Software Requirements Specification

for

Smart Inventory Purchases (SIP)

Version 1.0 Beta

Prepared by

Group 10:

|  |  |  |
| --- | --- | --- |
| David Cruz Beltrán | A01360416 | a01360416@tec.mx |
| Arturo Alfonso Gallardo Jasso | A01795510 | a01795510@tec.mx |
| Julio Cesar Mestas Delgado | A01794827 | a01794827@tec.mx |

|  |  |  |
| --- | --- | --- |
| Professors: | Dr. Alberto Aguilar González  Xóchitl Gabriela Margarita Zapata García | |
| Course: | Análisis, diseño y construcción de software | |
| Date: | | September 28th, 2024 |

Contents

Contents ii

Revisions iii

1 Introduction 1

1.1 Document Purpose 1

1.2 Product Scope 1

1.3 Intended Audience and Document Overview 1

1.4 Definitions, Acronyms and Abbreviations 2

1.5 Document Conventions 2

1.6 References and Acknowledgments 3

2 Overall Description 4

2.1 Product Overview 4

2.2 Product Functionality 4

2.3 Design and Implementation Constraints 5

2.4 Assumptions and Dependencies 5

3 Specific Requirements 7

3.1 External Interface Requirements 7

3.2 Functional Requirements 8

3.3 Use Case Model 9

4 Other Non-functional Requirements 11

4.1 Performance Requirements 11

4.2 Safety and Security Requirements 12

4.3 Software Quality Attributes 13

5 Other Requirements 15

5.1 Other Requirements 15

Appendix A – Data Dictionary 17

Appendix B - Group Log 18

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| Draft Type and Number | Full Name | Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded. | 00/00/00 |
| SRS 1.0 | Group 10 | Software Requirements Specifications. | 9/22/2024 |
| SRS 1.1 | Group 10 | SRS Updates. Added Use cases to SRS. | 9/28/2024 |
| SRS 1.2 | Group 10 | Update USE Case | 11/10/2024 |

# 

# Introduction

## Document Purpose

The purpose of this document is to outline the software requirements for a predictive inventory management system called “Smart Inventory Purchases”, designed to help a retail business to optimize their inventory management. This system leverages artificial intelligence to forecast demand based on past purchase trends, customer behavior and other variables.

This document serves to define application features, functionalities, restrictions and provides detailed description of the use cases intended for developers, stakeholders and users through the product lifecycle.

## Product Scope

The “Smart Inventory Purchases” system is meant to help a retail business to predict and recommend periodically, based on historical sales data, the products and quantities to purchase to minimize stock imbalances. This results in increased profitability, reduce overstocking, understocking and product turnover and improved cashflow for the company.

In its current version, the “Smart Inventory Purchases” system is not intended to be integrated with another existing software. The application operates as a stand-alone solution with its own user interface retrieving the data from a database or with an alternative option to manually load the data and create the recommendations for the purchase team to fill in its own system. Point-of-sale systems are out of the scope during this version.

## Intended Audience and Document Overview

This document is intended for the following audience:

* Clients.
* Developers.
* Professor.

The intended way to read this document is from beginning to end, as each section is designed to progressively answer any questions the reader may have.

For clients and the professor, it is recommended to start with an overview and scope to understand the system's purpose and goals. Developers should proceed to the detailed requirements and use case sections for guidance on the system's architecture, functionality, and performance expectations.

## Definitions, Acronyms and Abbreviations

**Definitions:**

* **Clients.** Retail business owners, managers and clerks, the users operating the software.
* **Developers.** The technical team responsible for designing, building, and implementing the system based on the outlined specifications.
* **Professor.** A Subject Matter Expert to guide the team to success.

**Acronyms:**

* **SIP.** Smart Inventory Purchases system.
* **UML.** Unified Modeling System.
* **COMET.**  Collaborative Object Modeling and Enterprise Tool.
* **RBAC.** Roled-based access control.
* **API.** Application Programming Interface.

## Document Conventions

**Formatting:**

* Font: Arial, size 11.
* Text: Single-spaced.
* Italics: Used for comments and notes.
* Headings: Follow the template for section and subsection titles.

**Naming Conventions:**

* Bold: Used for important technical terms and system components.
* Monospace font: For variables, code snippets, or configuration items.

**Highlighting:**

* Bold: For key points or definitions.
* Lists: Use bullets or numbering for clarity.

## References and Acknowledgments

Currently, there are no references to any external documents.

