**SALES AND INVENTORY SYSTEM FOR KAPEÑA CAFE**

A Design Document Presented to the

Faculty of Datamex College of Saint Adeline, Inc

In Partial Fulfillment of the Requirements for the

Degree of Bachelor of Science in Information Technology

By:

Fuentes, Marlon

Gasalao, John kean A.

Gentile, Eljay J.

Villas, Lalaine M.

**INTRODUCTION**

Running a café like Kapeña is more than just serving coffee and snacks it also means keeping the business organized and financially steady. Every day, products are sold and ingredients are used, and if these are not properly tracked, the store risks running into problems such as mismatched stock counts, delays in service, or overspending on supplies. Kapeña Café has long relied on manual methods like handwritten records and basic lists to manage its sales and inventory. While this approach works on a small scale, it often leads to errors and makes it difficult to keep up during busy hours. As demand grows, these limitations affect not only efficiency but also the quality of service that customers receive.

The purpose of the Kapeña Café Sales and Inventory System is to address these challenges by computerized the way sales and inventory are handled. Instead of scattered notes or guesswork, the system keeps everything in one place sales, stock levels, product categories, and reports making the process faster and more accurate. It also reduces the risk of human error and ensures that updates happen in real time, so managers and staff always have a clear picture of what is available and what needs restocking.

This system is designed to do more than just store records; it helps the café understand its daily operations better. With features like low-stock notifications, stock adjustments for returns, role-based user access, and reporting tools, the owner and staff can make smarter decisions whether it’s planning purchases, tracking sales trends, or identifying where improvements can be made. By automating these essential processes, the Sales and Inventory System allows Kapeña Café to focus on what it does best: serving quality products and creating a smooth, enjoyable experience for every customer.

.

**SYSTEM ARCHITECURE**

The Kapeña Café Sales and Inventory System is a desktop application that offers an organized way of managing daily sales and inventory tasks. Sales transactions, stock updates, and reports are all integrated into a single process thanks to the system's ability to centralize data. The stock levels are automatically updated and entered in the database each time a transaction is recorded. This ensures that documents will always be accurate, reliable, and easily accessible for reporting.

**High-Level Components and Interactions**

* Database: Holds and maintains all information, including users, sales transactions, and products. In order to save and retrieve data in real time, the logic layer communicates with the database.
* User Interface Provides forms and screens for users such as staff, and admin. This is where sales are recorded, products are managed, and reports are generated.
* Reporting: Produces detailed overviews of inventory and sales records. It offers reports that display the total sales for a given time period on a weekly, monthly, or custom date period.

**Deployment Architecture**

The Sales and Inventory System for Kapeña Café is deployed as a standalone desktop application that runs directly on a personal computer without requiring a server. The client application is developed in VB.NET Windows Forms, serving as the user interface for cashiers, inventory staff, and administrators, while the backend database is managed in SQL Server (SSMS).

The system can be installed on a single computer for standalone use or configured within the café’s local area network (LAN) for multiple-user access. In either case, the database may reside on the same machine as the client or on a designated local server. This setup ensures that sales transactions and inventory updates are processed quickly and recorded in real time, without the need for internet connectivity.

**Communication protocols and interface**

The communication between the Kapeña Café Sales and Inventory System (client application) and the SQL Server database is handled through the System.Data.SqlClient library. This library provides a direct way for the application to send SQL queries and receive responses over a secure connection. If needed, the system can also be configured to use ODBC drivers, depending on the deployment environment.

**ODBC/SQL Client Drivers**

* The system uses SqlConnection objects from the System.Data.SqlClient namespace to establish communication with SQL Server.
* These drivers translate client requests (such as SELECT, INSERT, UPDATE, DELETE) into SQL commands that the database can process.

**Local/Network Connectivity**

* When the database is stored on the same machine, the connection string points to localhost or the local SQL Server instance.
* For multi-user deployment, the client connects to a central server over TCP/IP, typically using port 1433.

|  |  |  |
| --- | --- | --- |
| **User** | **Application** | **Database** |
| * Interacts through the graphical interface built in VB.NET Windows Forms, including forms, DataGridViews, ComboBoxes, Buttons, and Menus * Roles: Administrator, Cashier, and Inventory Staff * Can log in, process sales, update stock, manage products, and generate reports * Views reports such as low-stock alerts. | * Business Logic Layer processes user actions (sales, inventory updates, reporting) * Real-time stock updates whenever a sale or adjustment is recorded. * Communication protocols (System.Data.SqlClient) handle queries sent to the SQL Server database. * Includes error handling, validation, and security features to ensure accurate and safe transactions. | * SQL Server database manages all data through Create, Update, Delete (CRUD) operations * Tables such as Products, Categories, Sales, Inventory and Users store structured data. * SQL queries support aggregation (e.g., daily sales totals, best-selling products) and filtering (e.g., low-stock reports). * Ensures data consistency and reliability with real-time updates. |

**DATABASE DESIGN**

The database design for the Kapeña Café Sales and Inventory System focuses on structuring and organizing sales and inventory data in a way that ensures accuracy, consistency, and reliability. It identifies the core entities, their relationships, and the rules that govern data management within the system. By creating a well-structured schema, the system can efficiently store transactions, update inventory levels, and generate reports with minimal redundancy.

