

Flow-Tech

S. Sankareswari¹, Sahil Vilas Bhatkar², Aditya Shankar Dhuri³, Mayur Balasaheb Garje⁴

¹ Assistant Professor, Department of Information Technology, Finolex Academy of Management and Technology, Maharashtra, India

^{2,3,4} B.E. Students, Department of Information Technology, Finolex Academy of Management and Technology, Maharashtra, India

DOI: 10.5281/zenodo.20628286

ABSTRACT

For contemporary agency-driven sales settings, the manual processing of orders and slow updates on inventory can commonly result in the existence of stock discrepancies, billing errors and decreased business efficiency. The given project shows the design and development of a real-time integrated Sales and Inventory Management System which interrelates a web-based Admin Dashboard and a mobile Salesman app called FlowTech. The system allows smooth interface between field operation and centralized work on the inventory use via secure APIs and role-based workflows. Admin Dashboard is developed based on Next.js 15, TypeScript, Tailwind CSS, and Shadcn/UI to provide the scalable and responsive user interface. SQLite and Drizzle ORM provides the dependable and type-safe database management whereas Better-Auth offers the secure session-based authentication. The system has an operational stock accounting ledger where all stock transactions can be followed and assists in creating invoices with the help of js PDF and processing CSV files in large volumes. FlowTech mobile application is designed by mobile-first principles and in the visual consistency of the dashboard, it allows the sales personnel to build and process the orders with ease. Mobile app orders must be approved by the administration before inventory changes can be implemented, eliminating unintentional stock modifications and enhancing the accuracy of the data. There exists strong validation and error-handling that will give consistency in the synchronization of systems. The offered solution helps to achieve the best level of transparency, reduce human error, promote operational control, and offer a base of further development in real-life enterprise conditions.

Keyword : - Inventory Management, Mobile App Integration, Real-Time System, Order Approval, Role-Based Access, Invoice Generation, Data Validation

1. INTRODUCTION

The effective sales and inventory management is crucial in smooth running of the agency based distribution systems. A lot of companies rely on the manual work or disjointed digital solutions to handle orders, stockings, and billing activities. These conventional practices tend to produce a slow inventory update, erroneous stock movement, human mistakes, and real-time invisibility, thus, impacting negatively on business performance and consumer contentment.

As more and more devices are adopting the mobile platform and modern web technologies, there is the increased necessity of having integrated platforms that bridge the gap between the field-level sales activities and the centralized administrative control. An efficient system must enable the sales staff to make orders with a mobile application and financial transactions and inventory management by administrators should be safe. There must be real-time synchronization of mobile applications with web dashboards to eliminate discrepancies in the data and conflict in operations.

The project will be a proposal of a real-time built Sales and Inventory Management System that will be made up of a web-based Admin Dashboard and a mobile Salesman application called FlowTech. The system is developed with modern full-stack technologies so that it can be scaled, become more secure and reliable. Role-based approval workflow is adopted whereby orders that have been made by the salesmen or the delivery individuals are checked and approved by the administrators, prior to deduction of inventory. This reduces unwanted stock variations and business regulations are in place.

A transactional stock ledger also forms part of the system to trace all stock operations, automated generation of invoices and handling of bulk data using CSV imports. The proposed solution, based on the principles of secure authentication, structured database design, and user-friendly interfaces, is expected to enhance the level of operational transparency, decrease the amount of manual work, and present an effective platform on which the further improvement of the system of sales and managing inventory at the enterprise level should be based.

2. LITERATURE REVIEW

1 History of Inventory management Systems.

The Inventory Management Systems (IMS) have developed beyond the traditional, manual record-keeping to

automated and computer-driven systems, which are used to address the multifaceted operations of supply-chain and retail. Recent surveys and reviews sum up the current IMS solutions focus on automation, traceability, and links with sales channels to eliminate stockouts and overstocking. These researches highlight the importance of ensuring that there is a good transactional logging and real-time visibility so as to ensure that there is a good inventory.