# Overall Description

## Product Overview

The Smart Inventory Purchases system is a new, self-contained product designed to help a retail business to optimize their purchasing strategies by predicting the most suitable products and quantities to stock. It utilizes historical sales data, machine learning algorithms, and user input to generate recommendations, aiming to improve inventory management, reduce unsold stock, and increase overall profitability. It is not part of an existing product family and operates as a standalone system with the potential for future integration into broader enterprise resource planning (ERP) systems.

The application will be primarily used by retail managers or business owners and clerks, who will input their historical sales data and receive actionable insights in return. The system will interact with an external data source, such as an existing inventory database, and provide a user-friendly interface for decision-making.

After analyzing the data, the application will show a detailed document specifying the products and quantities recommended for the purchasing department. Ultimately, it is the responsibility of the business to approve the recommendations before proceeding with the purchase process.

## Product Functionality

The SIP system offers several key functions that allow users to optimize their inventory purchases based on data analysis. The major functionalities include:

* **Account Management (System/User).** There are two level of user accounts to simplify operations: Operator (Employees), and Admin (Business Management).
* **Data Input Interface (User).** Users upload historical sales data in CSV or Excel format, which is then analyzed by the system.
* **Database Integration (System).** The system connects to external databases to retrieve or update inventory data.
* **Recommendation Generation (System).** After analysis, the system generates a document with product and quantity recommendations based on historical data and user preferences.
* **Download Interface (User).** Users can download the recommendation document in a specified format (PDF, Excel).
* **Feedback (System).** The system retrieves feedback from the users to improve reliability, this information can be delivered to update the model if necessary.

## Design and Implementation Constraints

The development of the SIP system will be guided by the following constraints:

* **Software Design Methodology.** The project will utilize the COMET method for iterative development and feedback.
* **Modeling Language.** UML (Unified Modeling Language) will be used for clear system design representation.
* **Hardware Limitations.** The application must operate effectively on standard retail hardware with limited processing power and memory. A system assessment is required to understand the capacity and communication with the Inventory system.
* **External Interfaces.** The system will require compatible APIs for integration with external databases.
* **Security Considerations.** Security protocols, including data encryption and secure authentication, will be implemented to protect sensitive information.
* **Programming Standards.** Development will adhere to established coding standards and documentation practices for maintainability.

## Assumptions and Dependencies

This section outlines key assumptions and dependencies that may impact the "Smart Inventory Purchases" system:

**Assumptions:**

* **External System Compatibility and Storage.** The system will integrate with the existing inventory system and the location of this inventory system has enough memory capacity to allocate all the data.
* **Machine Learning Accuracy.** It is assumed that the machine learning models will generate reliable recommendations based on accurate historical data.

**Dependencies:**

* **Development Tools.** Compatibility with specific programming languages and frameworks is necessary.
* **User Feedback.** Ongoing feedback will be essential for refining features during and after development.

# Specific Requirements

## External Interface Requirements

### User Interfaces

The SIP system offers a user-friendly interface with the following characteristics:

**Logical characteristics of user interface:**

* **Login/Account Creation.** Users must log in or create an account before accessing system features. Also requires a two-factor authenticator for added security. The interface is clear, offering easy access to login or account creation pages with minimal steps for account management.
* **Data Upload.** Users can upload historical sales data via a drag-and-drop interface, with real-time validation to ensure the file format is correct (CSV/Excel) before processing.
* **Data Analysis and Dashboards.** After data analysis, users can view the results in a dashboard format with visualizations, like graphs or charts, summarizing key insights for easy interpretation.
* **Recommendation Download.** Once analysis is complete, users can download a document containing product recommendations in a user-friendly format.

**Aspects of interface experience optimization:**

* **Screen Layout.** The system uses a structured layout with easy-to-navigate menus and clearly labeled buttons. Error messages are concise but informative, with the option to display detailed explanations for more advanced users.
* **User Actions.** Each user action, such as data upload, and report generation, is confirmed via notifications to ensure transparency and prevent errors.
* **Ease of Use.** Designed to be operable by retail managers with minimal training, the system should allow users to perform core functions (uploading data, downloading reports) within 10 minutes of first use.
* **Error Management.** Guidelines outline roles and responsibilities for error management, including how to review log files and follow troubleshooting instructions.

### Hardware Interfaces

The system will operate on standard retail hardware that meets basic specifications, including:

* **Device Support:** Compatible with desktops and laptops equipped with internet access.
* **Configuration Characteristics:** Support for standard input devices (keyboard, mouse) and output devices (monitor).
* **Protocol:** Communication with any required external storage devices or servers will follow standard protocols.
* No additional hardware interfaces related to point-of-sale systems will be necessary.

