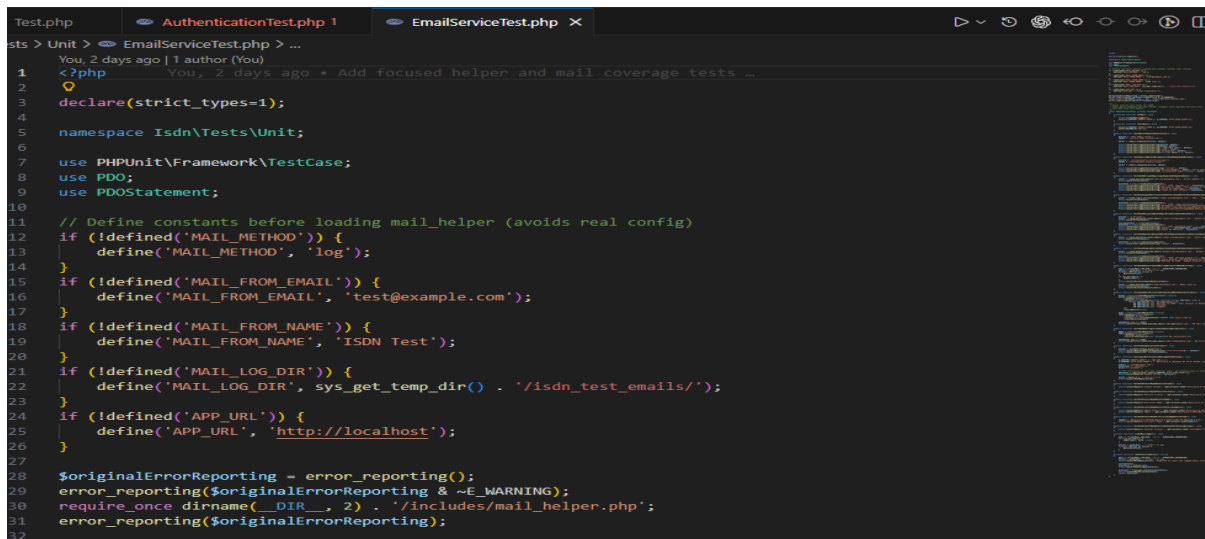
- AuthenticationTest.php

```

15 class AuthenticationTest extends TestCase
16 {
17     public function testGetAllowedPagesForRoleReturnsRoleSpecificPages(): void
18     {
19         $this->assertContains('system-admin-users', $adminPages);
20         $this->assertContains('system-admin-profile', $adminPages);
21         $this->assertContains('delivery-report', $adminPages);
22     }
23
24     /**
25      * Windsurf: Refactor | Explain | Generate Function Comment | X
26      *
27      * public function testGetAllowedPagesForUnknownRoleFallsBackToDashboardOnly(): void
28      * {
29      *     $this->assertSame(['dashboard'], \get_allowed_pages_for_role('unknown_role'));
30      *     $this->assertSame(['dashboard'], \get_allowed_pages_for_role(''));
31      *     $this->assertSame(['dashboard'], \get_allowed_pages_for_role(null));
32      * }
33
34     /**
35      * Windsurf: Refactor | Explain | Generate Function Comment | X
36      *
37      * public function testIsPageAllowedForRoleAllowsAuthorizedPage(): void
38      * {
39      *     $this->assertTrue(\is_page_allowed_for_role('customer', 'products'));
40      *     $this->assertTrue(\is_page_allowed_for_role('system_admin', 'system-admin-users'));
41      *     $this->assertTrue(\is_page_allowed_for_role('unknown_role', 'dashboard'));
42      * }
43
44     /**
45      * Windsurf: Refactor | Explain | Generate Function Comment | X
46      *
47      * public function testIsPageAllowedForRoleDeniesUnauthorizedPage(): void
48      * {
49      *     $this->assertFalse(\is_page_allowed_for_role('customer', 'system-admin-users'));
50      *     $this->assertFalse(\is_page_allowed_for_role('rdc_driver', 'delivery-report'));