2 Importance of Real-time Synchronization.

Real-time Synchronization is important since it improves real-time communication between the domestic and the outside markets. Numerous recent applications and researches point out that real-time data updates are a great way to enhance better decision-making, decrease lost sales, and timely replenishment. The systems, which give the real-time stock visibility to the warehouse administrators and field salespeople, minimize the discrepancies in the inventory filling processes due to long synchronization and manual reconciliation. Another aspect of the literature notes that real-time features are also being implemented with both web APIs and mobile clients so that consistency is ensured between the channels.

3 Web-Mobile Integration Patterns.

The research and practical projects suggest a distinct division between mobile frontends (to perform the field operations) and centralized backends (to contain the authoritative data), which are interconnected through properly designed APIs, which have authentication controls, validation controls, and idempotency controls. The backend should be presented as the source of truth and orders should be sent as pending transactions, and destructive actions on inventory should be postponed until server-side confirmation: this ensures that accidental inventory changes are minimized, and can be used to maintain audit trails. A number of case studies in the literature embrace this stock-update method of approval to be reliable and comply.

4 Auditability and Transactional Stock Ledger.

One of the common suggestions in the academic and industry literature is to have a transaction-type stock ledger where all "Stock In" and "Stock Out" incidents with their timestamps and user IDs and order or supplier receipts are recorded. Such ledger method aids in reconciliation, auditability, and other forensic tracking of inventory movement; its implementation is particularly necessary when mobile agents have the ability to trigger stock-related events. Ledger-backed model systems have more accuracy and accountability.

5 Next.js 5 Modern Web Stack and developer productivity.

Models like Next.js (App Router) encourage server-first rendering, route handlers, and unified API processing that is beneficial towards creating dashboards and secure APIs that the mobile clients consume. The conventions of the App Router (persistent layouts, server actions, and fine-grained caching) assist in the development of performant administration interfaces and predictable server-side behavior of important actions such as order approval and invoice generation.

6 Type-safe SQLite and Edge / Local Deployments (Drizzle ORM)

Type-safe ORMs which work well with TypeScript minimize runtime schema errors and accelerate development. The built-in SQLite support of drizzle ORM (through libsql/better-sqlite3) is ideal when the project needs optimization through the use of a small, file-based storage in the initial development or edge deployment. Its TypeScript schema description and migration tooling assist in having a stable data layer of ledger-like transactions.

7 Authentication and Secure Session Management (Better-Auth)

Any administrative dashboard should have secure and session-based authentication. Auth libraries like Better-Auth have Next.js integrations and have patterns of session and account management that can be used to easily implement role-based access controls (Admin vs Salesman). With an effective auth system, the probability of defective self-implemented security logic is minimalized, and the effort to configure protected routes/middleware is minimal.

8 Utility Libraries: Invoicing PDF and CSV Parsing.

The implementation of the guideline is often necessitated by trusted third-party assistants: PapaParse is popularized as a robust and high-speed CSV-parser in-the-browser and server-side, is handy when you need to import a large number of suppliers at once; jsPDF (and invoice template packages based on it) is popularized as an in-the-browser in-the-server PDF-generator in a fixed, GST-compliant format. These libraries are well documented, mature and are applicable in importing/ exporting and invoicing requirements that are detailed in this project.

9 Gaps, Challenges and Design Implications.

Despite the advantages of real-time sync and ledger-based tracking the literature is concerned about the reliability of the network, concurrency and the complexity of the integration. The most important issues are the need to be able to guarantee idempotent API endpoints, the behavior of an offline app (local caching and reconciliation), and ensuring that race conditions do not occur when interacting with different actors on the same stock items. The following caveats drive the design decisions in this project: (1) approving orders in advance before stock mutation, (2) strong server-side validation and transactional updates, and (3) effective audit trails using the stock ledger.

3. PROBLEM STATEMENT

The small and medium-sized enterprises (SMEs), especially those that are working in distribution-oriented business like retail sales, local agencies and others that rely on the supply in the business usually experience poor order handling and inventory tracking. Many of these businesses still rely on manual or semi-digital operations such as phone calls, handwritten logs, spreadsheets or informal messaging applications to run their day-to-day business. These techniques do not use standardization, centralized storage of data and real time synchronization.

