International Journal of Science, Technology, Engineering and Mathematics
Volume 4 Issue 4 December 2024
DOI: https://doi.org/10.53378/ijstem.353134

Transforming SME operations with real-time mobile POS and Firebase integration

[1]Farel Naufal Azhari & [2]Sutarman

Abstract

Small and medium enterprises (SMEs) often struggle to efficiently manage sales transactions and inventory due to a lack of affordable, real-time technological solutions. Existing systems typically fail to offer seamless synchronization and proactive inventory management, which are crucial for resource-constrained businesses. This study addresses these challenges by developing a Mobile Point of Sale (mPOS) application integrated with Firebase, aimed at improving operational performance, reducing data inconsistencies, and supporting decision-making through real-time synchronization. The application was developed using a descriptive research approach, React Native for cross-platform compatibility, and Firebase for its real-time database features. Notable innovations include an automated low-stock notification system and an intuitive user interface for both staff and administrators, designed to streamline inventory management. Comprehensive testing and validation confirmed the system's reliability, usability, and data consistency. However, the study was limited by its small-scale implementation and the absence of real-world retail trials. Future research should focus on its scalability, integration with advanced analytics, and application in larger retail settings.

Keywords: *digital retail systems, inventory optimization, low-stock notification, real-time backend solutions, SME efficiency, user interface design*

Article History:

***Received**: October 15, 2024* ***Revised**: November 24, 2024*
***Accepted**: November 27, 2024* ***Published online**: December 4, 2024*

Suggested Citation:
Azhari, F.N. & Sutarman (2024). Transforming SME operations with real-time mobile POS and Firebase integration. *International Journal of Science, Technology, Engineering and Mathematics*, 4(4), 108-135. https://doi.org/10.53378/ijstem.353134

About the authors:
[1]Corresponding author. University of Technology Yogyakarta, Yogyakarta, Indonesia. Email: farelnazhari.22@gmail.com

[2]University of Technology Yogyakarta, Yogyakarta, Indonesia. E-mail: sutarman@uty.ac.id

1. Introduction

In the current dynamic retail environment, managing sales and inventory effectively has become increasingly difficult due to rising customer expectations and fierce competition. Small and medium enterprises (SMEs) contribute approximately 40% to the GDP of developing countries and employ over 70% of the global workforce (World Bank, 2019). However, fewer than 30% of SMEs adopt digital tools for sales and inventory management, resulting in inefficiencies such as stockouts, overstocking, and untimely replenishment (Intal et al., 2022; Joshi et al., 2022). Hence, this research aims to develop a Mobile Point of Sale (mPOS) application integrated with Firebase, enabling real-time data sharing between sales transactions and inventory management. The study's objective is to design an innovative mPOS solution that improves operational performance, reduces data inconsistencies, and supports decision-making by providing timely information to retailers. This system is particularly crucial for SMEs, which often lack the funding and resources available to larger enterprises.

This research builds on existing literature about mobile point-of-sale systems, specifically addressing the gap in real-time data synchronization. Many current mPOS solutions lack essential features for SMEs, such as proactive inventory control and seamless real-time synchronization, both of which are vital for small-scale operations with limited resources. The proposed mPOS application integrates sales management and stock keeping functions, providing clients with up-to-date information at all times. A key feature of this system is an automated alert notification that activates whenever an item's stock drops below three units, addressing a major issue for SMEs by enabling proactive inventory management and reducing the risks of stockouts and overstocking (Ashita et al., 2020; Haidukevych & Doroshenko, 2022).

The rapid digital transformation of the retail sector, coupled with shifts in consumer behavior post-pandemic, further underscores the need for affordable and scalable technological solutions. Conventional POS systems are often inadequate for managing real-time inventory, resulting in inefficiencies such as stockouts and inaccurate data tracking (Lewis et al., 2022). Firebase's robust real-time database capabilities provide a cost-effective solution tailored to the needs of SMEs, enabling them to modernize their operations and adapt to changing market demands. By leveraging Firebase, the mPOS system enhances operational resilience, allowing

businesses to quickly adjust to market fluctuations, ultimately improving customer experience and productivity (Dantu et al., 2021).

This research arises from the observation that many SMEs lack access to sophisticated POS systems capable of real-time data updates. To address these gaps, the study proposes a system that integrates sales and inventory management with automated features like low-stock notifications. The goal is to provide SMEs with actionable insights into sales and inventory, transforming decision-making, eliminating operational inefficiencies, and improving customer satisfaction (Margate et al., 2020). Firebase integration ensures secure, synchronized data sharing across devices, making this system a viable solution for the digital economy (Joshi et al., 2022). This research introduces an innovative system that can help SMEs become more competitive and successful in the rapidly evolving retail landscape.

2. Literature Review

2.1. Point of Sale (mPOS) Application

The Mobile Point of Sale (mPOS) system has revolutionized business transactions, allowing payments to be processed anytime, anywhere, and on any device. This flexibility is especially beneficial for SMEs, which often lack the resources or infrastructure of larger businesses. mPOS systems enable SMEs to serve customers more efficiently, without the limitations of traditional cash registers and dedicated storefronts (Tsai, 2021). Despite the widespread adoption of mPOS, there is a notable gap in research addressing the integration of real-time data synchronization through Firebase, particularly in relation to inventory and sales transaction management for SMEs. This study fills that gap by integrating Firebase to provide real-time synchronization, addressing the specific challenges faced by SMEs in terms of data accuracy and operational efficiency.

As the demand for operational agility and efficient traffic management rises, the adoption of mPOS systems has accelerated, particularly in industries with high customer engagement and fluctuating demand (Lewis et al., 2022). A key feature that sets modern mPOS systems apart is the integration of Artificial Intelligence (AI). This technology enhances transaction completion rates by providing personalized product recommendations and fostering customer loyalty through real-time, data-driven promotions (Tsai, 2021). For SMEs, AI-powered mPOS systems allow businesses to track sales and inventory in real time, improving operational efficiency without the need for expensive infrastructure (Lewis et al.,

2022). As Naik (2023) highlights, AI not only attracts customers but also significantly boosts sales by offering tailored promotions based on customer preferences.

AI integration with mPOS systems also greatly enhances inventory management. Real-time stock tracking enables businesses to align inventory levels with actual demand, preventing stockouts or overstocking. By analyzing historical sales data, mPOS systems help retailers forecast inventory needs more accurately, ensuring products are stocked appropriately without exceeding budgetary constraints (Tandel et al., 2020). Furthermore, the integration of mPOS with Enterprise Resource Planning (ERP) systems enhances cross-functional coordination, enabling businesses to manage sales, inventory, and customer data seamlessly across various channels (Udayaprakash & Sujatha, 2020). This integration provides powerful tools for improving decision-making and strengthening competitive positioning (Melvin et al., 2023).
Consumer satisfaction is a critical factor in the success of mPOS systems. Research within the S-O-R (Stimulus-Organism-Response) framework has shown that system simplicity and ease of use significantly affect user satisfaction and business retention (Chopdar & Balakrishnan, 2020). Security development is also a key area of focus, with cloud computing integration expected to address data protection and privacy concerns (Alhazmi, 2020). As more SMEs adopt mPOS, future developments will likely focus on improving AI performance, system security, and ensuring smooth interactions across integrated platforms (Lewis et al., 2022).

2.2. Realtime Data Synchronization

Real-time data synchronization is a core feature of modern mPOS systems, enabling seamless updates of transaction and inventory data across devices. This capability is crucial for SMEs, where the ability to manage data in real-time can significantly improve operational efficiency and decision-making. By integrating Firebase, a robust backend service, mPOS systems provide accurate, up-to-date information on transactions, stock levels, and customer data, helping businesses respond quickly to changes and maintain optimal inventory levels (Ruiz et al., 2020).

In fast-paced environments like retail, where demand can fluctuate rapidly, real-time data synchronization enables mPOS systems to provide businesses with immediate insights into sales trends and inventory status. This data-driven approach helps SMEs forecast demand more effectively and adjust their operations accordingly (Afifi et al., 2022). Firebase's low-latency synchronization ensures that multiple users and devices can access and update data

simultaneously without compromising performance, which is essential for businesses requiring fast and reliable data (Pandit et al., 2023).

