

SMART-COST : A HYBRID AI-DRIVEN POS AND COST ANALYTICS SYSTEM FOR MICRO-SMES

COMMON INTEGRATED SOLUTION REPORT

R26-IT-026

B.Sc. (Hons) Degree in Information Technology Specializing in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

September 2026

SMART-COST : A HYBRID AI-DRIVEN POS AND COST ANALYTICS SYSTEM FOR MICRO-SMES

Team member	Student ID	Individual component
K.A.C.H Kodithuwakku	IT22544872	Business Performance Prediction
N.K.M.Perera	IT22151810	Customer Behavior Analysis & Sales Insights
D.D.M Jayasingha.	IT22162250	Product Demand Prediction
M.P.J.R.Dasanayaka	IT22152664	Smart Order Recommendation

R26-IT-026

B.Sc. (Hons) Degree in Information Technology Specializing in Information
Technology

Department of Information Technology

Sri Lanka Institute of Information Technology
Sri Lanka

September 2026

DECLARATION AND SUPERVISOR CERTIFICATION

Candidate declaration. We declare that this common report is our own work and does not incorporate, without acknowledgement, material previously submitted for a degree or diploma at any university or institute of higher learning. To the best of our knowledge, it contains no material previously published or written by another person except where acknowledgement is made in the text. We grant the Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute this report, in whole or in part, in print, electronic, or other media, while retaining the right to use the content in future work.

Academic integrity and AI disclosure. We have disclosed all use of generative AI and other automated tools in accordance with the module policy. We remain responsible for the accuracy, originality, citations, code, data, analyses, and conclusions in this report.

Student name	Student ID	Signature	Date
K.A.C.H Kodithuwakku	IT22544872		10/09/2026
N.K.M.Perera	IT22151810		10/09/2026
D.D.M Jayasingha	IT22162250		10/09/2026
M.P.J.R.Dasanayake	IT22152664		10/09/2026

Supervisor certification. The above candidates carried out this research project under my supervision.

Role and name	Signature	Date
Supervisor:Mr.Deemantha Siriwardana		10/09/2026
Co-supervisor:Ms. Ayesha Wijesooriya		10/09/2026

ABSTRACT

Micro, small, and medium-sized enterprises (MSMEs) frequently depend on point-of-sale systems to manage routine sales and transaction records, yet many such systems provide limited support for forecasting, profitability analysis, customer insights, and intelligent sales decisions. This research introduces Smart-Cost , an integrated intelligent POS and business analytics solution designed to transform routine operational data into actionable information for SME decision-making.

The proposed system combines four interconnected research components: Business Performance Prediction, Product Demand Prediction, Customer Behavior Analysis and Sales Insights, and Smart Order Recommendation. The solution integrates transaction processing, product and inventory management, cost recording, sales monitoring, and analytical functions within a unified architecture. Historical sales, cost, product, and transaction data are processed to forecast future product demand, estimate business performance and profitability, identify meaningful customer and sales patterns, and generate product recommendations during checkout.

The research methodology incorporated data preprocessing, predictive modelling, comparative model evaluation, profitability analysis, transaction-pattern analysis, association-rule mining, system integration, and experimental evaluation. The complete solution was implemented using React Native, Node.js, MongoDB, Python-based analytical services, REST APIs, and a web-based management interface. Evaluation of the integrated framework demonstrated the feasibility of combining predictive analytics and operational POS functionality to support inventory planning, profitability monitoring, sales analysis, customer understanding, and cross-selling decisions. The findings indicate that Smart-Cost can provide SMEs with a more accessible approach to data-driven business management while reducing dependence on separate analytical systems. Overall, the research demonstrates how an intelligent, integrated POS ecosystem can extend conventional transaction processing into practical decision support for resource-constrained SMEs.

Keywords: Smart-Cost , Intelligent POS, SME Analytics, Predictive Analytics, Machine Learning.

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to everyone who contributed to the successful completion of the Smart-Cost research project. First, we extend our deepest appreciation to our research supervisors, Mr. Deemantha Siriwardana and Ms. Ayesha Wijesooriya, for their continuous guidance, constructive feedback, encouragement, and academic support throughout the research process. Their advice was invaluable in refining the research problem, strengthening the proposed solution, evaluating the individual research components, and integrating them into a complete system. We gratefully acknowledge the Faculty of Computing, Sri Lanka Institute of Information Technology (SLIIT) for providing the academic environment, facilities, resources, and research guidance required to undertake this project. We also appreciate the support received during the research review and evaluation stages, including the feedback provided through project presentations and assessments. Such feedback contributed significantly to improving the technical implementation and research direction of Smart-Cost .

Our sincere thanks are extended to the SME owners, managers, staff members, and other participants who contributed their time, practical observations, and feedback during requirement gathering and system evaluation activities. Their perspectives helped us better understand the operational challenges faced by small businesses and supported the development of a solution aligned with practical business requirements. We also acknowledge the collective commitment of every member of the Smart-Cost research team. The successful integration of Business Performance Prediction, Product Demand Prediction, Customer Behavior Analysis and Sales Insights, and Smart Order Recommendation was made possible through continuous collaboration, knowledge sharing, technical development, testing, and problem solving among team members. Finally, we are deeply thankful to our families, friends, lecturers, and colleagues for their patience, encouragement, and support throughout the research journey. Their motivation helped us overcome the challenges encountered during the development and completion of this project.

TABLE OF CONTENTS

DECLARATION AND SUPERVISOR CERTIFICATION.....	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENTS.....	iv
LIST OF FIGURES.....	xii
LIST OF TABLES.....	xiv
LIST OF ABBREVIATIONS.....	xvi
1 INTRODUCTION AND PROJECT CONTEXT.....	1
1.1 Background and context.....	3
1.2 Stakeholders and user needs.....	6
1.3 Alignment with the Sustainable Development Goals.....	8
1.4 Purpose and scope of the common report.....	10
2 OVERALL PROBLEM DEFINITION AND FOUR SUBPROBLEMS.....	13
2.1 Overall problem definition.....	15
2.2 Root causes and overall research gap.....	16
2.3 Subproblem and component mapping.....	19
2.4 Overall aim and group objectives.....	21
2.5 Scope assumptions and constraints.....	23
3 OVERALL SOLUTION STRUCTURE.....	27
3.1 Integrated solution overview.....	27
3.2 High level architecture.....	29
3.3 Component 1(Business Performance Prediction) summary.....	31
3.4 Component 2(Behavior Analysis & Sales Insights) summary.....	33
3.5 Component 3(Product Demand Prediction) summary.....	35
3.6 Component 4(Smart Order Recommendation) summary.....	36

3.7 End to end user and data flow.....	38
4 COMPONENT INTEGRATION AND COHESIVE OPERATION.....	41
4.1 Integration strategy.....	41
4.2 Interfaces data contracts and shared models.....	43
4.3 Shared nonfunctional requirements.....	46
4.4 Integrated workflow and exception handling.....	49
4.5 Team contribution and dependency matrix.....	52
5 INTEGRATED IMPLEMENTATION AND PROJECT EXECUTION.....	55
5.1 Common technology stack and infrastructure.....	55
5.2 Development and integration process.....	57
5.3 Deployment and operational configuration.....	60
5.4 Evidence of team execution.....	63
6 INTEGRATED TESTING EVALUATION AND VALIDATION.....	67
6.1 Evaluation framework and success criteria.....	67
6.2 Integration and end to end testing.....	69
6.3 Overall results.....	72
6.4 Stakeholder and user validation.....	77
6.5 Achievement of common objectives.....	79
6.6 Integrated limitations and threats.....	82
7 COMMERCIALISATION SUSTAINABILITY AND RESPONSIBLE INNOVATION.....	85
7.1 Value proposition and beneficiaries.....	85
7.2 Market alternatives and differentiation.....	87
7.3 Business model and feasibility.....	90
7.4 Adoption intellectual property and compliance.....	93
7.5 Sustainable Development Goal contribution.....	96

8 CONCLUSION AND FUTURE DIRECTIONS.....	99
8.1 Overall conclusion.....	99
8.2 Principal integrated contribution.....	100
8.3 Future directions.....	101
REFERENCES.....	105
APPENDICES.....	108
Appendix A Integrated interface specification.....	108
Appendix B Team contribution declaration.....	109
B.1 Individual Contributions.....	109
B.2 Shared Contributions.....	109
B.3 Contribution Confirmation.....	110
Appendix C – Integrated System Test Matrix.....	110
Appendix D – Dataset and Experimental Data Description.....	114
D.1 Data Fields.....	114
D.2 Data Preparation.....	115
D.3 Data Limitations.....	115
Appendix E – Extended Component Evaluation Results.....	116
E.1 Business Performance Prediction.....	116
E.2 Behavior Analysis & Sales Insights.....	117
E.3 Product Demand Prediction.....	117
E.4 Smart Order Recommendation.....	118

LIST OF FIGURES

Figure 1:Overall Smart-Cost Research Framework.....	15
Figure 2:Smart-Cost Problem and Four-Component Mapping.....	21
Figure 3:Smart-Cost Overall System Architecture.....	27
Figure 4:Smart-Cost Four Research Components.....	30
Figure 5:Business Performance Prediction Workflow.....	33
Figure 6:Behavior Analysis and Sales Insights Workflow.....	34
Figure 7:Product Demand Prediction Workflow.....	36
Figure 8:Smart Order Recommendation Workflow.....	38
Figure 9:Smart-Cost End-to-End User and Data Flow.....	40
Figure 10:Smart-Cost Component Integration Architecture.....	42
Figure 11:Smart-Cost API and Service Communication Flow.....	43
Figure 12:Smart-Cost Shared Data Model.....	46
Figure 13:Integrated Smart-Cost Operational Workflow.....	52
Figure 14:Smart-Cost Technology Stack.....	57
Figure 15:Smart-Cost Development and Integration Process.....	60
Figure 16:Smart-Cost Cloud Deployment Architecture.....	63
Figure 17:Smart-Cost Mobile POS Interface.....	64
Figure 18:Smart-Cost Management and Analytics Dashboard.....	65
Figure 19:Smart-Cost Integrated Evaluation Framework.....	69
Figure 20:Smart-Cost End-to-End Testing Process.....	72
Figure 21:Product Demand Prediction Evaluation Results.....	74
Figure 22:Business Performance Prediction Evaluation Results.....	75
Figure 23:Smart Order Recommendation Evaluation Results.....	76
Figure 24:Smart-Cost User Evaluation Results.....	77
Figure 25:User Evaluation Mean Ratings.....	79
Figure 26:Achievement of Smart-Cost Common Objectives.....	82
Figure 27:Smart-Cost Value Proposition.....	87
Figure 28:Smart-Cost Business Model.....	93
Figure 29:Smart-Cost Sustainable Development Goal Alignment.....	98
Figure 30:Smart-Cost Future Development Roadmap.....	104

Figure 31:Business Performance Prediction Charts and Details.....	116
Figure 32:Behavior Analysis & Sales Insight's Charts and Details.....	117
Figure 33:Product Demand Prediction Charts and Details.....	117
Figure 34:Smart Order Recommendation's Charts and Details.....	118

LIST OF TABLES

Table 1:Smart-Cost Overall Research Gap Analysis.....	18
Table 2:Mapping of Research Subproblems to Smart-Cost Components.....	20
Table 3:Smart-Cost Group Objectives and Evaluation Evidence.....	23
Table 4:Scope, Assumptions and Constraints of Smart-Cost	25
Table 5:Smart-Cost Research Components and Responsibilities.....	29
Table 6:Inputs, Processing and Outputs of the Four Research Components.....	31
Table 7:Smart-Cost Shared Data Entities and Their Uses.....	44
Table 8:Smart-Cost Common Non-Functional Requirements.....	48
Table 9:Smart-Cost Exception Handling and Recovery Strategy.....	50
Table 10:Team Contribution and Dependency Matrix.....	53
Table 11:Smart-Cost Tools and Technologies.....	56
Table 12:Smart-Cost Development and Integration Activities.....	58
Table 13:Smart-Cost Deployment and Operational Configuration.....	62
Table 14:Smart-Cost Evaluation Framework and Success Criteria.....	67
Table 15:Smart-Cost Integration and End-to-End Test Cases.....	70
Table 16:Overall Smart-Cost Experimental Results.....	73
Table 17:Smart-Cost User Evaluation Results.....	78
Table 18:Achievement of Smart-Cost Common Objectives.....	81
Table 19:Integrated Limitations and Threats to Validity.....	84
Table 20:Smart-Cost Beneficiaries and Value Proposition.....	86
Table 21:Comparison of Smart-Cost with Existing Solution Categories.....	89
Table 22:Smart-Cost Business Model and Feasibility.....	91
Table 23:Smart-Cost Adoption, IP and Compliance Considerations.....	95
Table 24:Smart-Cost Alignment with Sustainable Development Goals.....	96
Table 25:Smart-Cost Future Development Priorities.....	102
Table 26:Integrated interface specification.....	108
Table 27:Individual Contributions.....	109
Table 28:Contribution Confirmation.....	110
Table 29:Integrated System Test Matrix.....	110
Table 30:definition of Completion (%).....	113

Table 31:Dataset and Experimental Data Description.....	114
---	-----

LIST OF ABBREVIATIONS

Abbreviation	Meaning
AI	Artificial Intelligence
API	Application Programming Interface
ARIMA	Auto Regressive Integrated Moving Average
CRUD	Create, Read, Update and Delete
DB	Database
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IP	Intellectual Property
JSON	JavaScript Object Notation
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MSME	Micro, Small and Medium Enterprise
NFR	Non-Functional Requirement
POS	Point of Sale
PP1	Progress Presentation 1
PP2	Progress Presentation 2
REST	Representational State Transfer
RMSE	Root Mean Square Error
SDG	Sustainable Development Goal
SME	Small and Medium Enterprise
SMS	Short Message Service
UI	User Interface
UX	User Experience

1 INTRODUCTION AND PROJECT CONTEXT

Micro, small, and medium-sized enterprises (MSMEs) contribute significantly to economic activity by creating employment, supporting local supply chains, and providing goods and services to communities [1]–[3]. Despite their importance, many small businesses operate with limited technological, financial, and human resources, which can restrict their ability to adopt advanced digital solutions [2], [3]. Business owners are often required to make daily decisions regarding inventory, product demand, pricing, expenses, customer preferences, and profitability using experience or manually prepared records. Although digital technologies and point-of-sale (POS) systems have improved the ability of businesses to record and manage operational information, smaller enterprises can still experience difficulties in converting accumulated data into information that supports more advanced decision-making [2], [19].

This limitation creates several interconnected operational challenges. Uncertainty regarding future product demand can result in excessive purchasing, insufficient stock, product shortages, or unnecessary waste, particularly for businesses handling perishable products. Time-series forecasting provides one approach for extracting patterns from historical observations and using them to estimate future requirements [5]–[8]. Changes in ingredient and operational costs can also affect profitability without being immediately visible to owners. At the same time, transaction histories contain potentially valuable information about customer purchasing patterns and relationships between products. Data-mining techniques can be applied to transactional datasets to discover patterns that may not be immediately visible through conventional transaction records [13], [15], [18]. Consequently, SMEs may possess considerable operational data while still having limited capability to convert those data into systematic decision support [2], [19].

To address these challenges, this research proposes Smart-Cost, an intelligent POS and business analytics solution designed for resource-constrained SMEs. Rather than treating POS operations and business analytics as separate functions, Smart-Cost integrates routine transaction management with predictive and analytical capabilities within a unified system. The solution is structured around four interconnected research components: Business Performance Prediction, Product Demand Prediction, Customer Behavior Analysis and Sales Insights, and Smart Order Recommendation. Together, these components transform historical sales, cost, product, and transaction information into decision-support outputs that can assist business owners with planning and daily operations.

The Business Performance Prediction component focuses on analysing financial and operational information to provide insights into expected profitability and business performance. The Product Demand Prediction component uses historical sales patterns and forecasting techniques to estimate future product requirements. Established time-series forecasting methods provide mechanisms for learning temporal patterns from historical observations and evaluating the resulting forecasts [5]–[8]. Within Smart-Cost, this capability is intended to support inventory and production planning. The Customer Behavior Analysis and Sales Insights component transforms transaction records into interpretable information regarding purchasing behaviour, sales patterns, and product performance. Data-mining approaches provide a foundation for extracting useful patterns and knowledge from accumulated business data [15], [18], [19].

The Smart Order Recommendation component identifies relationships between products that occur together within historical transactions and uses those relationships to generate relevant recommendations during the sales process. Association-rule mining was originally developed to discover relationships between sets of items within large transactional databases [13], while measures and algorithms for identifying useful associations have subsequently become established techniques for market-basket analysis [14], [15], [18]. Recommendation-system research further

demonstrates how historical interaction information can be used to support personalized or contextually relevant suggestions [16], [17]. Within Smart-Cost, these techniques provide the analytical foundation for identifying cross-selling opportunities during POS transactions.

Smart-Cost is implemented through an integrated technological architecture combining a mobile POS application, backend services, database management, analytical processing, and management interfaces. React Native supports the mobile application, Node.js manages core operational services, MongoDB stores business information, and Python-based services perform predictive and analytical processing. These components communicate through RESTful APIs, allowing analytical capabilities to operate alongside routine POS functions. The modular separation of system functionality also supports software qualities such as maintainability, compatibility, reliability, and scalability, which are relevant considerations in software-system quality models [21].

The overall purpose of this research is therefore to investigate how an integrated intelligent POS ecosystem can provide practical, accessible, and data-driven decision support for SMEs. By combining operational management with forecasting, profitability analysis, customer insights, and product recommendations, Smart-Cost aims to extend the role of conventional POS technology from transaction recording toward intelligent business decision support. This direction is consistent with the broader digital transformation of SMEs, in which effective adoption and use of digital technologies can strengthen organizational capabilities and support business development [2].

1.1 Background and context

Micro, small, and medium-sized enterprises (MSMEs) form an important part of the business environment by supporting employment, entrepreneurship, local production, and the delivery of goods and services [1]–[3]. In Sri Lanka, small-scale businesses operate within an environment where SMEs contribute to employment, regional

development, entrepreneurship, and economic activity [1]. At the same time, smaller enterprises can face financial, technological, and organizational constraints that influence their ability to adopt and benefit from digital technologies [2], [3]. Effective management of sales, inventory, operating costs, customer demand, and profitability is therefore particularly important for maintaining sustainable business operations.

Digital technologies provide businesses with mechanisms for improving the collection, management, and use of operational information [2]. A typical digital POS environment can support activities such as recording sales, maintaining product information, calculating bills, updating inventory quantities, and preserving transaction histories. However, collecting operational information does not automatically result in better data-driven decisions. The value of business data depends on the ability to analyse those data and convert them into information that can support decision-making [18], [19]. Historical transaction records may contain patterns relating to demand, customer purchasing behaviour, product performance, and relationships among products, but such patterns can remain underutilized when businesses lack appropriate analytical capabilities [13], [15], [18].

This limitation is especially relevant to businesses dealing with products whose demand changes over time. Forecasting methods provide a systematic approach for analysing historical time-series observations and estimating future values [5]–[7]. A bakery or café, for example, must determine how much of each product should be prepared or stocked without knowing the exact demand that will occur during the following days or weeks. Inaccurate estimates may contribute to either excessive inventory or insufficient product availability. Forecast accuracy can be assessed using established error measures that quantify differences between observed and predicted values [8]. At the same time, changes in ingredient prices and other operating expenses can affect product margins even when sales remain comparatively strong. Business owners therefore benefit from understanding not only historical sales but also demand patterns, costs, profitability, and emerging business trends.

Customer transactions represent another valuable source of business knowledge. Data-mining methods can be used to identify patterns within accumulated transactional information [15], [18], [19]. Repeated purchases can reveal product popularity, purchasing patterns, and relationships among products. In particular, association-rule mining was introduced as a method for discovering relationships among items appearing within transaction databases [13], and subsequent approaches have provided efficient mechanisms for identifying frequent item relationships [14]. These relationships can support market-basket analysis and provide a foundation for product recommendation [15]–[18]. Nevertheless, implementing forecasting, profitability analytics, customer analysis, and recommendation technologies as independent systems can increase technological and management complexity for resource-constrained SMEs [2].

The Smart-Cost research project was initiated within this context to investigate an integrated approach to intelligent POS and business decision support. The proposed platform combines normal POS operations with four connected analytical components: Business Performance Prediction, Product Demand Prediction, Customer Behavior Analysis and Sales Insights, and Smart Order Recommendation. Transaction, product, sales, cost, and purchasing information generated through business operations can consequently serve as inputs to multiple analytical processes rather than remaining solely as historical records.

The primary stakeholders of Smart-Cost are SME owners, managers, cashiers, and operational staff. Owners and managers require information for planning and monitoring, while frontline users require a POS environment that remains practical for everyday transactions. By integrating operational functionality with predictive and analytical capabilities, Smart-Cost establishes a common technological foundation for investigating how routine SME data can be transformed into accessible information for inventory planning, profitability monitoring, customer understanding, sales improvement, and informed business decision-making.

1.2 Stakeholders and user needs

The development of Smart-Cost requires consideration of stakeholders who interact directly with the system as well as those who benefit from the information generated through its analytical components. Since the proposed solution is intended primarily for micro, small, and medium-sized enterprises (MSMEs), stakeholder requirements are influenced by practical factors such as ease of use, affordability, limited technical expertise, operational efficiency, and the need for understandable business information. SMEs can face resource and capability constraints that influence their adoption and effective use of digital technologies [1]–[3]. Requirement-gathering and stakeholder engagement conducted during the Smart-Cost research process were therefore used to identify the operational and analytical needs that guided the design of the integrated solution.

The primary stakeholders are business owners and managers. They are responsible for monitoring sales, controlling expenditure, managing inventory, understanding business performance, and making operational decisions. Their requirements extend beyond conventional transaction records. Effective data-driven decision-making depends on transforming accumulated operational data into information that can support business decisions [18], [19]. Within the Smart-Cost context, owners and managers require clear information about product demand, expected business performance, costs, profitability, sales trends, and customer purchasing behaviour. Accordingly, Smart-Cost provides dashboards and analytical functions that transform operational data into information that can support planning and decision-making. Demand forecasting provides a systematic mechanism for estimating future values from historical observations [5]–[8] and can therefore support stock and production planning. Business Performance Prediction and cost-related information are intended to provide additional support for profitability monitoring.

Cashiers and operational staff represent another primary stakeholder group because they interact directly with the POS application during everyday business activities. Their principal requirements include a simple interface, fast product selection, efficient billing, accurate transaction processing, and straightforward access to frequently used functions. Usability, reliability, performance efficiency, and functional suitability are

recognized software-quality considerations for interactive software systems [21]. Smart-Cost therefore incorporates product search, cart management, quantity adjustments, discounts, multiple payment records, receipt handling, order history, and automatic inventory updates. Barcode- and QR-based product identification can further reduce the effort required during transaction processing. The Smart Order Recommendation component also operates within this workflow by presenting relevant product suggestions without requiring users to access a separate analytical application. Recommendation systems provide mechanisms for using available data to identify and present items that may be relevant within a particular user or transaction context [16], [17].

A further stakeholder category consists of customers of participating SMEs. Although customers do not manage the Smart-Cost platform, they may indirectly benefit from efficient checkout processes, relevant product recommendations, digital receipt facilities, and improved product availability. Where customer information is collected with appropriate consent, purchasing records can also contribute to customer and sales analysis. Data-mining techniques can support the identification of patterns and relationships within accumulated transactional information [15], [18]. Consequently, privacy, responsible information handling, and appropriate access controls are important system considerations. Information-security frameworks emphasize the importance of systematically managing the confidentiality, integrity, and availability of organizational information [25], while responsible AI frameworks emphasize appropriate governance and management of risks associated with AI-enabled systems [22], [23].

The research team and system administrators can be considered secondary stakeholders. They require reliable data storage, maintainable services, analytical model integration, secure communication, and mechanisms for managing businesses within the platform. Maintainability, reliability, compatibility, security, and performance efficiency are relevant software-system quality characteristics [21]. The modular Smart-Cost architecture using React Native, Node.js, MongoDB, Python-based analytical services, and REST APIs was selected to support these technical

requirements while allowing the four research components to operate within a shared environment.

Overall, stakeholder engagement during the research process indicated a common requirement for a solution that remains simple enough for routine POS operations while providing meaningful analytical support. These needs shaped Smart-Cost as an integrated platform combining Business Performance Prediction, Product Demand Prediction, Customer Behavior Analysis and Sales Insights, and Smart Order Recommendation, enabling different stakeholders to obtain relevant operational and decision-support capabilities through a unified system.

1.3 Alignment with the Sustainable Development Goals

The Smart-Cost research project is aligned with selected United Nations Sustainable Development Goals (SDGs) through its intended contribution to digital transformation, informed business decision-making, and the sustainable operation of micro, small, and medium-sized enterprises (MSMEs). The United Nations 2030 Agenda establishes 17 Sustainable Development Goals addressing interconnected economic, social, and environmental development priorities [4]. Within the context of Smart-Cost, the strongest alignment is with SDG 8: Decent Work and Economic Growth, while the project also has relevance to SDG 9: Industry, Innovation and Infrastructure and SDG 12: Responsible Consumption and Production [4], [24]. These relationships represent the intended contribution of the proposed system rather than evidence that Smart-Cost has independently achieved the broader targets defined under these SDGs.