Consequently, some of the problems that are often faced by businesses include the entry of incorrect orders and confirmation, deduction of stock accidentally, and inability to trace the movement of inventory. The lack of an organized approval system also enhances the likelihood of mistakes particularly in cases where there are several individuals taking part in the placement of orders and delivery. Also, a majority of the available digital solutions are web based or mobile based and this restricts their applications in various operational functions.

Most existing systems are single-vendor or domain specific in nature and cannot support role-based workflows specifically designed to work with administrators, sales staff and delivery agents. This poses a problem in the scaling of operations, transparency, and accountability. As such, it is necessary to have an integrated, role awareness, and real-time digital solution that would effectively handle orders, inventory, and communication among various stakeholders with a reduction in human error and enhanced operational controls.

4. PROPOSED SYSTEM

This project is a proposal of a real time integrated Sales and Inventory Management System that links a Web based centralized Admin Dashboard with mobile Salesman application by the name FlowTech. The suggested system will aim to computerize the order management, inventory update, and approval processes and have a tight control of access controls based on roles.

It is a modern full stack system with an integration-friendly architecture based on modern full-stack technologies. Admin Dashboard is built on Next / TypeScript and offers an accessibility and scalable web interface to administrative users. The mobile application is designed as a mobile-first approach and is aimed at sales and delivery staff where they submit their orders and confirmation of deliveries. A relational database is used on data storage to include a transactional stock-ledger to make it accurate and traceable. Session based authentication is also secure and the user is authenticated before gaining access to the system resources.

The key functional components of the proposed system are described below:

1. Role Management and Authentication of user

- Users have access to a secure authentication system.
- Every user has a predefined role which can be an Admin or Salesman.
- Access controls are of a very strict nature on the basis of the user roles.
- The salesmen are capable of making and placing orders but not changing inventory.
- Only administrative users can be able to give orders a green light or not.
- Role-based validation helps to avoid access to the restricted features that are not permitted to unauthorized persons.

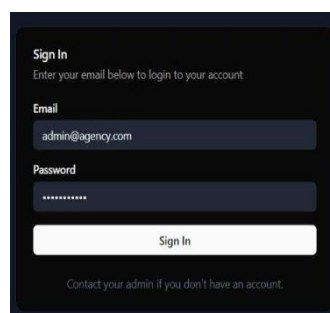


Fig -1: Admin Login

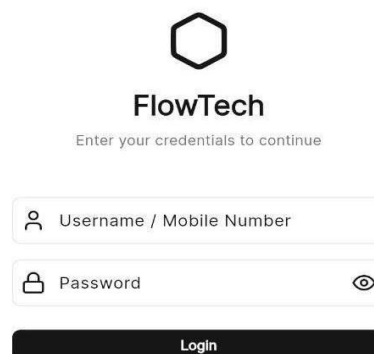
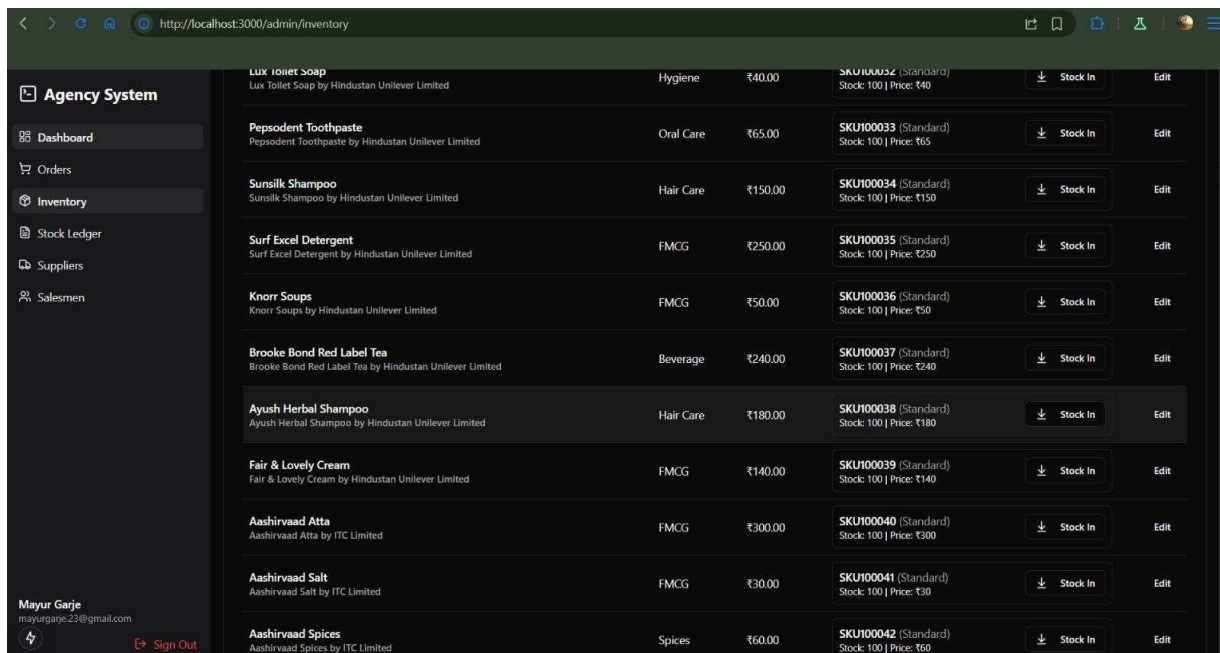