Additionally, Firebase's offline capabilities ensure that businesses continue to function smoothly even in areas with unreliable internet connectivity. Data is stored locally on devices and synced to the cloud once the connection is restored, ensuring uninterrupted service and data integrity (Kimm & Flynn, 2022). This is particularly beneficial for SMEs operating in regions with unstable internet access, enabling them to maintain consistent operations. By integrating real-time synchronization capabilities into mPOS systems, SMEs can efficiently manage inventory, sales, and customer data, minimizing errors and ensuring that businesses can make informed decisions based on accurate, up-to-date information. Real-time synchronization enables quicker responses to customer needs and market changes, which is essential for staying competitive in the fast-moving retail sector (Wiesner & Kovácsházy, 2023; Chowdhury, 2021; Hasan et al., 2020; Margate et al., 2020; Reddy Lakkireddy et al., 2022).

2.3. Firebase as a Backend Service

Firebase, a backend-as-a-service (BaaS) platform owned by Google, is widely used for mobile and web applications due to its comprehensive feature set, including a real-time database, user authentication, cloud storage, and analytics. Firebase's ease of use and scalability make it an ideal solution for SMEs seeking an affordable and efficient mPOS system. By enabling seamless real-time data synchronization, Firebase simplifies backend infrastructure for SMEs, allowing them to focus on developing user-friendly applications (Pandit et al., 2023).

The real-time capabilities of Firebase are particularly beneficial for mPOS systems, where fast data updates are essential. Whether for processing transactions or tracking inventory, Firebase ensures that data is synchronized instantly across all devices, providing businesses with real-time insights into their operations (Ashita et al., 2020). Additionally, Firebase's dynamic scaling capabilities ensure that SMEs can manage varying user loads without performance issues, making it ideal for businesses with fluctuating demand or seasonal spikes (Ahmed, 2022).

One of Firebase's key advantages is its comprehensive security features, including multi-factor authentication and OAuth, which help protect sensitive business data. However,

SMEs must follow best practices when configuring Firebase to avoid potential security vulnerabilities (Demissie & Ranise, 2021). Firebase's Firestore security rules and Firebase Authentication provide necessary safeguards to protect customer and business data, ensuring secure, real-time interactions (Vaz et al., 2021).

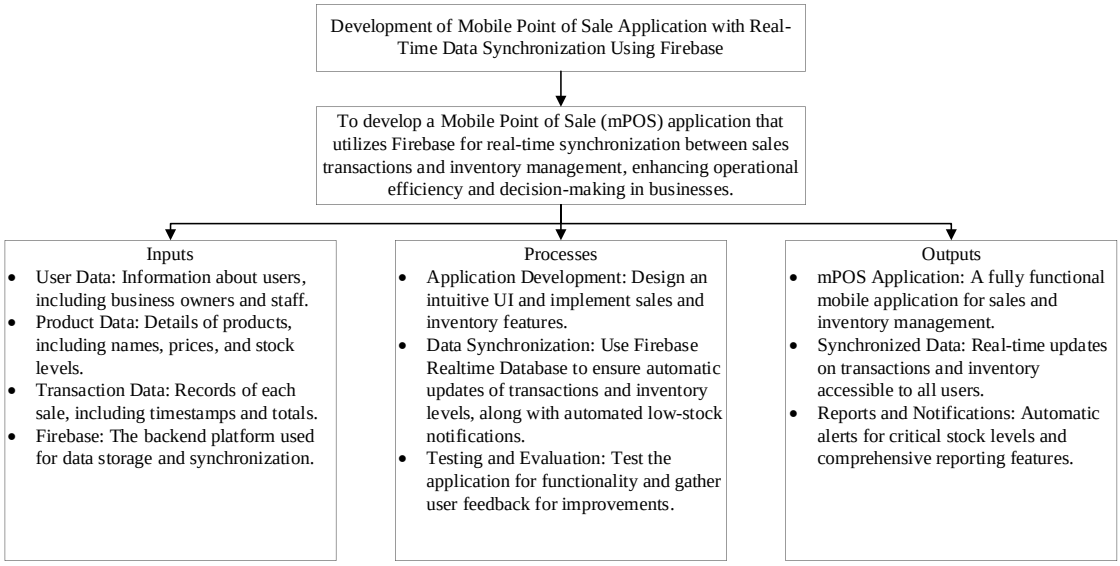
Firebase's scalability is particularly valuable for SMEs that operate across different regions and require cross-platform solutions. With Firebase, SMEs can manage data seamlessly across mobile and web platforms, ensuring that customer interactions and transactions are synchronized in real-time (Ahmed, 2022). This ability to scale and handle large amounts of data without compromising performance makes Firebase an ideal backend solution for mPOS applications supporting growing businesses (Pandit et al., 2023).

2.4. Research Framework

The conceptual framework in figure 1 illustrates the integration of Firebase into the Mobile Point of Sale (mPOS) system, emphasizing the relationships between key components such as user data, product data, and transaction data. These components drive core processes like real-time data synchronization and application development, forming the foundation of the mPOS system. By leveraging Firebase, mPOS systems ensure continuous data updates and maintain synchronization across multiple devices, which is crucial for SMEs managing fluctuating demand and ensuring accurate, up-to-date inventory and transaction records.

Figure 1

Conceptual framework of the study



Real-time synchronization is essential for accurate transaction processing and inventory management, both of which are critical to SME operations. The system's ability to provide live data enables businesses to make timely decisions, improving efficiency and responsiveness. By integrating Firebase, the mPOS system addresses common operational challenges faced by SMEs, including inventory mismanagement, delayed transaction processing, and the lack of actionable real-time insights.

This framework highlights the pivotal role of real-time data synchronization in empowering SMEs to manage their operations effectively, especially in dynamic retail environments characterized by fluctuating demand and fast-moving inventories. The integration of Firebase within mPOS systems offers a robust solution to these challenges, ensuring accurate, real-time data for improved decision-making and operational efficiency. This research demonstrates that incorporating Firebase into mPOS applications can help SMEs remain competitive in the increasingly digital marketplace. By transitioning from traditional point-of-sale and inventory control processes to real-time management, SMEs can enhance customer relations, optimize business operations, and boost overall productivity.

3. Methodology

3.1. Research Design

The research design provides a structured framework for data collection, measurement, and analysis, ensuring the research questions are systematically addressed. It serves as a roadmap, guiding the efficient execution of the study. This research adopts a descriptive methodology, which is particularly well-suited for documenting the development process, application features, and the implementation of real-time data synchronization. The descriptive approach enables a detailed recording of technical attributes, development challenges, and the solutions applied—specifically in the context of small and medium-sized enterprises (SMEs), which often face resource limitations that preclude the use of more sophisticated systems. This method allows for an in-depth examination of the critical factors influencing the success or failure of mPOS applications integrated with Firebase in practical SME scenarios. By systematically capturing these details, the study highlights the application’s functionality and real-world implications for SMEs.

Previous studies have demonstrated Firebase’s efficacy in synchronizing real-time data for mobile app development, particularly in data modeling, conflict resolution, and

performance enhancement (Pandit et al., 2023). Despite these advancements, a notable research gap exists regarding the specific needs of SMEs, particularly in inventory management and sales transaction systems. Current solutions often cater to larger enterprises or provide generalized mPOS functionalities, lacking the real-time synchronization and scalability needed for resource-constrained businesses.

By employing a descriptive approach, this research identifies key challenges faced during development and provides detailed solutions leveraging Firebase's features. It offers valuable insights into how Firebase improves operational efficiency, minimizes data inconsistencies, and supports real-time decision-making—essential capabilities for SMEs striving to remain competitive in the digital economy. This methodology ensures a thorough understanding of both technical and operational aspects, bridging the gap between existing solutions and the unique needs of SMEs.

3.2. System Analysis

System analysis involves evaluating the requirements, features, and user interactions of the POS application, with a particular focus on addressing the evolving needs of SMEs. The primary objective is to define the system's structure and ensure seamless interrelations between control flows and key actors. This is essential for developing a secure, intuitive, and effective mPOS application that supports the operations of both employees and administrators.

