A Technical Documentation 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:

Lorenzana,Joshua

**INTRODUCTION**

Purpose of the Document

This document explains the testing done on the Kape Kalinaw Order Management System (OMS). It shows the test cases, methods, results, bugs found, and feedback from users. The goal is to make sure the system works correctly and meets the needs of its users. It can also be used later for fixing, updating, or improving the system.

Overview of the Software System

The Kape Kalinaw OMS is a desktop program made with VB.NET 2010 to make coffee shop operations easier. It helps with taking orders, billing customers, managing user roles for Admins and Cashiers, showing real-time dashboards and reports, and processing transactions automatically. The system makes orders faster, reduces mistakes, and keeps accurate records for the café.

Scope of the Technical Documentation

This document focuses on testing the OMS, including:

Checking all main features like selecting products, billing, processing orders, and reports.

Testing the user interface to make sure both Admins and Cashiers can work smoothly.

Doing User Acceptance Testing (UAT) to see how real users use the system and what feedback they give.

Reporting bugs, tracking issues, and giving recommendations for improvement.

This document does **not** cover installing the system, database design, or the detailed programming code. It only focuses on checking that the system works as expected and meets user needs.

**SYSTEM OVERVIEW**

**System Architecture**

The Kape Kalinaw Order Management System (OMS) is a desktop application built using VB.NET 2010. It follows a client-side architecture where all the main program logic runs on the user’s computer, and it connects to a central SQL Server database to store orders, products, and user data.

High-Level Components and Interactions  
The system has three main components:

1. User Interface (UI) - This is what the user sees and interacts with. It includes product selection screens, billing panels, dashboards, and reports.
2. Business Logic - This part handles the main functions, such as calculating totals, processing orders, updating stock, and applying role-based access rules.
3. Database - A SQL Server database stores all data, including product information, transactions, users, and stock levels.

The UI communicates with the business logic, which in turn interacts with the database to retrieve or save information. For example, when a cashier adds a product to an order, the UI sends the selection to the business logic, which calculates the price and updates the database.

**Deployment Architecture**  
The system is installed on each coffee shop’s desktop computers. Each computer connects to the central SQL Server database over the local network. Admin users can manage products, view dashboards, and run reports, while Cashiers can process orders and payments. All data is stored centrally in the database to keep records consistent and secure.

**INSTALLATION GUIDE**

System Requirements

Hardware:

* Processor: Intel Core i5 or higher
* RAM: 8 GB minimum
* Hard Disk: 500 GB free space
* Display: 1024x768 resolution or higher

Software:

* Windows 10 or later
* Microsoft .NET Framework 4.0 or higher
* SQL Server (any version compatible with the system)

Dependencies:

* Microsoft Visual Basic 2010 Runtime
* Access to a local or network SQL Server database

Step-by-Step Installation Instructions

1. Install .NET Framework: Make sure your computer has .NET Framework 4.0 or higher installed.
2. Install SQL Server: Set up SQL Server on your computer or network to store system data.
3. Copy the OMS Program Files: Transfer the Kape Kalinaw OMS folder to your preferred location on the computer.
4. Run the Setup or Executable: Open the main executable file (KapeKalinawOMS.exe) to start the program.
5. Database Connection: When prompted, enter the SQL Server name, database name, and login credentials. Test the connection to ensure it works.
6. First Login: Use the default Admin account (provided in the system manual) to log in and start configuring products, users, and other settings.

Configuration Settings and Options

* User Accounts: Create Admin and Cashier accounts with role-based access.
* Products: Add products, set prices, and define sizes or variations.
* Dashboard: Configure which data to display (orders, revenue, stock levels).
* Reports: Set default report formats and date ranges for viewing transaction history.

**CONFIGURATION GUIDE**

**Configuring the Software**

**Database Setup:** Open the configuration file. Enter your SQL Server name, database name, username, and password.

Save the file and test the connection from the program.

**User Accounts:**

Create Admin and Cashier accounts from the “User Management” menu.

Assign roles carefully: Admin has full access, Cashier can only handle orders and payments.

**Products and Prices:**

Go to the “Products” or “Menu Management” section.

Add new products, set prices, sizes, and categories.

Save changes so the system displays updated products immediately.

**Dashboard Settings:**

Configure which data is displayed on the dashboard (e.g., total sales, active users, product stock).

Adjust charts and reports to show relevant time periods (daily, weekly).**Reports Configuration:**

Set default formats for invoices and sales reports.Choose date ranges or filters for transaction history and charts.

Configuration File Formats and Parameters

ServerName = [Your SQL Server Name]

DatabaseName = [Your Database Name]

UserName = [SQL User]

Password = [SQL Password]

Best Practices for Customization

* Always back up the configuration file before making changes.
* Limit Admin access to trusted personnel to avoid accidental changes.
* Test any changes in a safe environment before applying them live.
* Keep default templates for reports and dashboards to restore if needed.

