INVENTORY MANAGEMENT SYSTEM

Project By.: Soma Navadeep
Date : 30/07/2025

Situation:

- Service providers are trying to keep up with these changes by improving their services and integrating technology.
- Businesses are involved in managing product details, purchases, sales, and stock.
- The system currently tracks inventory, but may be limited or manual in some areas.
- Data related to products, stock, sales, and purchases is being stored in databases.
- Customer buying preferences are dynamic and change frequently.

Problem :

- **Manual or outdated systems** may result in: stockouts or overstock situations, delays in transactions and reporting and inaccurate tracking of inventory movement.
- Lack of real-time data visibility on stock status between specific dates.
- Inefficiencies in purchase and sales tracking, leading to reduced customer satisfaction and missed sales opportunities.
- No centralized actor or workflow for managing stock if roles are undefined.
- Frequent changes in customer preferences make it difficult to predict demand accurately.

Opportunity :

- Implement a centralized, tech-enabled inventory management system.
- Provide real-time visibility of stock levels, purchase history, and sales trends.
- Use the system to generate custom reports between dates for analysis and planning.
- Allow automation of stock updates based on purchases and sales.
- Improve decision-making through clear dashboards and data tracking.
- Empower businesses to respond quickly to market changes and align supply with customer demand.
- Use the system to generate custom reports between dates for analysis and planning.

Problem Statement

Businesses are struggling to keep up with rapidly changing customer preferences due to outdated or inefficient inventory systems, resulting in poor stock management and delayed services. The lack of real-time tracking and automation hampers decision-making, disrupts supply-demand balance, and affects customer satisfaction.

Key Issues Identified:

- Stock is not updated automatically and needs to be managed manually.
- It's hard to know how much stock is available at any given time.
- Sales and purchase details are not properly connected to stock updates.
- Business owners can't easily check stock reports between two dates.
- Sometimes there is too much stock, and other times it runs out unexpectedly.
- Customer billing and service take longer due to slow systems.
- No single system to manage product, purchase, and sales details together.

Project objectives :

1. **Improve Inventory Management**

   Ensure real-time tracking of stock - what's coming in, what's going out, and what's left (current stock)

   Reduce manual work and chances of errors in stock records.

2. **Faster and Accurate Transactions**

   Make customer billing and stock updates quick and smooth.

   Improve service quality by avoiding delays and stock mismatches.

3. **Better Decision Making with Reports**

   Generate clear reports on stock movement, sales, and purchases.

   Allow filtering stock data between any two dates.

4. **Support Business Growth**

   Make it easier to handle more products, stores, or customers as the business grows.

5. **Enable Role-Based Access**

   Allow different access levels for admin, sales staff, and purchase staff.

6. **Track and Improve Service Quality**

   Use data to understand which products are in demand.

   Improve customer satisfaction by always having popular items in stock.

7. **Create and Test a Prototype**

   Develop a basic working model of the system.

   Test its functionality, user-friendliness, and performance before final implementation

8. **Choose the Right Solution**

   Select a system that meets business goals, technical requirements, and user needs.

   Ensure it supports product, purchase, sales, and stock management in one place.

Success Criteria:

1. **Easy to Use**

   Staff can use the system without much training or confusion.

2. **Real-Time Stock Updates**

   Stock is updated instantly after every sale or purchase.

3. **All Data in One Place**

   Product, sales, purchase, and stock details are all stored in one system.

4. **Reports Are Accurate and On Time**

   Sales, purchase, and stock reports are correct and can be generated for any time period.

5. **No Stockouts or Overstock**

   The business can maintain just the right amount of stock by using the system.

6. **Secure and Role-Based Access**

   Only authorized users can access or change certain parts of the system.

7. **System Can Grow with Business**

   The solution can handle more data and users as the business grows.

8. **Positive Feedback from Users**

   Employees and management are happy with how the system works and supports their tasks.

Waterfall Method/Approach

The waterfall model is a linear and sequential approach where the project progresses through distinct phases. Each phase must be completed before the next begins, and there is typically no overlapping between phases.

1. **Requirement Gathering and Analysis**

   All system requirements are collected from the stakeholders or clients.

   Requirements are documented clearly and signed off before starting design.

   Example: products, stock, sales.., etc

   Deliverables: requirements specification document (RSD)

2. **System Design**

   Based on the requirements, a design is prepared that includes system architecture, data models, screen layouts, etc

   Example: Design of database tables for product, sales, and stock details.

   Deliverables: system design document (SSD)

3. **Development**

   Developers start writing code based on the finalized design.

   Each module (product entry, purchase, sales, reporting) is developed step by step.

   Deliverables : Fully coded and internally tested

4. **Testing**

   After coding, the system is tested to check if it meets the requirements.

   Bugs are identified and fixed.

   Example: Testing if stock updates correctly after sales.

   Deliverables: Testing report and bugs fixes

5. **Deployment**

   Once tested and approved, the system is installed for actual use.

   End users (like admin or staff) start using it.

   Deliverables: live and working model

6. **Maintenance and support**

   After deployment, updates, bug fixes, or small changes may be needed.

   Example: Adding a new report or fixing login issues.

   Deliverables : support document, updates

Resources

1. **People (HR)**

The project team will include members from both the client side and the IT/software development team.

Project manager (1) - Plans and manages the entire project

Business analyst (1) - Collects and documents all system requirements

Developers (3) - Build the core functions of the system (like product, stock, purchase, and sales modules)

UI/UX Designer (1) - Designs the look and layout of the system screens

Test Team/QA (1) - Tests the system to find and report errors

Deployment / Support Engineer (1) - Installs the system on client machines or cloud servers

2. **Timeline**

Estimated total project duration 3 – 4 months

Requirements gathering and analysis - 2 weeks

System design – 2 weeks

Development – 6 weeks

Testing – 2 weeks

Deployment – 2 weeks

Support – 2weeks

3. **Budget**

Estimated budget components

Development costs : Salaries

Infrastructure costs : Services, backup systems

Licensing costs : Software licenses

Maintenance and support : 1-2 months of post development support

Rough Budget 5-6 lakhs INR

4. **Others**

Hardware - Dedicated servers for hosting the application and database

Backup and disaster recovery systems

Software – Development tools (java, database tools)

Network - High speed internet connections

Risk and dependencies

1. **New system may be hard to use at first**

   If the new system Is not easy to understand, staff might make mistakes or avoid using it.

2. **Data Migration Risk**

   Errors or incomplete migration of data (from the old system to the new one) can lead to data loss or inaccuracies.

3. **Cost Justification Risk**

   If the benefits (like ease of use, speed, and better data) are not clearly visible, stakeholders may question the cost of development and implementation.

4. **Business Disruption During Transition**

   Switching to a new system might require some downtime, which can temporarily disrupt operations.

Dependencies

1. **User Training and Support**

   Successful adoption depends on how well users are trained and supported during the transition.

2. **Third-party Tools and Software**

   The project may rely on external tools for reporting, database, or UI components. Their availability and compatibility are critical.

3. **Client-side Participation**

   The client team must be available for requirement discussions, feedback, testing, and approvals at various stages.

4. **Hardware/Infrastructure Setup**

   The system depends on proper setup of servers, network, or cloud infrastructure on time for deployment.

5. **Data Availability and Accuracy**

   Accurate and complete historical data must be available from the existing system for smooth migration and setup.

Thank you.!