Given the dynamic nature of the retail environment, a comprehensive analysis is vital to meet the operational demands of staff (cashiers) and admins/store owners. This ensures that the system is not only functionally robust but also adaptable to the evolving requirements of the retail industry.

To enhance efficiency, the application incorporates role-specific functionalities. Admins are tasked with inventory management, transaction monitoring, and responding to stock alerts, while cashiers focus on processing sales, viewing product details, and maintaining sales records. By leveraging Firebase's real-time synchronization capabilities, the system ensures all users have access to the latest data, minimizing delays and reducing errors. For instance, when a cashier identifies low stock levels, the system immediately notifies the admin, facilitating prompt restocking and ensuring uninterrupted operations.

This structured approach to role management highlights both usability and security. Each user is granted access only to functionalities pertinent to their role, reducing complexity

while maintaining operational clarity. By aligning the system's features with distinct user responsibilities, the mPOS application effectively addresses the unique operational challenges of SMEs, fostering efficient workflows and secure interactions.

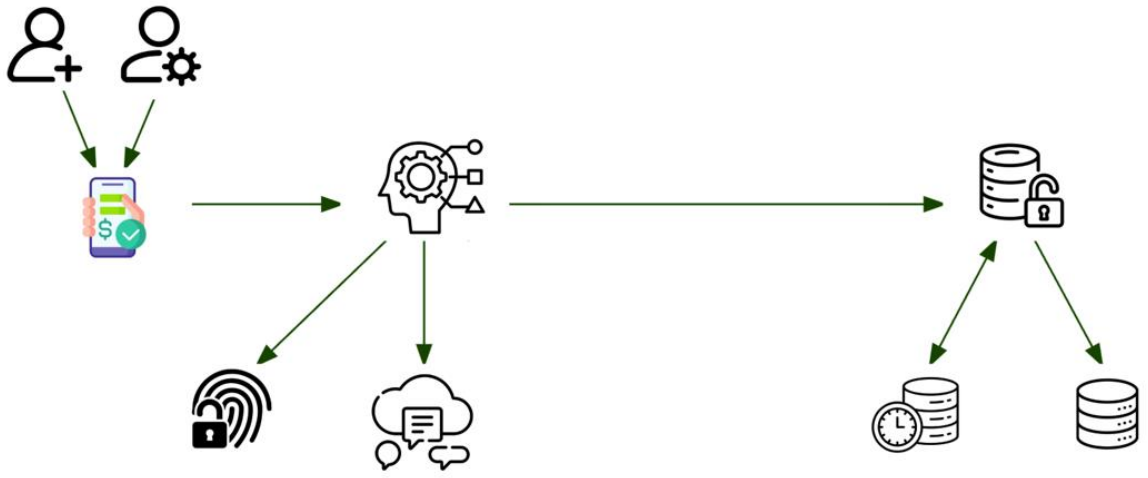
3.3. System Architecture

The mPOS application architecture is designed to support operational scalability, real-time data synchronization, and efficient system operation. Figure 2 presents a high-level diagram that outlines all the key components and their interconnections within the system. The architecture has been designed in a modular fashion, with distinct layers handling specific responsibilities to ensure flexibility, performance, and security.

At the core of the architecture is the User Interface (UI) layer, which serves as the main interface point for both the admin and cashier roles. This layer communicates with the Business Logic layer, which provides the application's core functionalities, such as sales transactions, inventory management, and user authentication. These components interact seamlessly with the Data Management layer, where Firebase manages real-time synchronization, local storage, and cloud services.

Figure 2

System architecture diagram of the mobile POS application



The system's modular design ensures that each component has a clear and focused responsibility, facilitating easier management and maintenance. The authentication services in the architecture ensure secure user logins, while cloud messaging (via Firebase Cloud

Messaging FCM) is used for real-time alerts and notifications—such as when stock levels are running low. This feature is essential for inventory management, as it keeps both the admin and cashier informed and responsive to stock changes, ensuring smooth operations with minimal downtime.

By integrating Firebase's real-time database along with local storage capabilities, the system guarantees that data is always synchronized across devices. This is especially important for businesses with multiple users who need access to the most up-to-date information at all times. Firebase ensures low-latency updates across devices, enabling immediate updates for inventory changes and transaction records, which is crucial for small businesses that need to make quick, data-driven decisions.

The architecture also emphasizes scalability, ensuring that the system can accommodate future growth. Whether the business expands in terms of users or data volume, Firebase's cloud-based infrastructure can scale dynamically without compromising performance. Furthermore, the architecture is designed with security in mind. Firebase's built-in security features, such as Firestore security rules and Firebase Authentication, ensure that only authorized users can access sensitive data, which is essential for maintaining data integrity and privacy.

This modular architecture, combined with real-time synchronization and scalable cloud infrastructure, enables us to build a responsive, secure, and reliable mPOS system capable of handling the demands of modern retail operations, particularly for SMEs that need an affordable yet robust solution.

3.4. Development Environment and Tools

The mPOS application development involves the integration of various hardware and software tools to ensure a smooth, cross-platform experience. The development environment and tools used in this project were carefully selected to support real-time data synchronization, seamless performance across devices, and the flexibility needed for SME-specific use cases. The key hardware and software components used for the development and testing of the mPOS system are summarized in tables 1 and 2.

Table 1

Hardware specifications

Device Type	Model	Specifications	Function
Development Computer	MacBook Pro 2020	M1 chip, 8GB RAM, 256GB SSD	Main development computer
Android Test Device (Phone)	Google Pixel 7 Pro	Android 13, Tensor G2	Testing application on Android smartphone
Android Test Device (Tablet)	Google Pixel Tablet	Android 13, Tensor G2	Testing application on Android tablet

Table 2

Software tools and technologies

Category	Tool/Technology	Version	Function
Programming Language	JavaScript/TypeScript	ES2024 / 5.2	Main development language
Main Framework	React Native (Expo)	0.72.0	Framework for cross-platform mobile app development
Development Platform	Expo Go	49.0.0	React Native development and testing platform
Backend Service	Firebase	10.5.0	Backend and realtime database service
IDE	Visual Studio Code	1.80.0	Primary Integrated Development Environment
Version Control	Git	2.42.0	Version control system
	GitHub	-	Repository hosting platform
Mobile Development	Expo CLI	6.3.0	Command Line Interface for Expo projects
	Expo DevTools	49.0.0	Web-based development tools for Expo
Testing	Jest	29.6.0	Testing framework
	Expo Jest	49.0.0	Jest configuration for Expo projects
Debugging	Expo Debug Tools	49.0.0	Built-in debugging tools for Expo
State Management	Redux Toolkit	2.0.0	Application state management
Code Quality	ESLint	8.45.0	Linter for maintaining code quality
	Prettier	3.0.0	Code formatter
Build & Deployment	Expo EAS	5.0.0	Expo build and submit services

The tables summarize the tools and technologies that were utilized to build the mPOS application, emphasizing both hardware and software selections crucial for cross-platform functionality and real-time synchronization.

React Native (with Expo) was selected as the main framework for cross-platform mobile app development due to its efficiency in creating apps that run on both Android and iOS devices, ensuring scalability and wide accessibility for SMEs. Firebase was chosen as the backend service for real-time data synchronization because of its robust database capabilities and real-time cloud messaging system, which are critical for seamless inventory management and real-time transaction processing across devices. This selection aligns with the study's focus on using Firebase to enhance operational efficiency, particularly in an SME context where cost-effective, scalable solutions are necessary. Furthermore, Expo DevTools and Jest were used to streamline the development and testing processes. Expo DevTools enables quick iterations and testing during development, making the integration of Firebase easier and more efficient. Jest is a testing framework that allows for unit and integration testing, ensuring that the application’s real-time synchronization and data integrity are maintained across different devices and user interactions.

The tools listed in table 2 ensure that the development of the mPOS application meets the requirements of modern mobile applications, providing high performance, scalability, security, and cross-platform compatibility, making it an ideal solution for SMEs looking to adopt mobile-based point-of-sale systems with real-time data synchronization.