Fig -2: Salesman Login

2. Inventory Management and Product.

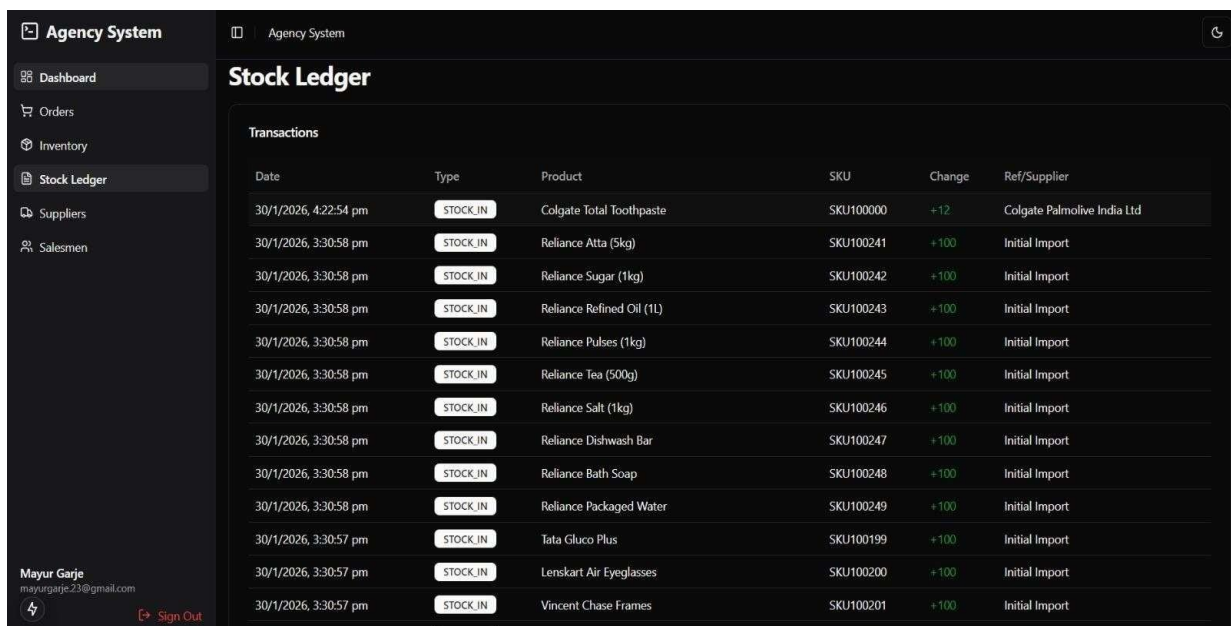
- The dashboard allows managers to maintain product categories, variants, prices, and supplier information.
- The data about the inventory are recorded in a central database and is processed through a transactional stock ledger.
- Added stocks are entered by means of suppliers.