### Software Interfaces

The SIP system shall interact with Google Firebase for security and the Company Database.

* **Data Availability.** Users will have access to accurate historical sales data.
* **Internet Connectivity.** Stable internet is required for data access and updates.

## Functional Requirements

### Data Input. The system shall allow users to upload historical sales data in CSV or Excel format. The system shall validate the format and content of the uploaded data to ensure it meets the required specifications.

### Database connection. The system shall connect to external databases to retrieve and update inventory data.

### Recommendation Generation. The system shall analyze the uploaded historical sales data and generate a recommendation document that includes suggested products and quantities. The recommendations shall be based on historical data trends and user preferences.

### Download Interface. The system shall provide users with an option to download the generated recommendation document in multiple formats (e.g., PDF, Excel). The system shall ensure that the download process is intuitive and confirms successful completion to the user.

### Account Management. The system shall support two levels of user accounts: Operator (Employees) and Admin (Business Management). Each account type shall have different permissions and functionalities:

### Operators shall be able to upload data, view recommendations, and download documents.

### Admins shall have additional capabilities to manage user accounts, oversee system settings, and access comprehensive analytics.

### Input Validation. The system shall validate all uploaded data to ensure it meets predefined criteria.

### User Authentication. The system shall require all users to authenticate using secure credentials (username and password) before accessing any features. Passwords must be stored using encryption to prevent unauthorized access.

### Data encryption in transit and at rest. All sensitive data, including historical sales data, inventory levels, and user information, shall be encrypted using industry-standard protocols during transmission (via HTTPS) and when stored in the database.

### Access control and role-based permissions. The system shall implement role-based access control (RBAC) to limit access to specific features based on user roles (e.g., administrator, manager, employee). Only authorized users shall have access to sensitive operations such as changing system configurations or accessing sales data.

### Logging and monitoring. The system shall log all user actions (such as logins, data uploads, and recommendation downloads) and system events for security auditing purposes. Logs shall be stored securely and be accessible only to system administrators.

## Use Case Model

|  |  |
| --- | --- |
| **Title:** | Smart Inventory Purchases . |
| **Description:** | Log into system to analyze historical data to create a list of new purchases and then logout. If data drift, sends feedback. |
| **Actor:** | Users (Clerk and Manager), System, Database, Developer. |

A diagram of inventory purchase

Description automatically generated

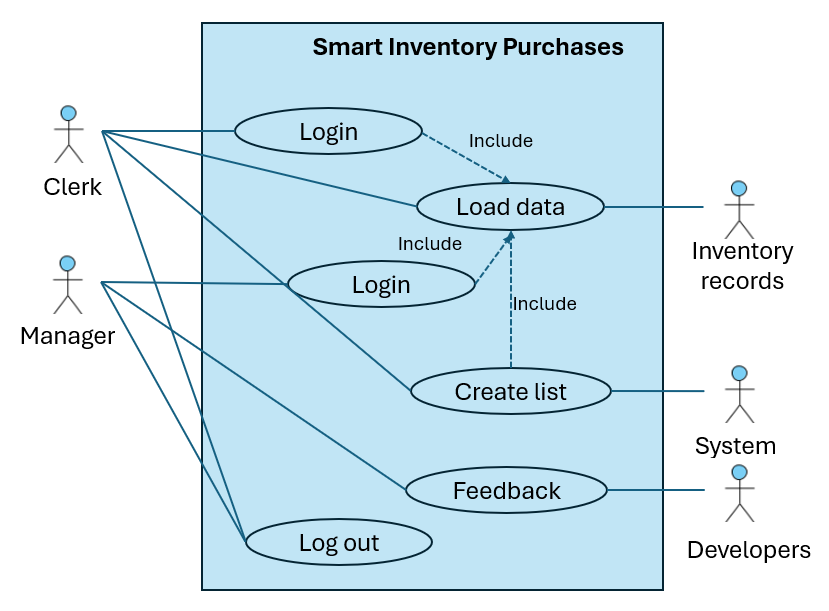


Figure 1. Use cases

### Login

|  |  |
| --- | --- |
| **Title:** | Log In |
| **Description:** | The system will let the users log in with their account. |
| **Actor:** | Users and System. |
| **Preconditions:** | The user must have an existing account with valid credentials. |
| **Postconditions:** | The user is granted access to the system. |
| **Basic Flow:** | 1. The user navigates to the login screen. 2. The user enters their username and password. 3. The system validates the username and password. 4. Access is granted to the system. |
| **Exemption Flow:** | **Invalid Username/Password:** The system displays an error message, and the user is prompted to re-enter their credentials. |