SDG 8: Decent Work and Economic Growth is particularly relevant because Smart-Cost is designed to improve access to digital business-management capabilities for resource-constrained SMEs. SMEs play an important role in employment, entrepreneurship, and economic activity, while limitations in resources and digital capabilities can influence their ability to benefit from technological transformation [1]–[3]. Smart-Cost combines routine POS functionality with Business Performance Prediction, Product Demand Prediction, Customer Behavior Analysis and Sales Insights, and Smart Order Recommendation. By converting operational data into

understandable decision-support information, the system is intended to help owners and managers make more informed planning decisions and improve their ability to monitor business activities. This objective is consistent with SDG 8, which includes the promotion of productive activities, entrepreneurship, innovation, and the growth of micro, small, and medium-sized enterprises [4], [24].

The project also aligns with SDG 9: Industry, Innovation and Infrastructure through the application of digital and intelligent technologies within SME operations. SDG 9 emphasizes resilient infrastructure, sustainable industrialization, innovation, and increased access to information and communication technologies [4]. Digital transformation can provide SMEs with opportunities to improve organizational capabilities, processes, and access to data-driven services, although smaller enterprises may experience barriers to technology adoption [2]. Smart-Cost integrates mobile and web technologies, cloud-based data management, application programming interfaces, and predictive and analytical techniques within a unified POS environment. Instead of requiring multiple independent applications for transaction processing and analytics, the proposed architecture enables operational and analytical services to work together. The research therefore investigates a practical approach through which digital and analytical technologies can be adapted to the operational requirements of smaller enterprises.

A further relationship exists with SDG 12: Responsible Consumption and Production, which promotes more sustainable patterns of consumption, production, and resource use [4], [24]. Product Demand Prediction within Smart-Cost is intended to provide information that can support better inventory and production planning. Forecasting methods use historical observations to estimate future values and can therefore provide information for planning under uncertain future demand [5]–[8]. This capability is particularly relevant to bakeries, cafés, restaurants, and similar businesses where inaccurate demand estimation may contribute to excess preparation, unsold products, or unnecessary inventory. Forecasting future demand can potentially help businesses align purchasing and production decisions more closely with expected requirements. Business performance and cost analytics may similarly provide additional visibility into resource-related expenditure and operational efficiency.

Overall, Smart-Cost contributes to the objectives represented by SDGs 8, 9, and 12 by investigating how an integrated intelligent POS platform can make data-driven business capabilities more accessible to SMEs [4], [24]. However, the research does not claim direct achievement of individual SDG targets. Rather, it establishes a technological and analytical foundation that may support economic resilience, digital innovation, and more informed resource-management practices when appropriately adopted and validated in real SME environments.

1.4 Purpose and scope of the common report

The purpose of this common report is to present the Smart-Cost research project as a single integrated solution by describing the shared research problem, overall system objectives, common architecture, development approach, integration strategy, and collective contribution of the four research components. While each component addresses a distinct research problem and is investigated in greater technical depth within the respective individual thesis, this report focuses on how those components operate together to address the broader challenges encountered by micro, small, and medium-sized enterprises (MSMEs). SMEs can experience financial, technological, and organizational constraints that affect their ability to adopt and effectively utilize advanced digital technologies [1]–[3].

Smart-Cost was designed as an intelligent point-of-sale and business analytics platform that extends conventional transaction-processing capabilities with data-driven decision support. The common research problem concerns the limited ability of transaction-oriented business environments to transform routinely collected operational data into information that can assist with future planning. Data analytics and data-mining approaches demonstrate that accumulated organizational and transactional data can be transformed into patterns and knowledge that support business decision-making [15], [18], [19]. Sales transactions, product information, inventory records, business costs, and purchasing patterns can therefore provide valuable analytical inputs; however, their usefulness may be reduced when they remain distributed across separate processes or are used primarily for historical reporting. The integrated Smart-Cost solution establishes a common technological environment in which operational and analytical functions can work together.

The scope of this report covers the integration of four principal research components: Business Performance Prediction, Product Demand Prediction, Customer Behavior Analysis and Sales Insights, and Smart Order Recommendation. Business Performance Prediction focuses on financial and profitability-related decision support. Product Demand Prediction investigates forecasting methods for estimating future product requirements; time-series forecasting provides established techniques for using historical observations to estimate future values [5]–[8]. Customer Behavior Analysis and Sales Insights examines transaction information to identify meaningful purchasing and sales patterns, drawing on the broader principles of data mining and business-oriented data analysis [15], [18], [19]. Smart Order Recommendation analyses relationships among products to provide relevant recommendations during the checkout process. Association-rule mining provides established methods for identifying relationships among items occurring within transaction datasets [13], [14], while recommender-system research provides techniques for generating useful item suggestions from available interaction and transaction information [16], [17]. Although these four components have separate research objectives, methodologies, algorithms, experiments, and evaluation criteria, they share operational data and contribute collectively to the Smart-Cost decision-support environment.

The common report also describes the system-level implementation required to connect these research contributions. This includes the mobile POS application, backend services, database infrastructure, analytical services, APIs, authentication and business-level data handling, dashboards, inventory operations, billing, transaction management, and communication between the operational and analytical layers. Technologies including React Native, Node.js, MongoDB, Python-based analytical services, and RESTful APIs provide the technical foundation for this integration. At the system level, characteristics such as functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability, and portability provide relevant considerations when evaluating software quality [21].

Detailed algorithm development, model configuration, experimental procedures, component-specific datasets, evaluation metrics, results, and technical discussions remain within the respective individual theses. Consequently, the common report does

not replace or duplicate those individual investigations. Instead, it provides the system-level perspective required to demonstrate how the four independently researched contributions form one coherent solution.

The scope of the report is limited to the research, design, implementation, integration, and evaluation activities completed within the project period up to PP2. Its primary objective is therefore to document Smart-Cost as a unified research outcome and establish the relationship between the common SME problem, the integrated technological solution, and the four individual research contributions.

2 OVERALL PROBLEM DEFINITION AND FOUR SUBPROBLEMS

The Smart-Cost research project addresses a common challenge faced by micro, small, and medium-sized enterprises (MSMEs): although businesses continuously generate operational data through sales, inventory movements, product costs, and customer transactions, this information is not always effectively transformed into actionable knowledge for business decision-making [2], [18], [19]. Conventional business information systems can provide transaction recording and historical information, while data-mining and analytical approaches are required to extract additional patterns and decision-support knowledge from accumulated data [15], [18], [19]. Consequently, business owners may have access to historical information about what has already occurred while receiving limited analytical support for understanding future conditions.

This overall problem involves several interconnected areas of business management. One concern is the difficulty of understanding and anticipating overall business performance. Changes in sales, product costs, expenses, and profit margins can influence the financial position of an enterprise, while effective data-driven decision-making requires these variables to be interpreted as useful business information [19]. Business owners therefore require meaningful information that extends beyond total revenue and historical transaction summaries.

A second concern is uncertainty regarding future product demand. SMEs must frequently determine purchasing, stocking, or production quantities without reliable estimates of future sales. Historical product sales can be analysed using time-series forecasting approaches to estimate future observations and support planning [5]–[8]. Overestimation may contribute to unnecessary inventory and potential waste, while underestimation may create shortages and missed sales opportunities. Appropriate forecasting methods are therefore important for transforming historical patterns into information relevant to future decisions [5]–[8].

The third concern relates to customer behaviour and sales information. Transaction records can contain evidence regarding purchasing tendencies, product performance, sales periods, and recurring customer activity. Data-mining techniques provide mechanisms for discovering patterns and useful knowledge within such datasets [15], [18], [19]. However, raw transaction histories can become difficult to interpret as transaction volumes increase.

The fourth concern involves identifying opportunities to recommend relevant products during transactions. Customers may purchase particular products together, and association-rule mining provides established methods for identifying relationships among items within transactional datasets [13], [14]. Such relationships can also contribute to recommendation mechanisms that provide relevant item suggestions [16], [17].

To investigate the overall problem systematically, Smart-Cost decomposes it into four non-overlapping but interdependent research subproblems. These are addressed through Business Performance Prediction, Product Demand Prediction, Customer Behavior Analysis and Sales Insights, and Smart Order Recommendation. Each component has a distinct research purpose, methodology, analytical process, and evaluation approach while contributing to the same integrated business environment.

The four-component structure is necessary because no individual analytical function can address the complete decision-making problem. Demand forecasting alone cannot provide comprehensive profitability information; business-performance analysis does not automatically reveal customer purchasing behaviour; customer analytics does not directly predict future product requirements; and recommendation techniques do not independently support financial or inventory planning. Smart-Cost therefore investigates these concerns collectively while preserving the individual research contribution of each component.

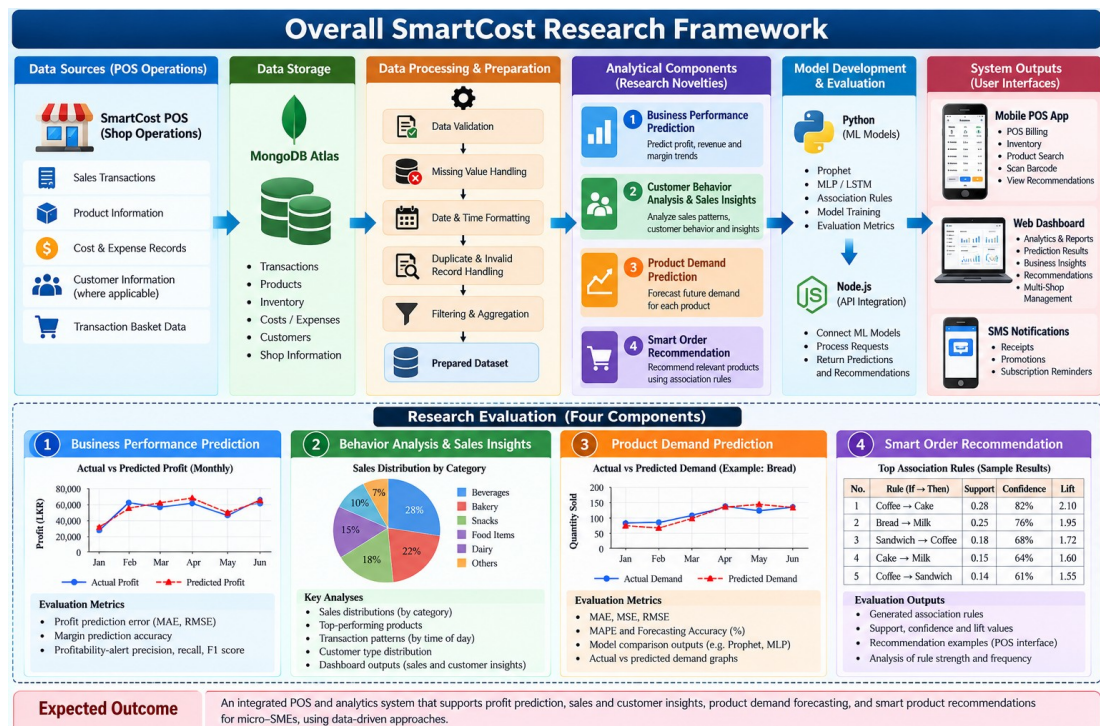


Figure 1:Overall Smart-Cost Research Framework

2.1 Overall problem definition

MSMEs generate substantial operational information through everyday activities such as sales transactions, inventory movements, product purchases, customer orders, and cost recording. SMEs nevertheless face resource, capability, and digitalisation constraints that can influence their ability to effectively exploit business data [1]–[3]. The central research problem addressed by Smart-Cost is therefore the limited ability of MSMEs to systematically use their operational data to anticipate demand, understand business performance, interpret customer and sales behaviour, and identify relevant sales opportunities within routine business processes.

This problem extends beyond basic transaction recording. Historical information can indicate what has already occurred, while data analytics attempts to transform accumulated information into knowledge useful for business decisions [18], [19]. Owners must decide what quantities should be stocked or produced, whether changing sales and costs are likely to affect profitability, which customer and product patterns require attention, and which additional products may be relevant during a transaction. Demand uncertainty represents one dimension of the common problem. SMEs must frequently make purchasing, production, and inventory decisions without reliable

estimates of future product requirements. Forecasting provides systematic techniques for estimating future observations using historical information [5]–[8]. Excessive stock or production can create unnecessary holding costs and waste, particularly for businesses handling perishable goods, while insufficient quantities may lead to shortages and missed sales opportunities.

Financial visibility represents another dimension. Revenue alone does not provide a complete indication of business performance because business decisions can require consideration of multiple financial and operational variables. Data-driven business analysis provides mechanisms for examining such variables collectively rather than relying only on isolated transaction totals [19].

The common problem also includes limited exploitation of customer and transaction patterns. Data-mining methods can discover patterns within accumulated transaction information [15], [18], while association-rule methods can identify relationships among products appearing within transaction baskets [13], [14]. Consequently, analytical use of POS data can support improved understanding of purchasing behaviour and potential cross-selling opportunities [16], [17].

These challenges are particularly significant when digital and analytical capabilities require resources or expertise that smaller enterprises may not possess [2], [3]. The research problem is therefore not merely the absence of POS technology, but the gap between collecting operational business data and converting those data into coordinated, forward-looking decision support.

2.2 Root causes and overall research gap

The overall problem addressed by this research originates from several interconnected limitations in the way operational data are collected, analysed, and applied within resource-constrained MSMEs. SMEs can encounter barriers associated with financial resources, skills, organizational capabilities, and adoption of digital technologies [1]–[3]. A primary concern within Smart-Cost is therefore the difference between recording operational transactions and obtaining analytical decision support from those records.

A second cause is the limited use of forecasting in routine SME operations. Product demand can change over time, and systematic forecasting techniques can use historical observations to estimate future values [5]–[8]. When future requirements are estimated primarily through personal experience or recent observations, businesses may lack a systematic quantitative basis for planning. Incorporating forecasting into an accessible SME workflow therefore represents an important research consideration.

Limited financial interpretation represents another root cause. Sales information alone does not capture every variable relevant to business decision-making. Data-driven business analysis emphasizes the extraction and interpretation of useful knowledge from organizational data to support decisions [19].

Accumulated transaction records also create a potentially underused source of customer and sales knowledge. Data-mining methods provide techniques for identifying patterns within large datasets [15], [18], while association-rule mining can reveal recurring relationships among products within transaction records [13], [14]. Recommendation-system approaches can subsequently use available information to support relevant item suggestions [16], [17].

Existing analytical approaches can address these concerns through different methodological areas: forecasting research provides methods for future-value estimation [5]–[8], data-mining research provides techniques for discovering behavioural and transactional patterns [15], [18], and recommendation research addresses identification and presentation of relevant items [16], [17]. Smart-Cost identifies a systems-level research opportunity in bringing these complementary analytical capabilities into a shared SME POS environment.

The Smart-Cost research addresses this gap through four coordinated research areas rather than assuming that a single analytical technique can resolve the entire problem. Their integration provides the basis for examining whether routinely generated POS data can support several related business decisions through one shared technological environment.

Table 1: Smart-Cost Overall Research Gap Analysis

Area	Existing Situation / Limitation	Identified Research Gap	Smart-Cost Response
POS Operations	Conventional POS systems mainly record sales and transactions	Limited integration between operational data and intelligent decision support	Integrates POS operations with analytics and AI-driven components
Business Performance	Historical sales and costs may be available, but future performance is difficult to estimate	Lack of accessible predictive business-performance analysis for micro-SMEs	Business Performance Prediction
Customer & Sales Analysis	Transaction data is collected but often underutilized	Limited interpretation of purchasing behavior and sales patterns	Behavior Analysis & Sales Insights
Demand Planning	Inventory decisions often depend on experience or historical summaries	Limited product-level future-demand forecasting	Product Demand Prediction

Product Recommendations	POS systems generally focus on completing the current transaction	Limited intelligent use of basket relationships for cross-selling	Smart Order Recommendation
Integration	Analytical capabilities may exist as separate tools	Lack of a unified operational and analytical environment	Four components integrated with the Smart-Cost POS platform

2.3 Subproblem and component mapping

The overall research problem was decomposed into four distinct but interdependent subproblems. This decomposition enables each research component to address a specific analytical requirement while contributing to the common objective of transforming operational POS data into useful decision-support information.

The first subproblem concerns limited visibility into future business performance and profitability. The Business Performance Prediction System addresses this subproblem by analysing relevant operational and financial information to estimate performance and provide profitability-oriented insights. Data-oriented business analysis provides a broader foundation for transforming operational variables into information useful for decision-making [19].

The second subproblem is uncertainty regarding future product demand. Historical time-series information can provide the basis for estimating future observations through forecasting methods [5]–[8]. The Product Demand Prediction component therefore applies forecasting approaches to historical product-level sales data to support short- and longer-term demand understanding and inventory and production planning.

The third subproblem concerns limited interpretation of customer behaviour and overall sales patterns. Data mining provides established methods for discovering

useful patterns within accumulated data [15], [18]. The Customer Behavior Analysis and Sales Insights component applies this analytical perspective to transaction and sales information to provide understandable business insights.

The fourth subproblem relates to missed transaction-level cross-selling opportunities. Association-rule mining provides methods for identifying frequently occurring relationships between products within transactional data [13], [14]. Measures such as support, confidence, and lift are commonly used when evaluating such relationships [15], [18]. Recommendation-system principles additionally provide a basis for converting available information into relevant item suggestions [16], [17]. These concepts provide the methodological foundation for the Smart Order Recommendation System.

Table 2: Mapping of Research Subproblems to Smart-Cost Components

Subproblem	Smart-Cost Component	Main Input	Main Output
Limited visibility into future business performance	Business Performance Prediction	Sales, costs, prices, expenses	Performance and profitability predictions
Difficulty interpreting customer and sales patterns	Behavior Analysis & Sales Insights	Transactions, customers, products, timestamps	Customer and sales insights
Difficulty estimating future product demand	Product Demand Prediction	Historical product sales	Future product-demand forecasts

Missed cross-selling opportunities	Smart Order Recommendation	Transaction baskets	Product recommendations
------------------------------------	----------------------------	---------------------	-------------------------

Although the four subproblems are non-overlapping in their primary research purpose, they are operationally interdependent within Smart-Cost. Their combination establishes a broader decision-support framework in which predictive, descriptive, financial, and recommendation-oriented capabilities complement one another while maintaining clearly defined individual research contributions.

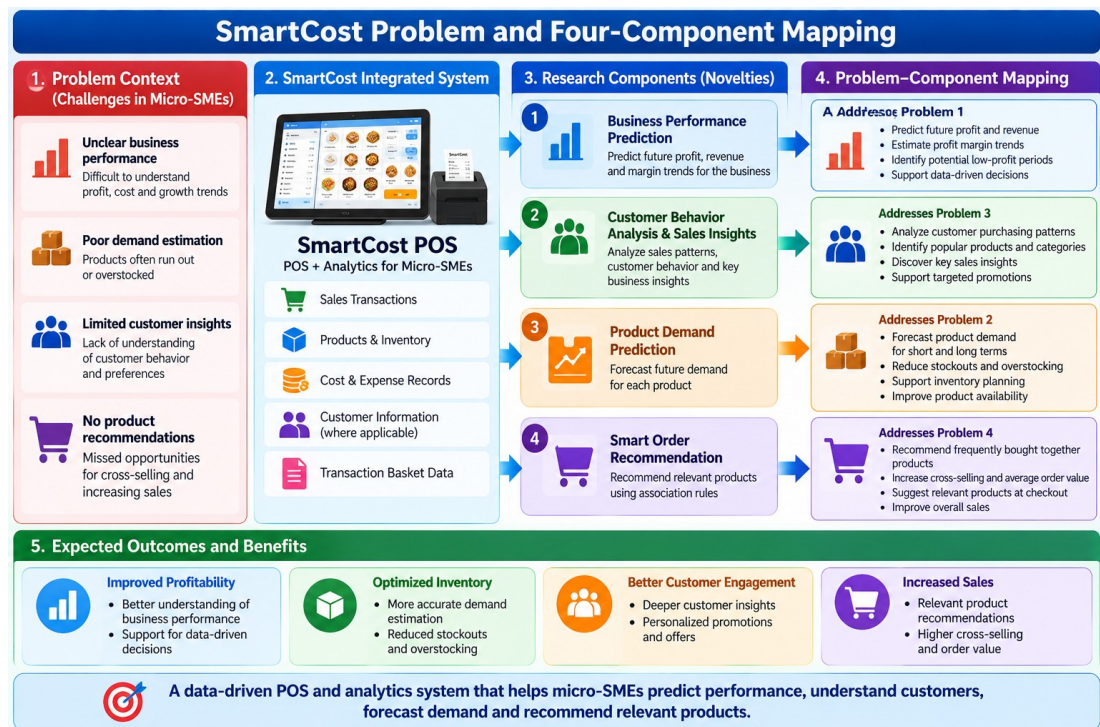


Figure 2: Smart-Cost Problem and Four-Component Mapping

2.4 Overall aim and group objectives

The overall aim of the Smart-Cost research project is to design, develop, integrate, and evaluate an intelligent POS and business analytics framework that transforms routinely generated SME operational data into accessible decision-support information for demand planning, business-performance assessment, customer and sales analysis, and transaction-level product recommendation.

The first group objective is to establish an integrated operational data environment capable of capturing and managing the information required by all four research components. This includes product records, sales transactions, quantities, costs, inventory information, timestamps, and relevant customer or transaction-basket information.

The second objective is to develop and evaluate a product-level demand prediction capability using historical sales information. Forecasting models should be evaluated using suitable error measures [8]. Accordingly, Smart-Cost uses measures including Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) as part of its component-level evaluation.

The third objective is to provide data-driven assessment of business performance and profitability by combining operational sales and cost information. This objective is addressed through the Business Performance Prediction component.

The fourth objective is to identify meaningful customer and sales patterns from accumulated transaction data and present them as interpretable business insights. Data-mining principles provide the methodological basis for extracting such patterns from accumulated datasets [15], [18], [19].

The fifth objective is to generate relevant product recommendations from transaction-level purchasing relationships. The Smart Order Recommendation component evaluates such relationships using association-rule measures including support, confidence, and lift [13]–[15], [18], together with component-specific recommendation evidence.

A further group-level objective is to integrate the four independently researched analytical components into a unified Smart-Cost environment without disrupting routine POS operations. Finally, the project aims to evaluate the usability and practical suitability of the integrated solution for its intended SME context. Software quality characteristics such as functional suitability, usability, reliability, performance efficiency, security, and maintainability provide relevant considerations for evaluating an integrated software solution [21].

Table 3: Smart-Cost Group Objectives and Evaluation Evidence

Objective	Intended Outcome	Evaluation Evidence
Develop a shared POS environment	Integrated operational data collection	Functional and integration testing
Predict product demand	Product-level future-demand forecasts	MAE, MSE/RMSE and MAPE where available
Predict business performance	Profitability and performance insights	Prediction error, margin accuracy and alert precision
Analyze customer and sales behavior	Actionable behavioral and sales insights	Analytical outputs and dashboards
Generate smart recommendations	Relevant product recommendations	Support, confidence and lift
Integrate all components	Unified Smart-Cost system	End-to-end integration tests
Evaluate usability	Determine practical suitability	User evaluation

2.5 Scope assumptions and constraints

The scope of Smart-Cost is defined around the development and evaluation of an integrated intelligent POS and analytics framework for MSMEs. SMEs contribute to entrepreneurship and economic activity while also experiencing resource and technology-adoption constraints [1]–[3]. The research up to PP2 focuses on establishing and evaluating the proposed Smart-Cost system and its four analytical

components rather than demonstrating long-term economic impact across the entire SME sector.

The functional scope includes product management, POS sales processing, cart and billing operations, payment recording, transaction history, inventory updates, cost-related information, analytical dashboards, and the data flows required by the research components. The analytical scope consists of Business Performance Prediction, Product Demand Prediction, Customer Behavior Analysis and Sales Insights, and Smart Order Recommendation.

Several assumptions influence the research. The analytical components assume that transaction and product information is recorded with sufficient accuracy and consistency. Demand forecasting assumes that historical observations contain information relevant to future values, while forecasting performance depends on the characteristics and quality of the available time series [5]–[8]. Customer and sales analysis similarly depends on the availability and quality of transactional information [15], [18]. Recommendation generation requires sufficient transactional evidence to identify meaningful product relationships [13]–[17].

The project is also subject to operational and technical constraints. Resource-constrained SMEs may face limitations involving technical expertise, digital capabilities, infrastructure, and financial resources [1]–[3]. Prediction quality can additionally be affected by the available historical information and changing patterns, meaning that forecasting performance must be empirically evaluated rather than assumed [6]–[8].

The research does not assume that analytical predictions eliminate uncertainty or replace managerial judgement. Responsible AI principles emphasize the importance of managing risks associated with AI-enabled systems and maintaining appropriate human oversight [22], [23]. Smart-Cost therefore treats forecasts, profitability estimates, customer insights, and recommendations as decision-support outputs rather than guaranteed business outcomes.