- Stock deductions are made after the approval of orders.
- The level of low stock is observed to help in the decision of restocking.
- The inventory changes are recorded in order to allow auditing.



Product Name	Category	Price	SKU	Stock	Price	Actions
Lux Toilet Soap	Hygiene	₹40.00	SKU100032 (Standard)	Stock: 100	Price: ₹40	Stock In Edit
Pepsodent Toothpaste	Oral Care	₹65.00	SKU100033 (Standard)	Stock: 100	Price: ₹65	Stock In Edit
Sunsilk Shampoo	Hair Care	₹150.00	SKU100034 (Standard)	Stock: 100	Price: ₹150	Stock In Edit
Surf Excel Detergent	FMCG	₹250.00	SKU100035 (Standard)	Stock: 100	Price: ₹250	Stock In Edit
Knorr Soups	FMCG	₹50.00	SKU100036 (Standard)	Stock: 100	Price: ₹50	Stock In Edit
Brooke Bond Red Label Tea	Beverage	₹240.00	SKU100037 (Standard)	Stock: 100	Price: ₹240	Stock In Edit
Ayush Herbal Shampoo	Hair Care	₹180.00	SKU100038 (Standard)	Stock: 100	Price: ₹180	Stock In Edit
Fair & Lovely Cream	FMCG	₹140.00	SKU100039 (Standard)	Stock: 100	Price: ₹140	Stock In Edit
Aashirvaad Atta	FMCG	₹300.00	SKU100040 (Standard)	Stock: 100	Price: ₹300	Stock In Edit
Aashirvaad Salt	FMCG	₹30.00	SKU100041 (Standard)	Stock: 100	Price: ₹30	Stock In Edit
Aashirvaad Spices	Spices	₹60.00	SKU100042 (Standard)	Stock: 100	Price: ₹60	Stock In Edit

Fig -3: Inventory



Date	Type	Product	SKU	Change	Ref/Supplier
30/1/2026, 4:22:54 pm	STOCK_IN	Colgate Total Toothpaste	SKU100000	+12	Colgate Palmolive India Ltd
30/1/2026, 3:30:58 pm	STOCK_IN	Reliance Atta (5kg)	SKU100241	+100	Initial Import
30/1/2026, 3:30:58 pm	STOCK_IN	Reliance Sugar (1kg)	SKU100242	+100	Initial Import
30/1/2026, 3:30:58 pm	STOCK_IN	Reliance Refined Oil (1L)	SKU100243	+100	Initial Import
30/1/2026, 3:30:58 pm	STOCK_IN	Reliance Pulses (1kg)	SKU100244	+100	Initial Import
30/1/2026, 3:30:58 pm	STOCK_IN	Reliance Tea (500g)	SKU100245	+100	Initial Import
30/1/2026, 3:30:58 pm	STOCK_IN	Reliance Salt (1kg)	SKU100246	+100	Initial Import
30/1/2026, 3:30:58 pm	STOCK_IN	Reliance Dishwash Bar	SKU100247	+100	Initial Import
30/1/2026, 3:30:58 pm	STOCK_IN	Reliance Bath Soap	SKU100248	+100	Initial Import
30/1/2026, 3:30:58 pm	STOCK_IN	Reliance Packaged Water	SKU100249	+100	Initial Import
30/1/2026, 3:30:57 pm	STOCK_IN	Tata Gluco Plus	SKU100199	+100	Initial Import
30/1/2026, 3:30:57 pm	STOCK_IN	Lenskart Air Eyeglasses	SKU100200	+100	Initial Import
30/1/2026, 3:30:57 pm	STOCK_IN	Vincent Chase Frames	SKU100201	+100	Initial Import

Fig -4: Stock Ledger

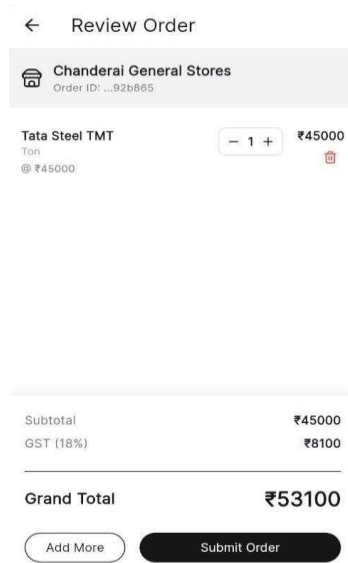
3. Creation of an Order (Mobile Application)

- Salesmen develop cart-based orders through FlowTech mobile application.
- The quantities and products are chosen out of the existing inventory.
- Intranet orders can still be modified before submission.
- When the order is received, it is changed to Pending.
- No inventory changes are implemented at this stage.