3.5. Application Development Process

The mPOS application development followed a structured and iterative process based on the Agile Development methodology to ensure that it meets the required functionality while remaining user-friendly. This process was designed to handle the complexity of modern POS systems and to remain flexible to changing requirements, methodologies, and technologies. Given the evolving needs of SMEs, the Agile methodology was chosen for its ability to accommodate iterative development and continuous feedback from users.

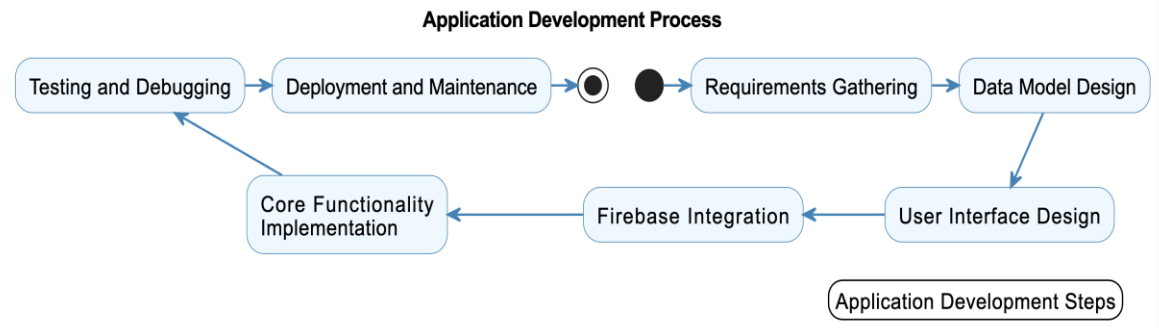
The process began with the planning phase, during which functional and non-functional requirements, such as real-time data synchronization and user-friendly interfaces, were identified through consultations with potential users, including administrators and cashiers. In the design phase, UI/UX elements were developed using Figma, and a robust data model was

constructed to support Firebase for real-time data management. During the implementation phase, the application was developed incrementally using React Native, with each sprint focusing on specific features such as transaction processing and low-stock notifications.

Throughout the development cycle, rigorous testing—including unit testing, integration testing, and performance testing—was conducted using Jest and Expo DevTools to ensure system reliability, data accuracy, and responsiveness across devices. Finally, in the deployment phase, the application was launched in a production environment, followed by continuous maintenance and updates based on user feedback and evolving operational needs. This iterative and user-centered approach ensured that the mPOS application is both robust and adaptable, meeting the practical requirements of SMEs while allowing for future enhancements.

Figure 3

System architecture diagram of the mobile POS application



The development of the mPOS application follows a structured, iterative process that begins with requirement analysis, where the needs of users (staff and administrators) are identified. The design phase focuses on creating a seamless user experience, with a data model that guides the user interface (UI) design. Firebase is integrated early in the process to enable real-time data synchronization across multiple devices. After the coding phase, extensive testing (unit, integration, and performance testing) ensures the application functions as intended, particularly regarding data accuracy and synchronization. The final phase involves deployment, followed by continuous maintenance to address evolving requirements. This flexible development methodology ensures the mPOS application meets the operational needs of SMEs, enhancing sales and inventory management through real-time synchronization.

3.6. Testing and Validation Procedures

The mPOS application underwent a rigorous and multidimensional testing and validation process to ensure its reliability, efficiency, and user-friendliness. This comprehensive approach was essential for verifying the system's readiness for deployment in a retail environment, where performance, accuracy, and real-time synchronization are critical. The testing process encompassed various phases, including unit testing, integration testing, performance testing, and user acceptance testing (UAT), to guarantee seamless functionality and robustness across all components.

To evaluate the system's usability and effectiveness, purposive sampling was employed to select participants for the UAT phase. This method focused on two primary user groups: store owners (admins) and cashiers, who represent the key end-users of the application. Respondents were chosen based on their familiarity with existing POS systems and their operational roles in SMEs. This selection ensured that the feedback collected was both relevant and actionable, providing insights into the system's usability, reliability, and alignment with the practical needs of SMEs.

Table 3

Testing and validation phases for mobile POS application

No.	Testing Phase	Description	Key Aspects Tested
1	Unit Testing	- Testing individual React Native components - Validation of core functions	- Item addition - Stock updates - Transaction total calculation
2	Integration Testing	- Testing front-end integration with Firebase backend - Verification of real-time data sync	- React Native-Firebase connection - Data synchronization accuracy
3	Performance Testing	- Evaluation of data sync speed - Multi-user load testing	- Performance under various network conditions - System scalability
4	User Interface Testing	- UI responsiveness testing - User flow validation	- UI consistency across devices - Ease of use
5	Real-time Notification Testing	- Verification of critical stock notification accuracy - Testing notification scenarios	- Timeliness of notifications - Accuracy of critical stock info
6	Data Consistency Testing	- Validation of app-Firebase data consistency - Testing data conflict resolution	- Data integrity - Handling of synchronization conflicts
7	User Acceptance Testing (UAT)	- Testing by end-users - Collection and implementation of feedback	- Alignment with user needs - Overall user experience

Table 3 summarizes the testing stages conducted throughout the development cycle. Each stage was meticulously designed to identify potential issues, validate system quality, and optimize performance prior to deployment.

Unit testing verified individual components, such as item addition, stock updates, and transaction total calculations. Integration testing validated the connection between the React Native front-end and Firebase backend, focusing on real-time data synchronization. Performance testing evaluated the system's scalability under various network conditions, ensuring reliable performance for SMEs with dynamic usage patterns.

User Interface (UI) testing assessed responsiveness and user flow across multiple devices to ensure ease of use for SME employees. Real-time notification testing confirmed the timeliness and accuracy of inventory alerts, such as low-stock warnings. Data consistency testing ensured that data synchronization between the mobile app and Firebase database remained intact, addressing any potential conflicts or discrepancies.

Finally, User Acceptance Testing (UAT) involved direct feedback from store owners and staff. Respondents evaluated the system based on their daily operational needs, providing critical insights into its functionality and ease of use. Their feedback informed iterative refinements, ensuring that the system met real-world requirements effectively.

By implementing this robust testing methodology, the mPOS application adheres to the highest standards of performance, usability, and reliability. This iterative approach guarantees that the system is well-suited for SMEs operating in dynamic retail environments, providing a scalable and efficient solution for real-time sales and inventory management.

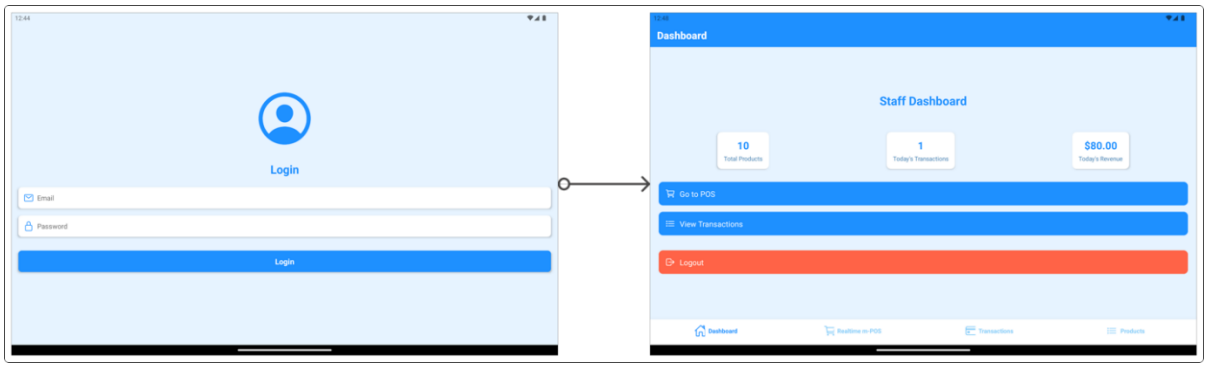
4. Findings and Discussion

4.1. Mobile Point of Sale (mPOS) Application (Staff / Cashier)

Applications designed for use by staff or cashiers, such as mPOS applications, serve as a critical interface in managing daily retail operations. The user interface of these applications simplifies and speeds up the transaction process, allowing staff to focus more on customer service rather than transaction administration. This simple yet effective system design provides efficiencies in managing retail tasks, while reducing the time wasted that typically occurs when staff search for needed devices or information.

