**SALES AND INVENTORY SYSTEM FOR KAPEÑA CAFE**

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:

Fuentes, Marlon T.

Gasalao, Khian A.

Gentile, Eljay J.

Villas, Lalaine

**INTRODUCTION**

The purpose of this document is to clearly define the functional and non-functional requirements for the development of a Sales and Inventory System customized for Kapeña Cafe. It is intended to serve as a comprehensive guide throughout the entire software development life cycle, ensuring that the system is designed and implemented according to the expectations of the client and the operational needs of the business. By establishing a clear foundation of requirements, this document supports the goal of building a reliable and efficient solution that addresses the specific challenges faced by Kapeña Cafe in managing its daily sales and inventory operations.

The Sales and Inventory System is envisioned as a desktop-based application that will automate and streamline the core processes of sales recording and inventory management within the café. Instead of relying on manual methods that are prone to delays and errors, the system will provide real-time sales tracking, automatic stock adjustments, and detailed reporting features. Through these capabilities, Kapeña Cafe will be able to maintain accurate records, reduce the workload on staff, and improve operational efficiency. In addition, the system is designed with usability in mind, offering a secure and user-friendly interface that ensures only authorized personnel can access its features.

This requirements specification defines the scope of the system by covering both functional and non-functional aspects. It includes essential features such as product management, inventory tracking, user authentication, and report generation. At the same time, it outlines important non-functional requirements including performance, security, and usability, which are necessary to guarantee that the system operates smoothly and reliably in a café setting. By documenting these requirements, the project ensures that the resulting Sales and Inventory System will meet the operational demands of Kapeña Cafe and provide long-term value to its daily business processes.

**FUNCTIONAL REQUIREMENTS**

This is the list of things the system needs to do. Each feature is described clearly so the developers and the cafe know what the system must be able to handle, like recording sales, tracking inventory, and managing products.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID** | **Requirement Description** | **Priority** | **Dependencies** | **Acceptance Criteria** |
| FR-01 | Record sales transactions in real-time and automatically update inventory levels. | High | None | All sales are accurately logged, and stock levels reflect changes immediately. |
| FR-02 | Track product inventory and notify users when stock is running low. | High | FR-01 | System generates timely alerts when inventory falls below predetermined thresholds. |
| FR-03 | Organize products by categories such as coffee, pastries, and beverages. | Medium | None | Users can view and filter products by category effortlessly. |
| FR-04 | Generate detailed reports on sales, stock movement, and inventory status. | Medium | FR-01, FR-02 | Reports contain accurate data and can be exported or printed as needed. |
| FR-05 | Support user authentication with role-based access control. | High | None | Access to sensitive functions is restricted based on user roles. |
| FR-06 | Allow manual adjustment of stock for incidents such as spoilage or returns. | Low | FR-02 | Users can manually update inventory, with the changes logged for audit. |

*Table 1: Functional Requirements*

**NON-FUNCTIONAL REQUIREMENTS**

This section is about how the system should work rather than what it does. It covers things like how fast it should be, how easy it is to use, how reliable it needs to be, and how safe and secure. Think of it as the qualities that make the system good to use.

**Performance**

The system shall process user inputs and transactions within 2 seconds and be capable of handling multiple concurrent operations without degradation in performance.

**Usability**

The system interface must be intuitive and simple enough for staff with minimal computer experience to operate effectively, requiring minimal training.

**Reliability**

The system must maintain 99% availability during business hours, with automatic daily backups to prevent data loss.

**Security**

The system shall process user inputs and transactions within 2 seconds and be capable of handling multiple concurrent operations without degradation in performance.

**Scalability**

Designed initially for a single branch, the system architecture should allow future scalability to support multi-branch operations if required.

**Maintainability**

The source code and database schema must be well-documented to facilitate easy maintenance, updates, and troubleshooting

**USES CASES**

This section shares practical stories about how different people will use the system in their daily tasks. It helps everyone from developers to cafe staff understand how the system is meant to work in real life and supports the workflow smoothly.

**Use Case 1: Record Sales Transaction**

* **Uses Case ID: UC - 01**
* **Use Case Name:** Record Sales Transaction
* **Description:** When a customer places an order, the cashier uses the system to enter the items and quantities requested. The system then calculates the total amount due, including any taxes or discounts. Once the customer pays, the system processes the payment and immediately updates the inventory, reducing the available stock for each sold product. This ensures that stock levels stay accurate and up to date without extra manual work.
* **Actors:** Cashier, Inventory System
* **Preconditions:** Before starting, the cashier must be logged into the system to have proper access. The products being sold should already be selected, and quantities specified for each item.
* **Postconditions:** After the transaction, the sale is fully recorded in the system, and the inventory stock reflects the updated quantities, ready for the next transaction.
* **Alternate Flows:** If payment is unsuccessful, the transaction is aborted and no data is saved.