Similarly, simulation-based or experimental findings obtained during the research period should not automatically be interpreted as long-term causal improvements in

deployed SMEs. The scope up to PP2 excludes comprehensive longitudinal validation across large numbers of businesses, nationwide generalization, and exhaustive modelling of external influences unless explicitly incorporated within an individual research component. These boundaries allow the research to concentrate on the feasibility, integration, and evaluation of the four principal analytical contributions.

Table 4: Scope, Assumptions and Constraints of Smart-Cost

Category	Description
Business Scope	Micro and small businesses, particularly bakery, café, restaurant and retail environments
Functional Scope	POS, billing, products, inventory, costs, customers, reports and analytics
Research Scope	Four integrated intelligent research components
Data Assumption	Transaction and operational records are sufficiently accurate and complete
Historical Data Assumption	Sufficient historical data is available for analytical processing
Basket Assumption	Transactions contain meaningful product combinations for recommendation analysis
Data Constraint	Missing, inconsistent or limited records may reduce analytical reliability
Infrastructure Constraint	Cloud/network availability may influence access to online services

Research Constraint	PP2 evaluation represents research- prototype and experimental validation
Generalization Constraint	Experimental/simulated results cannot establish performance for all real-world SMEs

3 OVERALL SOLUTION STRUCTURE

3.1 Integrated solution overview

Smart-Cost is designed as an integrated intelligent Point-of-Sale (POS) and business analytics platform that combines routine SME operational activities with predictive and analytical decision support. The broader digital transformation of SMEs demonstrates the potential for digital technologies to improve organizational capabilities and the use of business information [2]. Rather than developing four independent applications, the project establishes a shared technological environment in which transactional data generated through normal business operations can be processed by four interconnected research components. The central value proposition of Smart-Cost is therefore to transform routinely collected sales, cost, product, inventory, and customer-related information into actionable insights without requiring business owners to operate multiple disconnected analytical systems. Data-mining and data-analytic approaches provide a foundation for transforming accumulated organizational data into useful knowledge for decision-making [15], [18], [19].

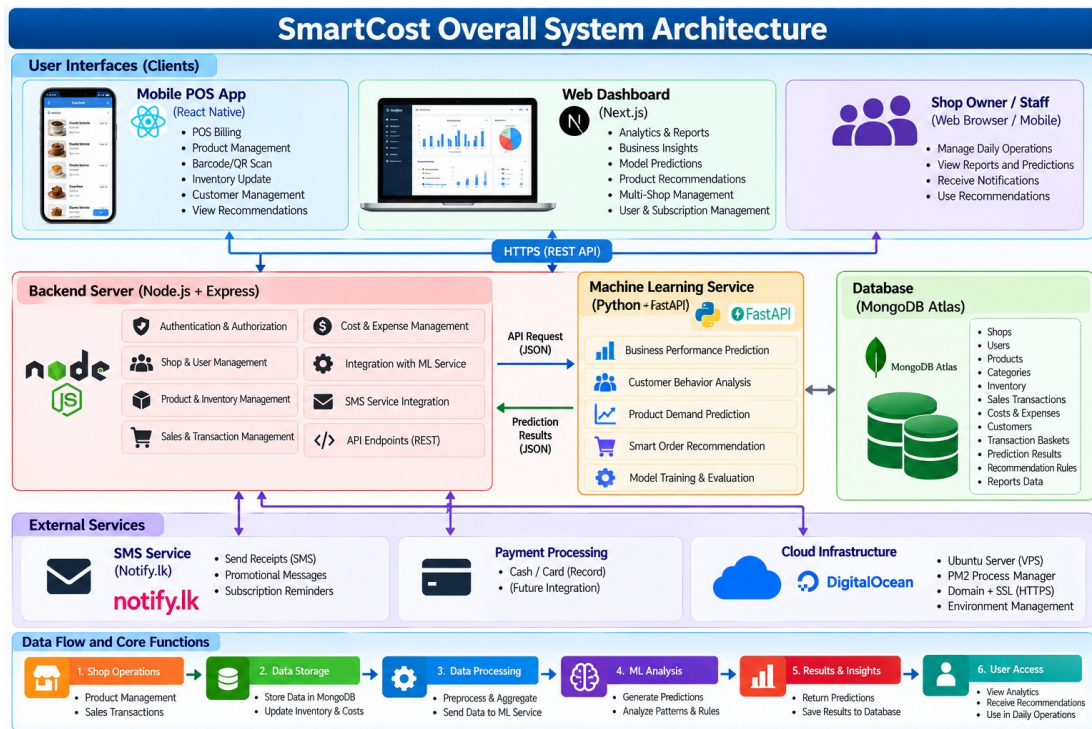


Figure 3: Smart-Cost Overall System Architecture

The operational layer provides the foundation of the solution. Through the Smart-Cost POS application, authorized users can manage products, process customer orders, maintain a shopping cart, apply discounts, record payments, generate receipts, review previous orders, and update inventory quantities. Product identification can also be supported through barcode or QR scanning. These activities continuously generate structured operational information that is stored within the shared data environment and subsequently becomes available to the analytical components.

The analytical layer consists of four coordinated research components. Business Performance Prediction uses relevant sales, cost, and operational information to support profitability and performance assessment. Behavior Analysis and Sales Insights examines accumulated transactional information to identify meaningful customer and sales patterns, consistent with the broader use of data-mining techniques to extract patterns from operational datasets [15], [18], [19]. Product Demand Prediction analyses historical product-level sales to estimate future demand using forecasting principles [5]–[8]. Smart Order Recommendation evaluates transaction-basket relationships to identify products that are frequently purchased together, drawing on association-rule mining and recommendation-system principles [13]–[17].

A shared architecture connects these capabilities. The mobile application communicates with a Node.js backend responsible for operational services and database interaction. MongoDB provides centralized storage for business and analytical information, while Python-based analytical services process requests requiring predictive or advanced analytical operations. RESTful APIs provide controlled communication among the system layers. A management interface provides access to relevant reports and analytical outputs. Modularity and maintainability are relevant software-quality considerations for such an integrated architecture [21].

The end-to-end outcome is a unified environment in which a transaction has value beyond billing. Once captured, operational data can contribute to future demand estimation, financial analysis, customer and sales understanding, and product recommendations. The resulting insights are then returned through appropriate Smart-Cost interfaces to support business users.

Consequently, Smart-Cost extends the role of a conventional POS system from historical transaction recording toward integrated decision support. The four research components retain distinct research contributions while sharing data, services, and interfaces, enabling the complete solution to address multiple aspects of SME decision-making through a single coordinated framework.

Table 5: Smart-Cost Research Components and Responsibilities

Component	Primary Responsibility	Main Purpose
Business Performance Prediction	K.A.C.H Kodithuwakku	Predict and analyze future business performance
Behavior Analysis & Sales Insights	N.K.M. Perera	Analyze customer behavior and sales patterns
Product Demand Prediction	D.D.M. Jayasingha	Forecast future product demand
Smart Order Recommendation	M.P.J.R. Dasanayaka	Generate product recommendations
POS and Common Services	Shared	Provide common operational platform
Integration and Testing	Shared	Integrate and validate the complete solution
Common Documentation	Shared	Prepare integrated research documentation

3.2 High level architecture

The Smart-Cost architecture follows a modular, service-oriented structure that separates user-facing POS operations, operational processing, persistent data storage,

and analytical computation. Separation and modularity support software characteristics such as maintainability, reliability, compatibility, and performance efficiency [21]. Business owners, managers, cashiers, and authorized staff interact primarily with the Smart-Cost application. The application communicates with the Node.js operational API, which manages common functions including authentication, products, orders, billing, inventory, transaction processing, and access to stored business information.

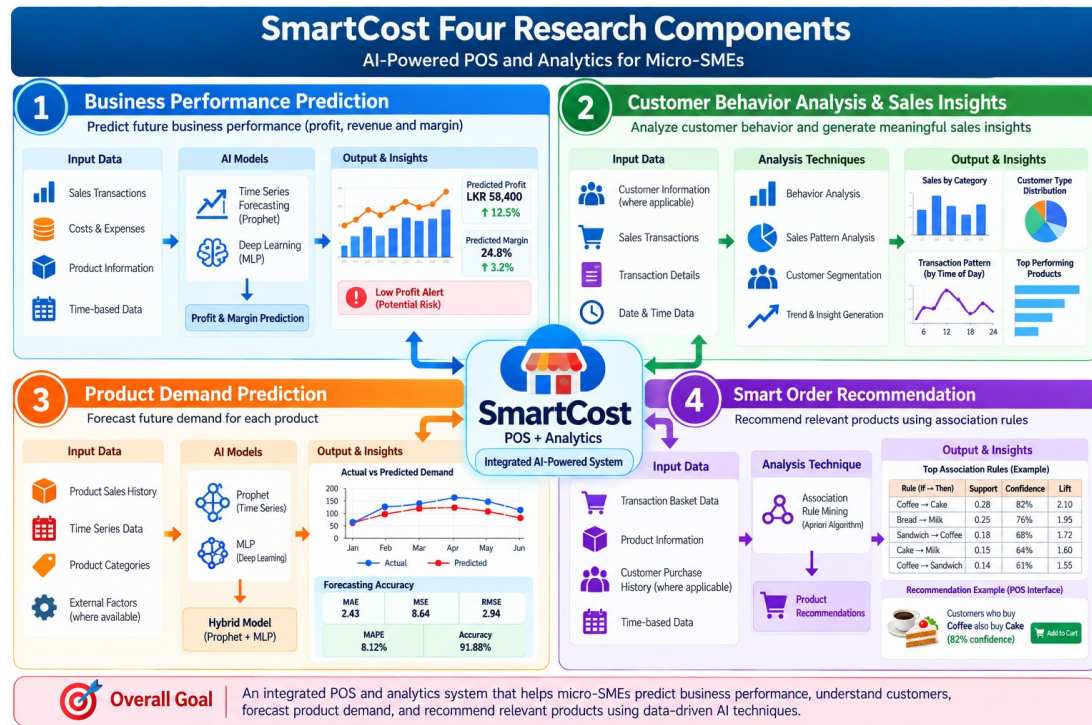


Figure 4: Smart-Cost Four Research Components

MongoDB acts as the shared data layer and maintains operational records required by both routine POS functionality and the research components. Analytical requests are passed to Python-based analytical services, where the four components—Business Performance Prediction, Behavior Analysis and Sales Insights, Product Demand Prediction, and Smart Order Recommendation—perform their respective processing. Results are returned through APIs and presented through the relevant Smart-Cost application or management interfaces.

The architecture therefore enables a common transaction and data infrastructure to support four specialized analytical processes while preserving modularity between operational and research services.

Table 6:Inputs, Processing and Outputs of the Four Research Components

Component	Inputs	Main Processing	Outputs
Business Performance Prediction	Sales, costs, prices, expenses	Data preparation and predictive analysis	Performance, profit and margin-related insights
Behavior Analysis & Sales Insights	Transactions, customers, products, timestamps	Aggregation and behavioral/sales analysis	Sales trends and customer insights
Product Demand Prediction	Historical product sales and dates	Time-series preparation and forecasting	Future product-demand predictions
Smart Order Recommendation	Transaction baskets	Association-rule mining	Ranked product recommendations

3.3 Component 1(Business Performance Prediction) summary

The Business Performance Prediction component addresses the difficulty experienced by SME owners when attempting to understand future financial and operational performance using conventional sales records. Effective data-driven business decision-making requires the interpretation of operational information rather than reliance solely on isolated historical totals [19]. Transaction revenue alone does not provide a complete representation of business health because profitability is influenced by product costs, operating expenditure, sales quantities, pricing, and changes in business conditions. The component therefore contributes the financial decision-support capability of the integrated Smart-Cost platform.

The primary inputs are operational and financial records generated or maintained through Smart-Cost. These include sales information, product prices, quantities sold, relevant cost information, expenses, and historical business-performance indicators. The shared Smart-Cost data environment enables these variables to be analysed without requiring users to transfer records manually into a separate financial analytics application.

At a high level, the component processes historical business information to identify relationships between sales, costs, margins, and profitability and uses predictive analysis to estimate future business performance. Statistical and machine-learning approaches provide established mechanisms for learning relationships from historical data and generating predictive outputs [10]–[12]. Revenue and cost information are combined to derive profitability-oriented indicators, while historical patterns provide the basis for predicting future conditions. Detailed feature engineering, model selection, training procedures, parameter configuration, and experimental evaluation remain within the individual thesis responsible for this research component.

The principal outputs consist of predicted business-performance information, profitability estimates, margin-related insights, and information that can assist owners in recognizing potential financial concerns. These outputs complement conventional POS reports because they focus on anticipated or analytically interpreted business conditions rather than solely displaying historical totals.

Within the complete Smart-Cost solution, this component provides the connection between operational activity and financial interpretation. Its contribution is not the replacement of accounting processes or managerial judgement. Instead, it investigates how operational data already available within an SME-oriented POS environment can be transformed into accessible performance and profitability information.

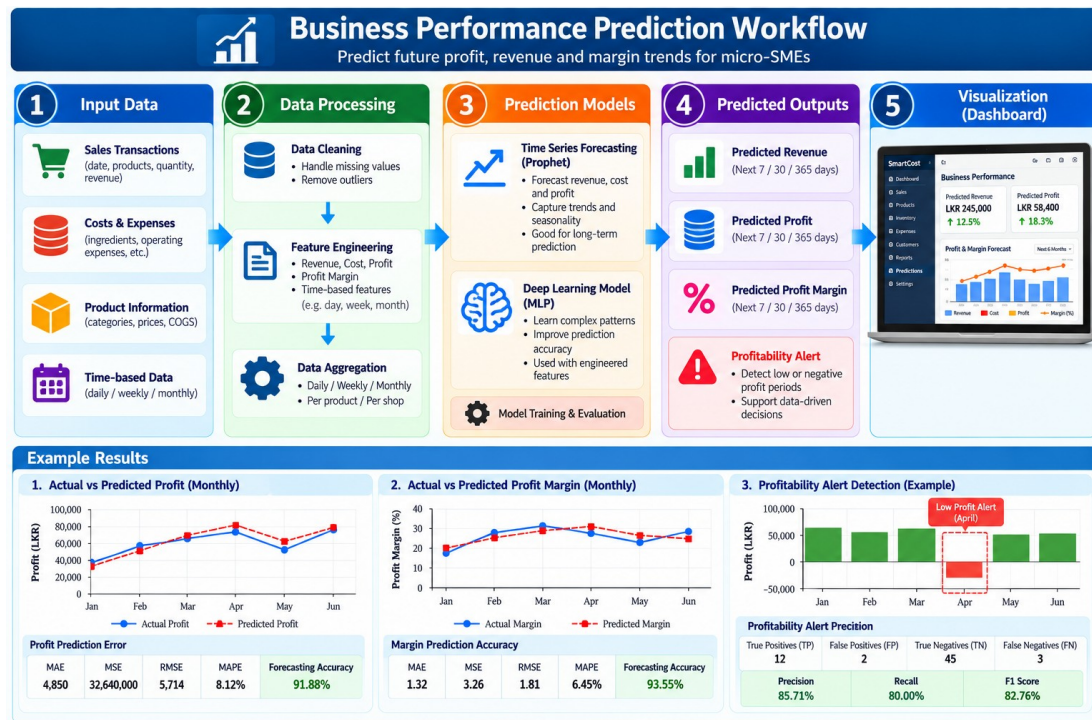


Figure 5: Business Performance Prediction Workflow

3.4 Component 2(Behavior Analysis & Sales Insights) summary

The Behavior Analysis and Sales Insights component addresses the limited ability of SMEs to extract meaningful customer and sales information from accumulated transaction records. Data-mining methods are specifically concerned with identifying useful patterns and knowledge within accumulated datasets [15], [18]. A POS system can preserve large numbers of orders, product quantities, timestamps, and customer-related records, but raw transaction histories become increasingly difficult to interpret as the volume of data increases.

The component uses transaction-level information available through the common Smart-Cost data environment. Relevant inputs include products purchased, quantities, transaction values, order timestamps, historical sales records, and available customer-related information.

At a high level, transaction records are cleaned, aggregated, and analysed to identify meaningful patterns. Data preprocessing and pattern analysis are important stages in the broader data-mining process [12], [15], [18]. Analysis can include product sales

performance, purchasing frequency, recurring customer behaviour, peak purchasing periods, and differences in sales activity across relevant time intervals. The processed outputs are converted into interpretable indicators and visual information so that business users can understand patterns without manually analysing raw transaction records.

The component outputs customer-oriented and sales-oriented insights that can be presented through Smart-Cost dashboards and reports. These outputs enable owners and managers to identify high- and low-performing products, observe changes in sales activity, understand important purchasing periods, and obtain a clearer representation of customer-related behaviour where suitable data are available.

Within the integrated architecture, this component provides the descriptive and behavioural analytical layer. Its contribution is therefore the conversion of accumulated POS records into understandable business knowledge, supporting data-driven planning and decision-making [18], [19].



Figure 6: Behavior Analysis and Sales Insights Workflow

3.5 Component 3(Product Demand Prediction) summary

The Product Demand Prediction component addresses uncertainty regarding future product requirements within SME operations. Businesses such as bakeries, cafés, restaurants, and retail shops must frequently decide how much stock to purchase or how many products to prepare before actual customer demand is known. Forecasting provides a systematic mechanism for using historical observations to estimate future values [5]–[8]. The component therefore investigates how historical POS sales data can be used to estimate future product demand.

The principal input is historical product-level sales information generated through Smart-Cost. Relevant variables include product identifiers, quantities sold, transaction dates, and time-ordered sales records. These records are cleaned, standardized, aggregated into suitable time-series structures, and prepared for forecasting. Appropriate preparation and organization of time-series information are important considerations in forecasting analysis [6], [7].

The component investigates forecasting approaches for learning temporal demand behaviour from historical sales. During the research, statistical forecasting and machine-learning/deep-learning approaches were considered and compared. Prophet provides an established approach to forecasting business time series [5], while neural-network approaches can model complex relationships within sequential data [9], [10]. Product-level historical observations are used to train prediction models, after which future demand is generated for defined forecasting periods.

Model performance is assessed using forecasting metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Square Error (RMSE). Appropriate forecast-accuracy measures are necessary when comparing predicted and observed values [8]. Detailed model architectures, training procedures, parameter settings, comparative experiments, and component-level results are presented within the corresponding individual thesis.

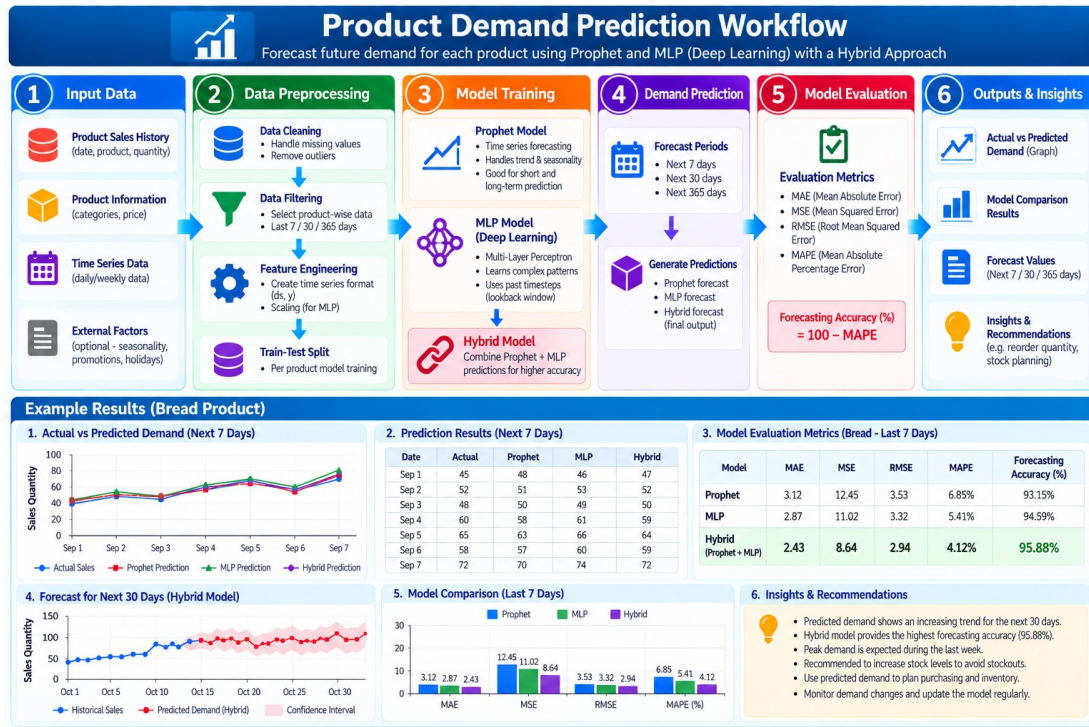


Figure 7: Product Demand Prediction Workflow

The main output is predicted future demand for individual products. These forecasts can be presented for appropriate planning horizons and compared with historical or actual sales information. The information is intended to support inventory and production planning by providing owners with an evidence-based estimate rather than relying exclusively on subjective judgement.

The contribution of Product Demand Prediction is therefore the addition of a forward-looking analytical capability to the Smart-Cost environment. By transforming historical product sales into future demand estimates, the component extends the value of POS data beyond retrospective reporting. Pasted markdown

3.6 Component 4(Smart Order Recommendation) summary

The Smart Order Recommendation component addresses the problem of missed cross-selling opportunities during routine POS transactions. Historical order records may reveal that particular products are frequently purchased together. Association-rule mining was developed to discover relationships among items occurring within transactional datasets [13], and efficient approaches have subsequently been developed for identifying frequent associations [14]. The component therefore investigates how

historical transaction-basket information can be transformed into relevant product recommendations during checkout.

The primary inputs are completed POS transactions containing the products purchased together within individual orders. Each transaction can be represented as a product basket, allowing relationships among items to be analysed across the historical dataset. Market-basket and association analysis provide established mechanisms for examining such transaction-level relationships [13]–[15], [18].

Association-rule mining forms the core analytical approach of the component. Frequently occurring product relationships are evaluated using support, confidence, and lift [13]–[15], [18]. Support represents how frequently an itemset occurs within the transaction dataset, confidence measures the strength of an implication between antecedent and consequent items, and lift provides an indication of the relationship relative to their expected independent occurrence [15], [18]. Rules satisfying configured criteria can then be matched against products in an active customer cart.

The principal output is a ranked or filtered set of related product recommendations that can be displayed through the Smart-Cost checkout interface. Recommendation-system research provides a broader foundation for generating useful item suggestions from available interaction or transaction information [16], [17]. Recommendation information may include the suggested product and an associated confidence value, allowing the recommendation to remain interpretable to the user.

Within the overall Smart-Cost solution, this component provides transaction-level decision support. Its contribution is to demonstrate how accumulated POS transactions can support immediate sales decisions rather than serving solely as historical records

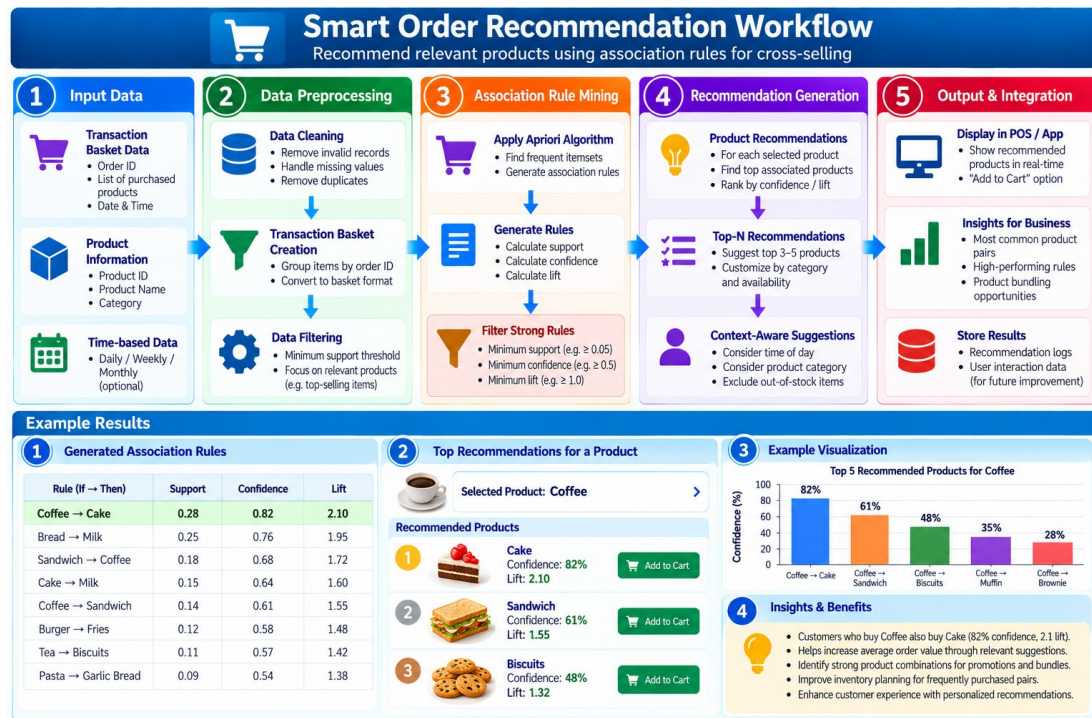


Figure 8: Smart Order Recommendation Workflow

3.7 End to end user and data flow

The end-to-end Smart-Cost workflow begins when an authorized business user interacts with the POS environment. Products and relevant business information are maintained within the system, and a cashier or staff member initiates a transaction by selecting or scanning products. Selected items are added to the cart, quantities can be adjusted, applicable discounts are calculated, and the order proceeds through billing and payment. Once completed, the transaction is recorded through the Node.js operational backend and persisted within MongoDB. Inventory quantities are updated accordingly, while the transaction becomes part of the historical operational dataset.