The database is implemented in SQL Server and serves as the centralized storage for all application data. Each table represents a key entity within the café’s operations, and relationships between these tables enforce data integrity. This design not only reduces the risk of inconsistencies but also enhances the overall performance of the system by ensuring that every transaction is immediately reflected in both sales and inventory records.

**Description of Database**

* Users – Stores information about administrators, cashiers, and inventory staff.
* Products – Contains details about items offered in the café such as product name, category, price, and stock level.
* Categories – Groups products into categories (e.g., Coffee, Non-Coffee, Pastries).
* Sales – Records each transaction, including product sold, quantity, date.
* Inventory – Tracks stock levels and updates automatically after sales or adjustments.
* Reports (generated) – Uses aggregated data from Sales and Inventory to produce summaries such as daily sales totals, and low-stock alerts.

**Entity–Relationship Diagram (ERD)**

users (Table)  
 {userId}  
  username: string  
  password: string  
  email: string  
  role: string (e.g., admin, user)  
  products (Table)  
 {productId}  
  category\_ref   
  product\_name: string  
  price: number  
  stock: number

sale\_items (Table)   
 {saleItemId}  
  sale\_ref ⟶ sales/{saleId}  
  product\_ref ⟶ products/{productId}  
  quantity: number  
  subtotal: number

inventory (Table)  
 {inventoryId}  
  product\_ref ⟶ products/{productId}  
  stock\_in: number  
  stock\_out: number  
  current\_stock: number  
  last\_updated: timestamp

**Data Normalization Techniques**

The database design for the Kapeña Café Sales and Inventory System adheres to the principles of Third Normal Form (3NF) to ensure accuracy, consistency, and efficiency. This normalization approach reduces redundancy, avoids anomalies, and makes the system more maintainable in the long run.

1. First Normal Form (1NF)

* All attributes are atomic, meaning each field contains only a single value.
* For example:

In the products table, attributes such as product\_name, price, and stock are stored separately, rather than combined into one field.In the users table, username, password are stored as distinct attributes.

**2.** Second Normal Form (2NF)

* Each non-key attribute is fully dependent on the table’s primary key.
* For example:

In the categories table, category\_name and description depend solely on the categoryId primary key.

In the sales table, total\_amount and date\_created depend only on the saleId.

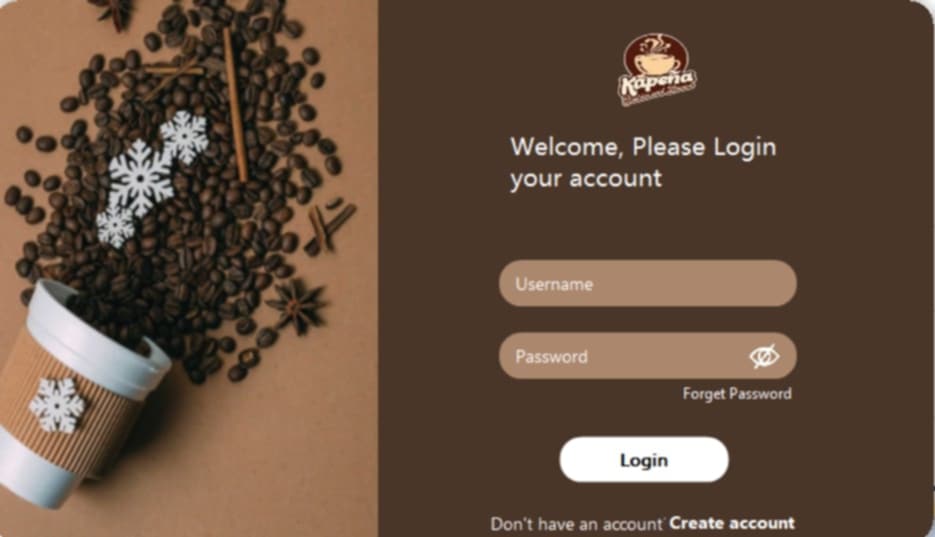
In the sale\_items table, the quantity and subtotal depend on the composite relationship of sale\_ref and product\_ref.

**3**. Third Normal Form (3NF)

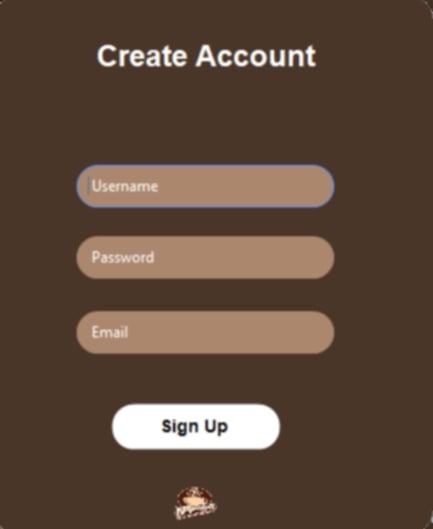
* Transitive dependencies are eliminated by introducing references instead of duplicating data.
* For example:
  1. The sales table references the users table (user\_ref) instead of repeating user details in every sales record.
  2. The products table references the categories table (category\_ref) instead of storing category names repeatedly.

