**A PROPOSED OFFERING OF A ORDER MANAGEMENT SYSTEM**

**FOR KAPE KALINAW COFFE SHOP**

A Requirement Specification 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:

Lorenzana,Joshua

**INTRODUCTION**

This document defines the requirements for the Order Management System (OMS) for Kape Kalinaw. The system is designed to make ordering and payment faster and easier for both cashier and admin. It reduces mistakes from manual record keeping and speeds up transactions. Admins can monitor real time revenue, total orders, users, and products sold, helping them make better business decisions. Only Admins can manage products, prices, order history, and user accounts, while Cashiers focus on fast and accurate sales transactions. Every order is automatically recorded with date, time, items, quantity, size, and payment. Overall, the OMS makes managing orders and revenue more efficient, secure, and transparent for Kape Kalinaw.

The Order Management System is a desktop-based application with two main user roles: Admin and Cashier. In the Admin interface, a dashboard displays revenue, total orders, total users, and total products sold. The Admin can add, update, or delete products with details such as type (coffee or non-coffee), size (hot or ice), price, and quantity. They can also update or delete past orders in the order history, and only the Admin has access to these management features. On the other hand, the Cashier interface focuses on sales transactions. The Cashier can view all available products and their details, select products, add them to the order summary, update quantities or remove items, and enter payment, where the system automatically calculates the change. The Cashier also has read-only access to the order history for reference. The system updates the dashboard metrics in real time whenever a new order is added or product details change, and it automatically records the date and time for each transaction. The interface is simple, clean, and easy to use to help streamline daily operations. Overall, the OMS provides a solution for more efficient order processing, accurate sales tracking, and secure role-based access for both Admin and Cashier.

**FUNCTIONAL REQUIREMENTS**

This section describes the specific functions the Kape Kalinaw Order Management System (OMS) must perform. Each feature ensures that both Admins and Cashiers can manage orders, products, and sales efficiently while maintaining accurate records.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | Requirement Description | Priority | Dependencies | Acceptance Criteria |
| |  | | --- | | FR-1 |  |  | | --- | |  | | The system must allow Admin login with a username and password. | High | |  | | --- | | User database |  |  | | --- | |  | | Admin can log in successfully with correct credentials. |
| |  | | --- | | FR-2 |  |  | | --- | |  | | The system must allow Cashier login with a username and password. | High | User database | Cashier can log in successfully with correct credentials. |
| FR-3 | Admin dashboard must display total revenue, number of orders, number of users, and products sold. | High | |  | | --- | |  |  |  | | --- | | OrderHistory  DB | | Dashboard displays accurate, real-time metrics. |
| |  | | --- | | FR-4 |  |  | | --- | |  | | Admin can add new products with details: name, type (coffee/non-coffee), size (hot/ice), price, and quantity. | High | Product DB | |  | | --- | | New product appears in the system and is available for Cashier. |  |  | | --- | |  | |
| |  | | --- | | FR-5 |  |  | | --- | |  | | Admin can update existing product details, including price, size, type, and quantity. | High | FR-4 | Changes are reflected immediately in Cashier interface. |
| |  | | --- | | FR-6 |  |  | | --- | |  | | |  | | --- | | Admin can delete products from the system. |  |  | | --- | |  | | |  | | --- | | High |  |  | | --- | |  | | |  | | --- | | FR-4 |  |  | | --- | |  | | Deleted products are no longer selectable by Cashiers. |
| FR-7 | |  | | --- | | Cashier can view the product menu with all details. |  |  | | --- | |  | | |  | | --- | | High |  |  | | --- | |  | | |  | | --- | |  |  |  | | --- | | Product DB | | Menu loads correctly with accurate product details. |
| FR-8 | |  | | --- | | Cashier can add selected products to an orders DataGrid |  |  | | --- | |  | | |  | | --- | | High |  |  | | --- | |  | | |  | | --- | | FR-7 |  |  | | --- | |  | | Order list updates accurately with selected products. |
| FR-9 | |  | | --- | |  |  |  | | --- | | The system automatically records the date and time for each order. | | |  | | --- | | High |  |  | | --- | |  | | FR-8 | |  | | --- | |  |  |  | | --- | | All orders have correct timestamp in history and dashboard. | |
| FR-10 | Cashier can update quantities or remove items in the order before finalizing. | Medium | |  | | --- | | FR-8 |  |  | | --- | |  | | Changes are accurately reflected in the DataGrid and total. |
| FR-11 | |  | | --- | |  |  |  | | --- | | Cashier can process payment and input the amount given by the customer. | | |  | | --- | | High | | |  | | --- | | FR-8 |  |  | | --- | |  | | System calculates change automatically and displays it. |
| FR-12 | |  | | --- | | Completed orders are saved to the order history automatically. |  |  | | --- | |  | | High | |  | | --- | | FR-11 |  |  | | --- | |  | | |  | | --- | | Orders are stored with all details: products, quantity, price, payment, and timestamp. |  |  | | --- | |  | |
| FR-13 | Admin can update or delete past orders from the order history. | Medium | |  | | --- | | FR-12 |  |  | | --- | |  | | Order history reflects changes immediately. |
| FR-14 | |  | | --- | | Cashiers cannot access Admin-only features. |  |  | | --- | |  | | |  | | --- | | High |  |  | | --- | |  | | |  | | --- | | User roles |  |  | | --- | |  | | Cashier interface only shows product menu and order entry options. |