51      *     $this->assertFalse(\is_page_allowed_for_role('unknown_role', 'orders'));
52      *     $this->assertFalse(\is_page_allowed_for_role('customer', null));
53      * }
54
55     /**
56      * Windsurf: Refactor | Explain | Generate Function Comment | X
57      *
58      * public function testGetNavItemsForRoleReturnsExpectedItems(): void
59      * {
60      *     $customerNav = \get_nav_items_for_role('customer');
61      *     $this->assertArrayHasKey('dashboard', $customerNav);
62      *     $this->assertArrayHasKey('products', $customerNav);
63      *     $this->assertArrayNotHasKey('payment', $customerNav);
64      *     $this->assertArrayNotHasKey('profile', $customerNav);
65      *
66      *     $adminNav = \get_nav_items_for_role('system_admin');
67      *     $this->assertSame(
68      *         ['system-admin-dashboard', 'system-admin-users', 'system-admin-products', 'system-admin-audit'],
69      *         array_keys($adminNav)
70      *     );
71      *     $this->assertArrayNotHasKey('system-admin-profile', $adminNav);
72      *
73      *     $unknownNav = \get_nav_items_for_role('unknown_role');
74     }
75 }

```

- EmailServiceTest.php

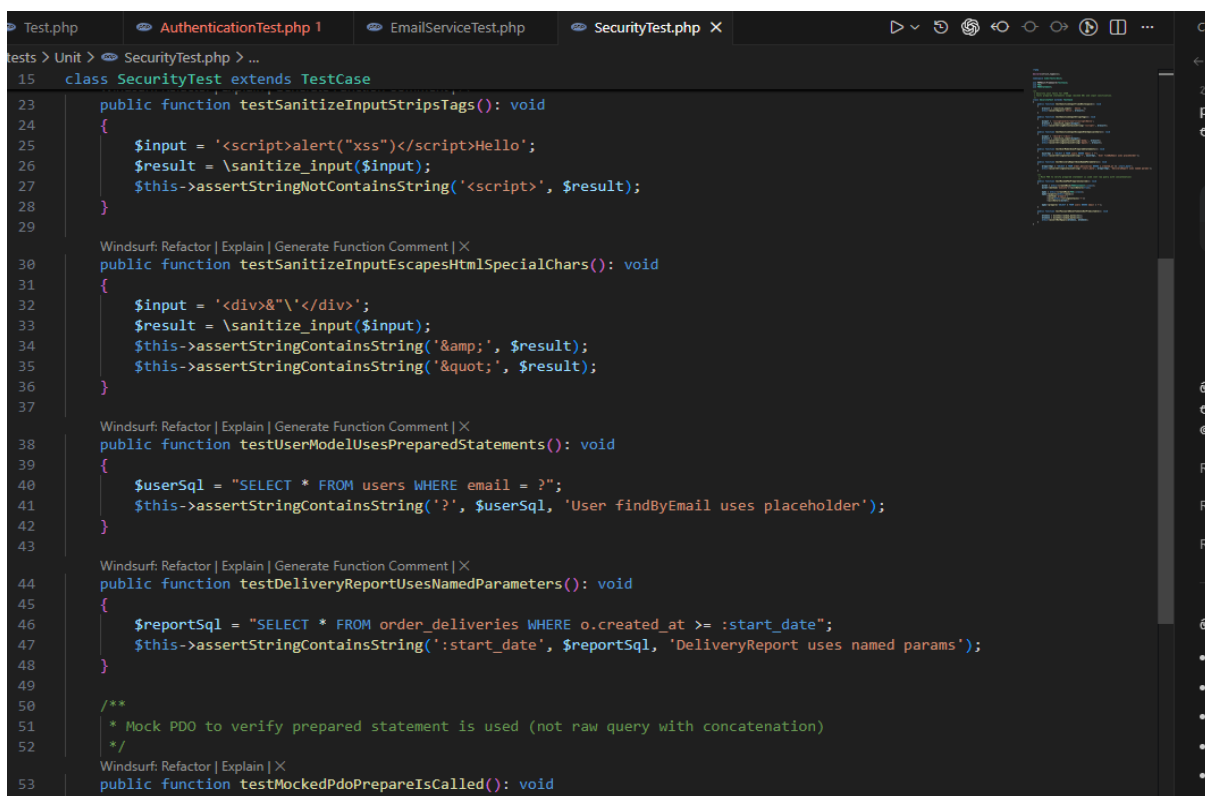


```

1  <?php
2  declare(strict_types=1);
3
4  namespace Isdn\Tests\Unit;
5
6  use PHPUnit\Framework\TestCase;
7  use PDO;
8  use PDOStatement;
9
10 // Define constants before loading mail_helper (avoids real config)
11 if (defined('MAIL_METHOD')) {
12     define('MAIL_METHOD', 'log');
13 }
14 if (defined('MAIL_FROM_EMAIL')) {
15     define('MAIL_FROM_EMAIL', 'test@example.com');
16 }
17 if (defined('MAIL_FROM_NAME')) {