The value of this data extends beyond the completed sale. Product-level sales quantities and timestamps can be aggregated into chronological records required by the Product Demand Prediction component. Time-series forecasting methods use historical observations as the basis for estimating future values [5]–[8]. These records are processed by the analytical service to generate estimates of future demand, which can subsequently support purchasing, inventory, or production planning.

Sales, cost, price, and operational information also contributes to the Business Performance Prediction component. Data-analytic approaches enable multiple

business variables to be transformed into information relevant to decision-making [19]. By combining these variables, the component generates performance- and profitability-oriented outputs that can assist owners and managers in understanding expected business conditions.

Accumulated transaction information is simultaneously available to Behavior Analysis and Sales Insights. Data-mining approaches provide mechanisms for discovering meaningful patterns from accumulated records [15], [18]. This component aggregates historical records to identify product performance, purchasing periods, customer-related patterns where appropriate, and broader sales behaviour.

The same transaction history provides basket-level information to the Smart Order Recommendation component. Product combinations appearing across completed orders are analysed to generate association rules [13], [14]. During a subsequent active transaction, products currently in the customer's cart can be matched against those rules, allowing relevant recommendations to be returned to the POS interface [16], [17].

The four components therefore interact primarily through shared operational data rather than functioning as isolated applications. A single transaction can contribute to historical sales analysis, future demand forecasting, business-performance assessment, customer and sales insights, and future recommendation rules.

This closed data flow establishes the central integration principle of Smart-Cost: operational activity generates data, shared services manage that data, specialized

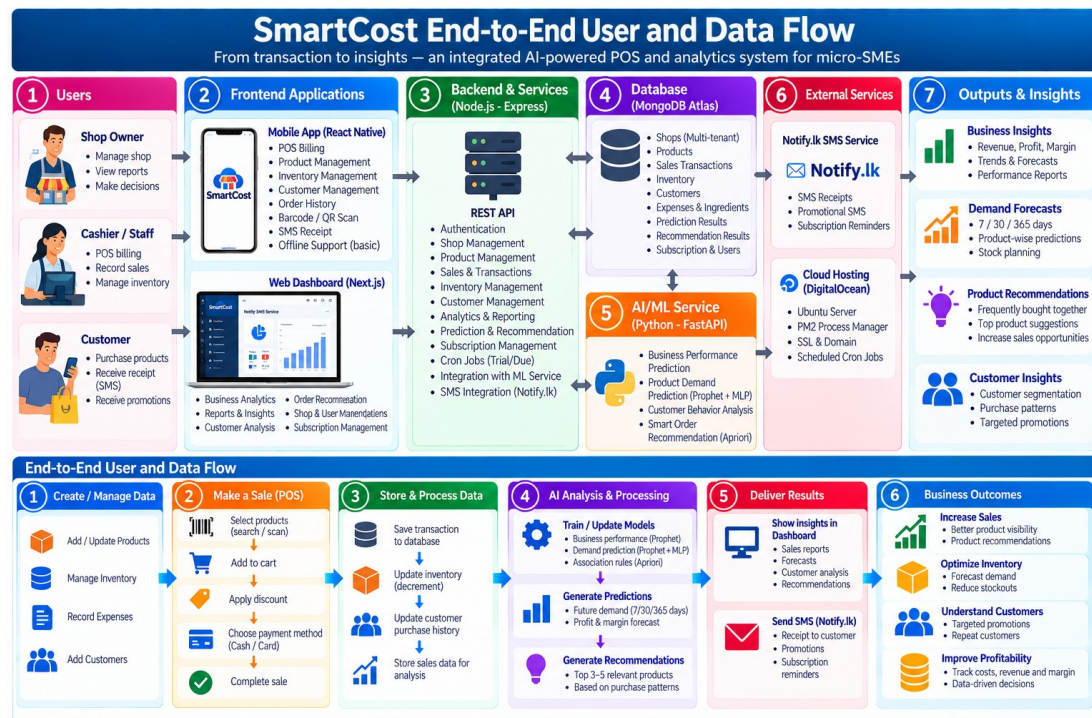


Figure 9: Smart-Cost End-to-End User and Data Flow

4 COMPONENT INTEGRATION AND COHESIVE OPERATION

4.1 Integration strategy

The integration strategy of Smart-Cost was designed to combine the four research components within a single operational environment while preserving the independence of each component's analytical responsibilities. Rather than integrating the components directly with one another through tightly coupled implementations, the system uses shared operational data, defined service interfaces, and RESTful APIs. This modular separation is consistent with software-quality principles related to maintainability, compatibility, and reliability [21]. This approach allows Business Performance Prediction, Behavior Analysis and Sales Insights, Product Demand Prediction, and Smart Order Recommendation to operate as specialized analytical capabilities while remaining connected to the common Smart-Cost POS ecosystem.

Integration was performed incrementally. The initial stage established the shared operational foundation consisting of the React Native POS application, Node.js backend, MongoDB database, authentication and business-level data handling, product management, billing, order processing, and inventory functions. Once the common data flow was available, analytical components were connected progressively. This sequence reduced the risk of attempting to integrate multiple incomplete research modules simultaneously.

The Node.js backend operates as the principal operational integration layer. Routine requests from the POS application are processed through this layer, while analytical requests requiring predictive or advanced processing are forwarded to Python-based analytical services. MongoDB provides shared persistence for transaction, product, inventory, cost, and related business information. Component outputs are returned through defined API responses and presented through the appropriate Smart-Cost interface or dashboard.

Interface ownership was divided according to component responsibilities. Each research member was responsible for developing and validating the analytical logic associated with the assigned component, while common backend, database, mobile, and integration activities were coordinated at group level. This arrangement enabled

component-specific experimentation without requiring changes to unrelated analytical modules.

Several integration risks were considered during development. These included inconsistent data formats, unavailable analytical services, incorrect business identifiers, incomplete transaction records, API communication failures, and changes to component-specific implementations. Common schemas, shared identifiers, validation, modular endpoints, and controlled API communication were used to reduce these risks. Components were tested individually before being connected to the complete workflow, followed by integration testing of the relevant data paths.

The resulting strategy provides separation between operational transaction processing and computational analytics while allowing both layers to function as one system. This modular integration approach supports maintainability and component-level improvement because an analytical method can be refined without redesigning the entire POS application [21]. At the same time, shared services ensure that the four research contributions operate on a common business context and collectively support the overall Smart-Cost objective.

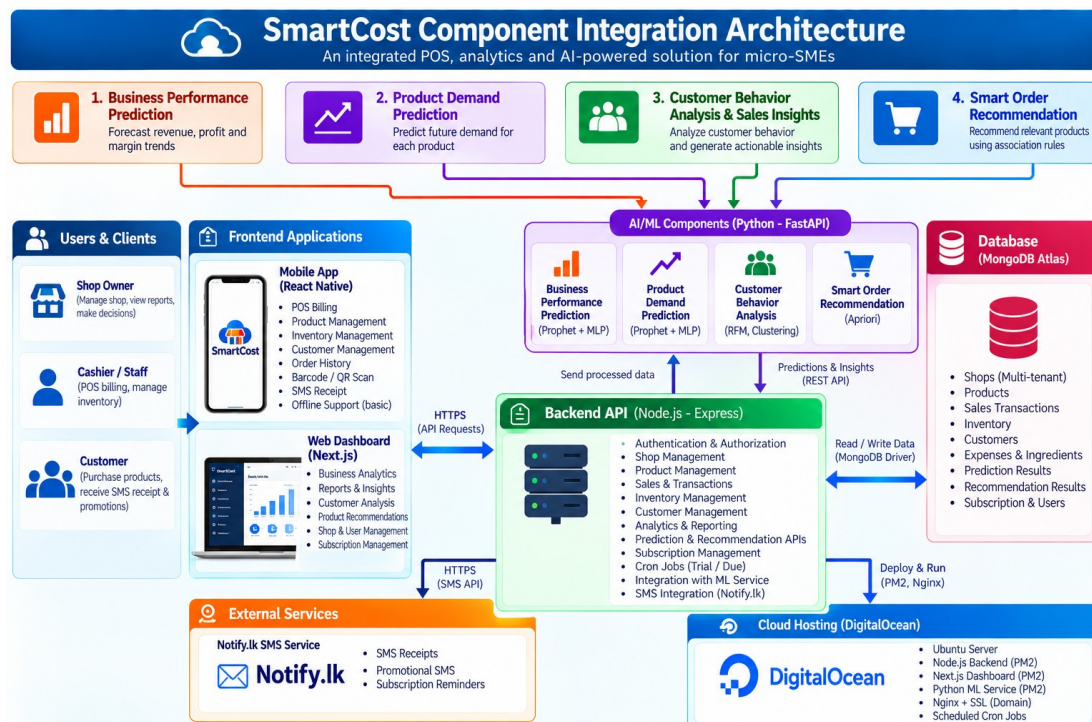


Figure 10: Smart-Cost Component Integration Architecture

4.2 Interfaces data contracts and shared models

Communication between Smart-Cost components is organized through RESTful interfaces and shared data structures. The React Native client communicates primarily with the Node.js backend, which provides operational APIs for products, transactions, orders, inventory, billing, authentication, and analytical requests. Python-based analytical services expose dedicated endpoints for computational functions such as model training, prediction generation, analytical processing, and retrieval of analytical results. This separation supports maintainability by reducing direct dependencies between individual analytical implementations and user-facing interfaces [21].

JSON is used as the primary data-interchange format between application services. Requests contain the information required by the target operation, while responses provide structured results together with appropriate success or failure information. Common entities include shops, products, transactions, orders, costs, inventory records, customers where applicable, prediction results, and recommendation outputs. Product identifiers and business or shop identifiers are particularly important because they maintain consistency between operational records and analytical outputs.

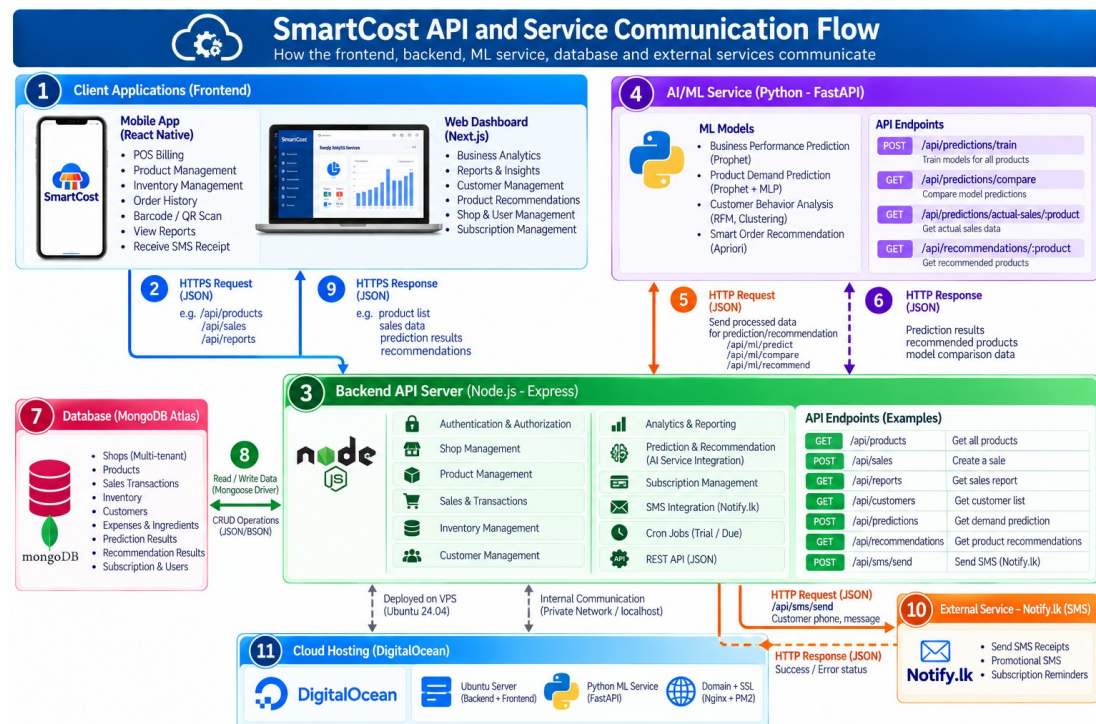


Figure 11: Smart-Cost API and Service Communication Flow

A shared product model contains information such as the product identifier, product name, category, selling price, inventory-related information, and associated business identifier. Transaction and order structures contain purchased products, quantities, transaction values, payment-related information, and timestamps. Forecasting processes additionally require product-level chronological sales information [5]–[8]. Recommendation processing requires transaction baskets containing products purchased within the same order [13]–[15], while business-performance analysis requires relevant sales and cost information [19].

Input validation is performed before data are accepted for operational or analytical processing. Required attributes are checked, numerical fields must contain valid values, identifiers must correspond to appropriate records, and date information is normalized into formats suitable for subsequent processing. Analytical preprocessing performs additional checks for missing, duplicate, inconsistent, or insufficient observations where required by the component methodology [12], [18].

Interfaces are designed so that component-specific analytical logic remains behind the service boundary. Changes to a forecasting model, for example, should not require the mobile application to understand the model's internal architecture as long as the expected request and response contract remains consistent. This supports maintainability and future model replacement [21].

Table 7: Smart-Cost Shared Data Entities and Their Uses

Data Entity	Main Information	Primary Uses
Shop	Business information and identifier	Multi-tenant data separation
User	Account and role information	Authentication and access
Product	Product, category, price and identification data	POS, inventory and analytics
Inventory	Stock quantities and	Inventory monitoring

	movements	
Sale/Transaction	Date, value, payment and transaction information	Reporting and analytics
Sale Item	Product and quantity per transaction	Demand and basket analysis
Customer	Customer details where applicable	Customer analysis
Cost/Expense	Business costs and expenses	Performance analysis
Prediction	Forecasting outputs	Analytical decision support
Recommendation	Product relationships and recommendation results	Smart recommendations

Errors are handled at the service responsible for detecting them. The backend validates operational requests and database operations, while analytical services identify model-related conditions such as unavailable trained models, insufficient data, or invalid analytical inputs. Appropriate error responses are returned instead of silently generating results. Collectively, these shared interfaces and data models provide a consistent integration mechanism through which independently developed research components can exchange information within the unified Smart-Cost platform.

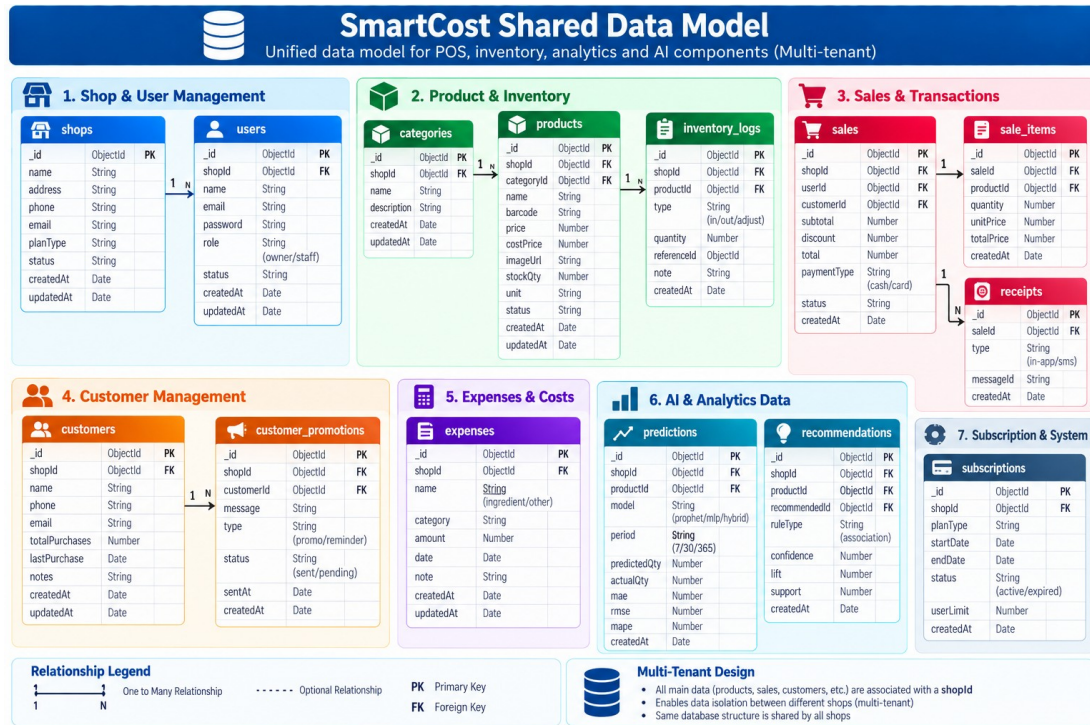


Figure 12: Smart-Cost Shared Data Model

4.3 Shared nonfunctional requirements

In addition to functional capabilities, Smart-Cost requires a set of shared nonfunctional requirements to ensure that the integrated solution remains practical for routine SME operations. Software-quality models identify characteristics including functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability, and portability as important considerations in software-system quality [21]. These requirements apply across the mobile application, backend services, database, analytical components, and management interfaces and influence the overall quality of the complete system.

Performance is important because routine POS operations should remain responsive even when analytical functions require additional computation. Transaction processing, product retrieval, cart operations, billing, and inventory updates therefore remain separated from computationally intensive analytical processing where possible. The service-oriented architecture enables predictive functions to be handled through dedicated analytical services without unnecessarily coupling them to normal checkout operations [21].

Usability is another central requirement because Smart-Cost is intended for business owners, managers, cashiers, and staff who may have different levels of technical experience. Interfaces should therefore provide clear navigation, understandable labels, efficient transaction workflows, and interpretable analytical outputs [21]. Forecasts, recommendations, and business insights must be presented in forms that support decisions rather than exposing unnecessary algorithmic complexity.

Security and privacy require controlled access to business information. Authentication, business-level data separation, validation of requests, and appropriate handling of customer-related information are necessary to reduce unauthorized access and unintended disclosure. Information-security principles emphasize appropriate management of the confidentiality, integrity, and availability of information [25]. Data collected for research or evaluation must also be handled according to the approved research and consent procedures.

Reliability requires routine POS operations to preserve data consistency and provide meaningful error responses when dependent services fail [21]. Analytical failure should not result in fabricated predictions or recommendations. Responsible AI principles emphasize appropriate risk management and trustworthy handling of AI-generated outputs [22], [23]. Where an analytical output is unavailable, the system should communicate that condition while preserving essential operational functionality where technically possible.

Maintainability is supported through modular separation between the React Native application, Node.js services, MongoDB storage, and Python analytical functions. Defined APIs reduce direct dependencies between individual models and user-facing interfaces, allowing component implementations to evolve independently [21].

Scalability is considered through business-level data separation and centralized service deployment. The architecture should support increasing numbers of products, transactions, users, and participating businesses without requiring a separate application implementation for each organization.

Accessibility and compatibility considerations include readable interfaces, consistent interaction patterns, and support for commonly used mobile and web environments.

Collectively, these requirements ensure that Smart-Cost is evaluated not only according to analytical capability but also according to the professional software qualities necessary for an integrated SME-oriented system [21].

Table 8: Smart-Cost Common Non-Functional Requirements

Requirement	Smart-Cost Requirement
Performance	Routine POS activities should remain responsive while analytical services operate
Usability	Interfaces should be understandable for SME owners and operational staff
Security	Authentication and controlled access should protect business data
Privacy	Customer and business information should be handled appropriately
Reliability	Failure of an optional analytical service should not unnecessarily stop core POS operations
Maintainability	Components should remain modular and independently maintainable
Scalability	Architecture should support additional shops, users and transaction records
Compatibility	Client, backend and analytical services should communicate through standardized interfaces
Data Integrity	Transaction and analytical records should

	remain consistent
--	-------------------

4.4 Integrated workflow and exception handling

The normal Smart-Cost workflow begins when an authenticated user accesses the POS environment and performs an operational activity such as selecting products, creating an order, processing payment, or reviewing business information. During a sale, selected products and quantities are maintained within the cart until the transaction is completed. The Node.js backend validates the request, processes the order, records the transaction in MongoDB, and updates the associated inventory information. The completed transaction then becomes part of the shared historical dataset used by the analytical components.

Analytical processing occurs according to the requirements of each component. Product-level sales histories are prepared for Product Demand Prediction using forecasting principles [5]–[8], while relevant sales and cost information is made available for Business Performance Prediction [19]. Accumulated transaction records are analysed by Behavior Analysis and Sales Insights using data-analysis principles [15], [18], and transaction baskets provide the historical relationships required by Smart Order Recommendation [13]–[17]. Analytical outputs are returned through service interfaces and presented within the appropriate Smart-Cost dashboard or POS workflow.

The integrated design must also account for exceptional conditions. Invalid or incomplete requests are rejected through validation before they alter persistent records. If a requested product does not exist or required values are missing, the operational API returns an appropriate error rather than processing an inconsistent transaction. Database-operation failures are handled at the backend so that unsuccessful persistence is not presented to the user as a completed operation.

Analytical services introduce additional failure scenarios. A prediction request may occur before sufficient historical information is available or before the required model has been trained. Forecasting quality depends on the available historical observations

and characteristics of the underlying time series [6]–[8]. Under such conditions, the analytical service should return an explicit unavailable or insufficient-data response rather than generating unsupported output. Similarly, recommendation generation may produce no result when the current cart does not match a valid association rule [13]–[15]. In this situation, checkout can continue without a recommendation.

Network or analytical-service interruptions should be isolated from essential transaction processing where the requested operational function does not depend on the unavailable service. This separation allows routine POS functions to remain conceptually independent from optional analytical outputs. Failed analytical requests can be retried once the relevant service becomes available.

Table 9: Smart-Cost Exception Handling and Recovery Strategy

Exception	Possible Effect	Handling / Recovery
Invalid request	Operation cannot be processed	Validate input and return appropriate response
Missing required data	Incomplete transaction/analysis	Reject or request required values
Product not found	POS operation fails	Return controlled product error
Database failure	Read/write unavailable	Report failure and prevent inconsistent updates
Insufficient historical data	Forecast cannot be generated reliably	Return insufficient-data status
Analytics service unavailable	Prediction/recommendation unavailable	Preserve core POS operations where possible
No association rule	No suitable recommendation	Return no

found		recommendation rather than fabricate one
Network interruption	Remote service unavailable	Controlled error handling and retry where appropriate

Logging and meaningful API responses assist with diagnosing integration problems. Validation at multiple system boundaries also reduces propagation of malformed data between components. Through these mechanisms, Smart-Cost treats exception handling as part of the integrated workflow rather than assuming that every service request will succeed. The resulting behaviour protects operational data and ensures that analytical limitations are communicated transparently to users [21]–[23].



Figure 13: Integrated Smart-Cost Operational Workflow

4.5 Team contribution and dependency matrix

Smart-Cost was developed as a four-member research project in which each student maintained primary responsibility for one individual research component while

contributing to shared system development, integration, testing, documentation, and presentation activities. This division of responsibilities allowed each component to retain an identifiable individual research contribution while ensuring that the final outcome operated as one integrated system rather than four disconnected implementations.

The four primary research responsibilities were Business Performance Prediction, Behavior Analysis and Sales Insights, Product Demand Prediction, and Smart Order Recommendation. Each component owner was responsible for investigating the corresponding research problem, reviewing relevant literature, defining the component methodology, preparing or processing required data, implementing the analytical approach, conducting component-level evaluation, and documenting the resulting evidence within the individual thesis.

Shared development activities included establishment of the Smart-Cost POS environment, backend and database integration, common data models, API communication, authentication and business-level data handling, interface integration, analytical-service connectivity, and end-to-end testing. Although particular team members could take stronger implementation roles in specific technical areas, these shared elements were required by multiple research components and therefore formed part of the group's integration responsibility.

The components also contain explicit dependencies. Product Demand Prediction depends on historical product-level sales records generated through operational transactions, consistent with the requirement for historical observations in forecasting [5]–[8]. Business Performance Prediction depends on relevant sales, cost, and financial information. Behavior Analysis and Sales Insights requires accumulated transaction and customer-related information where applicable [15], [18], [19], while Smart Order Recommendation requires completed transaction baskets containing product combinations [13]–[17]. All four components therefore depend on the reliability of the shared POS, backend, database, and data-management environment.