**USER INTERFACE**

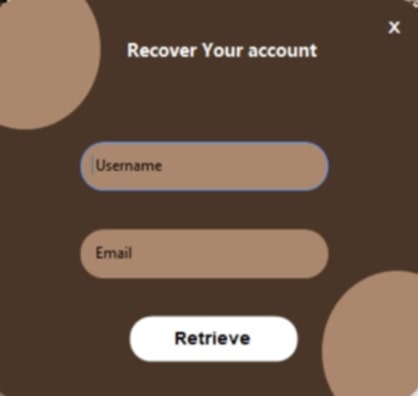
The system's desktop-based interface is designed to be simple to use. Employees have easy access to and flexibility in sales and inventory management. The organized menus, buttons, and labels are clear and free of errors, making it easier to access crucial tasks and completing transactions, stock updates, and report generation more quickly and accurately. For café employees, this design encourages a consistent workflow and smooth daily operations.

**Login Form**

This is the login interface where users enter their username and password to access the system. It also provides options for account creation and password recovery.

**Create Account Form**

This page allows users to recover their account by entering their username and registered email. It ensures users can retrieve access if they forget their login credentials.

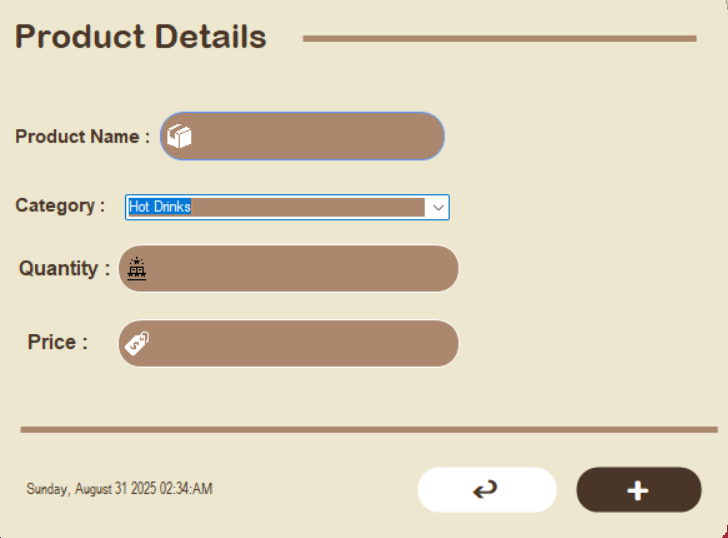
**Forgot Password Form**

This page allows users to recover their account by entering their username and registered email. It ensures users can retrieve access if they forget their login credentials.

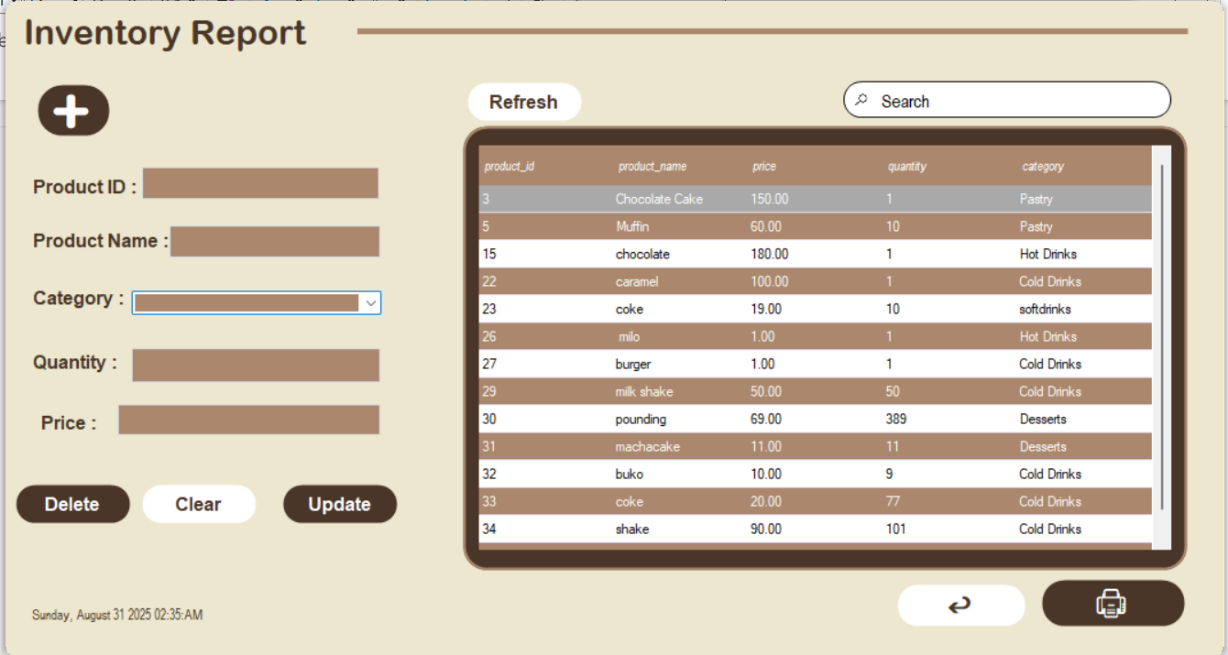
**Dashboard Form**

The Dashboard provides a centralized and organized interface that allows users to efficiently navigate through the system’s main functions. It’s clear and intuitive layout ensures a smooth and user-friendly experience, enabling staff to manage tasks effectively.