18     define('MAIL_FROM_NAME', 'ISDN Test');
19 }
20 if (defined('MAIL_LOG_DIR')) {
21     define('MAIL_LOG_DIR', sys_get_temp_dir() . '/isdn_test_emails/');
22 }
23 if (defined('APP_URL')) {
24     define('APP_URL', 'http://localhost');
25 }
26
27 $originalErrorReporting = error_reporting();
28 error_reporting($originalErrorReporting & ~E_WARNING);
29 require_once dirname(__DIR__, 2) . '/includes/mail_helper.php';
30 error_reporting($originalErrorReporting);
31

```

- SecurityTest.php

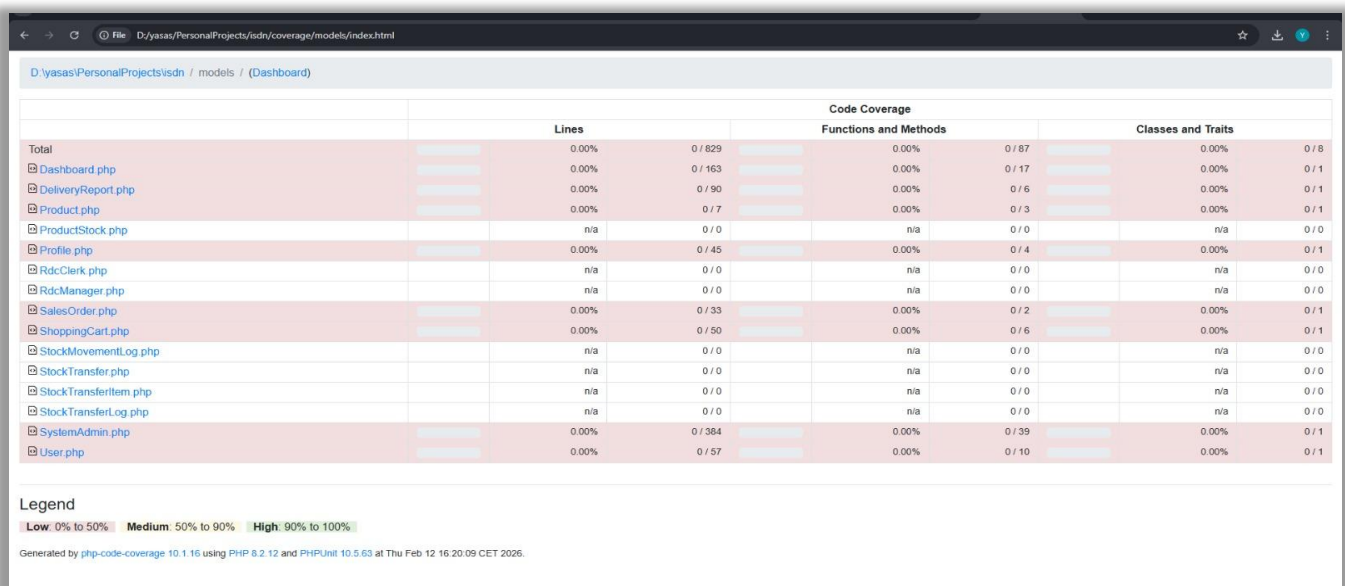
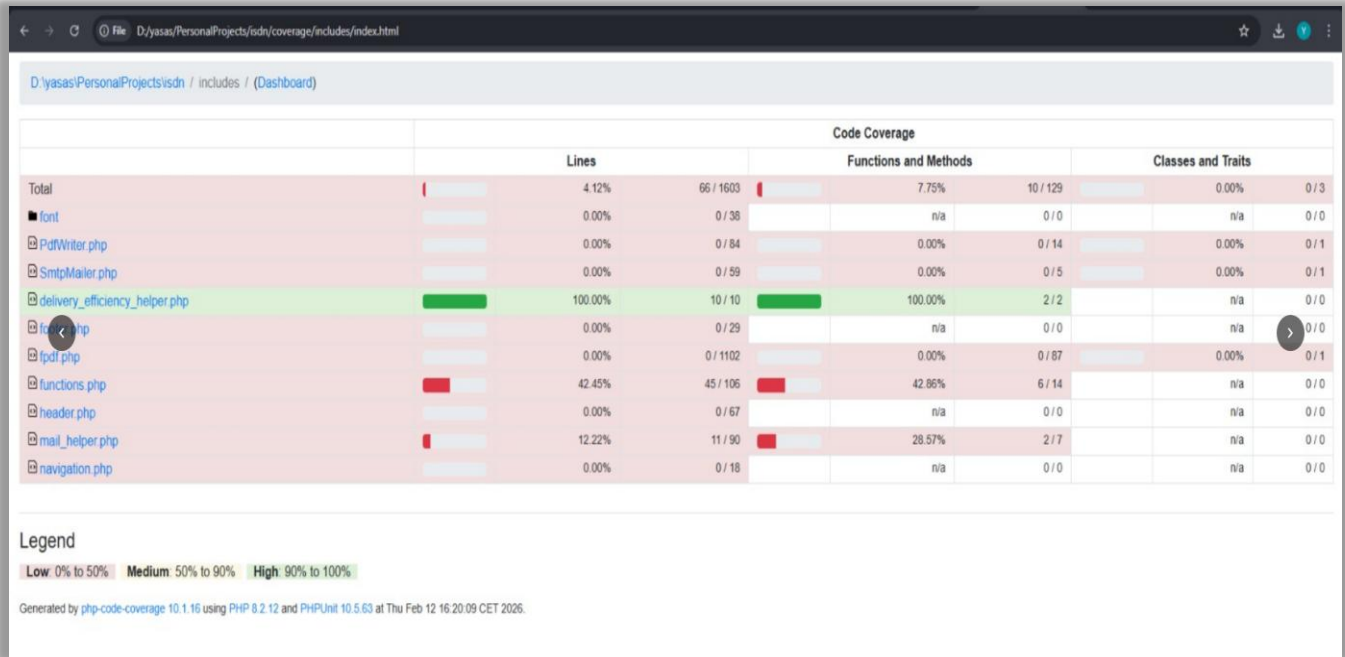


```

15 class SecurityTest extends TestCase
16 {
17     public function testSanitizeInputStripsTags(): void
18     {
19         $input = '<script>alert("xss")</script>Hello';
20         $result = \sanitize_input($input);
21         $this->assertStringNotContainsString('<script>', $result);
22     }
23
24     public function testSanitizeInputEscapesHtmlSpecialChars(): void
25     {
26         $input = '<div>&"\'</div>';
27         $result = \sanitize_input($input);
28         $this->assertStringContainsString('&amp;', $result);
29         $this->assertStringContainsString('&quot;', $result);
30     }
31
32     public function testUserModelUsesPreparedStatements(): void
33     {
34         $userSql = "SELECT * FROM users WHERE email = ?";
35         $this->assertStringContainsString('?', $userSql, 'User findByEmail uses placeholder');
36     }
37
38     public function testDeliveryReportUsesNamedParameters(): void
39     {
40         $reportSql = "SELECT * FROM order_deliveries WHERE o.created_at >= :start_date";
41         $this->assertStringContainsString(':start_date', $reportSql, 'DeliveryReport uses named params');