Figure 4

Login and dashboard screens of mobile POS application for staff/cashier



In figure 4, two main windows of the mPOS application are shown: the first window is for login, and the second window is for the staff operational interface. The login page is designed in a minimalist manner, consisting of only email and password fields, indicating that access is restricted by user authentication. The staff operational window contains a lot of important information, including the total number of products, activities performed throughout the day, and the total amount of sales. In addition, the application provides options to access key activities such as “Login to POS”, “View Transactions”, and “Log Out”. This easy and functional user interface design is perfect for portable devices commonly used in retail stores, as it minimizes wasted time and ensures that every activity is carried out quickly.

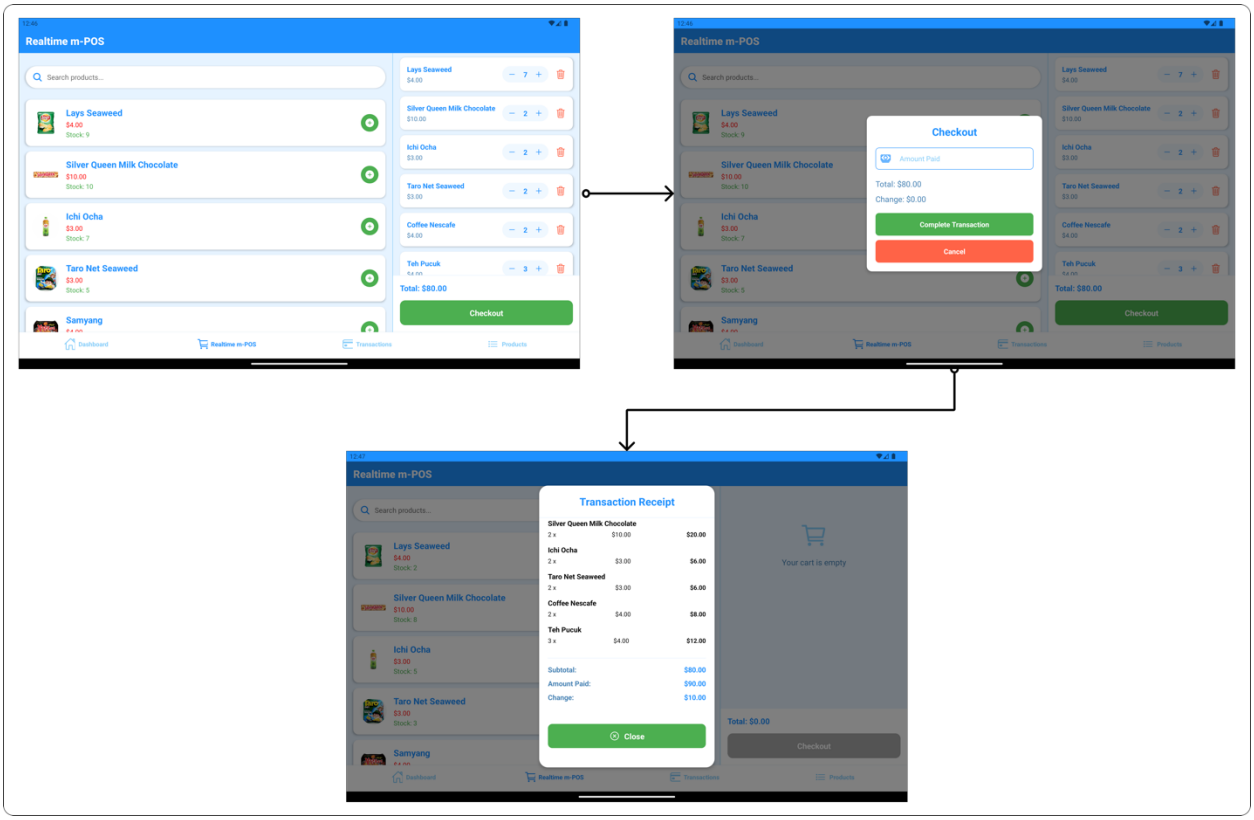
Process transaction module. The transaction processing module in this POS system aims to facilitate effective sales within the retail store and is crucial to the sales process itself. This module streamlines the entire sales sequence, from item selection to payment and receipt printing. With a focus on ease of use and user interaction, figure 5 shows how this application facilitates transactions step by step. The design of this module allows sales staff to complete transactions faster and more accurately, reducing the chances of errors. In addition, the system is equipped with a real-time stock update tool that ensures that stock figures are updated as soon as a transaction is completed.

The transaction process flow is depicted in figure 5, with special emphasis on the three main screens found in a POS application. The first screen serves as the product selection interface, where the staff can add products to the shopping cart. A list of products is displayed with the name, price, quantity adjustment options, and running total below it. The second screen, or “payment screen,” appears after the “checkout” button is pressed. Here, the cashier

can view the selected items and approve the payment after making any necessary changes. The final screen displays the transaction information in a receipt format, showing the number of items purchased, the price per item, and the total cost, which is not only used for record keeping but can also be given to the customer as proof of payment.

Figure 5

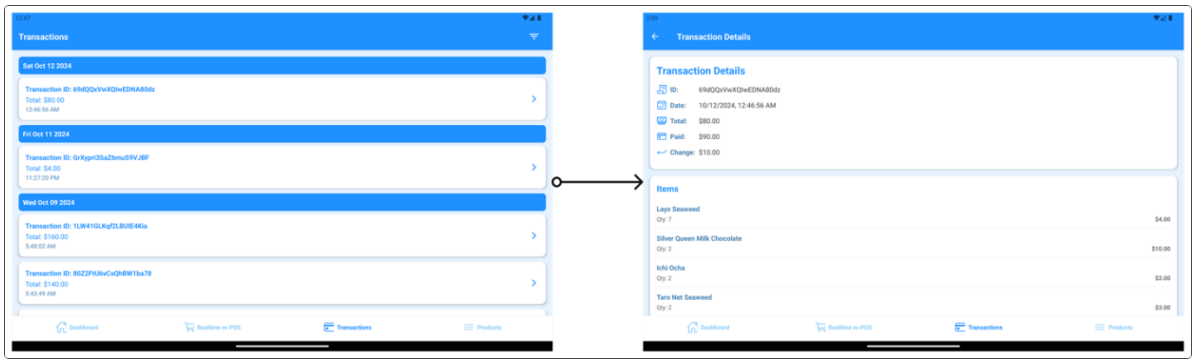
Transaction process workflow screen in mobile POS application



Transaction history module. The transaction history module is a key part of this POS application, providing staff and management with a comprehensive view of past sales activity. Figure 6 illustrates the interface and functionality of this module. It shows the two main screens of the transaction history module: the transaction list and the transaction detail view. The left screen displays transactions in chronological order with clear date separators, showing the transaction ID and total amount for quick reference. The right screen provides detailed information about a particular transaction, including the transaction ID, date, time, total amount, payment details, and a list of items purchased along with their respective quantities and prices.