**DATABASE DOCUMENTATION**

Entity-Relationship Diagram (ERD):

The database for Kape Kalinaw OMS stores all information related to products, orders, users, and transactions. The ERD shows how tables are connected: for example, an Orders table links to the Products table to record what was sold, and to the Users table to know who processed the order.

Database Tables and Fields:

Users: stores user accounts (UserID, Username, Password, Role).

Products: stores products available in the café (ProductID, ProductName, Price, Category).

Orders: stores customer orders (OrderID, UserID, OrderDate, TotalAmount).

OrderDetails: stores details of each order (OrderDetailID, OrderID, ProductID, Quantity, Size).

Relationships:

One User can have many Orders.One Order can have many OrderDetails.

Each OrderDetail corresponds to one Product.

Data Migration and Backup Procedures:

Migration: Export data from development/testing database and import into production using SQL scripts or backup files.

Backup: Regularly back up the database (daily or weekly) to prevent data loss. Store backups in a secure location.

**USER MANUAL**

Instructions for Using the Software

Launching the System

Double-click the Kape Kalinaw OMS icon on your desktop.

Enter your username and password, then click Login.

Logging Out

Click the Logout button on the dashboard to safely exit the system.

Switching Users

After logging out, a different user can log in with their own credentials.

User Interface Description & Navigation

1. Login Screen

Fields: Username, Password

Buttons: Login, Exit

2. Dashboard (Main Screen)

Shows real-time data:

Total orders

Total revenue

Products available

Active users

Navigation Panel: Quick links to Orders, Products, Reports

3. Orders Screen

Select products from the menu

Choose product size and quantity

Click Add to Order to build the customer’s list

Click Process Payment to complete the transaction

4. Products / Admin Module (Admin Only)

Add new products

Update prices

Delete discontinued items

Monitor product stock

5. Reports Screen

View order history by date, user, or product

See revenue trends over the last seven days with charts

Common Tasks & Workflows

Processing a Customer Order

Log in as Cashier.

Go to Orders Screen.

Select product, size, and quantity.

Add items to the order list.

Click Process Payment and confirm total amount.

System calculates change automatically.

Print receipt (if printer available).

Adding a New Product (Admin Only)

Log in as Admin.

Open Products Module.

Click Add Product.

Enter product name, size, price, and stock quantity.

Click Save.

Generating Sales Report

Go to Reports Screen.

Select date range or filter options.

View transaction history and revenue charts.

Export or print report if needed.

Viewing Dashboard Metrics

Check total sales, orders, and products at a glance.

Identify popular products or busy hours for better planning.

**TROUBLESHOOTING GUIDE**

Common Issues and Error Messages

1. Cannot connect to database - The system cannot open the database.
2. Invalid login - Username or password is incorrect.
3. Program crashes - The application stops working unexpectedly.
4. Orders not updating - New orders or changes are not showing in the bill or dashboard.
5. Images not showing - Product images fail to load.

Technical Support

If the problem continues, contact technical support:

* Email: support@kapekalinaw.com
* Phone: +63 912 345 6789
* Office Hours: Monday to Friday, 9:00 AM - 5:00 PM

**CODE DOCUMENTATION**

Code Structure and Organization

The Kape Kalinaw OMS is built in VB.NET 2010 as a desktop application.

The main form (Form3) handles:

Displaying product categories and cards.Managing billing and transactions.

Handling role-based access (Admin and Cashier).

Additional forms include:

Form4 - Manage products (add, update, delete).

frmReceipt - Shows the receipt after a transaction.

The system uses FlowLayoutPanels to organize product cards by category:

panelNonCoffee - Non-Coffee products

panelIcedCoffee - Iced Coffee products

panelHotCoffee - Hot Coffee products

A DataGridView (dgvBill) is used to display orders and calculate totals.

Inline Comments and Key Functions

LoadProductsFromDB() - Loads products from the database and creates cards dynamically.

AddProductCard() - Creates a product card with name, price, image, size buttons, quantity, and "Add to Billing" button.

ProductAdd\_Click() - Adds selected products from the card to the billing DataGridView and updates total price.

btnProcess\_Click() - Handles payment, updates stock, saves order to database, and shows receipt.

Clock Timer - Updates the current date and time on Label1 every second.

Search Function - Filters products in real-time based on search input.

Role-based Access - Admin can manage products and see the dashboard; Cashier focuses on processing orders only.

Coding Standards and Conventions

Naming:

Controls: Use meaningful names (e.g., btnProcess, panelHotCoffee, lblTotal).

Variables: Use camelCase for local variables (e.g., totalPrice, selectedSize).

Comments:

include inline comments for important logic and functions.

Sections are marked with clear headings for readability.

Formatting:

Indent blocks consistently.

Group related code together for clarity (e.g., all button click events in one section).

Error Handling:

Use Try...Catch blocks to handle exceptions when accessing the database or files.

Validate inputs before processing (e.g., quantity and payment amount).