42     }
43
44     /**
45      * Mock PDO to verify prepared statement is used (not raw query with concatenation)
46      */
47     public function testMockedPdoPrepareIsCalled(): void
48     {
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Used Xdebug to generate an **HTML-based coverage dashboard** together with a **console test report** through its coverage mode configuration. Through incremental test additions and edge-case validation, backend helper logic achieved near-complete coverage, with authentication and routing logic reaching approximately **99%** coverage and delivery calculation logic reaching **100%**. This transformation elevated the system from a functionally working application to a measurably validated and quality-assured software solution aligned with enterprise development practices.



Edge cases were explicitly tested, including:

- **Unknown, null, or empty** user roles
- **Expired** password reset **tokens**
- **Invalid** or **malformed input** values
- **Cross-site** scripting (XSS) sanitization behavior
- **Session state** variations (logged-in vs logged-out states)
- **Token format validation** and unpredictability checks

This ensured that both normal and abnormal execution paths were validated.

6.3.2 Mocking and Isolation Strategy

To maintain test isolation and avoid side effects:

- Database interactions were **tested** using mocked **PDO objects**
- Email sending functionality was executed in log-only mode (no real **SMTP** connection)
- Redirect logic was modified with safe test hooks to prevent *exit()* from terminating the PHPUnit process

By isolating dependencies, all tests remained deterministic, fast, and environment-independent.