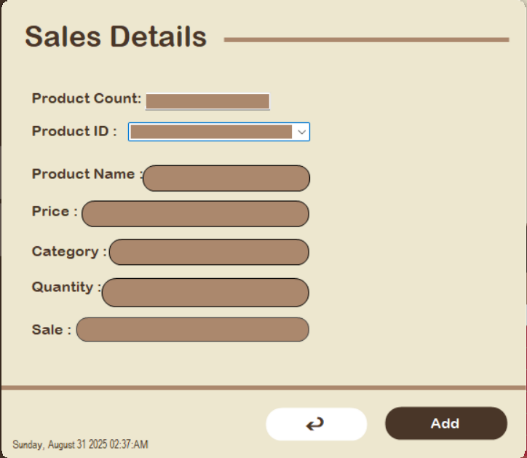
**Product Form**

.

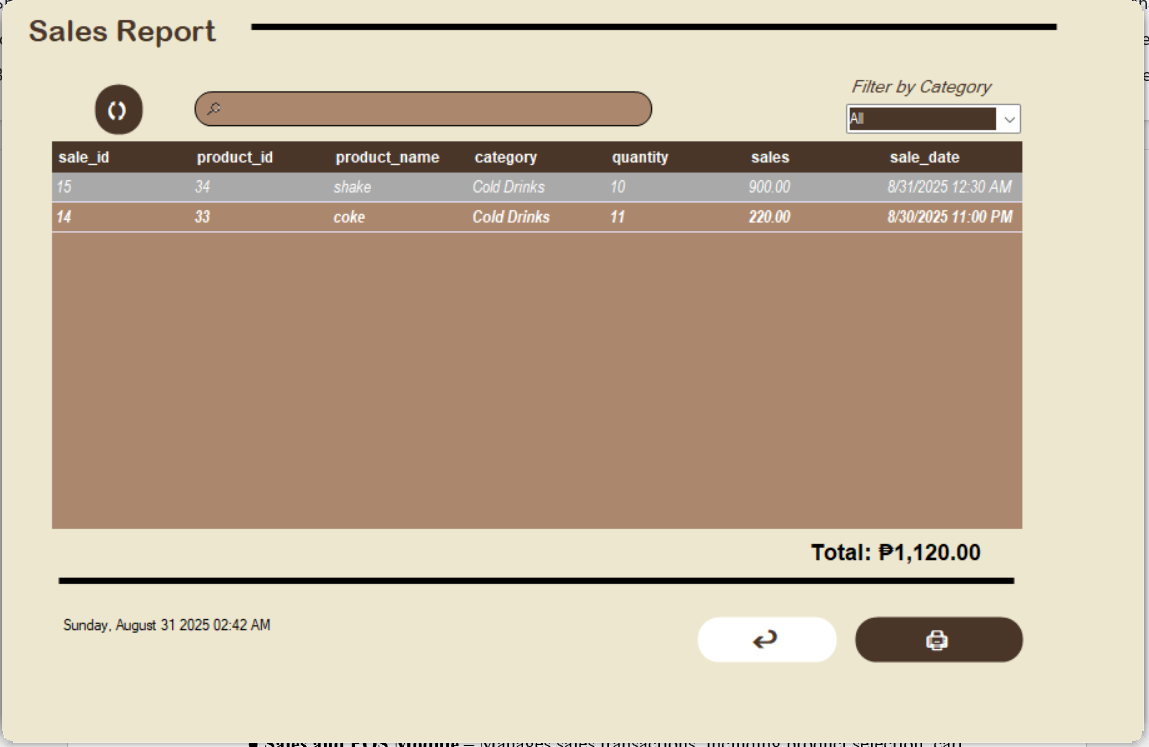
The Add Product form offers a simple and user-friendly interface that allows staff to easily enter and register new products into the system. Its clear layout ensures that users can perform the task quickly and accurately, supporting smooth inventory management.

**Inventory Form**

This is an inventory management interface that allows users to add, update, delete, and search products. It displays product details such as ID, name, price, quantity, and category in a structured table.

**Sales form**

The Stock form features a clean and organized interface that allows staff to manage inventory effectively. Its straightforward layout enables users to easily monitor and update stock, ensuring accurate records and smooth inventory operations.

**Sales Report Form**

The Sales Report form offers a simple interface for viewing, filtering, and summarizing transactions, helping staff and administrators track sales accurately and efficiently.