Figure 6

History transaction list screen and transaction detail screen in mobile POS application

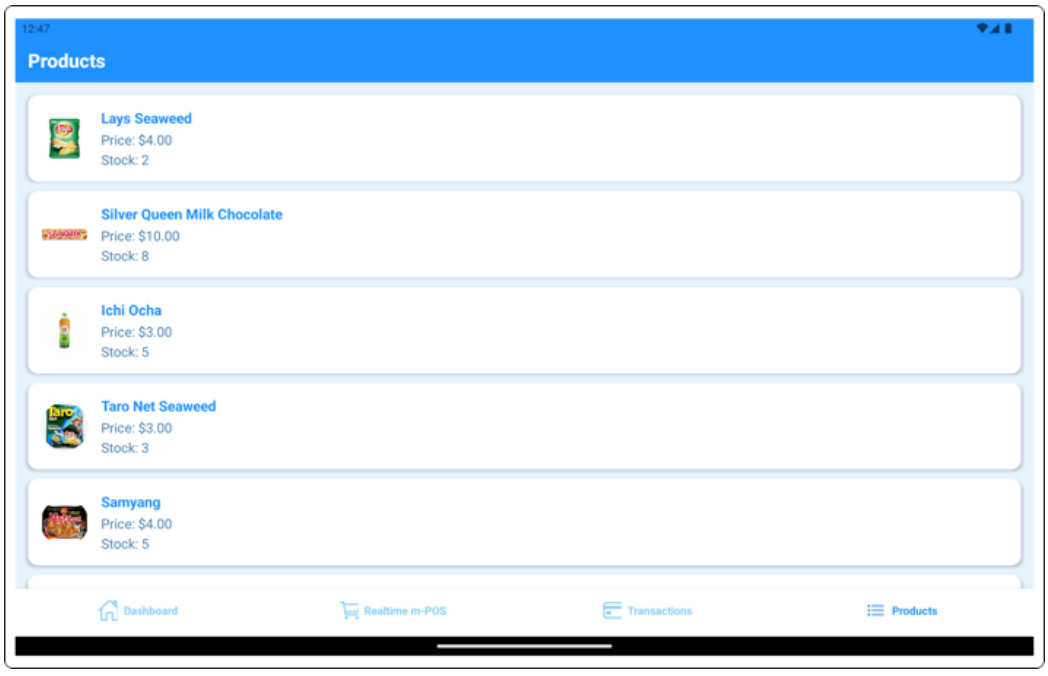


This design allows users to easily navigate and review past transactions, supporting operational tasks such as handling customer inquiries, sales reconciliation, and sales pattern analysis. The user-friendly interface and comprehensive functionality enhance daily operations and long-term business management, emphasizing clarity and efficiency.

Products module. The product module is a key component of this POS application, providing a comprehensive view of the available inventory. Figure 7 illustrates the interface of this module, showing its design and functionality.

Figure 7

Product screens in mobile POS application



The product screen in the POS application displays a complete list of available items, featuring clear product images, names, prices, and stock levels. This design uses individual product cards for easy visual scanning and quick identification. This intuitive interface supports smooth retail operations by allowing for easy product search and inventory tracking, which is critical to maintaining optimal stock and identifying items that need to be replenished.

4.2. Mobile Point of Sale (mPOS) Application (Admin)

The mPOS application for owners or administrators provides a comprehensive management interface to oversee the entire retail operation. This feature is essential to ensure efficient and real-time monitoring of inventory, transactions, and business metrics. Figure 8 illustrates a lock screen designed specifically for this advanced user.

Figure 8

Login and dashboard screens of mobile POS application for admin

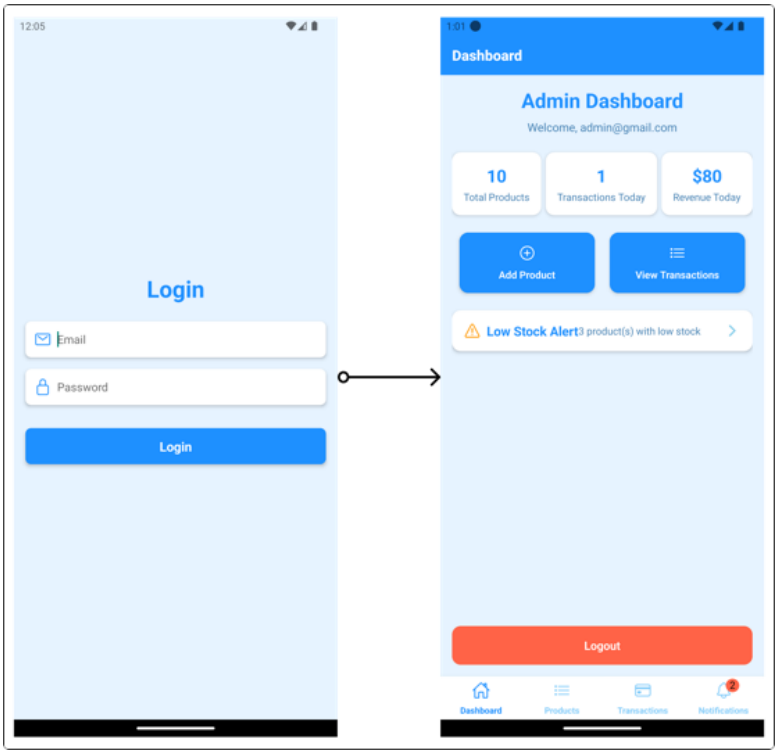


Figure 8 shows the admin/owner interface of the Mobile POS application. The left screen displays a secure login with email and password fields, ensuring access protected by authentication. The right screen displays the admin dashboard that includes key business metrics such as total sales, inventory levels, and recent transactions. The dashboard also provides quick access to reports, inventory management, and system settings. A notification

feature alerts the admin when stock levels drop below three items, allowing quick action to avoid shortages. The interface combines extensive functionality with intuitive navigation, keeping the owner informed of important updates in real time.

Process notification module. The notification process module in the POS application plays a vital role in ensuring that administrators are always updated on critical inventory changes in real-time. This feature supports quick and accurate restock decisions, ensuring that retail operations run smoothly. Figure 9 illustrates the notification process that occurs during a transaction.

Figure 9
Real-time low stock notification process in mobile POS application

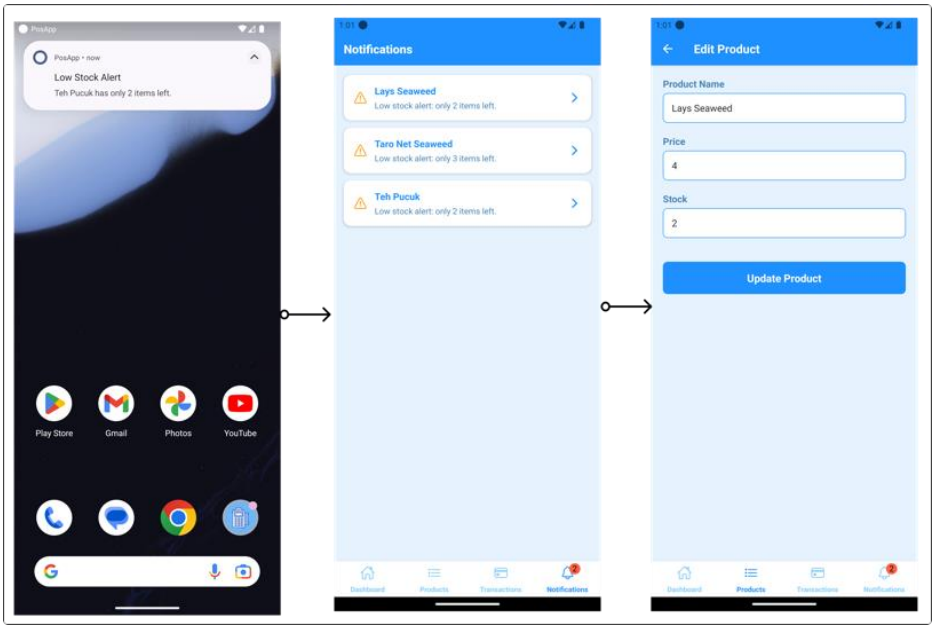


Figure 9 shows the flow of information from the point of sale to the admin notification system. As staff or cashiers process transactions, the system automatically monitors the stock levels of products sold. If a transaction causes a product's stock to drop below a critical threshold (e.g., 3 units), the application immediately triggers a notification to the admin. The notification includes important information such as the product name, current stock level, and recent sales velocity. This information allows the admin to take quick action, such as reordering items, to prevent stockouts.

Manage products module. The product management module gives administrators full control over their product inventory. With a user-friendly interface, this module allows administrators to manage product data efficiently, including adding new products, updating

product information, and deleting products that are no longer available. Figure 10 illustrates the main interface of this module.