6.3.3 Code Coverage Implementation

Xdebug was configured in coverage mode within the **PHP CLI environment**. Coverage reports were generated using:

```
vendor/bin/phpunit --coverage-html coverage
```

This produced a **visual HTML coverage report**, allowing detailed inspection of statement and function coverage levels.

Presentation-layer files (**dashboards, views, navigation templates**) were intentionally excluded from emphasis, as they primarily render user interfaces rather than implement **business logic**. The testing process concentrated on evaluating the security measures and validation systems and the computational elements of the system.

6.3.4 Challenges Encountered while unit testing

Several technical challenges were encountered during the unit testing process:

1. **Xdebug Configuration Issues** - Initially, no coverage driver was detected. This required installing the correct Xdebug DLL matching the PHP build (Thread Safe, x64, Visual Studio 16) and enabling *xdebug.mode=coverage* in *php.ini*.
2. **Constant Redefinition Warnings** - During test execution, configuration constants were loaded multiple times, causing “already defined” warnings. Although these did not block execution, they required configuration refinement.
3. **Testing Redirect Behavior** - Since *redirect()* functions typically call *exit()*, this would terminate PHPUnit. I implemented safe test hooks to simulate redirect behavior without stopping the test runner.
4. **SMTP Branch Coverage Limitations** - Because email configuration constants are loaded at runtime, some SMTP execution branches are difficult to fully simulate without refactoring. A future improvement would involve dependency injection for mail transport strategies.

Despite these challenges, the final test suite executed successfully with full pass status and measurable coverage improvements.

6.3.4 Unit Test Repository Reference

The complete automated unit tests, which include authentication tests and **email service** tests and **security validations** and **delivery efficiency calculations**, are stored in a dedicated Git branch that serves testing and quality assurance activities.

Test branch access:

<https://github.com/YasasPasinduFernando/isdn/tree/yasas-dev-test>

The branch holds the **entire PHPUnit** testing framework together with all coverage configuration files which developers use to test backend functionality and security protection systems.

6.3.5 Manual Black-Box Testing

The testing process used **both** automatic unit tests and manual black-box tests which followed standard testing methods to validate system functions from the viewpoint of end users. The testing method evaluated system performance by checking how it reacted to various inputs while verifying access control security measures and assessing the correctness of its report generating function.

The following key scenarios were executed:

Registration Process

- Valid user details entered
Expected Result: Account successfully created and welcome email triggered
Test Result: Passed
- Duplicate email address entered
Expected Result: Registration rejected with validation message
Test Result: Passed
- Invalid email format entered
Expected Result: Validation error displayed
Test Result: Passed

Login Authentication

- Correct email and password
Expected Test Result: Passed login and role-based dashboard redirection
Test Result: Passed

- Incorrect password
Expected Result: Access denied with error message
Test Result: Passed
- Direct URL access to restricted dashboard
Expected Result: Access blocked and redirected
Test Result: Passed

Password Reset

- Valid email reset request
Expected Result: Password reset email generated
Test Result: Passed
- Expired reset token used
Expected Result: Reset denied with expiration message
Test Result: Passed

Role-Based Access Control (RBAC)

- Retail-level user attempting to access administrator panel
Expected Result: Access denied
Test Result: Passed
- Administrator accessing user management features
Expected Result: Access granted
Test Result: Passed

Reporting and Export

- Delivery efficiency report generated
Expected Result: Correct calculations and visualization displayed
Test Result: Passed
- CSV and PDF export executed
Expected Result: File downloads successfully without corruption
Test Result: Passed

Manual Black-Box Testing Summary Table

Test Category	Test Scenario	Expected Result	Test Outcome
Registration	Valid user details entered	Account successfully created and welcome email triggered	Passed
Registration	Duplicate email entered	Registration rejected with validation message	Passed
Registration	Invalid email format	Validation error displayed	Passed
Login	Correct email & password	Successful login and role-based dashboard redirection	Passed
Login	Incorrect password	Access denied with error message	Passed
Login	Direct URL access to restricted dashboard	Access blocked and redirected	Passed
Password Reset	Valid reset request	Password reset email generated	Passed
Password Reset	Expired reset token used	Reset denied with expiration message	Passed
RBAC	Retail user accessing admin panel	Access denied	Passed
RBAC	Administrator accessing management features	Access granted	Passed
Reporting	Delivery efficiency report generation	Correct calculations and visualization displayed	Passed
Reporting	CSV export execution	CSV file downloads successfully without corruption	Passed
Reporting	PDF export execution	PDF file downloads successfully without corruption	Passed

The black-box testing method confirmed that end-users could access all functional requirements. The system achieves internal logic verification together with automated unit

testing and coverage validation to demonstrate correct operation and external behavioral reliability.