**NON-FUNCTIONAL REQUIREMENTS**

These describe how the system should perform rather than what it does. They ensure the OMS is fast, reliable, secure, and easy to use.

**Performance**

The system should process order transactions in under 3 second. The system should handle by cashier adding or updating orders without noticeable delay.

**Usability**

The interface should be simple, clean, and easy to navigate for both Admin and Cashier.

**Reliability**

The system should be available at least 99% of business hours without crashes.

**Security**

Users must create passwords that include at least one uppercase letter, one lowercase letter, and one numeric or special character. Passwords must be protected in database to prevent unauthorized access. Usernames are case-sensitive and must contain both uppercase and lowercase letters.

**Scalability**

The database and application should be designed to easily accommodate an increasing number of products, users, and orders over time.

**Maintainability**

The code should be modular, documented, and easy to update or expand in the future. The system should include logging and error-handling mechanisms to help identify and troubleshoot issues efficiently.

**USE CASES**

**Admin Login**

* Actor: Admin
* Precondition: Admin account exists.
* Scenario: Admin opens the system → Enters username and password → System validates credentials → Admin dashboard opens.
* Postcondition: Admin can access management features such as product management, order history, and dashboard metrics.
* Alternate Flows:
  + Invalid username/password → System displays error message.

**Cashier Login**

* Actor: Cashier
* Precondition: Cashier account exists.
* Scenario: Cashier opens the system → Enters username and password → System validates credentials → Cashier interface opens.
* Postcondition: Cashier can access the menu and process sales transactions.
* Alternate Flows:
  + Invalid username/password → System displays error message.

**Product Management (Admin Only)**

* **Actor:** Admin
* **Precondition:** Admin is logged in.
* **Scenario:** Admin selects a product → Adds/updates/deletes product details (name, type, size, price, quantity) → System updates the database and dashboard metrics.
* **Postcondition:** Product information is updated in the system, and changes are reflected in the Cashier menu.
* **Alternate Flows:**
  + If required fields are missing → System shows validation error.

**Process Order (Cashier and Admin)**

* **Actor:** Cashier
* **Precondition:** Cashier is logged in and products exist in the database.
* **Scenario:** Cashier selects products → Adds to order DataGrid → Updates quantity or removes items → Enters payment → System calculates total and change → Order is saved with date/time stamp.
* **Postcondition:** Order is completed and recorded in order history.
* **Alternate Flows:**
  + Payment amount insufficient → System displays error.

**Update or Delete Past Orders (Admin Only)**

* **Actor:** Admin
* **Precondition:** Admin is logged in and order history exists.
* **Scenario:** Admin selects past order → Updates details or deletes the order → System updates order history and dashboard metrics.
* **Postcondition:** Order history reflects changes accurately.
* **Alternate Flows:**
  + Admin attempts to delete a non-existent order → System displays error.

**Dashboard Metrics (Admin Only)**

* **Actor:** Admin
* **Precondition:** Admin is logged in.
* **Scenario:** Admin opens dashboard → System displays real-time metrics (revenue, total orders, total products sold, number of users).
* **Postcondition:** Admin can monitor business performance

**View Menu (Cashier and Admin)**

* **Actor:** Cashier
* **Precondition:** Cashier is logged in.
* **Scenario:** Cashier opens menu → System displays products with details (type, size, price, quantity available).
* **Postcondition:** Cashier can select items for order processing.