4. Order Review and Approval

- The orders on hold are shown on the Admin Dashboard.
- Order details viewed by the administrators include items, quantities and pricing.
- Administrators may:

- Approve the order Marks the inventory as changed and the status of the order alters to Approved.
- Reject the order → The inventory does not change and the order status is changed to Rejected.
- All decisions are registered in the database, which is used to maintain records.



5. Fig -5: Cart with subtotal



Fig -6: GST Invoice

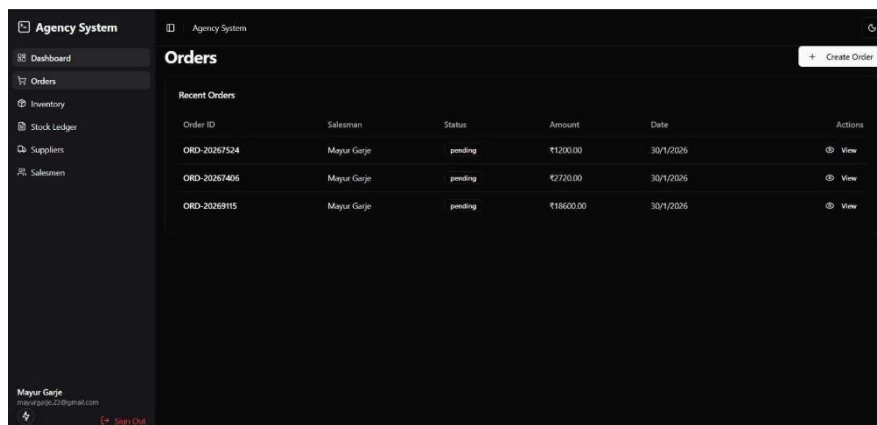


Fig -7: All Orders

6. Delivery Confirmation and Status Update.

- Delivery employees place delivery confirmation orders via the mobile application.
- The system provides an update on progress of orders on the basis of approval.
- When an order is delivered successfully, it is tracked as Completed.
- Finished orders are entered in the historical records so that they can be reported and used as reference.

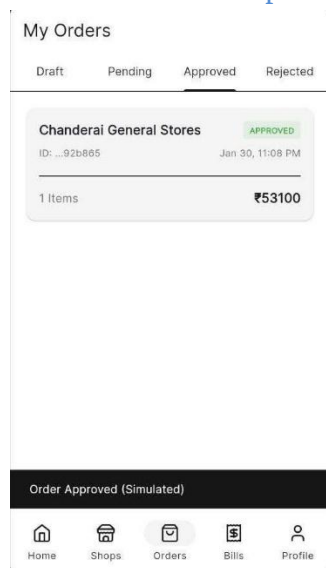


Fig -8: Order status

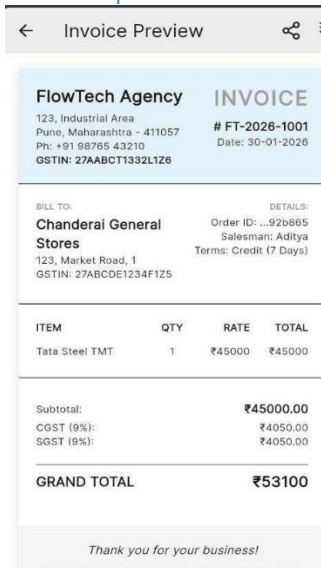


Fig -9: Invoice

7. Invoice Generation

- Once the orders have been approved, a GST compliant invoice is automatically documented by the system.
- Bills are developed in a standardized manner so that their billing is the same.
- The Admin Dashboard can be used to look at invoices and download them.
- This eradicates the process of manual billing and also reduces error.