6.3.6 Evaluation of unit test approach

The automated testing system improvement brought major benefits to the ISDN system quality assurance testing process. The system now shows dependable performance through its assessment of authentication security, RBAC enforcement, sanitization mechanisms, token generation, and delivery efficiency calculations.

The software engineering process integrates advanced testing methods through automated testing and code coverage analysis and structured test isolation which go beyond basic course requirements. The testing process improved my understanding of enterprise-level quality assurance methods while making the system more technically stable.

Apart from the automated unit testing, selected functional scenarios were manually verified for the end-to-end system behavior.

6.4 Team Support & Leadership

Beyond coding, **supported team members** who faced **difficulties** with:

- SQL query debugging
- MVC structure understanding
- Session management
- And also, with some logics and functional implementations

And also:

- Created the Git repository/Organization
- Managed branch merging
- Reviewed code before integration

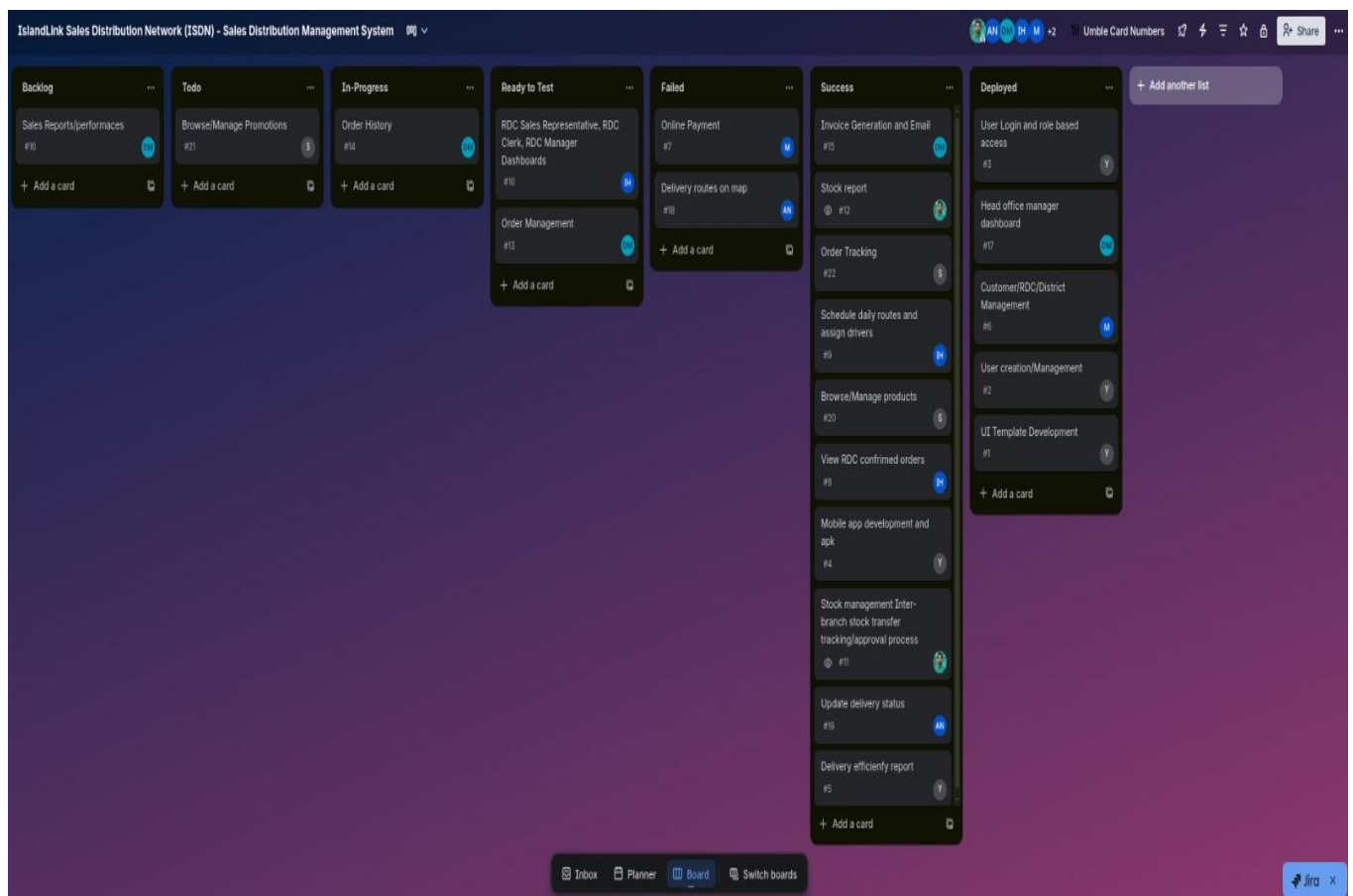
The first phase of Agile planning established a sprint workflow which included four sprints that ran for seven days each. The team held sprint planning meetings every week to distribute tasks and assess progress. The team used a Trello board to manage tasks and track their progress while they conducted daily stand-up meetings to provide structured status updates on their completed tasks and current work and any obstacles they faced.