Figure 10

Manage product screens in mobile POS application

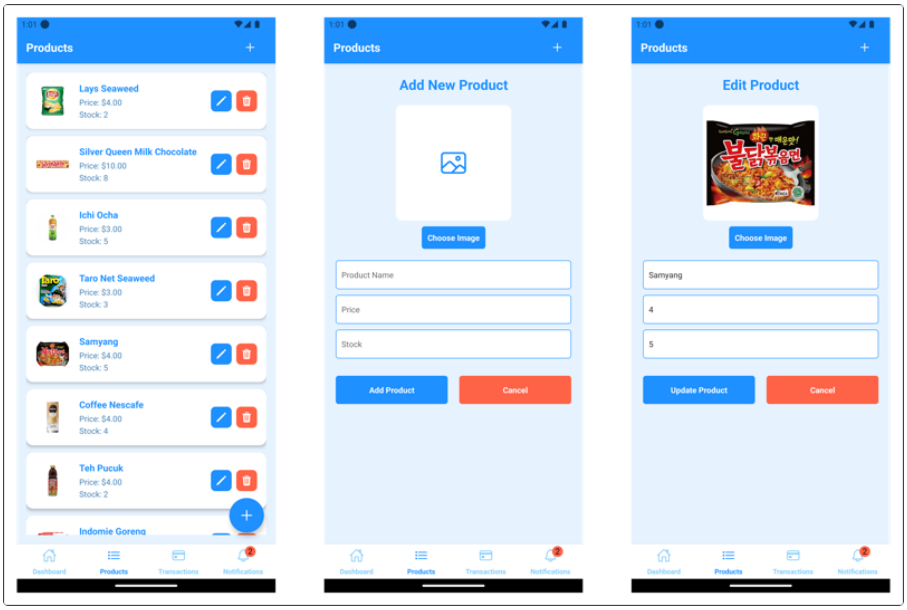


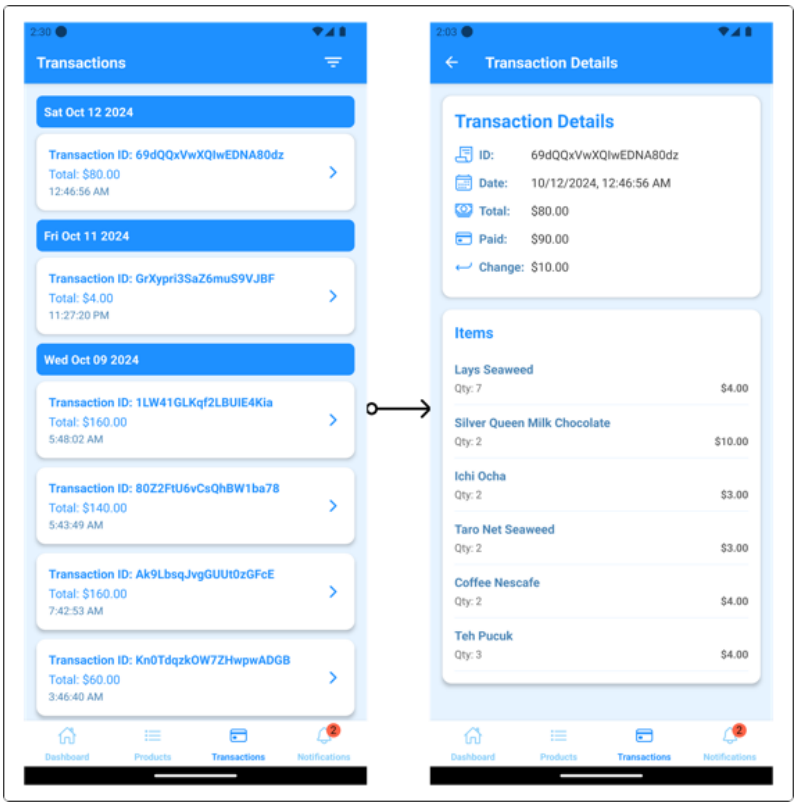
Figure 10 shows the three main screens of the product management module: the product list, the add new product interface, and the product editing interface. The product list screen displays all inventory items with images, names, prices, and stock levels, allowing admins to easily edit or delete product data. The add new product interface includes fields for entering name, price, and initial stock, as well as an option to upload images. The product editing interface allows admins to update product information, including images, prices, and stock quantities. With this intuitive design, the module supports efficient product management, helping administrators keep inventory accurate and up-to-date.

History transaction module. The transaction history module provides a comprehensive view of past sales activity, which is essential for financial tracking and business analysis. This module is designed to make it easier for admins to review past transactions and gain insights into customer purchasing patterns. Figure 11 illustrates the main interface of this module.

Figure 11 shows the two main screens of the transaction history module: the transaction list view and the transaction detail view. The transaction list view shows a chronological record of all transactions, complete with key details such as date, time, and total amount, allowing for quick identification and access to more detailed information. The detail view provides a complete breakdown of the selected transaction, including items purchased, price, quantity,

total amount, payment method, and discounts applied. This module is designed for easy navigation, allowing admins to review sales history, analyze purchasing patterns, and handle customer inquiries with high efficiency. With real-time synchronization, transaction data is always accurate and up-to-date, ensuring reliable financial reporting and decision-making.

Figure 11
transaction list and detailed transaction view screens in mobile POS application



4.3. Validation and Testing Results

Testing and validation are essential steps to ensure the reliability and efficiency of the mPOS Application. This process includes various testing phases to evaluate performance, data synchronization, interface responsiveness, and system stability under various operational conditions. Table 4 summarizes the test results, interpreted using the following 5-point scale: This table also provides average scores for each aspect of the test, indicating the application's success in meeting the expected quality standards.

Table 4

Testing and validation phases for mobile POS application

No.	Testing Phase	Main Results	Interpretation	Key Observations	Primary Actions
1	Unit Testing	4.8	Excellent	CRUD functions working as expected	Optimize CRUD performance
2	Integration Testing	4.8	Excellent	Minor synchronization delays	Implement retry mechanism
3	Performance Testing	4.6	Excellent	Average response 1.2 seconds	Refine Firebase queries
4	UI Testing	4.9	Excellent	UI responsive across of tested device	Continue UI optimization for edge cases
5	Notification Testing	4.7	Excellent	Notifications working well	Continue monitoring performance
6	Data Consistency Testing	4.6	Excellent	Rare conflicts, mostly auto-resolved	Improve conflict resolution algorithm
7	User Acceptance Testing	4.5	Excellent	Users request inventory report feature	Plan inventory report development
Overall Average		**4.7**	**Excellent**		

Legend: 1 (1.00-1.79): Poor, 2 (1.80-2.59): Fair, 3 (2.60-3.39): Satisfactory, 4 (3.40-4.19): Very Good, 5 (4.20-5.00): Excellent

The application achieved a high overall average score of 4.7, classified as excellent. This reflects its strong performance across various dimensions. UI responsiveness was rated 4.9, highlighting its effectiveness in providing a seamless user experience. The unit testing and integration testing also scored 4.8, indicating that core functionalities and real-time synchronization with Firebase are operating effectively with minimal delays. User acceptance testing (UAT) scored 4.5, showing that the application aligns well with user expectations, although users have suggested further development in reporting features.

The high ratings in testing also align with key principles outlined in the ISO/IEC 25010:2011 standard, specifically regarding functional suitability, usability, and reliability. The real-time synchronization and user-friendly interface of the system meet the standard's criteria for these qualities. For example, the application’s real-time stock notifications and data

consistency reflect its compliance with ISO standards, ensuring operational efficiency and high customer satisfaction. Further, the performance testing and integration assessments underline the system's ability to scale and perform under varying conditions, validating its robustness and resilience, as recommended in ISO/IEC 25010:2011.

These results support previous studies on Firebase's effectiveness in providing scalable and reliable back-end solutions for SMEs (Pandit et al., 2023), emphasizing its role in improving operational performance. The positive user feedback regarding UI responsiveness and notification reliability aligns with findings from Ashita et al. (2020), who emphasized that mPOS systems significantly enhance customer satisfaction and inventory control.

4.4. Potential Impact and Effectiveness

The mPOS application significantly enhances retail operations for SMEs by addressing common challenges such as inventory discrepancies and decision-making inefficiencies through real-time data synchronization (Lewis et al., 2022). A key feature, real-time stock notifications, improves customer satisfaction by ensuring timely restocking and reducing sales losses (Ashita et al., 2020).

Powered by Firebase, the application offers scalability and cost efficiency, aligning with prior studies highlighting Firebase's suitability for retail solutions (Pandit et al., 2023). Its integrated features, including inventory management, transaction processing, and notifications, enhance business efficiency and customer satisfaction. Additionally, by meeting ISO/IEC 25010:2011 standards for functional suitability and reliability, the mPOS application supports SMEs in embracing digital transformation and competing in a data-driven economy.