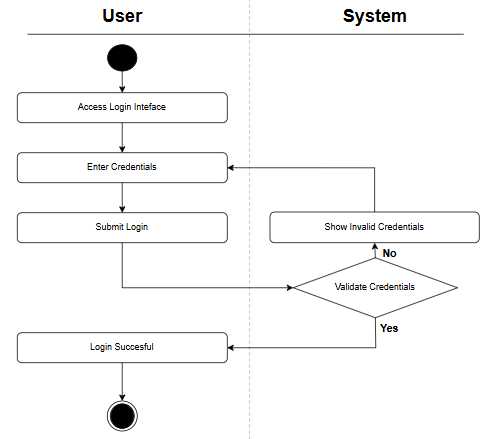


Figure 2. Activity diagram with swimline

### Logout

|  |  |
| --- | --- |
| **Title:** | Log Out. |
| **Description:** | The system allows users to log out securely, ending their session. |
| **Actor:** | Users and System. |
| **Preconditions:** | The user is logged in to the system. |
| **Postconditions:** | The user is logged out, and the session is terminated securely. |
| **Basic Flow:** | 1. The user clicks the "Log Out" button. 2. Enter credentials. 3. The system logs the user out and terminates the session. 4. The user is redirected to the login screen. |
| **Exemption Flow:** | **Invalid credentials.** |

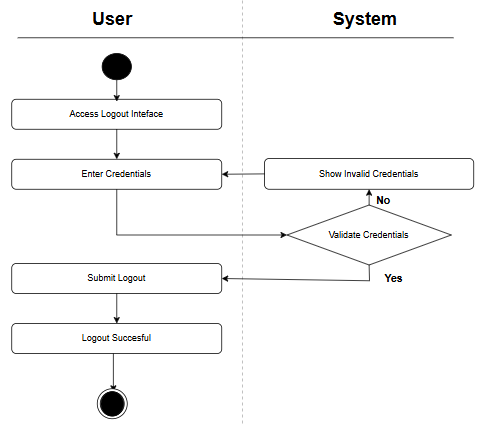


Figure 3. Activity diagram with swimline

### Load data file manually.

|  |  |
| --- | --- |
| **Title:** | Load data file manually |
| **Description:** | The system allows the user to upload their data manually |
| **Actor:** | Users and System. |
| **Preconditions:** | The user is logged in to the system. |
| **Postconditions:** | The data is uploaded and the system analyze it |
| **Basic Flow:** | 1. The user clicks the "Upload data” button. 2. The system retrieves a form where the user can upload his document. 3. The user uploads the document with the data in the form 4. The user hits “Submit” button to confirm the data upload 5. The data is processed by the system |
| **Exemption Flow:** | **Feedback prompt:** The system can’t read the document or the format is not correct |

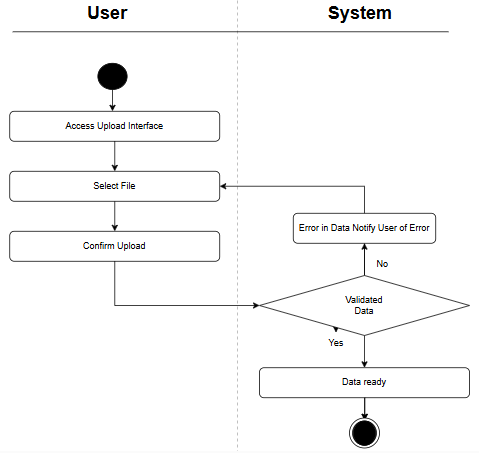


Figure 4. Activity diagram with swimline

### Load data file from database.

|  |  |
| --- | --- |
| **Title:** | Load Data File from Database. |
| **Description:** | The system allows users to load sales data directly from an external database. |
| **Actor:** | Users, System and Database. |
| **Preconditions:** | The user must be logged in.  The external database must be properly connected and authorized for access. |
| **Postconditions:** | Data is retrieved from the external database and made available for analysis. |
| **Basic Flow:** | 1. The user selects the “Load from Database” option. 2. The system establishes a connection to the external database. 3. The user selects the relevant dataset. 4. The data is retrieved and loaded into the system. |
| **Exemption Flow:** | **Database Connection Failure:** The system notifies the user that the connection failed and provides an option to retry or cancel.  **Data Retrieval Error:** If data cannot be retrieved, the system shows an error message and allows the user to attempt again. |