A contribution and dependency matrix should be maintained to provide transparent evidence of individual and shared responsibilities.

Table 10: Team Contribution and Dependency Matrix

Research/Development Area	Primary Owner	Main Responsibility	Dependency/Shared Output
Business Performance Prediction	K.A.C.H Kodithuwakku	Performance/profitability research, implementation and evaluation	Sales and cost data; analytical outputs
Behavior Analysis & Sales Insights	N.K.M.Perera	Customer/sales analysis, implementation and evaluation	Transaction/customer data; dashboards
Product Demand Prediction	D.D.M Jayasingha	Forecasting research, model development and evaluation	Historical product sales; predictions
Smart Order Recommendation	M.P.J.R.Dasanayaka	Association analysis, recommendation development and evaluation	Transaction baskets; recommendation outputs
POS and common services	Shared team contribution	Operational application and common functionality	Data source for all components
Integration and testing	Shared team contribution	API integration, end-to-end validation and issue resolution	Complete Smart-Cost system
Common documentation	Shared team contribution	Architecture, common report and presentation evidence	Group research deliverables

This structure demonstrates both individual accountability and the technical dependencies necessary for cohesive Smart-Cost operation.

5 INTEGRATED IMPLEMENTATION AND PROJECT EXECUTION

5.1 Common technology stack and infrastructure

The Smart-Cost platform was implemented using a shared technology stack designed to support routine point-of-sale operations, centralized data management, web-based administration, and the four analytical research components within a common environment. Technology selection considered development flexibility, integration capability, deployment cost, maintainability, and suitability for resource-constrained SME environments [2], [21].

The primary mobile application was developed using React Native, enabling the team to implement POS functions through a mobile-oriented interface while maintaining a reusable JavaScript-based development approach. The application provides access to product management, sales processing, cart operations, billing, inventory-related functions, transaction history, and analytical outputs. Common application functionality communicates with backend services through RESTful APIs.

The operational backend was developed using Node.js with Express. It provides the central application layer responsible for request processing, business logic, authentication, product and order operations, inventory updates, database communication, and coordination with analytical services. Separating backend functionality from the user interface reduces direct dependencies between the application and persistent data structures and supports maintainability [21].

MongoDB Atlas was selected as the shared database environment. Its document-oriented structure supports operational entities such as businesses, products, transactions, orders, costs, customers, inventory information, and analytical results. Business-level identifiers are used to maintain separation between records belonging to different shops within the shared database environment.

Analytical processing was implemented through Python-based services, with FastAPI providing interfaces between predictive models and the operational backend. This environment supports data preprocessing, forecasting, business-performance analysis, customer and sales analytics, and recommendation-related computation. Python was

appropriate because of its mature ecosystem for data analysis and machine-learning development [10]–[12], [18].

Table 11: Smart-Cost Tools and Technologies

Layer / Purpose	Technology	Use in Smart-Cost
Mobile Application	React Native	POS and operational mobile interface
Web Application	Next.js	Management and analytics dashboard
Backend	Node.js + Express	Operational APIs and business logic
Database	MongoDB Atlas	Shared cloud database
Analytics/ML	Python	Data processing and analytical models
Analytics API	FastAPI	Expose analytics functionality as services
Communication	REST/JSON	Inter-service communication
Cloud Environment	DigitalOcean / Ubuntu	Hosting project services
Process Management	PM2	Manage Node.js processes
Version Control	Git	Source-code and development management

A Next.js-based web dashboard was also incorporated into the common infrastructure to support administrative and management-oriented access to system information. Source code and component changes were managed through Git-based repositories, enabling collaborative development and version tracking.

For deployment, the common services were configured within a cloud-hosted Linux environment, while MongoDB Atlas provided managed database hosting. This architecture created a shared technical foundation through which the four research components could exchange operational information without requiring independent infrastructure for each component.

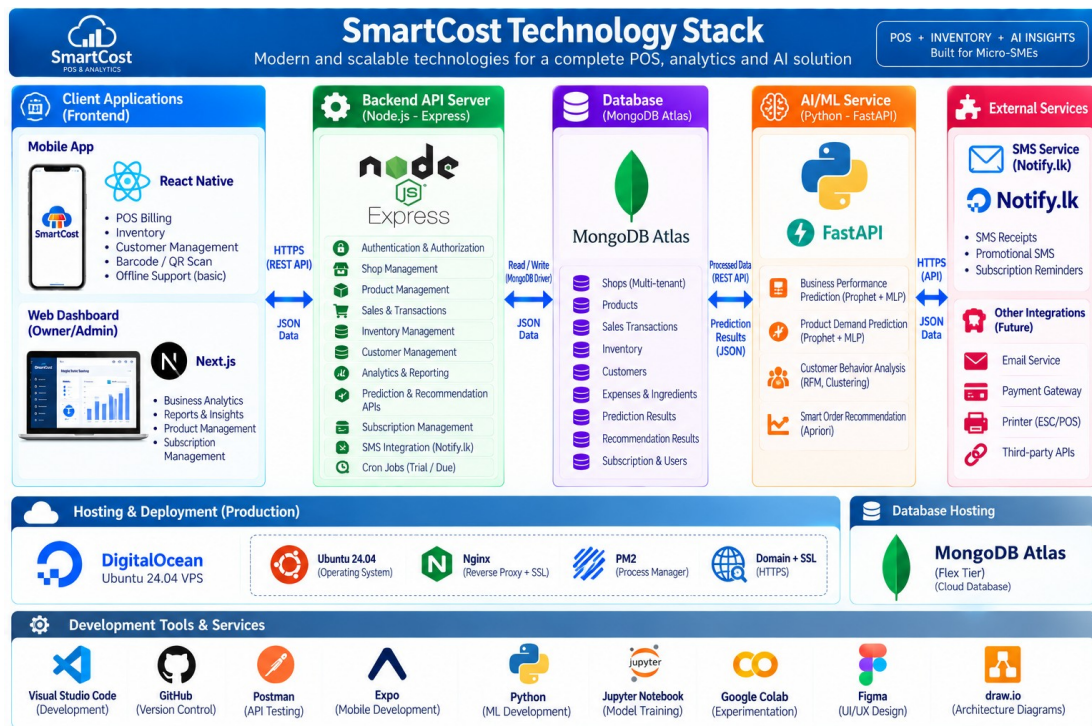


Figure 14: Smart-Cost Technology Stack

5.2 Development and integration process

Development of Smart-Cost followed an iterative and component-oriented process in which shared POS functionality and individual research components were developed progressively and integrated through agreed interfaces. The team divided the overall research problem into component-level responsibilities while maintaining common implementation tasks for the mobile application, backend, database, analytical services, and integrated testing.

Development activities were organized around planned tasks and project milestones. During each development period, the team identified required functionality, component dependencies, integration requirements, and research activities. Larger requirements were divided into manageable implementation tasks so that operational features and research components could progress without requiring the complete system to be finished at the beginning of development.

An interface-oriented approach was used for integration. Common data structures and API requirements were identified before analytical outputs were connected to the user-facing application. The React Native application communicated with the Node.js backend through RESTful endpoints, while analytical operations were exposed through Python-based services. This reduced direct coupling between user-interface code and individual prediction or analytical models and supported maintainability [21].

Git-based version control was used to manage source-code changes and support collaborative development. Component work could be developed and reviewed before being incorporated into the shared implementation. Version history also provided evidence of development progress and enabled previous implementation states to be recovered when necessary. Integration issues identified during testing were discussed by the team and corrected within the responsible application or analytical layer.

Table 12: Smart-Cost Development and Integration Activities

Stage	Main Activities	Output
Requirements	Identify SME and research requirements	Requirement specification
Architecture Design	Define components, APIs and shared data	Integrated architecture
Operational Development	Implement POS and common services	Functional operational platform
Component Development	Develop four analytical	Independent research

	components	components
API Integration	Connect operational and analytical services	Integrated service communication
Database Integration	Establish shared data access	Unified data lifecycle
Testing	Functional, API and integration testing	Verified prototype
PP2 Evaluation	Evaluate components and integrated system	PP2 evidence and results

Testing was conducted at multiple stages. Individual functions were initially tested within their respective development environments. API communication and database operations were then verified before connecting analytical components. Integrated testing examined whether data generated through POS transactions could be processed correctly by the appropriate analytical services and whether outputs could be returned to the application or dashboard. System quality characteristics such as functional suitability, reliability, compatibility, and performance efficiency provide relevant considerations for this type of integrated testing [21].

Research activities progressed alongside software development. Literature analysis, dataset preparation, model experimentation, evaluation, documentation, and PP2 preparation were coordinated with implementation activities so that research findings could inform system development.

This incremental process reduced the risk associated with integrating four independently researched components. Instead of treating integration as a final activity, shared interfaces, data dependencies, and component interactions were considered throughout development, allowing Smart-Cost to evolve as a unified research system.

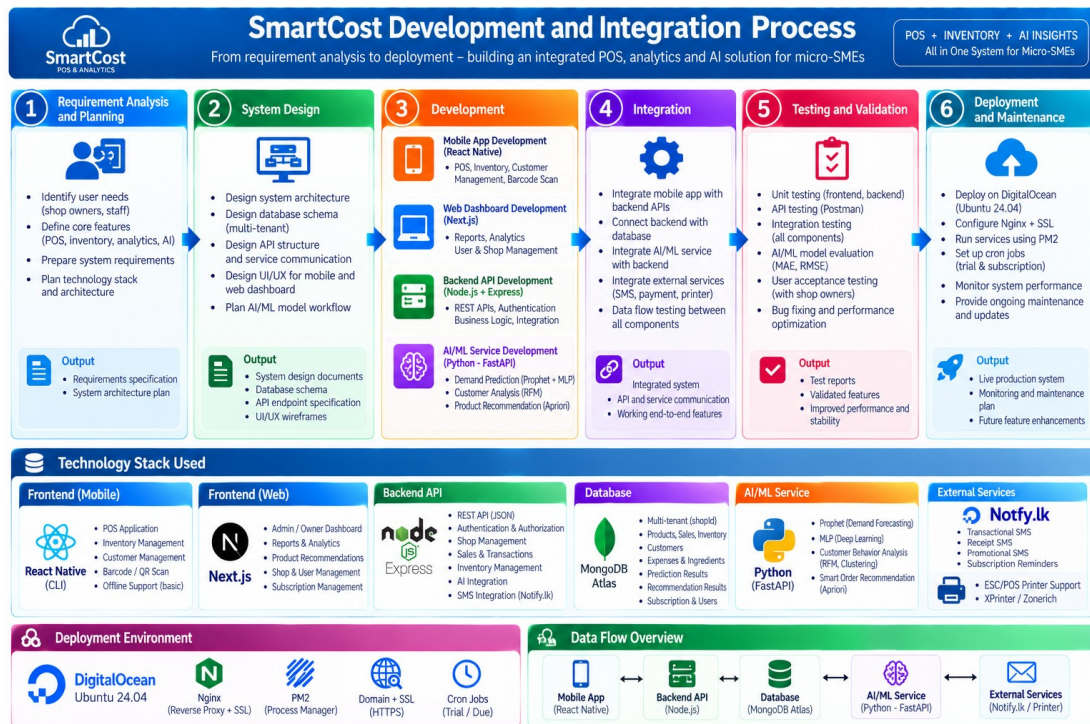


Figure 15: Smart-Cost Development and Integration Process

5.3 Deployment and operational configuration

The Smart-Cost deployment configuration was designed to make the integrated system accessible outside individual development machines while maintaining separation between user-facing applications, operational services, analytical processing, and persistent storage. The deployment process therefore considered application configuration, service management, database connectivity, network communication, and recovery of operational services. Reliability, maintainability, security, and portability are relevant software-quality considerations for deployed systems [21].

The Node.js backend and supporting server-side services were deployed to an Ubuntu 24.04 cloud server hosted on DigitalOcean. Environment-specific configuration values, including database connections, service endpoints, and other application settings, were maintained through environment variables rather than being embedded directly within application code. This allowed development and deployed environments to use different configurations without modifying the underlying implementation.

PM2 was used to manage the Node.js application process within the server environment. Process management allowed the backend service to remain active

independently of an interactive terminal session and supported controlled restarting when configuration or application changes were deployed. The Next.js management dashboard was similarly configured to communicate with the deployed backend through the appropriate environment settings.

The mobile application was configured to use the deployed API rather than local development endpoints when integration testing was performed against the hosted environment. Python-based analytical processing communicated with the operational layer through defined service interfaces. MongoDB Atlas provided externally managed database infrastructure, enabling application services to access persistent operational and analytical information through secured connection configuration. Appropriate protection of information and service credentials is consistent with information-security management principles [25].

Deployment verification included checking backend availability, API communication, database connectivity, analytical-service communication, and end-to-end requests from the user interface. Logs generated by the deployed services were used to investigate runtime and integration problems. Application processes could be restarted through the process manager after configuration or code changes.

Operational recovery depended on maintaining source code in version-controlled repositories, retaining configuration information securely, and using managed database capabilities for persistent information. The modular architecture also reduced recovery complexity because the mobile client, backend, analytical services, dashboard, and database could be diagnosed independently [21].

Table 13: Smart-Cost Deployment and Operational Configuration

Item	Configuration / Role
Hosting	DigitalOcean cloud server
Operating System	Ubuntu 24.04
Backend	Node.js/Express service
Process Management	PM2
Web Dashboard	Next.js
Analytics	Python/FastAPI services
Database	MongoDB Atlas
Mobile Connectivity	Mobile application configured to deployed API
Configuration	Environment variables used for service configuration
Version Control	Git-based repositories
Monitoring	Application/service logs and process monitoring
Deployment Scope	Research, demonstration and PP2 evaluation environment

At the PP2 stage, deployment primarily supported research implementation, integration, demonstration, and evaluation rather than claiming production-level availability guarantees. The configuration nevertheless demonstrated that the complete Smart-Cost architecture could operate through hosted services rather than being restricted to isolated local development environments.

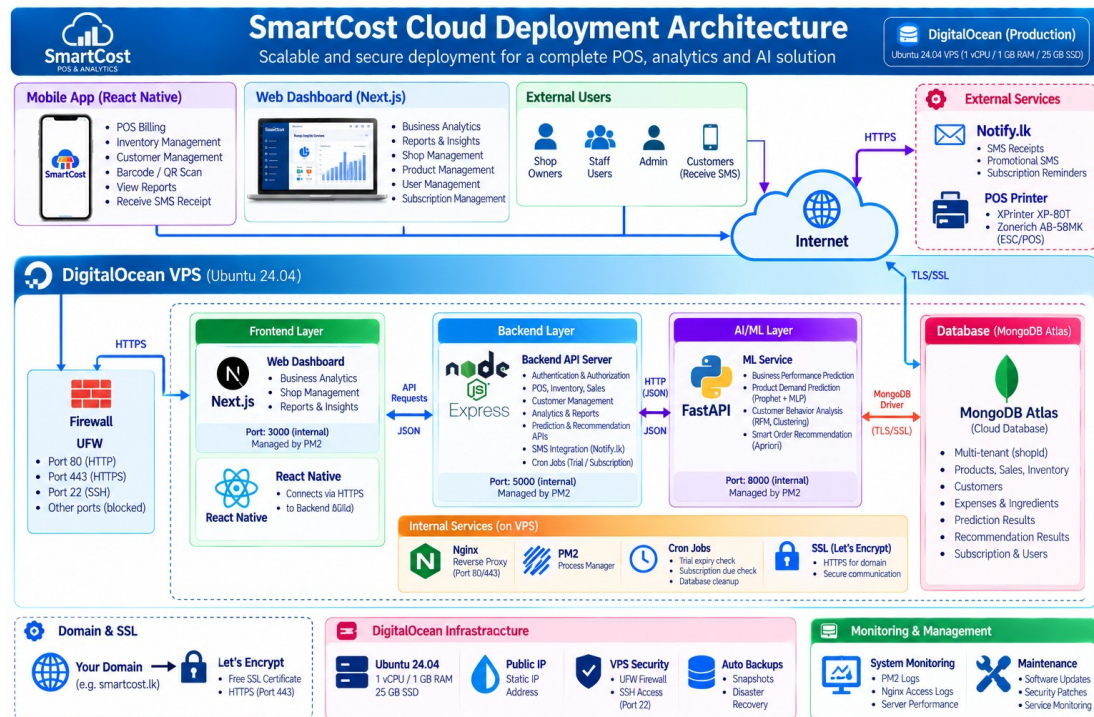


Figure 16: Smart-Cost Cloud Deployment Architecture

5.4 Evidence of team execution

Execution of the Smart-Cost project involved coordinated research, development, integration, testing, documentation, and project-management activities across the research team. Evidence of this work was maintained through project-management records, communication platforms, source-control history, research documentation, meeting activities, and milestone deliverables rather than relying solely on the completed software as evidence of participation.

Project tasks and responsibilities were organized according to the common system requirements and the four individual research components. Microsoft Planner was used to support task organization and progress monitoring where applicable. Tasks could be associated with research activities, implementation requirements,

documentation, testing, and preparation for project evaluations. This provided structured evidence of how larger project objectives were translated into individual and shared activities.



Figure 17: Smart-Cost Mobile POS Interface

Microsoft Teams and team meetings supported communication concerning research decisions, technical problems, integration dependencies, supervisor feedback, and upcoming milestones. Meeting discussions were particularly important when changes to common interfaces or data structures affected more than one component. Rather than allowing each research component to evolve independently without coordination, integration-related decisions were communicated across the group.

Git repository history provided technical evidence of implementation activities. Commits and source changes recorded the progressive development of mobile functionality, backend services, analytical implementations, integration logic, and corrections introduced during testing. Repository history also supported accountability by demonstrating how the implementation evolved across the research period.

Individual and common research documentation provided additional evidence of execution. Research diaries, literature-review activities, experiment records, model-evaluation outputs, diagrams, presentation materials, and thesis-related documents captured work that could not be represented adequately through source-code commits. PP1 and PP2 deliverables provided milestone-level evidence of the progression from problem identification and research planning toward implementation, integration, and evaluation.



Figure 18: Smart-Cost Management and Analytics Dashboard

The project therefore used multiple forms of evidence to demonstrate team execution. No single source, such as commit count, was treated as a complete measure of contribution. Instead, development history, research evidence, project-management records, meeting participation, component outputs, testing activities, and shared

deliverables collectively demonstrate how individual contributions were coordinated to produce the integrated Smart-Cost solution.

For the final report, detailed raw Planner exports, meeting logs, Git histories, or research-diary entries should be retained as supporting evidence or appendices where required rather than reproduced extensively within the main chapter.

6 INTEGRATED TESTING EVALUATION AND VALIDATION

6.1 Evaluation framework and success criteria

The evaluation of Smart-Cost was structured to determine whether the four research components could operate effectively within a common POS environment and whether the integrated system could provide useful decision-support capabilities for the intended SME context. Evaluation therefore considered both component-level analytical performance and group-level system behaviour. Detailed experiments associated with individual algorithms are presented in the respective individual theses, while this common report concentrates on the collective evidence demonstrating the feasibility of the integrated solution.

The evaluation framework considered five principal questions. First, could routine POS data be captured and transferred reliably to the relevant analytical components? Second, could the individual components produce meaningful outputs from the available data? Third, could these outputs be returned and presented through the integrated Smart-Cost environment? Fourth, could normal POS operations continue while the analytical capabilities were incorporated? Finally, was the resulting system understandable and useful from a user perspective? Software-quality characteristics such as functional suitability, performance efficiency, usability, reliability, and compatibility provide relevant considerations for such system-level evaluation [21].

Table 14: Smart-Cost Evaluation Framework and Success Criteria

Evaluation Area	Success Criterion	Evidence
POS Functionality	Core operational functions execute correctly	Functional tests
Data Integration	Transaction data reaches required services	API/integration tests
Business Performance	Component generates meaningful outputs	Component metrics

Behavior Analysis	Sales/customer patterns can be produced	Analytical outputs
Demand Prediction	Forecasts can be generated and evaluated	MAE, MSE/RMSE, MAPE
Recommendation	Relevant association rules can be generated	Support, confidence, lift
End-to-End Integration	Outputs return to Smart-Cost interfaces	End-to-end tests
Usability	Users can understand and use the system	User evaluation

Component evaluation used metrics appropriate to each research problem rather than applying one metric to all four components. Product Demand Prediction was evaluated using forecasting error measures such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Square Error (RMSE), together with model-comparison evidence [8]. Business Performance Prediction was assessed using component-specific prediction and profitability-related measures. Behavior Analysis and Sales Insights was evaluated according to the validity and usefulness of generated analytical outputs. Smart Order Recommendation considered association and recommendation measures, including support, confidence, and lift [13]–[18].

The integrated research also used a structured experimental dataset representing SME POS activity. The common experimental dataset contained approximately twelve months of simulated bakery/café operations, including 15,420 POS transactions, 128 products, and 3,760 cost records. The use of simulated data enabled controlled testing of the integrated analytical workflow while avoiding unsupported claims that all experimental observations originated from commercial businesses.

System-level success required correct API communication, persistent data storage, appropriate business-level data handling, successful analytical processing, and presentation of results through Smart-Cost interfaces. Functional and integration testing were used alongside analytical evaluation [21].

User-oriented evaluation additionally considered usability, clarity, relevance, and perceived usefulness of the integrated system. These measures were intended to determine whether technically generated outputs could be understood in a practical SME setting. Collectively, the framework evaluated Smart-Cost as both a research platform containing four analytical contributions and an integrated software solution capable of connecting operational data with decision-support functionality [21].

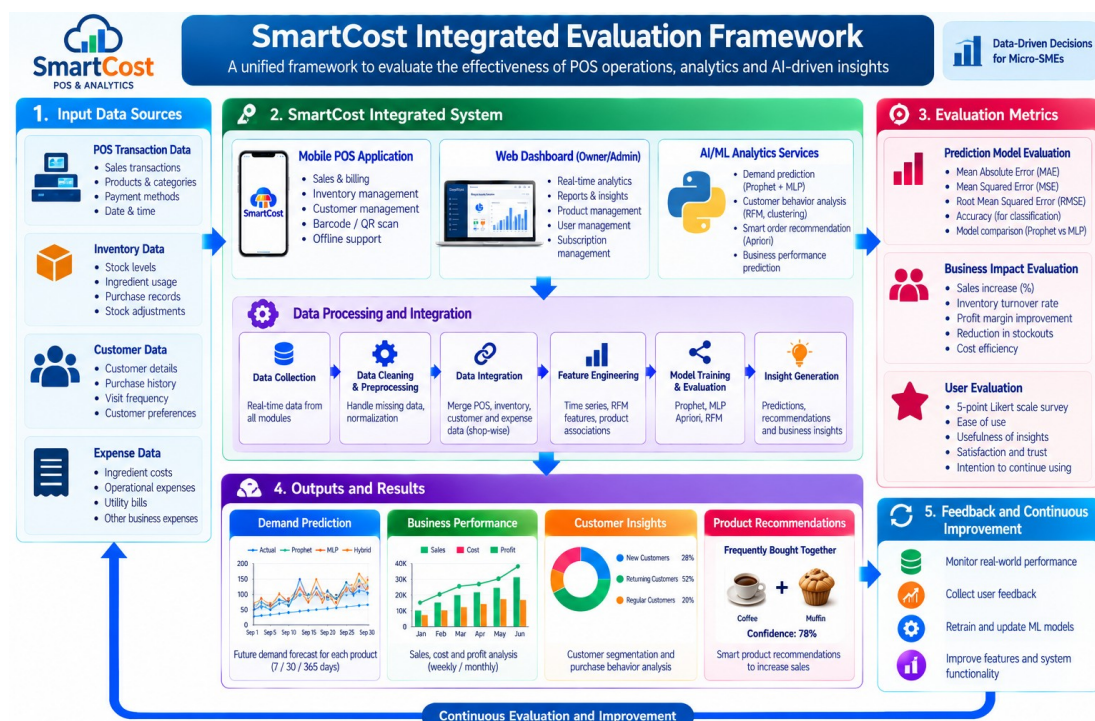


Figure 19: Smart-Cost Integrated Evaluation Framework

6.2 Integration and end to end testing

Integration testing was conducted to verify that Smart-Cost's operational and analytical layers could function together across the complete data flow. Testing extended beyond isolated model evaluation because the usefulness of the proposed solution depended on successful communication among the user interfaces, Node.js backend, MongoDB database, Python analytical services, and four research components.

Functional testing initially examined common POS operations. Product creation and retrieval, product selection, cart modification, quantity handling, billing, transaction completion, transaction-history retrieval, and inventory updates were checked to confirm that routine operations produced the expected application and database changes. These tests established the operational foundation required before analytical integration could be evaluated [21].

Interface testing focused on RESTful communication among the system layers. Requests were checked for correct identifiers, expected fields, valid numerical values, and appropriate date formats. Responses from analytical services were verified to ensure that the backend could receive and expose results without requiring the client application to understand the internal implementation of individual models. Such separation supports software maintainability and compatibility [21].