5. SYSTEM ARCHITECTURE

The proposed system follows a structured and layered architecture that connects a web-based Admin Dashboard with a mobile Salesman application called FlowTech. The architecture is designed to ensure security, scalability, and real-time data synchronization between all users.

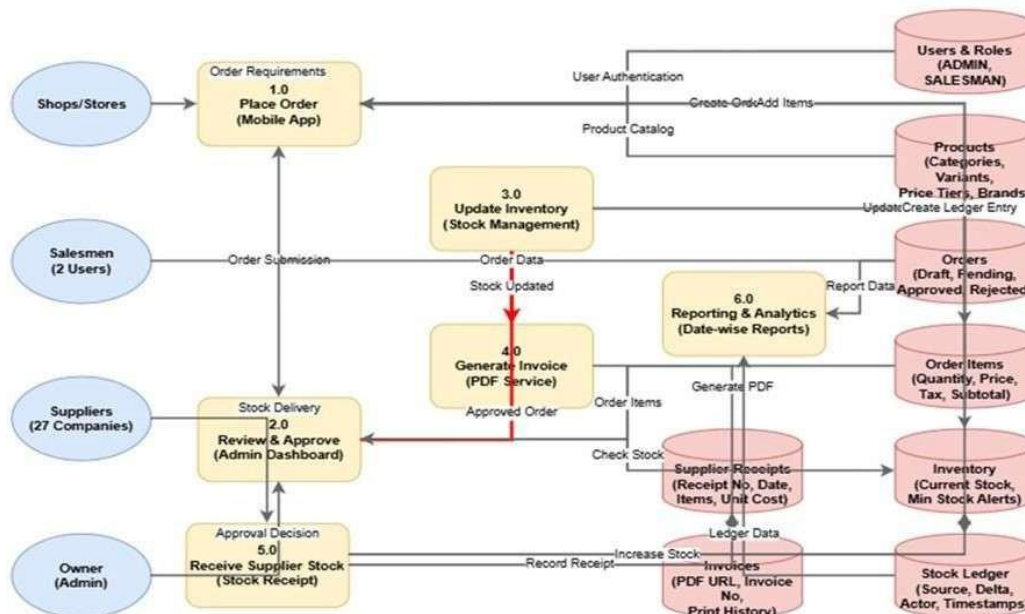


Fig -10: System Architecture

The system is divided into three main layers: the Presentation Layer, the Application Layer, and the Data Layer.

1. Presentation Layer

This is the user interaction layer of the system. It includes:

- The Admin Dashboard (Web Application)
- The FlowTech Mobile Application

Administrators use the web dashboard to manage products, approve orders, monitor inventory, and generate reports. Salesmen use the mobile application to create orders, track shop visits, and submit delivery updates. Both interfaces are designed to provide role-specific access, meaning users can only see and use features

allowed for their role.

2. Application Layer

This layer acts as the brain of the system. It processes all business logic and handles communication between the frontend applications and the database.

When a salesman submits an order from the mobile app, the request is sent to the backend server through secure APIs. The server validates the request, stores it as a pending order, and waits for admin approval. Once the admin approves the order from the dashboard, the system automatically updates the inventory and generates the invoice.

Role-based access control is enforced in this layer to prevent unauthorized actions. For example, salesmen cannot directly modify stock quantities or approve orders.

3. Data Layer

The data layer is responsible for storing all system information in a structured database. It includes:

- User data (Admin and Salesman accounts)
- Product and inventory details
- Shop and territory information
- Order records and statuses
- Stock ledger transactions
- Invoice records

A transactional stock ledger is maintained to record every stock addition and deduction. This ensures transparency, traceability, and accurate inventory tracking.

6. CHALLENGES

There are several technical and operational issues when creating and developing a combined sales and inventory management platform connecting a web-based administrator control program and a mobile app. The ability to keep real-time data synchronization of the mobile app and the backend system is one of the most crucial challenges. Request delays, network failure, or multiple order entries can be inconsistent unless it is addressed properly.

Strict role-based access control is another significant limitation that has to be imposed. The system should be such that the salespeople can create and place orders only and the administrative users should be given all the power of order acceptance and updating inventory. To ensure that there are no excessive or unintended stock changes in the system, there is a necessity to introduce an approval-based workflow, which complicates the entire system design.