**Interface Elements**

**Login Form**

* TextBoxes: txtUsername, txtPassword
* Buttons: btnLogin, btnExit
* Labels: system title, error messages
* Password masking: applied to txtPassword

**Main Dashboard Form**

* MenuStrip or Navigation Buttons: btnSales, btnInventory, btnReports, btnCategories, btnUsers, btnLogout
* Labels & Icons: quick summary of total sales, total products, and low-stock alerts

**Product Form**

* + - DataGridView (dgvSales): displays recorded
    - ComboBoxes: select product and quantity
    - TextBoxes: subtotal, total amount
    - Buttons: btnAddSale

**Inventory Form**

* DataGridView (dgvInventory): displays products with stock levels
* ComboBox (cmbCategory): filter by product category
* TextBoxes: product name, price, quantity
* Buttons: btnAddProduct, btnEditProduct, btnDeleteProduct,
* Labels: product details

**Sales Form**

* DataGridView (dgvCategories): lists product categories
* TextBoxes: category name, description
* Buttons: btnAddCategory, btnEditCategory, btnDeleteCategory
* Labels: field indicators

**Sales Reports Form**

* DataGridView: displays sales and inventory reports in tabular format
* Export Buttons: btnPrint, btnExportExcel, btnExportPDF

**Interface Navigation**

* Login Form → Main Dashboard Form (after successful login)
* Main Dashboard → Sales Form (manage transactions, linked to Sales & Sale\_Items tables)
* Main Dashboard → Inventory Form (manage stock, linked to Products & Inventory tables)
* Main Dashboard → Categories Form (manage product categories, linked to Categories table)
* Main Dashboard → Reports Form (generate sales/inventory reports,)
* Main Dashboard → Logout (return to Login Form)

**Design Principles for Sales & Inventory System**

1. Consistency

* Use the same color scheme and font across all forms (Login, Sales, Inventory, Reports, Categories, Users).
* Buttons like Add / Edit / Delete / Save / Cancel should have the same size, style, and color across all modules.
* Align input fields (Labels + TextBoxes) uniformly — e.g., in Inventory Form and Sales Form, labels are left-aligned, inputs are right beside them.
* Naming conventions should be standard: always use “Save” instead of mixing “Submit” or “Add.”

2. Simplicity & Clarity

* Keep each form focused:
* Sales Form → Only shows product selection, quantity, total.
* Inventory Form → Only product details (name, price, stock).
* Avoid clutter by hiding advanced options until needed (e.g., reports filters).
* Labels should be descriptive: “Product Price” instead of just “Price”.

3. Feedback

* Product added successfully (Inventory)
* Sale transaction recorded (Sales)
* Use Error Provider for invalid inputs (e.g., entering text instead of numbers in price).
* Red borders or highlighted fields for missing/invalid entries.
* Confirmation dialogs: “Are you sure you want to delete this sale.

4. Visibility of System Status

* Display a Loading indicator or status message when fetching large reports.
* Dashboard Form should update totals in real time:
* Total Products
* Total Sales
* Low Stock Alerts
* Show current logged-in user and role at the top (e.g., “Welcome, Admin”).

5. User Control & Freedom

* Provide Cancel buttons on every form to clear/reset fields.
* Confirmation dialogs for destructive actions: deleting sales, products, or categories.
* Allow users to log out anytime and return to the Login Form.

6. Accessibility

* Support keyboard shortcuts:
  + Enter = Login
  + Esc = Cancel / Close Form
  + Ensure text has good contrast (black text on white or light background).
  + Password masking in the Login Form for security.

7. Flexibility & Efficiency

* Provide search and filters:
  + Inventory Form → search by product name or category.
  + Sales Form → filter sales
* Export options in Reports Form: Excel, Print.

8. Error Prevention & Recovery

* Validate inputs before saving:
  + Product price and stock must be numeric.
  + No empty product name allowed.
* If database connection fails, show clear error:
  + “Cannot connect to database. Please check server settings.”
* Always require confirmation before updating or deleting records.

**COMPONENT DESIGN**

The Sales and Inventory Management System is composed of several key components that work together to make inventory tracking, sales recording, and reporting accurate and efficient. Each component has its own role, but when combined, they provide a smooth and reliable experience for both staff and administrators. By keeping the system structured and straightforward, users can easily manage products, monitor stock levels, record sales, and generate reports.