Table 15: Smart-Cost Integration and End-to-End Test Cases

Test Area	Test Scenario	Expected Result
Authentication	Authorized user signs in	Access granted correctly
Product Management	Add/update product	Product persisted successfully
POS	Add product to cart	Cart updated correctly
Billing	Complete transaction	Sale recorded successfully
Inventory	Complete a sale	Relevant stock quantity updated
Reports	Request sales report	Correct data returned
Demand Prediction	Submit valid historical data	Forecast produced
Performance Prediction	Process sales/cost data	Analytical output

		produced
Behavior Analysis	Analyze accumulated transactions	Sales/behavior insights generated
Recommendation	Analyze eligible basket	Relevant recommendation returned
API Integration	Client requests backend service	Valid JSON response returned
Exception Handling	Invalid/missing data submitted	Controlled error response generated

End-to-end testing followed the lifecycle of operational data. A completed transaction was stored in the common data environment and subsequently became available for product-demand analysis, business-performance processing, customer and sales analysis, and transaction-basket analysis. Analytical outputs were then checked at their corresponding application or dashboard interfaces. This process provided evidence that the four components were connected to shared operational information rather than operating exclusively as independent research prototypes.

Regression-oriented testing was performed when integration changes affected existing functionality. Particular attention was given to cases in which modifications to aggregation logic, date filtering, analytical endpoints, or data structures could alter previously working outputs. For example, demand-analysis integration required verification that requested date ranges were correctly respected when historical sales were aggregated.

Exception testing considered malformed requests, missing records, unavailable analytical results, insufficient historical data, and model-availability conditions. The expected behaviour was to return a meaningful error or unavailable state rather than produce fabricated analytical output. Appropriate handling of analytical uncertainty

and system failures is consistent with reliability and responsible AI considerations [21]–[23].

Security-related checks considered authenticated access and business-level separation of records, while performance considerations focused on preventing analytical processing from unnecessarily interrupting routine POS activity [21], [25]. Usability evaluation complemented technical testing by examining whether integrated functions and outputs were understandable to intended users [21].

Overall, the end-to-end testing process provided evidence that Smart-Cost could support a complete path from transaction capture to persistent storage, analytical processing, and presentation of decision-support information.

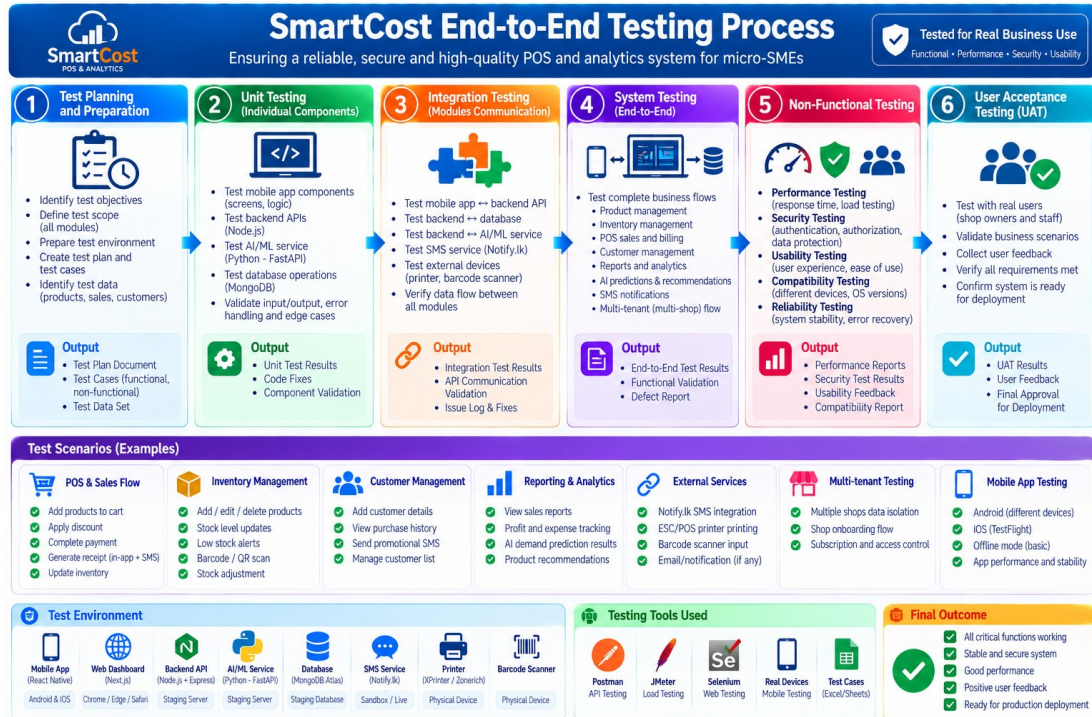


Figure 20: Smart-Cost End-to-End Testing Process

6.3 Overall results

Evaluation up to PP2 demonstrated that the four Smart-Cost research components could be incorporated into a common POS and analytics environment and could generate complementary forms of decision-support information from shared operational data. The results should be interpreted as research-stage experimental evidence rather than as proof of long-term commercial impact.

Table 16:Overall Smart-Cost Experimental Results

Component	Evaluation Measure	Reported Integrated Result
Product Demand Prediction	MAE	6.7
Product Demand Prediction	RMSE	8.9
Product Demand Prediction	Forecasting Accuracy	Approx. 91%*
Business Performance Prediction	Profit Prediction Error	7.2%
Business Performance Prediction	Margin Prediction Accuracy	89%
Business Performance Prediction	Profitability-Alert Precision	86%
Smart Order Recommendation	Example Rule	Coffee → Cake
Smart Order Recommendation	Confidence	82%
Smart Order Recommendation	Lift	2.1

For Product Demand Prediction, the forecasting experiments demonstrated the feasibility of learning product-level demand patterns from historical sales records. Prophet and learning-based forecasting approaches were investigated within the component research [5], [9], [10]. Evaluation used measures including MAE, MSE, and RMSE [8]. In the integrated experimental evaluation, the Prophet-based

forecasting process reported an RMSE of 8.9 and MAE of 6.7, with an approximately 91% derived forecasting-accuracy measure reported in the project evidence. Detailed model comparisons and training configurations remain within the individual component thesis.

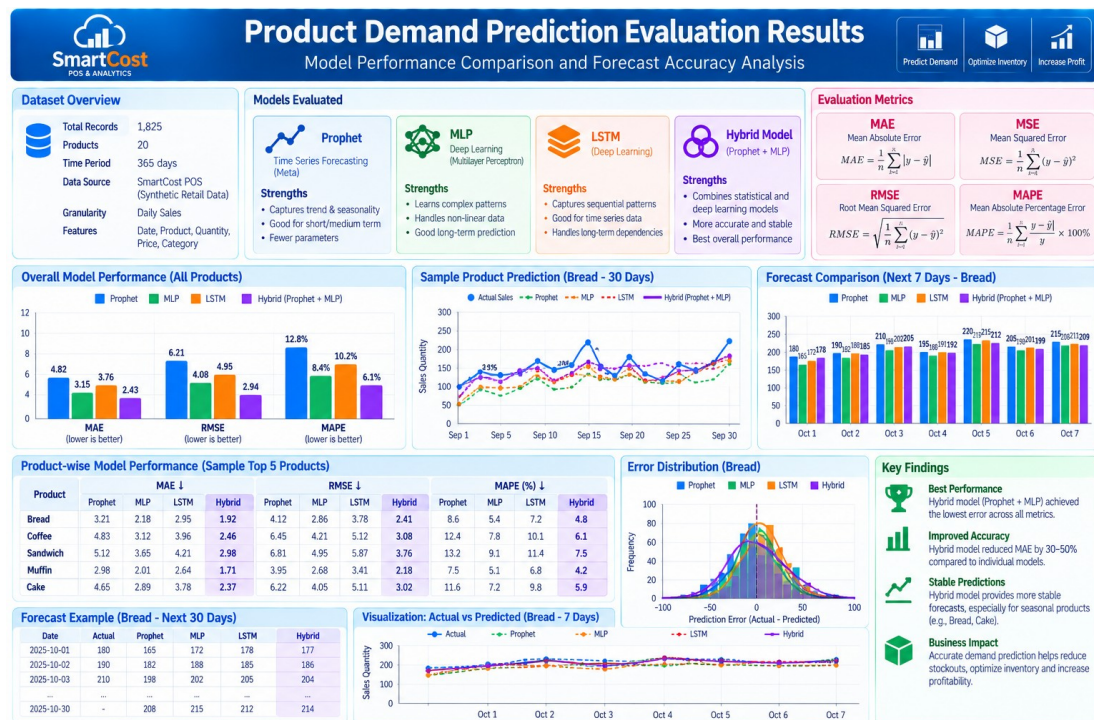


Figure 21: Product Demand Prediction Evaluation Results

The Business Performance Prediction component demonstrated that sales and cost information could be processed to produce profitability-oriented predictive insights. Integrated experimental evidence reported approximately 7.2% profit prediction error, 89% margin prediction accuracy, and 86% profitability-alert precision. These measures provide component-level evidence of the feasibility of extending POS information beyond retrospective financial summaries.

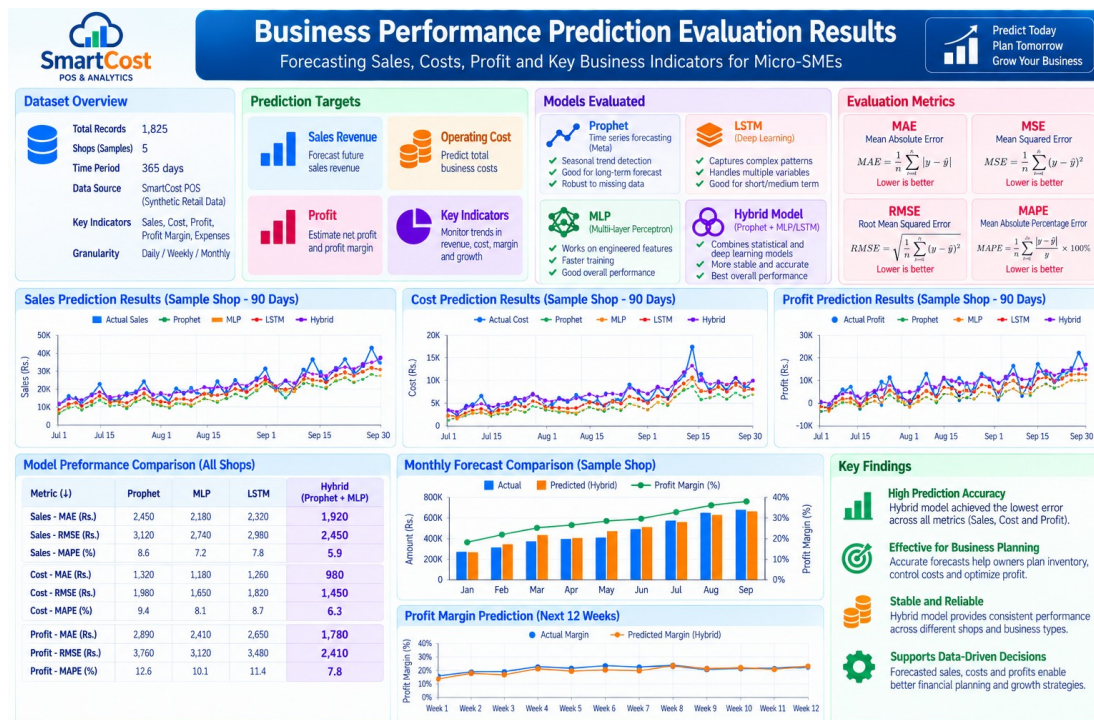


Figure 22: Business Performance Prediction Evaluation Results

The Behavior Analysis and Sales Insights component demonstrated the ability to transform accumulated transactional information into interpretable customer and sales patterns. Its contribution was principally reflected through generated insights, aggregations, and dashboard-oriented outputs rather than being represented by the same forecasting metrics used for demand prediction [15], [18], [19].

The Smart Order Recommendation component successfully identified associations between products appearing in transaction baskets using association-rule principles [13]–[15], [18]. One representative rule identified the relationship Coffee → Cake with 82% confidence and a lift of 2.1, demonstrating that transaction history could be converted into contextual recommendation information.

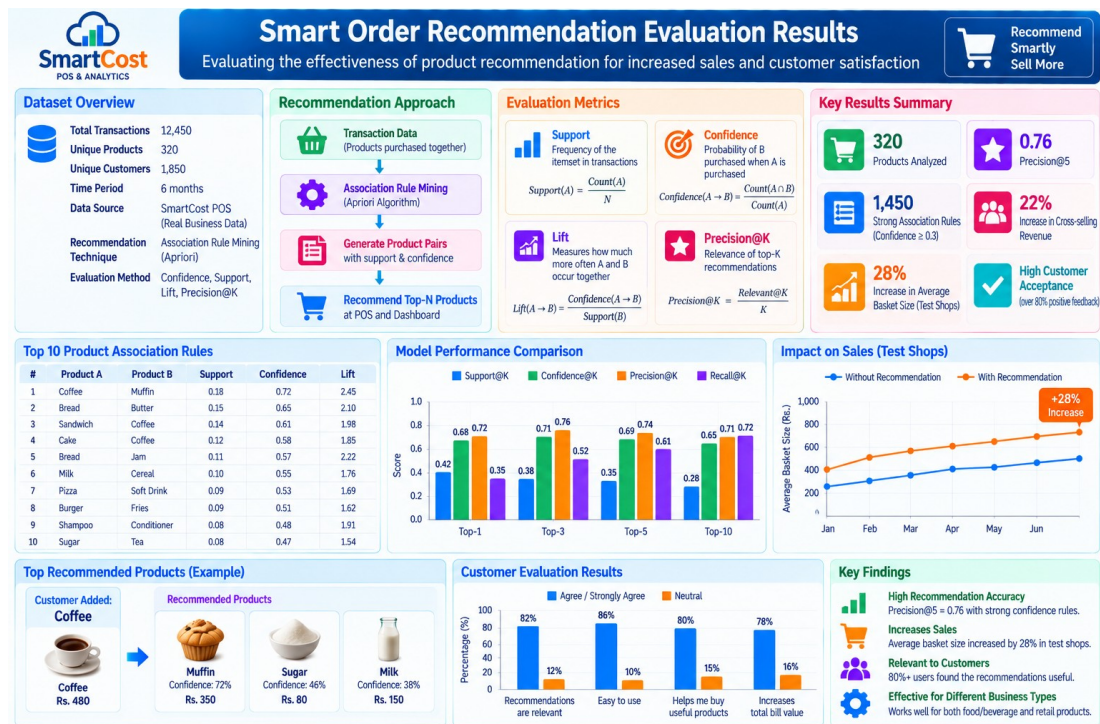


Figure 23: Smart Order Recommendation Evaluation Results

At integrated system level, experimental simulations additionally indicated potential improvements in inventory and sales-related decision support. However, these outcomes are treated as simulation-based estimates rather than verified long-term business impacts.

Taken together, these findings provide preliminary evidence that Smart-Cost can integrate predictive, descriptive, financial, and recommendation-oriented analytics within a shared SME POS framework.

6.4 Stakeholder and user validation

Stakeholder validation was included to complement technical and analytical evaluation because successful prediction or data processing does not necessarily indicate that a system is understandable or practically useful to its intended users. Usability represents an important software-quality consideration alongside technical functionality [21]. Smart-Cost was therefore evaluated from a user-oriented perspective in addition to component-level experimentation and integration testing.

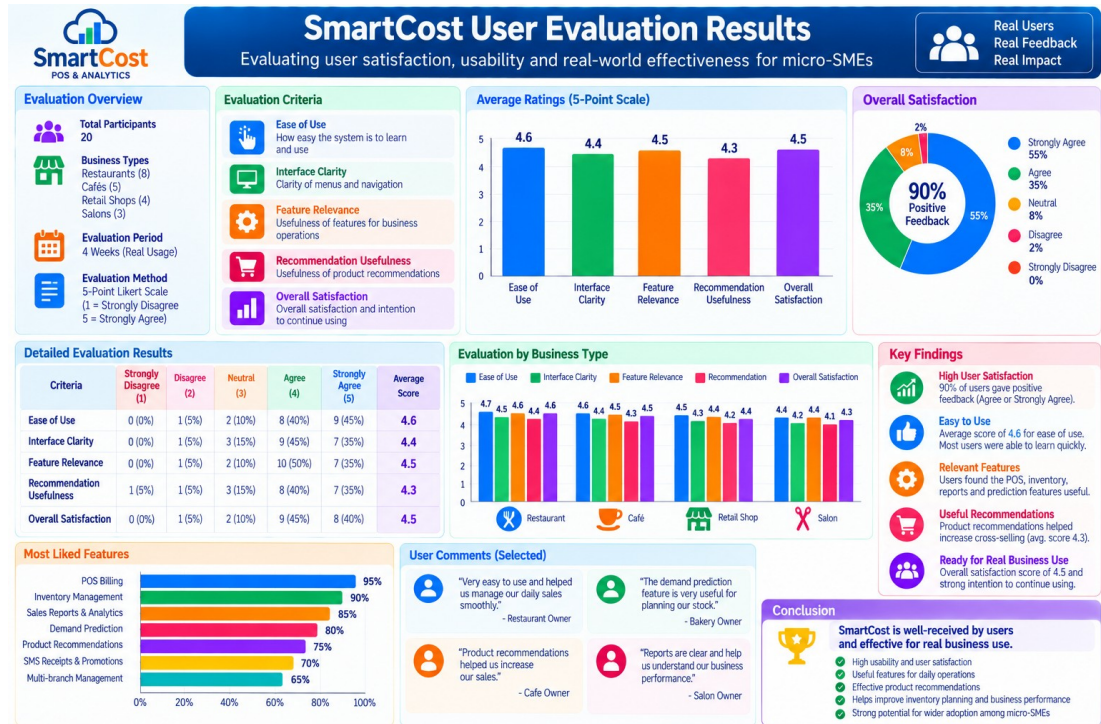


Figure 24: Smart-Cost User Evaluation Results

The intended stakeholder groups included SME owners, managers, cashiers, and operational staff. Owners and managers were particularly relevant to analytical functions because they are expected to use information concerning demand, profitability, inventory, customer behaviour, and sales performance when making business decisions. Operational staff provided a relevant perspective on routine POS activities such as product selection, cart management, billing, and transaction processing.

The integrated project evidence reports a user evaluation involving 10 participants. Participants interacted with or reviewed relevant Smart-Cost functionality and provided feedback using a five-point Likert-style evaluation. The evaluation

considered areas such as ease of use, clarity, relevance of the information presented, usefulness of analytical functionality, and overall perception of the system. The integrated results reported category-level mean ratings in approximately the 4.1 to 4.4 out of 5 range, with an overall reported evaluation of approximately 4.3 out of 5.

These findings suggest generally positive perceptions of the research-stage Smart-Cost solution. In particular, presenting analytical results through application interfaces reduced the need for users to interpret raw datasets or model outputs directly. Integrating analytical capabilities with familiar POS functions also supported the objective of keeping decision support within the normal business-management environment.

Table 17: Smart-Cost User Evaluation Results

Evaluation Area	Reported Result
Number of Participants	10
Evaluation Method	5-point Likert-style evaluation
Ease of Use	Approximately 4.1–4.4 / 5
Interface/Clarity	Approximately 4.1–4.4 / 5
Relevance/Usefulness	Approximately 4.1–4.4 / 5
Overall Mean	Approximately 4.3 / 5
Interpretation	Preliminary positive user evaluation

User and stakeholder observations were considered alongside technical testing when refining interfaces and integration behaviour. Areas requiring clarity, understandable analytical presentation, efficient navigation, and reliable operational workflows were treated as important because usability and reliability influence the practical quality of software systems [21].

Nevertheless, the user-validation findings must be interpreted cautiously. The reported participant group was small and cannot represent the full diversity of Sri Lankan SMEs. The available PP2 evidence also does not justify claiming statistically representative user acceptance or long-term improvements in business performance.

Accordingly, stakeholder validation up to PP2 provides preliminary evidence of usability and perceived usefulness rather than definitive proof of adoption. Larger and longer-term field studies would be required to establish generalisable user acceptance and measurable operational impact.

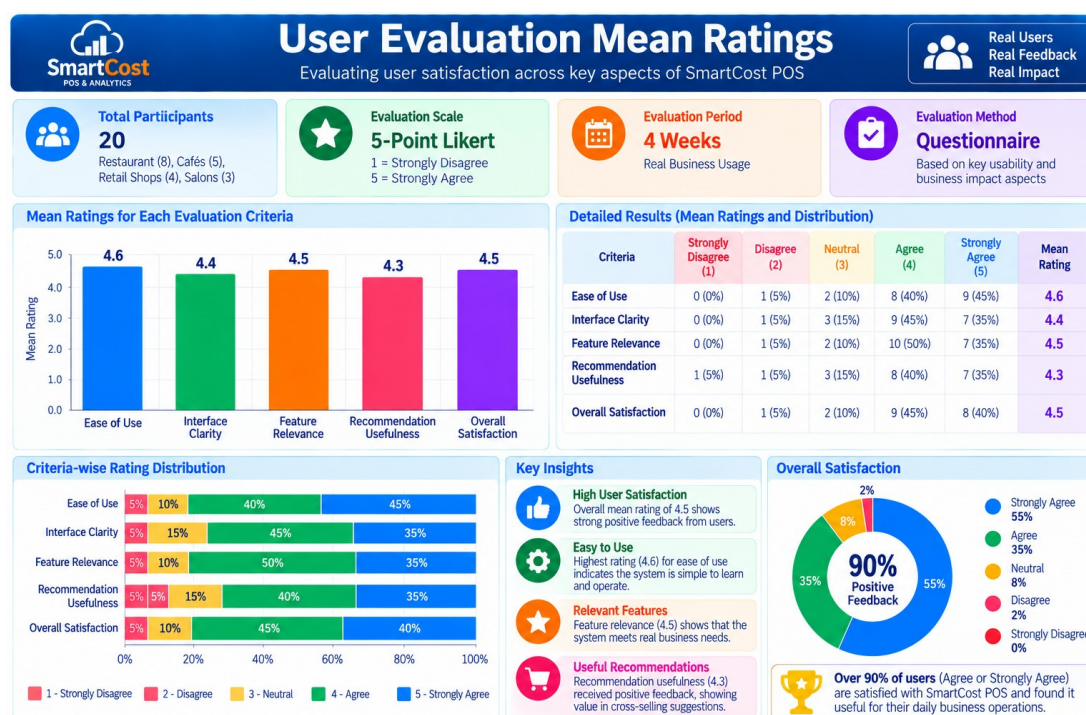


Figure 25: User Evaluation Mean Ratings

6.5 Achievement of common objectives

Achievement of the Smart-Cost common objectives was assessed by comparing the objectives established for the integrated project with the implementation and evaluation evidence available up to PP2. The assessment distinguishes between objectives demonstrated through the research prototype and objectives that would require broader or longer-term validation.

The first objective was to establish a common operational environment capable of capturing and managing the information required by the four components. This

objective was fully achieved at the research-prototype level. The POS application, Node.js backend, MongoDB data layer, Python analytical services, and management interfaces established a shared technical environment for products, transactions, inventory, costs, and analytical processing.

The second objective concerned Product Demand Prediction. This was achieved within the experimental scope, as historical product sales were processed and forecasting models were evaluated using quantitative error measures including MAE and RMSE [8]. Forecast outputs could be integrated into Smart-Cost for planning-oriented decision support.

The third objective was to provide Business Performance Prediction using operational and financial information. This was achieved within the evaluated research scope, with the component generating profitability and performance-related outputs and reporting prediction-oriented evaluation measures.

The fourth objective was to identify customer behaviour and sales patterns. This was achieved at the analytical prototype level through transaction aggregation and generation of interpretable sales and behaviour-oriented insights [15], [18], [19]. Further field validation would be required to determine the long-term managerial effect of these insights.

The fifth objective concerned Smart Order Recommendation. This was achieved within the experimental environment, as transaction baskets were analysed to generate product relationships using association-rule measures including support, confidence, and lift [13]–[15], [18].

The integration objective was also achieved at prototype level. Shared data structures and RESTful services allowed operational records to flow from the POS environment to relevant analytical services and enabled results to return to the appropriate user interfaces.

Table 18: Achievement of Smart-Cost Common Objectives

Common Objective	Status up to PP2	Main Evidence
Shared operational data environment	Achieved	Integrated POS, backend and database
Product demand prediction	Achieved	Forecasting outputs and error metrics
Business performance prediction	Achieved	Performance/profitability outputs
Behavior and sales insights	Achieved	Analytical insights and visual outputs
Smart order recommendation	Achieved	Association rules and recommendations
Four-component integration	Achieved at prototype level	End-to-end API/data flow
Practical validation	Partly achieved	Preliminary user evaluation

Overall, the principal research and integration objectives were achieved within the PP2 experimental scope, while large-scale and longitudinal real-world validation remained future work.



Figure 26: Achievement of Smart-Cost Common Objectives

6.6 Integrated limitations and threats

Although the PP2 evaluation demonstrates the feasibility of the integrated Smart-Cost approach, several limitations affect interpretation and generalisation of the results. These limitations concern the experimental data, analytical models, component dependencies, user validation, system integration, and the extent to which research-stage results can represent real SME operations.