**Use Case 2: Manage Inventory**

* **Uses Case ID: UC - 02**
* **Use Case Name:** Manage Inventory
* **Description:** The inventory manager uses the system to keep the stock information accurate. This can include adding new products, updating the quantities of existing products, or organizing items into categories for easier tracking. The system helps by providing forms and inputs for these updates and checks that the entered data is correct.
* **Actors:** Inventory Manager
* **Preconditions:** The inventory manager needs to log into the system first to gain access to the inventory management functions.
* **Post-conditions:** Once the updates are saved, the system ensures the inventory database is accurate, reflecting the latest stock quantities and product details.
* **Alternate Flows:** If the manager enters incorrect or incomplete data, the system will detect the errors and prompt the user to fix them before saving. This way, the inventory information stays clean and reliable.

**DATA REQUIREMENTS**

This section details the types of information that the system will manage to support the daily operations of Kapeña Cafe. It explains specifically what data will be stored, the important details attached to each type of data, and how they relate to and interact with one another within the system to enable smooth business processes.

**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 (for example, coffee or baked goods), 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.
* **Sales Transaction:** This records every sale made in the cafe. For each transaction, important attributes include the Date Time when the sale occurred, the Product ID(s) involved, the quantity of each product sold, and the total amount charged to the customer. These details are crucial for accurate sales tracking and financial reporting.
* **User:** This data entity contains information about the people who use the system. Each user is assigned a unique User ID along with a username and password for logging in. The user's role is also recorded, such as whether they are staff members or managers, which controls their access rights and permissions to various functions within the system.

**Relationship:**

* Each Sales Transaction can involve multiple products because customers may buy more than one item at a time. This relationship allows the system to handle and break down each transaction into the individual items sold and update stock levels accordingly.
* Users have different roles assigned to them, and these roles govern what operations they are authorized to perform in the system. For example, a manager may have access to inventory reports and user management, while a regular staff member may be limited to recording sales only. This ensures proper access control and security within the system.

**ASSUMPTIONS AND CONSTRAINTS**

This part of the document shares some important things we're counting on to be true while building and using the system, as well as the boundaries or limits that shape what the system can do. Knowing these helps everyone understand the environment around the project and manage expectations properly.

**Assumptions:**

* We assume that all staff members who will use the system have at least basic computer skills. This means they know how to use a keyboard and mouse and can navigate simple software interfaces without trouble. This allows us to design the system to be user-friendly but doesn't require highly technical knowledge.
* The system primarily operates locally on the cafe’s computer without requiring an internet connection. All sales and inventory data are stored directly on the local machine. Internet access is not required for daily operations but can be optionally used for manual data backup or software updates if desired.

**Constraints:**

* The system is designed to be used only at a single cafe location initially. It doesn't yet support managing multiple branches or locations. If the business expands, additional features will be needed to handle multi-location operations.
* Features like staff payroll processing, customer online orders, and delivery tracking aren't part of the current system. These kinds of functions will be considered in future phases but are out of scope for now to keep the project focused.
* Advanced sales forecasting tools, which predict future sales based on previous trends, won’t be included at this stage. Building these features needs more data and more complex analysis, which will be planned for later development.

**GLOSSARY**

This section serves as a small dictionary for the important terms used in the document, helping everyone understand exactly what the words mean so there is no confusion.

**Stock In -** This means adding new items or products into the cafe's inventory. For example, when fresh coffee beans or ingredients arrive at the cafe, they are recorded as "stock in" so the system knows how much is available.

**Stock Out -** This happens when products leave the inventory, usually because they were sold but sometimes because they spoiled or were damaged. The system keeps track of these removals to keep stock levels accurate.

**User Authentication -** This is the process where the system verifies who the user is before giving access. Typically, this means entering a username and password to make sure only authorized people can use the system.

**Real-time Processing -** This means the system updates information immediately when something happens. For example, as soon as a sale is made, the inventory changes right away so the staff always has the latest information.

**REVISION HISTORY**

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. | |

*Table 2: Revision History*

**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.

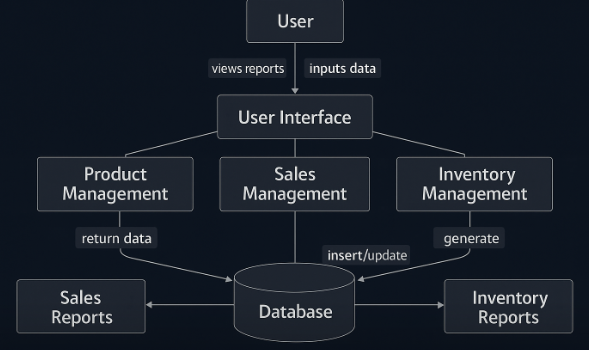
## **Appendix A - Supporting Diagrams**

**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*