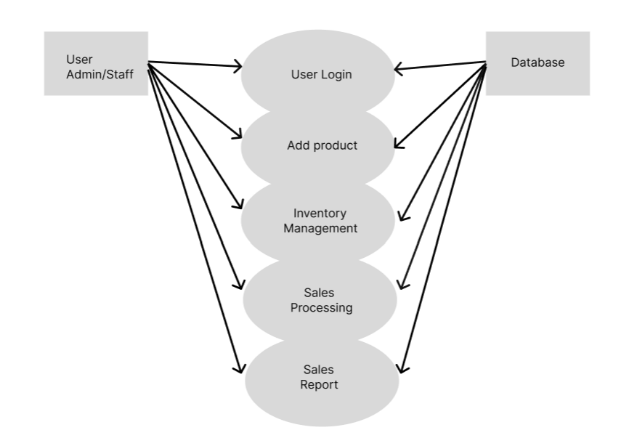
**Key Components and Modules**

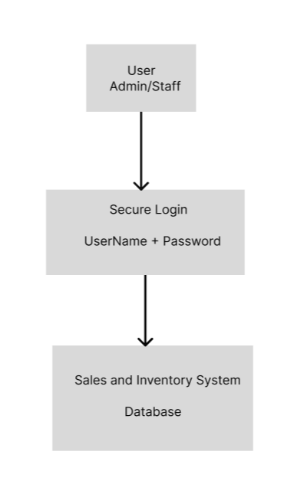
* User Interface (UI) Component – Provides staff and administrators with a clean and easy-to-use interface through forms, buttons, menus, and dashboards. It enables users to quickly access different functions such as sales, inventory, reports, and user management.
* Inventory Management Component – Handles the recording, updating, and monitoring of product details, including product name, category, price, and stock quantity. It ensures accurate tracking of available items and helps prevent stock shortages or overstocking.
* Sales Management Component – Manages sales transactions by recording customer purchases, calculating totals, and updating stock levels in real-time. This ensures that sales are accurately logged and inventory remains up to date.
* Report and Analytics Component – Generates tabular reports and visual charts for both sales and inventory. Users can filter by date ranges, categories, or products, and export data to Excel for better decision-making and performance tracking.
* Database Component – Stores and secures all system data, including product records, sales transactions, user accounts, and reports. It ensures data consistency, reliability, and accessibility across the entire system.

**Dependency Management**

The architecture of the Sales and Inventory Management System is designed with modular components that communicate primarily through database operations and well-defined functions, ensuring a clear separation of features and promoting flexibility. Each module — such as inventory management, sales transactions, reporting, and user accounts — can be updated, enhanced, or scaled independently without affecting the rest of the system.

The system supports scalability, making it capable of handling more users, products, and sales records as the business grows. A centralized database is implemented as the single source of truth, storing all product, sales, and user information in one secure location. This centralized approach guarantees consistency, synchronization, and accuracy of data across all forms and modules. It reduces redundancy and prevents errors in stock levels, sales calculations, and report generation.

**DATA FLOW DIAGRAM**

**SECURITY DESIGN**

The system is designed with strict security measures to ensure the confidentiality, integrity, and availability of all financial and inventory records it processes. This includes sensitive information such as sales transactions, expense entries, budget records, and stock data. Security practices are based on the principles of least-privilege access, ensuring that only authorized personnel can modify or view critical data. Accountability is enforced through audit trails and activity logs that record key user actions, providing transparency and traceability. Preventive measures such as input validation, database access restrictions, and secure authentication help protect against unauthorized access and data corruption.

**SYSTEM DESIGN**

The system ensures data security by using parameterized queries and input validation to prevent SQL injection and accidental errors. Critical operations, such as adding, updating, or deleting records, are monitored through simple audit logs. Regular database backups are recommended to avoid data loss, while error messages are kept minimal to prevent exposing sensitive technical details. Since the system is designed for single-computer use, additional protection relies on maintaining strong device security through passwords and updated antivirus software.

**Authentication and Authorization Mechanisms**

As a single-user system, the design does not include multi-user logins or role-based permissions. Instead, security focuses on data validation and safe database access through parameterized queries. Unauthorized access is minimized by restricting direct database exposure. Since the system is installed on a single device, device-level security measures such as strong passwords, limited access to the computer, and updated antivirus serve as the primary layer of protection.

**Data Encryption and Protection Measures**

The system does not implement advanced data encryption but protects records through secure data handling practices. All inputs are validated to prevent invalid or malicious entries, and transactions are processed with parameterized queries to safeguard against attacks. To further protect financial and inventory data, regular backups are encouraged, ensuring data can be recovered in case of corruption or hardware failure. In addition, securing the host device with strong credentials, antivirus software, and controlled access ensures that sensitive records remain safe and reliable**.**

**PERFORMANCE DESIGN**

The Sales Inventory System is designed to deliver fast, reliable, and responsive performance, ensuring that users can efficiently manage products, sales, and inventory records. The system optimizes database queries to minimize response times and reduce unnecessary data loads. To maintain smooth operation, indexing is applied to frequently accessed records such as product IDs, sale dates, and categories, supporting efficient retrieval of sales and inventory data. The modular design prevents performance bottlenecks by allowing separate components such as sales management, product inventory, and reporting to operate independently. Scalability is also considered, allowing the system to handle a growing number of transactions and products without compromising speed or reliability.

To maintain high performance, several strategies are implemented at the database level. Indexing on key fields improves search and filter operations, while selective denormalization and archiving of outdated sales records reduce the load on transactional tables and enhance reporting efficiency. Queries are carefully structured to retrieve only the necessary data, and stored procedures or caching mechanisms are used to avoid repetitive computations.

From a coding perspective, the system employs modular and reusable functions to avoid duplication and improve maintainability. Unnecessary loops and nested conditions are eliminated to enhance execution speed, and asynchronous operations are applied where possible to allow tasks such as report generation or inventory updates to run without delaying user interaction. Resource management practices, including connection pooling, proper disposal of unused objects, and optimized file handling, ensure minimal memory usage and maintain system stability.