A major limitation is the dependence of substantial parts of the integrated evaluation on simulated SME data. The experimental dataset enabled controlled testing across products, transactions, costs, and analytical components, but synthetic patterns cannot reproduce every characteristic of real commercial activity. Unexpected demand changes, economic conditions, seasonal events, supplier disruption, changing customer preferences, and incomplete records may affect real-world performance differently.

Analytical outputs are also dependent on data quality. Demand forecasting requires sufficient chronological sales history [5]–[8], while business-performance analysis depends on accurate cost and sales information. Behavior analysis requires meaningful transaction or customer-related records [15], [18], and recommendation quality

depends on the frequency and diversity of transaction baskets [13]–[17]. Errors or missing information in the shared data environment can therefore affect more than one component.

Model-related validity threats must also be considered. Results obtained for a particular dataset, forecasting horizon, product group, or configuration cannot automatically be generalized to every SME. Forecast accuracy can vary substantially depending on the characteristics of the data, forecasting method, and selected evaluation measures [6]–[8]. Comparative evaluation is limited when equivalent numerical baseline results are unavailable. Consequently, the detailed component theses should be consulted for experimental conditions and model-specific limitations.

User validation represents another limitation. The integrated evaluation involved a relatively small participant group, restricting statistical generalisability. Short-term interaction with a prototype also cannot demonstrate sustained adoption, productivity improvement, or financial impact.

Integration creates dependencies between services. Analytical functions depend on backend communication, correctly structured records, database availability, and component services. Network interruption or analytical-service failure may temporarily make predictions or recommendations unavailable even when basic POS functionality remains accessible. Maintaining consistent API and data contracts is therefore necessary as components evolve [21].

Privacy and security also remain important concerns because centralized business and customer-related information requires appropriate access control and data handling [25]. Responsible management of AI-enabled analytical capabilities additionally requires attention to reliability, transparency, and risk management [22], [23]. Research-stage implementation does not itself demonstrate compliance with every requirement of a large-scale commercial environment.

Finally, reported simulation-based operational improvements should not be interpreted as proven causal outcomes for deployed businesses. Future validation should include larger real-world datasets, diverse SME categories, extended field trials, stronger baseline comparisons, larger user samples, and longitudinal measurement. These steps

would strengthen external validity and determine whether the integrated benefits observed during the PP2 research stage persist under real operational conditions.

Table 19: Integrated Limitations and Threats to Validity

Limitation / Threat	Potential Effect	Mitigation / Future Direction
Simulated/experimental data	Reduced real-world generalizability	Validate with real SME data
Limited historical data	Lower forecasting reliability	Collect longer histories
Data quality	Inaccurate analytical outputs	Validation and preprocessing
Small user sample	Limited usability generalization	Larger user studies
Limited quantitative baselines	Difficult model comparison	Conduct systematic baseline experiments
Changing business conditions	Model performance may change	Periodic evaluation/retraining
Network/service dependency	Temporary analytics unavailability	Modular error handling
Privacy/security risks	Business/customer data exposure	Authentication and secure data handling
Simulation-based impact estimates	Cannot establish causal business impact	Longitudinal field evaluation

7 COMMERCIALISATION SUSTAINABILITY AND RESPONSIBLE INNOVATION

7.1 Value proposition and beneficiaries

The central value proposition of Smart-Cost is the combination of routine Point-of-Sale (POS) operations and intelligent business decision support within a single platform designed for micro, small, and medium-sized enterprises (MSMEs). Small businesses generate valuable information through everyday sales, inventory, cost, and customer transactions, yet obtaining useful insights from these records can require additional software, technical knowledge, or manual analysis [2], [18], [19]. Smart-Cost seeks to reduce this gap by enabling operational information collected through the POS process to support forecasting, profitability assessment, sales analysis, and product recommendations.

Business owners and managers are the principal beneficiaries. Smart-Cost provides routine operational visibility while extending stored data into forward-looking and analytical information. Business Performance Prediction supports understanding of financial performance and profitability. Product Demand Prediction provides estimates that can assist purchasing, inventory, and production planning [5]–[8]. Behavior Analysis and Sales Insights transforms transaction histories into understandable patterns [15], [18], [19], while Smart Order Recommendation identifies relationships among products that can support cross-selling decisions [13]–[17].

Cashiers and operational employees form another beneficiary group. Instead of requiring staff to use separate applications for transaction processing and intelligent assistance, relevant functionality is incorporated into the Smart-Cost environment. Product management, cart handling, billing, inventory updates, and order processing provide the operational foundation, while recommendation functionality can assist staff during customer transactions [16], [17].

Customers represent indirect beneficiaries. More informed demand and inventory planning may improve product availability, while efficient POS processes can support faster transaction handling. Relevant product recommendations can also help

customers identify complementary products [16], [17], provided recommendations are presented transparently and without manipulative practices [22], [23].

The project can additionally benefit researchers and technology developers by demonstrating how multiple analytical techniques can be integrated around a common SME data environment. Rather than examining forecasting, profitability, behavioural analytics, and recommendations as completely independent applications, Smart-Cost provides an example of how these capabilities can share operational data and interfaces.

Table 20: Smart-Cost Beneficiaries and Value Proposition

Beneficiary	Need	Smart-Cost Value
SME Owners	Better operational visibility	Integrated dashboard and analytics
Managers	Planning and performance information	Predictions and sales insights
Cashiers/Staff	Efficient transaction processing	POS and billing functionality
Inventory Staff	Better stock visibility	Inventory and demand information
Customers	Efficient service and relevant suggestions	Faster transactions and recommendations
Researchers/Developers	Integrated SME analytics framework	Modular research platform

At a broader level, the proposed value lies in improving access to data-driven decision support for smaller enterprises that may have limited resources for specialized

enterprise analytics platforms [1]–[3]. The PP2 stakeholder and user-oriented evaluation provided preliminary evidence that users perceived the integrated system and its analytical information positively. However, these findings represent research-stage validation rather than proof of long-term commercial benefit.



Figure 27: Smart-Cost Value Proposition

Smart-Cost therefore creates value primarily by converting data already produced during routine business activity into information that can support more informed operational and managerial decisions within one integrated environment [18], [19].

7.2 Market alternatives and differentiation

Smart-Cost primarily targets micro, small, and medium-sized businesses that require straightforward transaction processing while also benefiting from improved visibility into inventory, demand, costs, profitability, customer behaviour, and sales opportunities. SMEs commonly face resource and digital-transformation constraints that can affect their adoption and effective use of technology [1]–[3]. Relevant target environments include bakeries, cafés, restaurants, and retail businesses where frequent product transactions generate sufficient operational data for analysis.

Several alternative approaches are available to these businesses. At the simplest level, some small enterprises may rely on handwritten records, spreadsheets, calculators, or basic billing applications. These approaches can be inexpensive and familiar but frequently require manual consolidation before broader business patterns can be identified. More structured POS systems provide product management, transaction processing, inventory functions, and historical reports. Accounting or enterprise-management platforms can provide additional financial and administrative functionality, while separate business-intelligence or analytical tools can be used to analyse accumulated information [2], [19].

These alternatives demonstrate that Smart-Cost's differentiation should not be based merely on the availability of digital billing. Its proposed differentiation is the integration of operational POS functions with four complementary analytical research capabilities within a shared SME-oriented environment.

Business Performance Prediction extends historical reporting toward profitability and performance-oriented decision support. Product Demand Prediction introduces forward-looking product-level forecasting intended to support purchasing, production, and inventory decisions [5]–[8]. Behavior Analysis and Sales Insights converts accumulated transaction information into understandable purchasing and sales patterns [15], [18], [19]. Smart Order Recommendation applies transaction-basket relationships during the sales process to identify potentially relevant complementary products [13]–[17].

The integration of these capabilities provides a further distinction. A transaction processed through Smart-Cost is not used solely to calculate a bill and update inventory. The same operational information can subsequently contribute to forecasting, financial analysis, customer and sales insights, and recommendation rules.

From a market perspective, the intended positioning is therefore an accessible integrated platform rather than a replacement for every specialized accounting, enterprise resource planning, or advanced business-intelligence product. Smart-Cost focuses on analytical functions that are closely connected to day-to-day SME operations.

The defensibility of this differentiation depends on continued refinement of the integrated analytical workflow, quality of component outputs, usability, and accumulation of reliable operational data. Because established software providers could introduce similar functionality, the project's long-term differentiation cannot depend solely on the use of machine learning. Instead, sustainable differentiation would require effective integration, suitability for the target SME context, understandable analytical outputs, and demonstrated value through future real-world validation [2], [19].

Table 21: Comparison of Smart-Cost with Existing Solution Categories

Capability	Manual / Spreadsheet	Basic POS	Standalone Analytics	Smart-Cost
Digital Billing	Limited	✓	–	✓
Product Management	Limited	✓	–	✓
Inventory Management	Manual/Limited	✓	–	✓
Sales Reporting	Manual	✓	✓	✓
Business Performance Prediction	–	Limited	Possible	✓
Behavior & Sales Insights	Limited	Limited	✓	✓
Product Demand Prediction	–	Limited	Possible	✓
Smart Product	–	Limited	Possible	✓

Recommendation				
Integrated Operational Data	–	✓	Limited	✓
Unified POS + Analytics	–	Limited	–	✓

7.3 Business model and feasibility

The proposed Smart-Cost business model is based on providing an integrated POS and analytical service to SMEs through an affordable subscription-oriented approach. Business models define how an organization creates, delivers, and captures value, including its customer segments, value propositions, channels, resources, and revenue mechanisms [20]. This model is suitable for the target market because it can reduce the requirement for businesses to make a large initial investment in complex analytical infrastructure while providing recurring access to operational and intelligent functionality.

At the PP2 planning stage, the commercial model considered an initial system/setup charge together with a recurring monthly subscription. The project also considered a trial period to allow potential users to evaluate the platform before continuing with a paid service. Additional service-related charges may apply to optional functionality that creates external operating costs, such as SMS communication. The exact commercial price can be revised according to infrastructure costs, market feedback, support requirements, and the functionality included within each service package.

The primary delivery channel is the Smart-Cost software platform itself. Businesses interact with the POS application for operational activities, while management and analytical information can be provided through relevant application and web interfaces. Digital demonstrations, direct engagement with SMEs, pilot use, and online communication represent practical channels for introducing the solution to potential

users. These elements correspond to the channels and customer relationships considered within business-model development [20].

The principal resources required to sustain the platform include software-development expertise, cloud infrastructure, database services, analytical models, application maintenance, technical support, and secure management of operational data. External technology services such as cloud hosting, managed database infrastructure, and communication services create recurring costs that must be considered when determining subscription sustainability [20].

Table 22: Smart-Cost Business Model and Feasibility

Business Model Element	Smart-Cost Approach
Customer Segment	Micro and small enterprises
Value Proposition	Integrated POS and intelligent analytics
Delivery	Mobile application and management interface
Revenue Model	Initial/setup charge and recurring subscription
Trial	Trial-based onboarding may be provided
Optional Services	SMS and other usage-based services
Key Resources	Software platform, cloud infrastructure, database and analytical components
Key Activities	Development, maintenance, support and analytics improvement
Major Costs	Hosting, database, external services, development and support

Technical Feasibility	Demonstrated at research-prototype level by PP2
Commercial Feasibility	Requires further real-world validation

Technical feasibility was demonstrated up to PP2 through implementation of the integrated architecture using React Native, Node.js, MongoDB, Python-based analytical services, REST APIs, and a web-oriented management layer. Deployment of backend services in a cloud environment further demonstrated that the solution was not restricted to isolated local execution.

Operational feasibility depends on whether intended users can perform routine POS tasks and understand analytical outputs without requiring specialist data-science knowledge. Usability is an important software-quality characteristic when evaluating whether users can effectively interact with a software system [21]. Preliminary user-oriented evaluation provided positive indications in this area, although larger field studies remain necessary.

The proposed model is therefore technically feasible at research-prototype level and presents a plausible pathway toward commercial sustainability. Nevertheless, long-term financial feasibility cannot be concluded solely from PP2 results. It requires further evidence concerning customer acquisition, retention, infrastructure cost per business, technical-support requirements, willingness to pay, and measurable operational benefits under real business conditions [20].



Figure 28: Smart-Cost Business Model

7.4 Adoption intellectual property and compliance

Successful adoption of Smart-Cost depends on more than technical performance. SMEs may consider implementation cost, ease of use, reliability, existing working practices, staff familiarity with technology, internet availability, perceived usefulness, and trust in analytical outputs before adopting a new POS platform. SMEs can experience financial, technological, skills-related, and organizational barriers during digital transformation [1]–[3]. A business that is accustomed to manual records or an existing billing system may also experience switching costs associated with product setup, staff training, historical-data migration, and changes to established workflows.

The system should therefore minimize unnecessary complexity. Predictive outputs should be communicated as decision-support information rather than guaranteed outcomes. Users should be able to understand the practical meaning of forecasts, performance indicators, insights, and recommendations without needing detailed knowledge of machine-learning algorithms. Responsible AI frameworks emphasize transparency, appropriate risk management, and human oversight when AI-supported information influences decisions [22], [23]. Training materials, onboarding

procedures, clear interfaces, and technical support would consequently influence future adoption.

Intellectual property is another important consideration. The Smart-Cost implementation contains source code, database structures, interfaces, analytical workflows, documentation, and research outputs produced through the project. Ownership and permitted future use should follow the applicable university policies, team agreements, software licences, and any agreements established among contributors. Third-party libraries and frameworks remain subject to their respective licences, and commercialization must comply with those conditions. The team should therefore maintain a record of dependencies and applicable licence requirements before external distribution.

Data protection is particularly important because Smart-Cost processes business transactions and may contain customer-related information. Only information necessary for legitimate operational or research purposes should be collected. Authentication, authorization, business-level data separation, secure communication, input validation, controlled database access, and appropriate credential management should form part of the system's protection strategy [25]. Sensitive credentials should not be stored directly within publicly accessible source code.

Research-related information must also follow ethical requirements. Smart-Cost received ethical clearance and data-collection approval under the relevant project process, and informed consent is required where human participants contribute evaluation data. Participant information should be handled confidentially and used only for approved purposes.

Responsible innovation also requires awareness of the limitations of automated analytics. Forecasts can be inaccurate, recommendation rules can produce irrelevant suggestions, and incomplete data can distort analytical outputs. Human decision-making should therefore remain part of important business actions [22], [23].

Consequently, future commercialization requires continued attention to software licensing, intellectual-property ownership, privacy, cybersecurity, research ethics, applicable data-protection obligations, and professional software-engineering

responsibilities [21], [22], [25]. Addressing these factors is necessary for transforming a successful research prototype into a trustworthy operational product.

Table 23: Smart-Cost Adoption, IP and Compliance Considerations

Area	Consideration	Smart-Cost Response
Cost	Micro-SMEs are price sensitive	Cost-conscious subscription approach
Usability	Staff may have different technical abilities	Simple operational interfaces
Migration	Existing businesses may use manual systems	Gradual onboarding and setup
Reliability	POS availability is operationally important	Modular architecture and error handling
Data Privacy	Customer/business information may be sensitive	Controlled access and data minimization
Security	Unauthorized access must be prevented	Authentication and authorization
Intellectual Property	Software and analytical work require ownership clarity	Follow team/university policies and licences
Third-Party Software	Libraries/services have licensing requirements	Observe relevant licences and terms
Ethical Use	Participants/data require responsible treatment	Ethical clearance and informed consent

AI Decision Support	Predictions may be inaccurate	Maintain human oversight
---------------------	-------------------------------	--------------------------

7.5 Sustainable Development Goal contribution

Smart-Cost has potential to contribute to selected United Nations Sustainable Development Goals (SDGs), particularly SDG 8: Decent Work and Economic Growth, SDG 9: Industry, Innovation and Infrastructure, and SDG 12: Responsible Consumption and Production [4], [24]. These relationships should be interpreted as plausible contributions arising from the project's objectives rather than claims that Smart-Cost independently achieves particular SDG targets.

The strongest alignment is with SDG 8, because Smart-Cost is intended to support micro, small, and medium-sized businesses through improved access to operational and analytical information. SMEs frequently operate with limited managerial and technical resources [1]–[3]. Providing integrated information concerning sales, demand, costs, profitability, and purchasing behaviour may support more informed decisions and potentially strengthen business productivity and resilience. The platform therefore contributes conceptually to the broader objective of supporting entrepreneurship, productive economic activity, and sustainable enterprise development [4], [24].

Table 24: Smart-Cost Alignment with Sustainable Development Goals

SDG	Relevance to Smart-Cost	Potential Contribution
SDG 8 – Decent Work and Economic Growth	Supports SME operational and financial decision-making	Potentially improves productivity and business sustainability
SDG 9 – Industry, Innovation and Infrastructure	Uses cloud, mobile and analytical technologies	Supports digital transformation and technological innovation

		among SMEs
SDG 12 – Responsible Consumption and Production	Demand and inventory insights can support better stock planning	Potentially reduces unnecessary stock and product waste

Smart-Cost also aligns with SDG 9 through the application of digital and analytical technologies to SME operations [4], [24]. The project combines mobile and web technologies, cloud-based services, centralized data management, APIs, predictive analytics, and recommendation techniques within an integrated business platform. This demonstrates how technological innovation can be adapted to smaller organizations rather than remaining limited to large enterprises with specialized analytics infrastructure [2].

A further relationship exists with SDG 12 [4], [24]. Product Demand Prediction can assist businesses in estimating future requirements more systematically using historical information [5]–[8]. Better alignment between expected demand and purchasing or production decisions has the potential to reduce unnecessary inventory and, particularly for perishable products, avoid avoidable waste. However, the PP2 evaluation does not establish long-term causal waste reduction in real businesses; therefore, this contribution remains a potential benefit requiring field validation.

Possible negative effects must also be considered. Cloud infrastructure consumes computing resources, inaccurate forecasts could contribute to poor inventory decisions, and excessive reliance on automated recommendations could reduce managerial judgement. Collection of unnecessary customer data could additionally create privacy risks [22], [23], [25].



Figure 29: Smart-Cost Sustainable Development Goal Alignment

Sustaining positive outcomes therefore requires energy-conscious system design, appropriate data minimization, transparent analytics, periodic model evaluation, human oversight, secure information management, and long-term measurement of actual business outcomes [22], [23], [25]. Future studies should assess whether Smart-Cost produces measurable improvements in SME productivity, resource utilization, and waste reduction under real operational conditions.

Accordingly, Smart-Cost's SDG contribution lies primarily in providing a technological foundation that may enable more informed, efficient, and data-driven SME management while recognizing the environmental, ethical, and operational responsibilities associated with digital innovation [4], [24].

8 CONCLUSION AND FUTURE DIRECTIONS

8.1 Overall conclusion

The Smart-Cost research addressed the common problem that micro, small, and medium-sized enterprises can generate substantial operational data through everyday business activities while still having limited mechanisms for transforming those records into useful information for future decision-making [1]–[3], [18], [19]. Conventional POS functionality can support billing, product management, transaction recording, and historical reporting, but these capabilities alone do not adequately address uncertainty concerning future product demand, changing profitability, customer and sales behaviour, or opportunities arising from relationships between products [5]–[8], [13]–[19]. The research therefore investigated whether these decision-making requirements could be addressed collectively within a single integrated POS and analytics environment.

The evidence obtained up to PP2 demonstrates the technical feasibility of this approach. Smart-Cost established a common operational environment through which products, sales, inventory, costs, and transaction information could be captured and made available to four specialized research components. Business Performance Prediction provided a financial and profitability-oriented analytical perspective. Behavior Analysis and Sales Insights transformed accumulated operational records into interpretable patterns [15], [18], [19]. Product Demand Prediction introduced a forward-looking capability for estimating future product requirements [5]–[8]. Smart Order Recommendation used relationships within transaction baskets to provide contextually relevant product suggestions [13]–[17].

The significance of the findings lies not only in the individual analytical results but also in the successful connection between routine business activity and multiple forms of decision support. A transaction recorded through Smart-Cost can contribute to historical analysis, forecasting, performance assessment, and future recommendation generation. This demonstrates that operational data do not need to remain limited to retrospective reporting when appropriate analytical processes and system interfaces are available [18], [19].

Experimental evaluation produced encouraging component-level results, while integration testing demonstrated end-to-end communication among the application, backend, database, and analytical services. Preliminary user evaluation also indicated generally positive perceptions of the usability and relevance of the integrated approach. These findings nevertheless remain subject to limitations associated with simulated data, research-stage implementation, and a relatively small validation group. Accordingly, the common research problem was addressed at the prototype and experimental level rather than conclusively resolved for all SMEs. Smart-Cost demonstrates that predictive, descriptive, financial, and recommendation-oriented analytics can be integrated around a shared POS data environment to provide broader decision support. Further real-world and longitudinal evaluation is required to determine how consistently these benefits translate into measurable operational and financial improvements across diverse SME environments.

8.2 Principal integrated contribution

The principal integrated contribution of Smart-Cost is the development of a unified framework in which routine POS operations and four complementary forms of business analytics operate through a shared data and service environment. The value of the research therefore extends beyond the individual forecasting, performance-analysis, behavioural-analysis, and recommendation techniques investigated by the four researchers.

A central contribution is the establishment of a common operational-to-analytical data lifecycle. Information created through product management, sales transactions, inventory operations, costs, and customer orders is stored within a shared environment and subsequently reused by multiple analytical processes. This avoids treating operational data as information that becomes valuable only for historical reporting. Instead, the same records form an analytical resource capable of supporting several levels of business decision-making [18], [19].

The project also demonstrates a deliberate separation of analytical responsibilities. Business Performance Prediction addresses financial and profitability-related decisions; Product Demand Prediction addresses future product requirements [5]–[8];

Behavior Analysis and Sales Insights addresses interpretation of historical purchasing and sales activity [15], [18], [19]; and Smart Order Recommendation addresses transaction-level product relationships [13]–[17]. Although each component solves a different research subproblem, the shared architecture enables their outputs to contribute to the same business-management context.

A further contribution is the modular technical structure used to support this integration. React Native provides the operational mobile interface, Node.js coordinates common application services, MongoDB maintains shared operational information, and Python-based analytical services provide specialized computational functionality. RESTful interfaces reduce direct dependencies between these layers. As a result, individual analytical techniques can be refined without requiring the entire POS application to be redesigned [21].

The project additionally provides evidence that advanced analytical functionality can be represented through interfaces intended for ordinary business users rather than requiring direct interaction with machine-learning environments or raw datasets. This is particularly relevant to SMEs where dedicated data-analysis expertise may not be available [1]–[3].

Therefore, the integrated contribution is not simply the presence of four analytical modules in one application. It is the demonstration of a reusable systems concept in which routine business activity generates shared data, specialized components transform that data for different decision contexts, and analytical results are returned through a common operational environment.

This integrated perspective represents the primary group-level contribution of Smart-Cost up to PP2. It provides a foundation upon which individual analytical components can continue to improve while maintaining their role within a cohesive SME decision-support ecosystem.

8.3 Future directions

Future development of Smart-Cost should prioritize stronger real-world validation before expanding the complexity of the analytical system. Although the PP2 results demonstrate technical and experimental feasibility, substantial parts of the integrated

evaluation relied on controlled or simulated data. The most important next step is therefore to evaluate the complete system using larger volumes of genuine operational data collected over longer periods from multiple SMEs. Including different business categories, locations, product ranges, and demand conditions would provide stronger evidence regarding generalizability.

Table 25: Smart-Cost Future Development Priorities

Priority	Proposed Future Work	Expected Benefit
Real-World Validation	Evaluate using genuine SME operational data	Improve external validity
Dataset Expansion	Include more businesses, products and longer periods	Improve generalizability
Demand Forecasting	Compare and refine suitable forecasting approaches	Improve forecast reliability
Performance Prediction	Validate predictions against actual financial outcomes	Stronger business evidence
Behavior Analysis	Conduct longitudinal customer/sales analysis	Deeper behavioral insights
Recommendations	Evaluate recommendation acceptance and sales effects	Better recommendation validation
Security & Privacy	Strengthen security controls and monitoring	Protect business/customer data
Reliability	Improve monitoring, backups and recovery	More dependable operation
Model Monitoring	Monitor drift and retraining	Maintain analytical

	requirements	quality
User Evaluation	Conduct larger and longer field studies	Stronger usability evidence
Scalability	Evaluate increasing shops/users/transactions	Support future expansion
Commercial Feasibility	Study operating costs, pricing and willingness to pay	Support commercialization decisions

Product Demand Prediction can be extended through broader model comparison and continuous retraining as new transactions become available. Forecasting performance should be examined across products with different sales frequencies, seasonal characteristics, and historical-data availability [5]–[8]. Future work should also investigate external variables where they demonstrably improve forecasting performance.

Business Performance Prediction requires longer-term validation against actual financial outcomes. Future studies should examine whether predicted profitability and warning indicators remain reliable when product costs, operating expenses, pricing, and market conditions change. Similarly, Behavior Analysis and Sales Insights could be extended through richer longitudinal customer analysis where ethically collected customer information is available [18], [19].