There are also issues of platform differences and other network conditions, which means that having a reliable communication between the mobile application and the web dashboard is difficult. Validation and error-handling procedures should be properly validated to deal with invalid inputs, duplicates as well as missing data transfers. Moreover, to have a transactional stock ledger and produce correct and standardized invoices automatically, a thorough database design and coherent business logic is essential to provide system reliability and integrity of data.

7. FUTURE DIRECTION

The Sales and Inventory Management System suggested is a sufficient basis of digital order processing and immediate inventory control, but several improvements could be made in the future to make the system even more functional and scalable. The push notifications can be integrated as one of the potential directions to deliver an immediate notification on order approval, delivery, and low-stock status to enhance the responsiveness of both administrator and sales staff.

The next iterations of the system may incorporate the offline support in the mobile application, where salesmen can create and store the orders in the locations with low network connectivity and synchronize the data when connection is restored. More sophisticated analytics and reporting can also be implemented to give information on the sales trends and demand forecasting as well as inventory optimization.

The system has the potential of being expanded to accommodate other user functions like the warehouse personnel or regional managers as well as multi-branch and multi-location inventory control. Monetary transactions would also be automated by having the ability to integrate with payment gateways and accounting software. Moreover, the cost-effectiveness and scalability of the large-scale deployment could be improved by migration of the system to cloud-based infrastructure, which increases its performance, reliability, and scalability.

All in all, these improvements of the future would make the suggested system an enterprise-level platform, which can potentially meet the expanding business demands.

8. CONCLUSION

This project is effective in documentation with regard to highlighting the design and development of a real-time integrated Sales and Inventory Management System to indicate the bridge between the mobile field operations and the centralized administrative control. Offering a combination of a web-based Admin Dashboard and a mobile Salesman Application, the system lets offer a well-organized and effective order processing, inventory, and an approval-dependent working process.

Role-based access control introduces administration and protection over system use as well as ensuring the accuracy of the available data, and the approval feature prevents the occurrence of accidental update of the inventory.

Transactional stock ledger is better because it enhances the level of transparency and offers effective audit trails on inventory flow. Automated invoice processing and dashboard analytics further make work in business much smoother and manual work is minimized.

All in all, the suggested system can enhance performance efficiency, reduce human mistakes, and provide a scalable system intended to be used in small and medium-scale businesses. Its flexible architecture and design that is easily integrable provides the system with a strong background of future improvements and implementation to the real world.

9. REFERENCES

- [1] Next.js Team, Next.js Documentation, Vercel Inc., 2024.
- [2] Drizzle ORM Contributors, Drizzle ORM Official Documentation, 2024.
- [3] SQLite Consortium, SQLite Database Engine Documentation, 2024.
- [4] Better-Auth Contributors, Better-Auth Authentication Framework Documentation, 2024.
- [5] Tailwind Labs, Tailwind CSS Documentation, 2024.
- [6] Radix UI Team, Radix UI and Accessible Component Design, 2024.
- [7] PapaParse Contributors, PapaParse: CSV Parsing Library Documentation, 2024.
- [8] Mr. Rio, jsPDF Documentation, 2024.
- [9] Android Developers, Material Design Guidelines, Google LLC, 2024.
- [10] I. Sommerville, Software Engineering, 10th ed., Pearson Education, 2016.
- [11] R. S. Pressman, Software Engineering: A Practitioner's Approach, McGraw-Hill Education, 2014.
- [12] A. Silberschatz, H. F. Korth, and S. Sudarshan, Database System Concepts, McGraw-Hill, 2019.
- [13] M. Christopher, Logistics & Supply Chain Management, 5th ed., Pearson Education, 2016.
- [14] E. A. Silver, D. F. Pyke, and D. J. Thomas, Inventory and Production Management in Supply Chains, 4th ed., CRC Press, 2016.
- [15] P. Helo and Y. Hao, "Cloud-Based Inventory Management System," International Journal of Information Systems and Supply Chain Management, vol. 10, no. 2, pp. 1–15, 2017.