The user interface is designed for optimal performance, featuring pagination for large datasets so that records load gradually rather than all at once. Search and filter capabilities allow users to quickly locate products, sales records, or inventory items, and progress indicators are provided for longer tasks to keep the user informed. Scalability is further supported by a modular system architecture, enabling individual components such as the sales module, inventory module, or reporting tools to be updated or scaled independently.

**ERROR HANDLING AND LOGGING**

The Sales Inventory System implements structured error handling and robust logging practices to ensure reliability, security, and transparency. Errors are detected, caught, and managed to prevent system crashes or interruptions in core functions such as sales processing, inventory updates, product management, and user authentication. When an exception occurs, the system logs the event with detailed technical information while presenting the user with a simple, clear, and non-technical message that guides them on how to proceed.

Critical operations, including login attempts, sales recording, inventory updates, and database transactions, are equipped with safeguards such as validation checks, retry mechanisms, and fallback procedures to minimize disruption. For example, if a sales record fails to save due to a database timeout, the system will automatically retry the operation, log the incident, and notify the user appropriately.

Logging is tightly integrated with error handling to ensure that every error or significant system event is properly recorded. System logs include timestamps, error codes, severity levels (Info, Warning, Error, Critical), and contextual details to facilitate troubleshooting. Logs are securely stored in the database and periodically archived to support performance monitoring, incident response, and long-term analysis. Sensitive information, such as user credentials or financial details, is never exposed in logs, ensuring compliance with privacy and security standards.

**Error Codes and Messages**

* E101 – Invalid Input: Triggered when the user enters incomplete or incorrect information in fields such as product details, sales entries, or inventory updates. The system will prompt the user to review and correct the highlighted fields.
* E201 – Authentication Failed: Occurs when login credentials are invalid, expired, or when an unauthorized user attempts to access the system. The system will display a clear message instructing the user to re-enter credentials or contact the administrator if access issues persist.
* E301 – Transaction Conflict: Raised when a sales transaction or inventory update conflicts with existing records, such as duplicate sales entries or insufficient stock. The system will guide the user to review and correct the entry before proceeding.
* E401 – Database Error: Generated when there is a failure in retrieving, saving, or updating critical data such as product information, sales records, or inventory quantities. The system will attempt a retry, and if unsuccessful, notify the user to try again later while logging the error for administrator review.
* E501 – System Error: Indicates an unexpected failure in the application, such as a server crash or unhandled exception. The user will be advised to retry the action, and if the issue persists, contact technical support. The system will log all relevant details for immediate troubleshooting.

**THIRD-PARTY INTEGRATIONS**

The Sales Inventory System utilizes several third-party tools and frameworks to enhance functionality, usability, and scalability

* **SQL Server (Database):** Serves as the main backend storage, managing structured data such as products, sales transactions, inventory records, and reports. Although a Microsoft product, it is considered an external dependency since the system relies on it for data persistence.
* **Guna UI2 Framework:** Provides modern and customizable UI components (buttons, panels, textboxes, and grids) that improve the appearance and usability of the application. This third-party library replaces standard Windows Forms controls with visually appealing alternatives, enhancing user experience without changing the core logic of the system.
* **NET Framework (WinForms):** The platform that runs the application, offering essential libraries, event handling, and system-level communication. While not strictly third-party, it is an external dependency that must be installed on client machines.

**DEPLOYMENT PLAN**

The deployment of the Kapeña Café Sales and Inventory System follows a structured and step-by-step process to ensure smooth installation, reliable performance, and maintainability within the café environment

1. Ensure the client machines or server have the necessary software installed, including:
   * + Microsoft Windows 10/11 (64-bit preferred)
     + Microsoft Visual Basic 2010 Redistributable
     + SQL Server with SQL Server Management Studio (SSMS)
     + Additional libraries for reporting and barcode generation (e.g., iTextSharp, EPPlus, Guna.UI2.WinForms)
2. Database Preparation:

* Set up the SQL Server database by creating the DBSalesInventory database.
* Initialize all required tables, relationships, and stored data, including sales, inventory, and product records.
* Configure stored procedures or initial data imports as needed.

1. Application Installation:

* Deploy the compiled VB.NET desktop application (.exe) along with all required DLLs to the designated client computers or server.
* Ensure all files are placed in a secure directory with proper permissions.

1. System Configuration:

* Configure the database connection by updating the connection string in the App.config file (e.g., " "Server=DESKTOP-060V3M5\SQLEXPRESS; Database=DBSalesInventory; Trusted\_Connection=True;").
* Ensure the application can successfully connect to the SQL Server database and perform CRUD operations.

1. Initial Testing:

* Conduct thorough testing of all modules, including:
  + Sales entry and transaction recording
  + Inventory management and stock updates
  + Reporting and analytics
* Confirm that all functionalities operate correctly and data is properly stored in the database.

1. Official Deployment:

* Once testing is complete and verified, release the system for daily café operations.
* Provide user training and documentation as needed.