UI Updates:

Use safe checks before updating controls (e.g., check Label1.IsDisposed before updating).

**TESTING DOCUMENTATION**

Objective:

* Ensure that the Kape Kalinaw OMS works correctly and efficiently for both Admin and Cashier users.
* Verify that all orders are accurately recorded, payments are calculated automatically, and transaction data is securely stored.
* Identify any errors or issues in the system and propose solutions.

Testing Strategies:

* Functional Testing: Check that each feature (login, adding products, billing, payment processing, reporting) works as expected.
* Non-Functional Testing: Test system performance, usability, and responsiveness.
* User Acceptance Testing (UAT): Confirm that end-users (staff) can operate the system easily and without errors.
* Regression Testing: Ensure that new updates or bug fixes do not break existing features.

**Summary of Test Results:**

* Core features such as billing, payment processing, and reporting function correctly.
* Minor bugs detected in login error handling and input validation.
* User acceptance testing confirmed that both Admin and Cashier users find the system easy to navigate.

**Conclusion:**  
The system is mostly stable and functional, with a few minor issues that need to be fixed before full deployment. Recommendations include performing regression tests after bug fixes and providing user training for smooth operation.

**11. Maintenance Guide**

Software Maintenance Procedures

To keep the Kape Kalinaw OMS running smoothly and up to date:

Regular Backups:

Backup the database daily to prevent data loss.

Save copies of the application files before applying updates.

Database Maintenance:

Check for data inconsistencies or errors weekly.

Optimize database performance by cleaning up old logs or unused records.

Application Updates:

Apply bug fixes or enhancements following a test plan.

Always test changes in a separate development environment before deploying to production.

User Account Management:

Review user roles periodically.

Remove inactive accounts to maintain security.

Version Control & Release Management

Version Numbering:

Follow the format: v[Major].[Minor].[Patch]

Major: Big updates or changes in functionality.

Minor: New features or enhancements.

Patch: Bug fixes or small improvements.

Release Management:

Maintain a release log detailing version number, changes, release date, and responsible personnel.

Test every release in a staging environment before deploying to live systems.

Bug Fixes & Enhancements

Reporting Bugs:

Users should report bugs via a standard form with the description, steps to reproduce, and screenshots if applicable.

Fixing Bugs:

Developers reproduce the issue in a test environment.

Apply the fix, test it, and update the release version.Deploy only after verifying that the fix does not affect other features.

Adding Enhancements:

Enhancements should be reviewed and approved by management.

Follow the same process: develop → test → document → deploy.

Documentation:

Update user manuals and internal code comments whenever new features or fixes are applied.

Maintain a changelog for reference.

12. Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Revision No. | Date | |  | | --- | |  |  |  | | --- | | Author | | Description of Change |
| 1.0 | 2025-10-11 | Joshua Lorenzana | Admin UI and Cashier Ui |
| 1.2 | 2025-10-13 | Joshua Lorenzana | Others Documentation and Updating the Documentation |
| 1.3 | 2025-10-19 | Joshua Lorenzana | Finalize All |

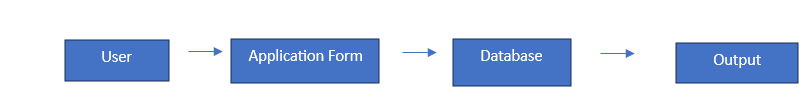
**APPROVAL**

| **Name** | **Position/Role** | **Signature** | **Date** |
| --- | --- | --- | --- |
| Joshua Lorenzana | Project Lead / Author |  |  |
| Joshua Lorenzana | Project Manager |  |  |
| Joshua Lorenzana | QA / Testing Lead |  |  |
| Kape Kalinaw | Client / End User |  |  |

**APPENDIX**

This section contains additional supporting materials, references, and diagrams that complement the main content of the Kape Kalinaw Order Management System (OMS) documentation.

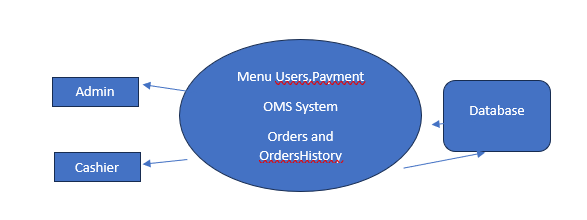
**System Diagrams**

****

**Database Schema Diagram** - Shows tables, relationships, and key fields for products, orders, and users.



**System Architecture Diagram** - Illustrates the structure of the OMS, including client interface, business logic, and database interactions.



**Workflow Diagram** - Visualizes the steps for order processing, payment handling, and reporting.

**Reference Materials**

VB.NET 2010 official documentation.

SQL Server management guides for database configuration.

User manuals and training guides for café staff.

**Glossary of Terms**

**OMS** - Order Management System

**Admin** - User role with full access to the system features

**Cashier** - User role focused on processing orders and payments

**Dashboard** - Visual interface displaying real-time sales and operational data