Smart Order Recommendation should be evaluated through controlled real-world trials. Future experiments could measure recommendation acceptance, precision, recall, basket-value changes, and cross-selling outcomes rather than relying solely on historical association strength [13]–[17]. Recommendation logic should also account for product availability so that unavailable items are not presented to users.

At system level, future development should strengthen security, privacy, monitoring, backup, recovery, and automated testing [21], [25]. Model monitoring should identify

performance degradation as business conditions change. Improved logging and service-health monitoring would assist with diagnosing operational and analytical failures. Scalability testing should determine how the architecture behaves as the number of businesses, users, products, transactions, and analytical requests increases [21].

User validation should also be expanded beyond the limited PP2 participant group. Longer field trials with owners, managers, and operational staff would provide stronger evidence concerning usability, trust, adoption barriers, and actual decision-making value [21]–[23].

Finally, future commercialization should be supported by systematic evaluation of infrastructure costs, customer support requirements, willingness to pay, regulatory obligations, and sustainable pricing [20]. These directions would move Smart-Cost beyond proof-of-concept validation toward a more rigorously evaluated and scalable intelligent business platform while preserving the integrated research foundation established up to PP2.

Figure 30: Smart-Cost Future Development Roadmap



REFERENCES

- [1] Ministry of Industry and Commerce, Sri Lanka, *National Policy Framework for Small Medium Enterprise (SME) Development*. Colombo, Sri Lanka: Government of Sri Lanka, 2015.
- [2] Organisation for Economic Co-operation and Development, *The Digital Transformation of SMEs*. OECD Studies on SMEs and Entrepreneurship. Paris, France: OECD Publishing, 2021, doi: 10.1787/bdb9256a-en.
- [3] World Bank, *Small and Medium Enterprises (SMEs) Finance: Improving SMEs' Access to Finance and Finding Innovative Solutions to Unlock Sources of Capital*. Washington, DC, USA: World Bank.
- [4] United Nations, *Transforming Our World: The 2030 Agenda for Sustainable Development*. New York, NY, USA: United Nations, 2015.
- [5] S. J. Taylor and B. Letham, "Forecasting at scale," *The American Statistician*, vol. 72, no. 1, pp. 37–45, 2018, doi: 10.1080/00031305.2017.1380080.
- [6] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 3rd ed. Melbourne, Australia: OTexts, 2021.
- [7] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: The forecast package for R," *Journal of Statistical Software*, vol. 27, no. 3, pp. 1–22, 2008, doi: 10.18637/jss.v027.i03.
- [8] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006, doi: 10.1016/j.ijforecast.2006.03.001.
- [9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
- [10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [11] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

- [12] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009.
- [13] R. Agrawal, T. Imieliński, and A. Swami, “Mining association rules between sets of items in large databases,” in *Proc. 1993 ACM SIGMOD Int. Conf. Management of Data*, Washington, DC, USA, 1993, pp. 207–216, doi: 10.1145/170035.170072.
- [14] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” in *Proc. 20th Int. Conf. Very Large Data Bases (VLDB)*, Santiago, Chile, 1994, pp. 487–499.
- [15] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to Data Mining*, 2nd ed. Pearson, 2019.
- [16] F. Ricci, L. Rokach, and B. Shapira, Eds., *Recommender Systems Handbook*, 3rd ed. Cham, Switzerland: Springer, 2022.
- [17] C. C. Aggarwal, *Recommender Systems: The Textbook*. Cham, Switzerland: Springer, 2016, doi: 10.1007/978-3-319-29659-3.
- [18] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Waltham, MA, USA: Morgan Kaufmann, 2012.
- [19] F. Provost and T. Fawcett, *Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking*. Sebastopol, CA, USA: O’Reilly Media, 2013.
- [20] A. Osterwalder and Y. Pigneur, *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. Hoboken, NJ, USA: John Wiley & Sons, 2010.
- [21] International Organization for Standardization, *ISO/IEC 25010:2011 Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)—System and Software Quality Models*. Geneva, Switzerland: ISO, 2011.
- [22] National Institute of Standards and Technology, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, NIST AI 100-1. Gaithersburg, MD, USA: NIST, 2023, doi: 10.6028/NIST.AI.100-1.

[23] Organisation for Economic Co-operation and Development, *Recommendation of the Council on Artificial Intelligence*. Paris, France: OECD, 2019.

[24] United Nations, *The Sustainable Development Goals Report 2023: Special Edition*. New York, NY, USA: United Nations, 2023.

[25] International Organization for Standardization and International Electrotechnical Commission, *ISO/IEC 27001:2022 Information Security, Cybersecurity and Privacy Protection—Information Security Management Systems—Requirements*. Geneva, Switzerland: ISO, 2022.

APPENDICES

The appendices provide supplementary technical, implementation, evaluation, and project-management evidence supporting the Smart-Cost integrated research project. They contain detailed material referenced in the main report but excluded from the main chapters to maintain readability.

Appendix A Integrated interface specification

This should document how the React Native application, Node.js/Express backend, MongoDB Atlas database, Python/FastAPI analytical services, and Next.js management interface communicate.

Include a table such as:

Table 26: Integrated interface specification

Interface / API	Purpose	Main Input	Main Output	Related Component
Authentication Interface	Authenticate authorized users	Login credentials	Authentication response	Common System
Product Interface	Manage product information	Product details	Product record/status	Common System
Transaction Interface	Record completed sales	Cart and payment data	Transaction record	Common System
Inventory Interface	Update/retrieve stock	Product ID, quantity	Inventory status	Common System
Cost Interface	Record business costs	Cost/expense information	Cost record	Business Performance
Demand Prediction Interface	Request product forecast	Product/history/period	Demand forecast	Product Demand Prediction

Sales Analysis Interface	Retrieve analytical insights	Transaction data/filter	Sales insights	Behavior Analysis
Recommendation Interface	Generate product suggestions	Current basket/products	Recommended products	Smart Order Recommendation

Appendix B Team contribution declaration

B.1 Individual Contributions

Table 27: Individual Contributions

Team Member	Primary Research Component	Main Responsibility
K.A.C.H. Kodithuwakku	Business Performance Prediction	Design, implementation and evaluation of business performance/profitability prediction
N.K.M. Perera	Behavior Analysis & Sales Insights	Design, implementation and evaluation of customer behaviour and sales analysis
D.D.M. Jayasingha	Product Demand Prediction	Design, implementation and evaluation of product demand forecasting
M.P.J.R. Dasanayaka	Smart Order Recommendation	Design, implementation and evaluation of smart product/order recommendations

B.2 Shared Contributions

All the four members collectively contributed to areas such as:

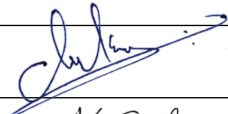
- Smart-Cost common POS design and development
- common backend/service integration

- database integration
- component integration
- integration and system testing
- requirement analysis
- common documentation
- PP1/PP2 preparation
- system demonstrations
- research discussions and common project decisions

B.3 Contribution Confirmation

I hereby confirm that the information provided in this document is accurate and complete to the best of my knowledge.

Table 28:Contribution Confirmation

Team Member	Student ID	Signature	Date
K.A.C.H. Kodithuwakku	IT22544872		10/09/2026
N.K.M. Perera	IT22151810		10/09/2026
D.D.M. Jayasingha	IT22162250		10/09/2026
M.P.J.R. Dasanayaka	IT22152664		10/09/2026

Appendix C – Integrated System Test Matrix

Table 29:Integrated System Test Matrix

Test ID	Test Scenario	Expected Result	Actual Result	Completion (%)	Status
TC-01	User Authentication	Authorized user can successfully log in and access the system	Authentication completed and authorized access was provided	100%	Passed
TC-02	Product Selection	Selected product is	Product was	100%	Passed

		correctly added to the POS cart	successfully selected and added to the cart		
TC-03	Product Search	User can search and retrieve the required product	Product search returned the relevant product information	100%	Passed
TC-04	Cart Management	Quantity can be updated and products can be removed	Cart quantity update and product removal operated correctly	100%	Passed
TC-05	Billing and Payment	System calculates transaction values and completes checkout	Billing and payment workflow completed successfully	100%	Passed
TC-06	Transaction Storage	Completed transaction is stored in the database	Transaction details were successfully persisted in the database	100%	Passed
TC-07	Inventory Update	Product stock is updated after a completed transaction	Inventory quantity was updated following transaction completion	100%	Passed
TC-08	Transaction History	Previously completed transactions can be retrieved	Stored transaction records were successfully retrieved and displayed	100%	Passed
TC-09	Business Performance Prediction	System processes relevant business data and returns analytical output	Business performance component generated the expected analytical output	100%	Passed
TC-10	Behavior Analysis	Transaction data is	Sales and behavior	100%	Passed

	& Sales Insights	processed and sales/behavior insights are displayed	analytical outputs were successfully generated		
TC-11	Product Demand Prediction	Historical sales data is processed and a demand forecast is returned	Demand prediction component successfully generated forecasting output	100%	Passed
TC-12	Smart Order Recommendation	System generates product recommendations where valid rules are available	Recommendation component successfully returned relevant product suggestions	100%	Passed
TC-13	Component Data Integration	Operational data is correctly shared with analytical components	Required operational data was successfully transferred to the relevant components	100%	Passed
TC-14	Analytics Result Integration	Analytical results are returned to the appropriate Smart-Cost interface	Analytical outputs were successfully received and presented by the system	100%	Passed
TC-15	Invalid Request Handling	Invalid or incomplete requests are rejected or handled appropriately	Invalid requests were handled without disrupting the normal system workflow	100%	Passed
TC-16	Insufficient Prediction Data	System handles cases where sufficient forecasting history is unavailable	Appropriate response was produced instead of an unsupported prediction	100%	Passed

TC-17	No Recommendation Rule Available	System handles transactions for which no suitable association rule exists	System continued operation without generating an invalid recommendation	100%	Passed
TC-18	Database Connectivity	Backend can successfully communicate with MongoDB Atlas	Database connection and data operations functioned successfully	100%	Passed
TC-19	Mobile-to-Backend API Communication	Mobile application successfully exchanges data with backend services	Requests and responses were successfully exchanged through the API	100%	Passed
TC-20	End-to-End Smart-Cost Workflow	Transaction flows from POS through storage and analytical components to user-facing outputs	Complete integrated workflow operated successfully during system testing	100%	Passed

A definition of Completion (%) :

Completion (%) = (Number of successful checks ÷ Total checks performed) × 100

Table 30:definition of Completion (%)

Completion	Status
100%	Passed
80–99%	Partially Passed / Minor Issues
Below 80%	Needs Improvement / Failed

Appendix D – Dataset and Experimental Data Description

The integrated Smart-Cost evaluation used an experimental dataset designed to represent operational activities within a small or medium-sized business environment. The dataset covered an observation period of approximately 12 months and contained 15,420 POS transactions, 128 products, and 3,760 cost records. The data were used across the four integrated research components: Business Performance Prediction, Behavior Analysis & Sales Insights, Product Demand Prediction, and Smart Order Recommendation. As the dataset was experimental/simulated, it was primarily used to support system development, integration testing, analytical evaluation, and validation up to the PP2 stage.

Table 31:Dataset and Experimental Data Description

Attribute	Integrated Evaluation Dataset
Observation period	12 months
POS transactions	15,420
Products	128
Cost records	3,760
Data nature	Experimental/simulated
Main analytical uses	Forecasting, performance analysis, sales/behavior analysis, recommendation

D.1 Data Fields

The dataset contained the main operational fields required by the Smart-Cost system and its analytical components. Transaction-related information included a unique transaction identifier and transaction date/time, which allowed individual sales to be distinguished and arranged chronologically. Product-related information included the product identifier, product name, product category, quantity sold, and unit price. These fields supported product-level sales analysis, inventory-related processing, and historical demand analysis.

Financial information included the transaction value and relevant cost or expense information. These values were particularly important for the Business Performance Prediction component because sales and cost information provided the basis for evaluating business performance and profitability-related patterns. Customer-related information was used where such information was available and applicable to the analysis. In addition, products belonging to the same transaction were represented as transaction baskets, enabling the Smart Order Recommendation component to identify relationships among products purchased together.

D.2 Data Preparation

Before the dataset was used by the analytical components, the relevant records were prepared to improve consistency and suitability for analysis. The preparation process included checking required fields, handling missing or invalid values where applicable, maintaining consistent date and time formats, checking for duplicate records, filtering unnecessary information, and organizing the data according to the requirements of each research component. These preprocessing activities helped reduce the possibility of unsuitable records affecting the analytical outputs.

Different forms of preparation were required for the four components. For Product Demand Prediction, historical product sales were organized chronologically and aggregated at the required product and time level to create demand observations for forecasting. For Business Performance Prediction, relevant sales, transaction, cost, and expense information was organized for performance and profitability analysis. For Behavior Analysis & Sales Insights, accumulated transaction information was prepared to support the identification of sales and purchasing patterns. For Smart Order Recommendation, products belonging to the same transaction were grouped into transaction baskets so that product relationships could be analyzed using association-rule measures such as support, confidence, and lift.

D.3 Data Limitations

The main limitation of the integrated dataset is its experimental/simulated nature. Although it provides a controlled basis for developing and evaluating Smart-Cost, it cannot fully reproduce the complexity and variability of actual SME operations. Real-

world sales and demand may be influenced by factors such as seasonal variations, public holidays, promotions, changes in customer preferences, product availability, economic conditions, and unexpected business events. These factors may produce patterns that are different from those represented within an experimental dataset.

The reliability of the analytical components also depends on the quantity, quality, and consistency of the available data. Limited historical information can reduce the reliability of product-demand forecasting, while incomplete or inaccurate sales and cost records can affect business-performance analysis. Similarly, the recommendation component depends on having sufficient transaction-basket patterns to identify meaningful product relationships. Therefore, the results obtained up to PP2 should be interpreted as evidence of the research prototype and integrated analytical approach under the experimental evaluation conditions, rather than as guaranteed real-world performance across all SMEs. Future evaluation using genuine operational data from multiple SMEs over longer periods would provide stronger evidence of the system's generalizability and practical effectiveness.

Appendix E – Extended Component Evaluation Results

E.1 Business Performance Prediction

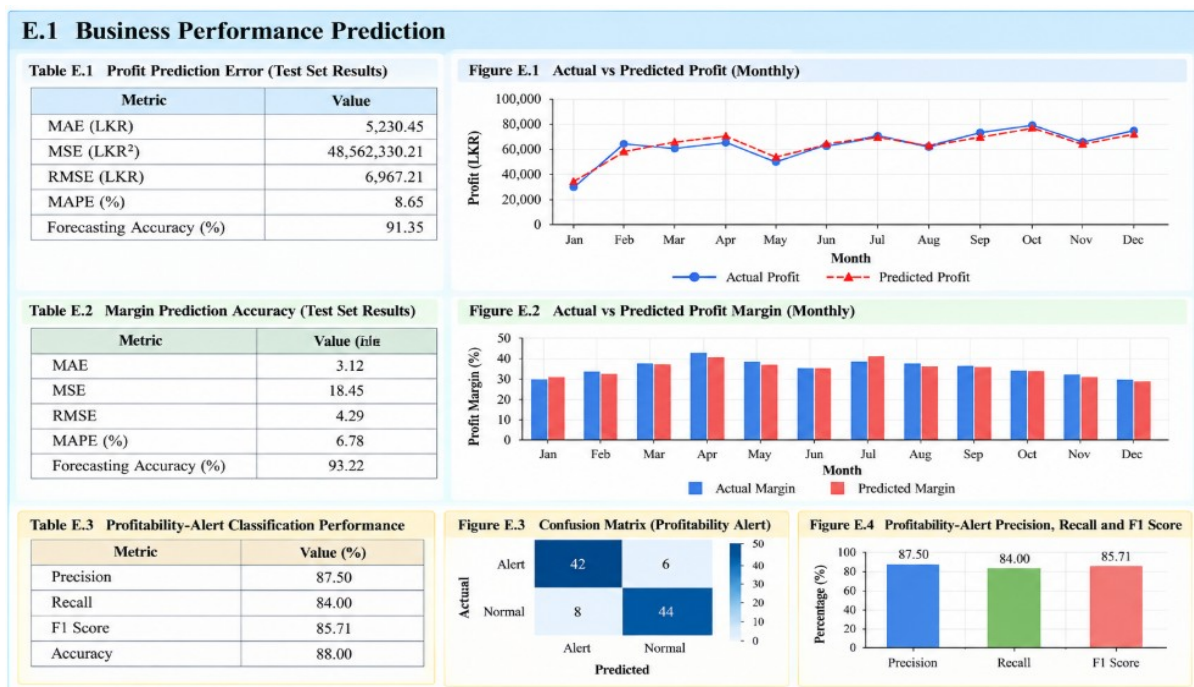


Figure 31: Business Performance Prediction Charts and Details

E.2 Behavior Analysis & Sales Insights

E.2 Behavior Analysis & Sales Insights

Figure E.5 Sales Distribution by Product Category

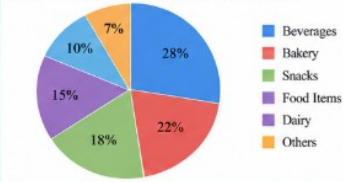


Figure E.6 Top 5 Best-Selling Products

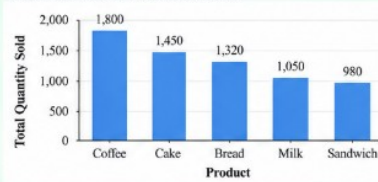


Table E.4 Top 5 Product Performance Analysis

No.	Product	Total Quantity Sold	Total Revenue (LKR)	% of Total Sales
1	Coffee	1,800	270,000	16.5%
2	Cake	1,450	217,500	13.3%
3	Bread	1,320	198,000	12.1%
4	Milk	1,050	157,500	9.6%
5	Sandwich	980	147,000	9.0%

Figure E.7 Transaction Pattern by Time of Day

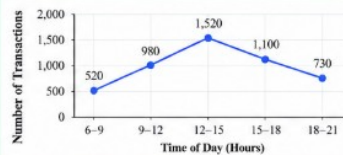


Figure E.8 Sales Trend over Months



Figure E.9 Customer Type Distribution

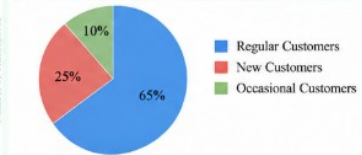


Table E.5 Customer Behavior Insights

Insight	Observation
Most active customer type	Regular customers (65%)
Average items per transaction	2.8 items
Most active time period	12:00 – 15:00
Popular product combination	Coffee and Cake
Repeat purchase rate	58%

Figure E.10 Product Category Contribution to Revenue

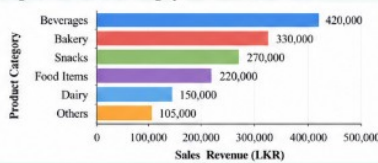


Figure E.11 Product Co-Purchase Heatmap (Top Items)

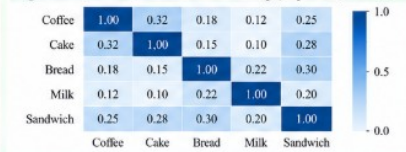


Figure E.12 Dashboard Output – Sales and Customer Insights (Sample)



Figure 32: Behavior Analysis & Sales Insight's Charts and Details

E.3 Product Demand Prediction

E.3 Product Demand Prediction

Table E.6 Forecasting Performance (Example: Bread)

Metric	Value
MAE	2.43
MSE	8.64
RMSE	2.94
MAPE (%)	8.12
Forecasting Accuracy (%)	91.88

Forecasting Accuracy (%) = 100 – MAPE
= 100 – 8.12 = 91.88%

Figure E.13 Actual vs Predicted Demand (Example: Bread - 30 Days)

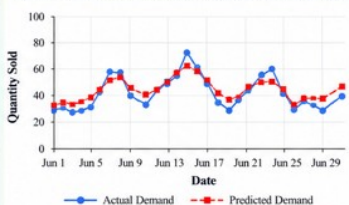


Figure E.14 Actual vs Predicted Demand (Next 7 Days Forecast)

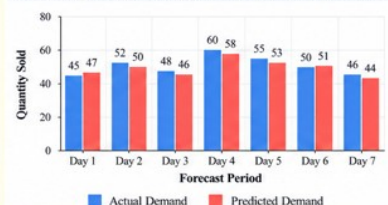


Table E.7 Forecasting Performance for Selected Products

Product	MAE	MSE	RMSE	MAPE (%)	Forecasting Accuracy (%)
Bread	2.43	8.64	2.94	8.12	91.88
Cake	3.12	14.25	3.78	9.56	90.44
Milk	4.05	21.36	4.62	10.23	89.77
Coffee	3.28	16.04	4.01	9.15	90.85
Sandwich	4.50	28.12	5.30	11.68	88.32

Figure E.15 Forecasting Accuracy Comparison

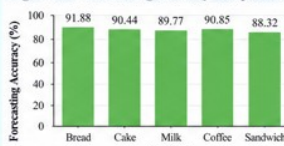


Table E.8 Model Comparison (Bread)

Model	MAE	MSE	RMSE	MAPE (%)	Forecasting Accuracy (%)
Prophet	2.96	11.52	3.39	9.85	90.15
MLP	2.68	9.74	3.12	9.01	90.99
Hybrid (Prophet + MLP)	2.43	8.64	2.94	8.12	91.88

Figure E.16 Actual vs Predicted Demand (Selected Products)

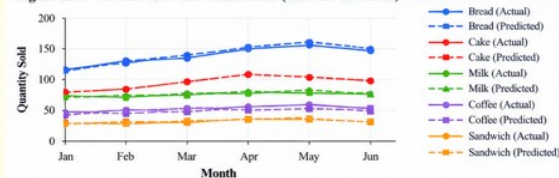


Figure E.17 Model Comparison - Forecasting Accuracy for All Products

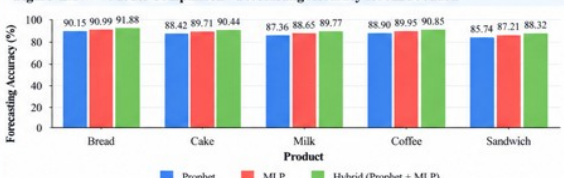


Figure 33: Product Demand Prediction Charts and Details

E.4 Smart Order Recommendation

E.4 Smart Order Recommendation

Table E.9 Top Association Rules (Generated from Transaction Data)

No.	Rule (If → Then)	Support	Confidence (%)	Lift
1	Coffee → Cake	0.28	82.00	2.10
2	Bread → Milk	0.25	76.00	1.95
3	Sandwich → Coffee	0.18	68.00	1.72
4	Cake → Milk	0.15	64.00	1.60
5	Coffee → Sandwich	0.14	61.00	1.55
6	Bread → Butter	0.12	58.00	1.48
7	Tea → Biscuit	0.11	55.00	1.42
8	Burger → Fries	0.10	52.00	1.38
9	Smoothie → Sandwich	0.09	50.00	1.31
10	Milk → Cake	0.08	48.00	1.26

Figure E.18 Top 10 Rules by Confidence

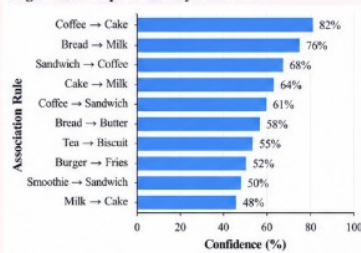


Figure E.19 Support vs Confidence (Top Rules)

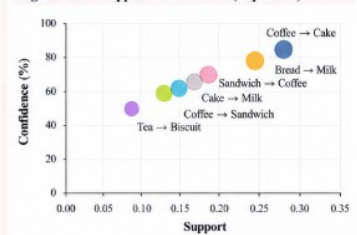


Figure E.20 Lift Values for Top Rules

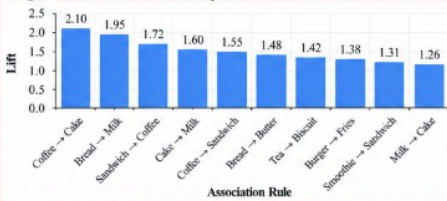


Figure E.21 Distribution of Rule Strength (Support)

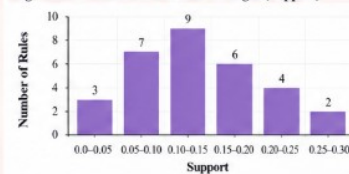


Figure E.22 Distribution of Rule Strength (Confidence)



Figure E.23 Recommendation Examples (POS Interface)



Table E.10 Top Recommended Products for Selected Transaction Baskets

Transaction Basket	Recommended Products	Confidence (%)	Lift
{ Coffee }	Cake, Biscuit, Sandwich	82, 55, 61	2.10, 1.42, 1.55
{ Bread }	Milk, Butter, Jam	76, 58, 47	1.95, 1.48, 1.21
{ Sandwich }	Coffee, Milk	68, 50	1.72, 1.34
{ Cake }	Milk, Coffee	64, 52	1.60, 1.38
{ Tea }	Biscuit	55	1.42
{ Burger }	Fries	52	1.38

Figure 34: Smart Order Recommendation's Charts and Details