**Hardware Requirements (Minimum)**

* Processor: Intel i3 or higher
* RAM: 4 GB (8 GB recommended for smoother performance)
* Storage: 250 GB minimum (includes space for the database and system files)
* Display: 1366x768 resolution or higher

**Software Requirements**

* Operating System: Microsoft Windows 10 / 11
* Database: SQL Server with SSMS installed
* Frameworks: Microsoft Visual Basic 2010 Redistributable, .NET Framework (as required by VB.NET version)
* Libraries: iTextSharp, EPPlus, Guna.UI2.WinForms
* Print Support: Windows Print Services for printing reports

**Configuration Management and Version Control Procedures**

To maintain system stability and facilitate updates:

* Version Control: All source code is stored in a dedicated Git repository. Updates are documented with detailed commit messages describing bug fixes, new features, or UI changes.
* Configuration Management:
  + The database connection string and system settings are stored in configuration files rather than hard-coded.
  + Separate configurations are maintained for development, testing, and production environments.
* Release Management: Each deployment package is labeled with a version number (e.g., v1.0.0, v1.1.0) for easy tracking of updates.
* Backup and Recovery:
  + Prior to deployment or system updates, the existing database and application files are backed up.
  + A rollback plan is maintained to revert to the previous stable version in case of errors during deployment.

**MAINTENANCE AND SUPPORT**

The Kapeña Café Sales and Inventory System follows structured guidelines for maintenance and support to ensure reliability, stability, and continuous operational efficiency. Regular monitoring is conducted to assess system performance, security, and data integrity. Preventive maintenance activities, such as database optimization, inventory audits, and server health checks, are performed routinely to avoid performance degradation or unexpected downtime.

**Procedures for Handling Software Updates, Patches, and Bug Fixes**

* Minor Updates and Patches: Small releases, such as UI improvements, report fixes, or query optimizations, are deployed as needed using signed installation packages (e.g., ClickOnce). Each patch is developed from a dedicated branch, tested on a staging environment with a recent database backup, and deployed during off-peak hours to minimize disruption. Any required database changes (SQL scripts) are versioned, idempotent, and include a rollback plan to restore previous states if needed.
* Major Updates: Planned updates, typically quarterly or as required, may introduce new features such as enhanced reporting, additional inventory management tools, or updated sales modules. These updates follow a structured lifecycle: design → development branch → code review → QA on staging → User Acceptance Testing (UAT) → scheduled production deployment. Database migrations are handled with versioned SQL scripts (e.g., V2025\_08\_31\_AddIndexes.sql) and are tested for both forward and rollback scenarios.

**SUPPORT FRAMEWORK**

● **Helpdesk Support** – A dedicated support channel will be available to handle inquiries and issues reported by users. Response times will be categorized by severity, with urgent issues addressed immediately.  
● **Documentation and Training** – Updated manuals, user guides, and technical documentation will accompany each release. Training sessions may be provided for staff to ensure proper use of new system features.  
● **Backup and Recovery Plan** – Regular database backups will be maintained, with defined recovery steps to restore services quickly in the event of unexpected failures or data loss.  
● **Security and Compliance** – Ongoing security monitoring, vulnerability testing, and compliance checks will be performed to safeguard sensitive data and ensure adherence to regulatory requirements.

**REVISION HITORY**

This section documents all changes made to the design document over time. Each revision entry includes the version number, date of update, and a summary of modifications. Maintaining this log ensures proper tracking of progress, accountability, and clarity in the evolution of the system’s design.

|  |  |  |
| --- | --- | --- |
| Version | Date | Change Mode |
| 1.0 | 8/17/2025 | Prepared the first draft of the document. |
| 1.1 | 8/18/2025 | Added the system requirements and use case details. |
| 1.2 | 8/19/2025 | Included data flow diagrams and outlined the security setup. |
| 1.3 | 8/20/2025 | Updated the database design and revised the queue process. |
| 1.4 | 8/22/2025 | Adjusted the design of the interface added some reporting |
| 1.5 | 8/24/2025 | Made changes to diagrams; improved parts of the layout. |
| 1.6 | 8/26/2025 | Added the testing plan, corrected minor inconsistencies, and finalized revisions. | |

**APPENDIX**

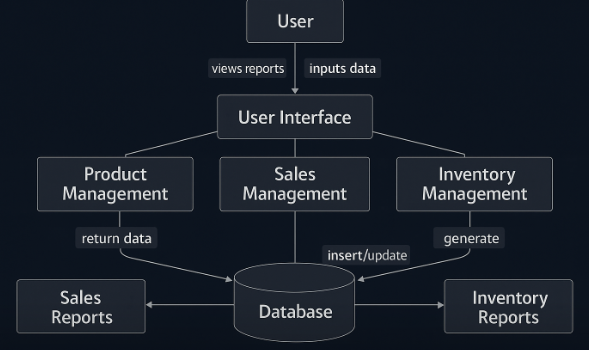
This section presents the screenshots of the developed system, showcasing its user interface and main functionalities. Each figure highlights the design and interaction flow of the Sales and Inventory System for Kapeña Café.

## **Appendix A - Supporting Diagrams**

**A.1 Entity Relationship Diagram (ERD)**

****

*Figure A.1: Complete ERD of the database schema*

**A.2 Data Flow Diagram (DFD)**

*Figure A.2: Data Flow Diagram of the system*