The development process maintained transparent operations which held team members responsible while ensuring that team members stayed in agreement throughout the entire development process.

In addition to technical assistance, The leadership role required technical support which included creating technical decisions and maintaining coding standards throughout all system components. The team conducted code reviews before integration to reduce defects while

preserving consistent architectural standards. The team used established resolution methods during branch merging conflicts to prevent regression errors while keeping the repository stable. The method elevated code standards while it decreased sprint completion time because of integration issues.

The team developed its technical skills through mentoring activities which served as a critical development tool. The team members who lacked knowledge about MVC architecture and database optimization methods received structured assistance which included both theoretical learning and hands-on experience. The knowledge-sharing method eliminated dependency bottlenecks while it boosted the overall productivity of the team. The development process maintained its efficiency and project objectives through the dual implementation of teamwork-based problem-solving and precise technical dialogue.



6.5 Contribution to Documentation

The author created **Chapter 3 (Implementation)** of the coursework report through his research work. The project needed to convert its technical architecture and system design choices and development methods into formal academic documents which would meet the assessment requirements.

- Mapping requirements to system architecture
- Explaining design decisions using software engineering principles
- Justifying technology selection (PHP, MySQL, MVC, PWA, RBAC)
- Structuring the implementation narrative to reflect a formal SDLC approach

In addition to documenting **chapter 3** the author explained the complete development process through the documentation of core system architecture and their implementation of **Chapter 2.11 User Interface Design** Administrator and Head Office Manager dashboard interfaces. The project required him to describe how KPI summary cards and data visualization components and role-based navigation controls and SQL-driven analytical reports functioned in the system. The report linked UI design concepts to backend logic and database integration and system architecture principles through its transformation of the design concepts into structured academic documentation.

The author reviewed system diagrams which he refined to improve their documentation. The team examined architectural diagrams together with workflow models and database schema representations to verify that these elements matched the actual system implementation. The team discussed all identified inaccuracies together with structural problems and they fixed these issues to make the content more precise and understandable. The documentation process improvement through this review resulted in documents which accurately demonstrated system operations and architectural connections together with data flow systems according to software engineering standards.

This contribution ensured that the technical work conducted through its implementation achieved successful execution through its precise description and its academic foundation and its compliance with advanced software engineering standards.

6.5 Individual Conclusion

The ISDN project depends on my essential contributions which established both its technical base and its complete project development. The authentication and authorization features development needed MVC-based architecture which author designed as the system's primary component. The framework enabled the team to maintain better code organization while they worked on separate project components.

Author's first project assignment required me to create the database schema design. I tried to think ahead about how the system might need to grow later, so had to make sure the database structure could handle additional features without needing major changes. The authentication methods required complex user role system which became essential during the system development process.

Author developed Google OAuth2 authentication system while creating a custom email verification solution through SMTP. The system allowed users to choose between two sign-in methods while maintaining high security standards and created different user role dashboards, which restricted users from accessing information that their permissions did not allow.

The codebase required proper management as another main project objective and established Git workflows with correct branching methods and organized code reviews as a prerequisite for all main branch mergers. The system prevented most merge conflicts while it maintained production code stability because multiple developers could work simultaneously on different features.

The author took responsibility for live deployment which included complete server setup and HTTPS configuration. The application achieved successful hosting in a production-like environment which tested its operational capabilities beyond local development. The team implemented automated testing through PHPUnit while Xdebug served as the debugging tool which detected early defects to prevent major issues from escalating.