5. Conclusion

This study successfully developed an mPOS application utilizing Firebase for real-time data synchronization between sales and inventory, addressing common challenges faced by SMEs in managing sales and inventory. The application improves operational efficiency by minimizing inventory discrepancies and reducing operational delays, with key features such as real-time stock notifications ensuring timely restocking and preventing both stockouts and overstocking. Comprehensive testing validated the application's performance, reliability, and user-friendliness, confirming its suitability for small-scale retail environments. This research contributes to the body of knowledge on real-time data systems for SMEs, demonstrating the

feasibility and benefits of integrating Firebase into mPOS applications. It is recommended for SMEs looking to enhance their sales and inventory management with a cost-effective and scalable solution. Future research should explore the system's scalability in larger retail settings, investigate its integration with other backend services, and evaluate its effectiveness in various business contexts to further validate its potential for broader implementation.

Disclosure statement
No potential conflict of interest was reported by the author(s).

Funding
This work was not supported by any funding.

ORCID
Farel Naufal Azhari – https://orcid.org/0009-0008-1962-0097
Sutarman – https://orcid.org/0000-0003-2171-9260

References

Afifi, H., Karl, H., Gburrek, T., & Schmalenstroeer, J. (2022). Data-driven time synchronization in wireless multimedia networks. *2022 International Wireless Communications and Mobile Computing (IWCMC)*. https://doi.org/10.1109/IWCMC55113.2022.9824980

Ahmed, M. (2022). Backend as a service cloud computing integrated with cross-platform mobile development framework to create an e-learning application that works in mobile and web with a single codebase. *4th International Conference on Communication Engineering and Computer Science (CIC-COCOS'2022)*, 50–57. https://doi.org/10.24086/cocos2022/paper.511

Alhazmi, R. M. (2020). System architecture of point-of-sale debt system using mobile technology. *2020 International Conference on Computing and Information Technology (ICCIT-1441)*, 1–4. https://doi.org/10.1109/ICCIT-144147971.2020.9213723

Ashita, Bahl, V., Goel, A., & Sengar, N. (2020). Android based instant messaging tool using firebase as backend. *International Journal for Modern Trends in Science and Technology*, *6*(12), 198–201. https://doi.org/10.46501/IJMTST061238

Chopdar, P. K., & Balakrishnan, J. (2020). Consumers response towards mobile commerce applications: S-O-R approach. *International Journal of Information Management*, *53*, 102106. https://doi.org/https://doi.org/10.1016/j.ijinfomgt.2020.102106

Chowdhury, D. D. (2021). Synchronization for Smart Grid Infrastructure. In *NextGen Network Synchronization* (pp. 181–207). Springer International Publishing. https://doi.org/10.1007/978-3-030-71179-5_10

Dantu, V. V. R., Sai Dasaradha, V. V. S., & Sasikumar, P. (2021). Unified automotive location tracking using android things (IoT). *Wireless Personal Communications*, *120*(1), 63–79. https://doi.org/10.1007/s11277-021-08434-y

Demissie, B. F., & Ranise, S. (2021). Assessing the effectiveness of the shared responsibility model for cloud databases: The case of Google's Firebase. *2021 IEEE International Conference on Smart Data Services (SMDS)*, 121–131. https://doi.org/10.1109/SMDS53860.2021.00026

Haidukevych, Y. O., & Doroshenko, A. Yu. (2022). Automated inventory management system on Android using barcodes and QR-codes. *Problems in Programming*, *1*, 013–022. https://doi.org/10.15407/pp2022.01.013

Hasan, M. K., Ahmed, M. M., Hashim, A. H. A., Razzaque, A., Islam, S., & Pandey, B. (2020). A Novel Artificial Intelligence Based Timing Synchronization Scheme for Smart Grid Applications. *Wireless Personal Communications*, *114*(2), 1067–1084. https://doi.org/10.1007/s11277-020-07408-w

Intal, G. L., Miranda, L. M. P., Sitoy, J. I., & Matoza, R. (2022). iBiz: An android mobile application with business intelligence for retail micro and small enterprises (MSEs). *Proceedings of the 6th International Conference on E-Commerce, E-Business and E-Government*, 145–151. https://doi.org/10.1145/3537693.3537755

Joshi, J., Bhirud, D., Shinde, G., Avhale, V., Vispute, S. R., & Rajeswari, K. (2022). Inventory and attendance management system for construction firm with voice assistant. *2022 6th International Conference On Computing, Communication, Control and Automation (ICCUBEA*, 1–5. https://doi.org/10.1109/ICCUBEA54992.2022.10010844

Kimm, H., & Flynn, D. (2022). Android application for tracking pedestrian movements in realtime with firebase. *2022 IEEE Cloud Summit*, 91–96. https://doi.org/10.1109/CloudSummit54781.2022.00020

Lewis, E. C., Harper, K. M., Poirier, L. K., & Gittelsohn, J. (2022). Feasibility of using mobile point-of-sale technology in Baltimore City corner stores for tracking sales: A brief report. *Journal of Public Health Research*, *11*(1). https://doi.org/10.4081/jphr.2021.2607

Margate, A. M. N., Ravina, Ma. C. F., Pido, J. J. G., & Young, M. N. (2020). Seiton: A mobile inventory management system application for micro, small and medium-sized enterprise. *2020 IEEE 7th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, 1–5. https://doi.org/10.1109/ICETAS51660.2020.9484183

Melvin, M., Wiratama, J., Sutomo, R., & Sanjaya, S. A. (2023). A web-based point of sales for automotive component industry using rapid application development model. *JOINS (Journal of Information System)*, *8*(2), 167–176. https://doi.org/10.33633/joins.v8i2.9383

Naik, G. R. (2023). AI based inventory management system using Odoo. *International Journal of Scientific Research in Engineering and Management*, *07*(08). https://doi.org/10.55041/ijsrem25510

Pandit, V. K. S., Yogeshwar, S. K., Priya, R., Ramesh, T., & Ranjitha, P. (2023). Real-time database synchronization techniques in firebase for mobile app development. *International Journal of Scientific Research in Engineering and Management (IJSREM)*, *7*(5), 1–7. https://doi.org/10.55041/IJSREM22021

Reddy Lakkireddy, S. N., Thomas, A. A., Shree, T. S., & Mamatha, T. (2022). Web-based application for real-time chatting using firebase. *2022 International Conference on Knowledge Engineering and Communication Systems (ICKES)*, 1–4. https://doi.org/10.1109/ICKECS56523.2022.10060845

Ruiz, A. P., Rivas, M. A., & Harbour, M. G. (2020). Non-blocking synchronization between real-time and non-real-time applications. *IEEE Access*, *8*, 147618–147634. https://doi.org/10.1109/ACCESS.2020.3015385

Tandel, T., Wagal, S., Singh, N., Chaudhari, R., & Badgujar, V. (2020). Case study on an android app for inventory management system with sales prediction for local

shopkeepers in India. *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 931–934. https://doi.org/10.1109/ICACCS48705.2020.9074234

Tsai, W. L. (2021). An intelligent point-of-sale system for mobile retailors. *Journal of Physics: Conference Series*, *1927*(1). https://doi.org/10.1088/1742-6596/1927/1/012015

Udayaprakash, U., & Sujatha, S. (2020). *Mobile Application Integrated with on SAP Business one Software based on Sales and Collection Management System.* International Journal of Engineering Research & Technology, 9(9), 400-402. https://doi.org/10.17577/IJERTV9IS090289

Vaz, D., Matos, D., Pardal, M., & Correia, M. (2021). MIRES: Recovering mobile applications based on backend-as-a-service from cyber attacks. *MobiQuitous 2020 - 17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 262–272. https://doi.org/10.1145/3448891.3448932

Wiesner, A., & Kovácsházy, T. (2023). Methods of peripheral synchronization in real-time cyber-physical systems. *2023 24th International Carpathian Control Conference (ICCC)*, 465–470. https://doi.org/10.1109/ICCC57093.2023.10178979