**DATA REQUIREMENTS**

The Kape Kalinaw Order Management System manages all essential data using four core database tables: Users, Products, Orders, and OrderHistory. This data model consolidates related information for efficient processing while maintaining clear entity boundaries and relationships.

**Users Table**

The Users table stores authentication and security information for Admins and Cashiers who access the system. It includes the following columns:

* **id**: A unique numeric identifier assigned to each user. This serves as the primary key.
* **username**: The user’s login name, required for authentication. Must include both uppercase and lowercase letters for added security.
* **password**: The securely hashed password associated with the username, ensuring credentials are not stored in plaintext. Passwords must also include both uppercase and lowercase letters.
* **role**: Specifies whether the user is an Admin or Cashier, controlling access to system features.
* **status**: Indicates the current status of the user account (e.g., Active, Inactive), managing access control.

This centralized table governs all user authentication, password recovery, and access management functions, forming the foundation of system security. It ensures that only authorized Admins and Cashiers can access their respective interfaces, enforces strong username and password rules, and maintains an audit trail of user activity for accountability and transparency.

**ProductsTable**  
The Products table stores all details about the items available for sale in the coffee shop. It includes the following columns:

* Name: The name of the product (e.g., Cappuccino, Latte, Espresso).
* Size: Specifies the size of the product (e.g., 16oz, 22oz).
* Price: The selling price of the product.

This table centralizes product information, allowing the Admin to add, update, or delete products while ensuring Cashiers always see accurate menu data.

**Orders Table**  
The Orders table records all customer transactions. It includes the following columns:

* OrderID: A unique identifier for each order, serving as the primary key.
* Temperature: Indicates Hot or Ice for beverages.
* Type: Specifies whether the product is Coffee or Non-Coffee.
* OrderDate: The date of the order.
* OrderTime: The time the order was placed.
* GrandTotal: The total amount for the order.
* Payment: The amount paid by the customer.
* Change: The change returned to the customer.

This table tracks all transactions and provides data for reporting, revenue calculation, and order history management.

**Order History Table**  
The OrderHistory table records all completed orders in detail. It includes the following columns:

* OrderID: A unique identifier for each order, serving as the primary key.
* Category: Specifies the product category (e.g., Coffee, Non-Coffee).
* HotIced: Indicates whether the drink is Hot or Iced.
* Product: The name of the product purchased.
* Size: The size of the product (e.g., 16oz, 22oz).
* Quantity: The number of units purchased.
* Price: The price per unit of the product.
* Total: The total price for that line item (Quantity × Price).
* OrderDate: The date the order was placed.
* GrandTotal: The total amount for the entire order.
* Payment: The amount paid by the customer.
* Change: The amount of change returned to the customer.
* OrderTime: The exact time the order was placed.

This table centralizes all order details, allowing the Admin to monitor sales, track revenue, manage inventory insights, and maintain a complete audit trail of all transactions. It is also used to populate dashboards, generate reports, and support order history managemen

**Data Relationships and Integrity**

**Data Entities:**

* **Product:**  
  This refers to all the items that the cafe sells, such as various coffee types, pastries, and beverages. Each product is identified by a unique ProductID. Additional information recorded for each product includes the product name, the category it belongs to (e.g., Coffee or Non-Coffee), the size (Hot or Iced), the price per unit, and the current stock quantity available in the inventory. This information helps track what products are sold and what remains in stock at any point in time.
* **OrderHistory:**  
  This records every sale made in the cafe. For each order, important attributes include OrderID, the user who processed the order (UserID), category, whether the product is hot or iced (HotIced), the product name, size, quantity, price, total per line item, OrderDate, GrandTotal, payment amount, change, and OrderTime. These details are crucial for accurate sales tracking, inventory updates, and financial reporting.
* **User:**  
  This data entity contains information about the people who use the system. Each user is assigned a unique UserID along with a Username and Password for logging in. The user's role is also recorded (Admin or Cashier), which controls their access rights and permissions to various functions within the system.