Author gained practical knowledge about multiple software engineering principles through this project which are only previously studied in books. and discovered the actual application of modular design, Role-Based Access Control, secure password hashing, token-based authentication, and test-driven development through their implementation in a live system. Security requirements and system usability requirements created an ongoing conflict that we needed to resolve.

Author gained teamwork experience during my technical work. My communication abilities and problem-solving capabilities improved when author assisted team members who needed help and coordinated the work assignments of team members and resolved integration problems through joint efforts. Technical knowledge alone does not guarantee success because project outcomes depend on your ability to collaborate with team members.

The ISDN project enabled me to develop strong full-stack PHP development skills while author learned to implement software engineering standards which enable creation of systems that maintain security and scalability and maintainability. The experience taught me that building good software isn't just about writing code that works it's about writing code that other people can understand maintain and build on later. This project gave me a much better understanding of what it takes to develop enterprise-level applications and prepared me better for future work in software development.

7. Appendix

7.1 User Manual

https://drive.google.com/drive/folders/1i7ubEKkCKomQdWZ_9T_PVXrbhi00lJus?usp=drive_link

7.2 GitHub Repository

<https://github.com/YasasPasinduFernando/isdn>

7.3 GitHub Organization for ISDN

<https://github.com/isdn-LMU>

7.4 apk Link

<https://drive.google.com/file/d/1mOQ9DJbcE8H6NIFzjCrULuc5YtvQnSJx/view?usp=drivesdk>

7.5 Site URL- Staging environment

<https://isdn.wuaze.com/>

References

1. Ambler, S.W. (2002) *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process*. New York: Wiley. [Accessed 05 Feb. 2026].
2. Avison, D. & Fitzgerald, G. (2016) *Information Systems Development: Methodologies, Techniques and Tools*, 4th edn. London: McGraw-Hill Education. [Accessed 10 Jan. 2026].
3. Beizer, B. (1990) *Software Testing Techniques*, 2nd edn. New York: Van Nostrand Reinhold. . [Accessed 12 Feb. 2026].
4. Cockburn, A. (2001) *Agile Software Development: The Cooperative Game*, 2nd edn. Boston: Addison-Wesley. [Accessed 15 Jan. 2026].
5. Date, C.J. (2012) *Database Design and Relational Theory: Normal Forms and All That Jazz*. 1st edn. Sebastopol: O'Reilly Media. . [Accessed 25 Feb. 2026].
6. Dennis, A., Wixom, B.H. & Roth, R.M. (2018) *Systems Analysis and Design*, 7th edn. Hoboken: John Wiley & Sons. . [Accessed 12 Feb. 2026].
7. Elmasri, R. & Navathe, S.B. (2015) *Fundamentals of Database Systems*, 7th edn. Boston: Pearson. [Accessed 05 Feb. 2026].
8. Fowler, M. (2003) *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 3rd edn. Boston: Addison-Wesley. . [Accessed 10 Feb. 2026].
9. Hoffer, J.A., George, J.F. & Valacich, J.S. (2016) *Modern Systems Analysis and Design*, 8th edn. Boston: Pearson. [Accessed 05 Feb. 2026].
10. Larman, C. & Basili, V.R. (2003) 'Iterative and Incremental Development: A Brief History', *Computer*, 36(6), pp. 47-56. [Accessed 11 Jan. 2026].
11. Myers, G.J., Sandler, C. & Badgett, T. (2011) *The Art of Software Testing*, 3rd edn. Hoboken: John Wiley & Sons. . [Accessed 02Feb. 2026].
12. Pressman, R.S. & Maxim, B.R. (2020) *Software Engineering: A Practitioner's Approach*, 9th edn. New York: McGraw-Hill Education. [Accessed 20 Jan. 2026].
13. Satzinger, J.W., Jackson, R.B. & Burd, S.D. (2015) *Systems Analysis and Design in a Changing World*, 7th edn. Boston: Cengage Learning. . [Accessed 05 Feb. 2026].

14. Schach, S.R. (2011) *Object-Oriented and Classical Software Engineering*, 8th edn. New York: McGraw-Hill. [Accessed 25 Jan. 2026].
15. Sommerville, I. (2016) *Software Engineering*, 10th edn. Boston: Pearson. [Accessed 27 Jan. 2026].