**Relationships:**

* Each OrderHistory record can involve multiple products because customers may buy more than one item in a single transaction. This relationship allows the system to break down each transaction into individual items sold and update stock levels accordingly.
* **Users** have different roles assigned to them, which govern what operations they are authorized to perform in the system. For example, an Admin may have access to product management, order history modifications, and user management, while a Cashier is limited to recording sales and viewing order history. This ensures proper access control and system security.
* The system enforces referential integrity:
  + OrderHistory.UserID → Users.UserID ensures every order is linked to a valid user.
  + Products in OrderHistory must exist in the Product table (ProductID).

**ASSUMPTION AND CONSTRAINS**

This section outlines the key assumptions and constraints considered during the development of the Order Management System (OMS) for Kape Kalinaw. These factors help define the scope, guide realistic planning, and anticipate potential limitations affecting design, implementation, and deployment.

**Assumptions:**

* The OMS will operate on a local Windows-based system within Kape Kalinaw.
* Admin and Cashier accounts will be pr
* e created, with proper usernames and passwords.
* Cashiers will have basic computer literacy and access to the OMS interface.
* All products and pricing details will be maintained by Admin and kept up-to-date.
* Internet connectivity is not required for daily operation, but may be needed for future updates or remote backups.
* Each order will be processed individually, and orders are recorded in real-time with automatic timestamps.

**Constraints:**

* Only Admin users can access product management, price updates, and order history modification.
* Cashiers are restricted to adding, updating, and processing orders, with read-only access to order history.
* The system must function reliably within the hardware and software available at Kape Kalinaw, without additional infrastructure.
* Budget limitations restrict the OMS to a desktop application with a simple local database (SQL Server).
* The system is designed for single-branch operation, with multi-branch support considered only for future versions.
* Regulatory compliance for data security and user privacy must be maintained, including secure password storage and role-based access.

**GLOSSARY**

This section provides definitions of key terms used within the Kape Kalinaw Order Management System. It ensures clarity and consistency by explaining important concepts related to orders, products, users, payments, and system security. These definitions serve as a reference for both system users and developers to maintain a shared understanding of terminology.

* **Admin**: A user role with full access to the system. Admins can manage products, update prices, monitor revenue, view and edit order history, and manage user accounts.
* **Cashier**: A user role focused on sales transactions. Cashiers can select products, add or update orders in the DataGrid, process payments, and view order history in read-only mode.
* **Dashboard**: The main interface for Admin, showing real-time metrics such as total orders, revenue, total products sold, and total users.
* **Order**: A record of one or more products purchased by a customer, including quantity, size, category, price, total amount, date, and time.
* **Order History**: A log of all past transactions saved in the system. Admin can update or delete entries; Cashiers can view them in read-only mode.
* **Product**: Any item sold at Kape Kalinaw, including coffee, beverages, or snacks. Each product has attributes like name, category, size, price, and available stock.
* **DataGrid**: A VB.NET interface element used to display order items and order history in a tabular format for easy viewing and manipulation.
* **Revenue**: The total sales income generated over a specific period. Displayed in the Admin dashboard and updated automatically when orders are processed.
* **User Account**: A record containing login credentials, role (Admin or Cashier), and status. Secures access to the system based on role.
* **Payment**: The amount of money provided by the customer for an order. The system calculates and displays any change automatically.
* **Access Control**: A security feature ensuring that only authorized users can perform specific actions (e.g., Admin-only features).

**REVISION HISTORY**

This section documents all changes made to the Kape Kalinaw OMS design, requirements, and documentation throughout the project lifecycle. Each revision entry includes a unique version number, the date of the change, and a brief description of the modifications.

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 2025-08-26 | 1.0 | Prepared the first draft of the document. | Lorenzana.Joshua |
| 2025-8-31 | 1.1 | |  | | --- | |  |  |  | | --- | | Final review and formatting for Kape Kalinaw | | Lorenzana,Joshua |

**Table 1.** Revision History

**APPENDIX**

This section includes supplementary materials and supporting information for the Kape Kalinaw Order Management System:

* UI Mockups:
  + Admin Dashboard showing revenue, total orders, users, and products.
  + Cashier Order Form with product selection, DataGrid for current orders, and payment section.
  + Order History DataGrid for Admin review and management.
* Possible Future Enhancements:
  + Integration with inventory stock alerts.
  + Daily, weekly, or monthly automated sales report export.
  + Multi-branch support.
  + Support for promotions, discounts, and loyalty programs.
* References:
  + VB.NET
  + SQL Server relational database guides
  + Guna.UI2 controls for enhanced UI components