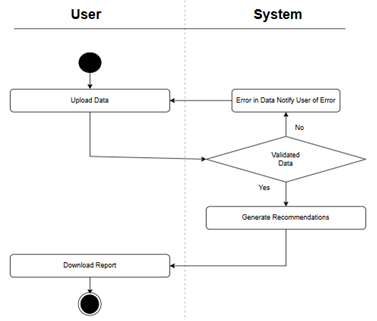


Figure 5. Activity diagram with swimline

### List of recommended products to purchase.

|  |  |
| --- | --- |
| **Title:** | List of recommended products to purchase. |
| **Description:** | The system generates a personalized product recommendation list, suggesting optimal quantities for each item. |
| **Actor:** | User, System. |
| **Preconditions:** | The file is already uploaded, or the system already retrieved the data from the database. |
| **Postconditions:** | The system drafts a csv document with product\_id, product\_name, recommended\_qty, qty\_in\_inventory |
| **Basic Flow:** | 1. The system analyzes the data using the machine learning model. 2. The system drafts a csv document with product\_id, product\_name, recommended\_qty, qty\_in\_inventory. 3. The system downloads automatically the document into the user’s computer. 4. The user opens the document. |
| **Exemption Flow:** | **Feedback prompt:** The system can’t read the document or it doesn’t have enough information to give a proper response |

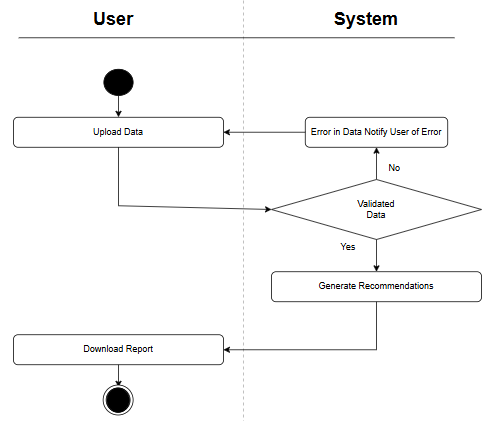


Figure 6. Activity diagram with swimline

### System’s feedback.

|  |  |
| --- | --- |
| **Title:** | System Feedback on Model Drift. |
| **Description:** | The system allows the user to upload their data manually |
| **Actor:** | Users (Admin) and System. |
| **Preconditions:** | The model is actively making predictions based on historical and new data. |
| **Postconditions:** | Feedback on model performance and drift status is provided by the user and shared to developer. |
| **Basic Flow:** | 1. The user reviews the list of recommended products to purchase. 2. The system prompts the user to indicate whether they have noticed any drift in the recommendations. 3. The user selects "Yes" or "No" to provide feedback. 4. If the user selects "Yes," the system asks for additional details or context on the drift. 5. The feedback is shared for further review. |
| **Exemption Flow:** | **The user does not provide feedback:** The system logs the absence of feedback for that session. |

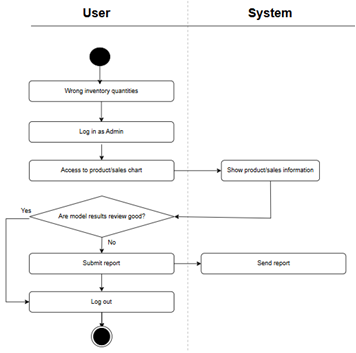


Figure 7. Activity diagram with swimline

# Other Non-functional Requirements

## Performance Requirements

The performance requirements outline the expected behavior of the system under different conditions to guarantee efficiency and proper functionality in various scenarios. These requirements focus on processing large amounts of historical sales data, delivering timely and accurate recommendations, and maintaining system performance in multi-user environments. Adhering to these standards will enable the SIP system to provide retail businesses with real-time, optimized inventory management. The following are the key performance requirements for the Smart Inventory Purchases system:

### Data Upload Response Time. The system shall process and upload historical data files (in CSV or Excel format) containing up to 1,000 records within a maximum of 60 seconds. *Rationale*: This requirement ensures that the system can handle large data volumes without affecting user experience, facilitating quick decisions and real-time inventory optimization.

### Inventory Recommendation Generation. The system shall generate purchase recommendations based on the analyzed data within a maximum of 40 seconds after data upload. *Rationale*: It is essential for the system to process information quickly to allow users to make timely adjustments and decisions regarding inventory management.

### Concurrent Workload Handling. The system shall support up to 5 concurrent users performing simultaneous operations (such as data uploads and recommendation generation) without exceeding a response time of 60 seconds.

### *Rationale*: The ability to handle multiple users is crucial to avoid delays in collaborative work environments where several employees may be using the system simultaneously.

### Real-Time Data Updates from External Sources. When using APIs to integrate with point-of-sale systems or external databases, inventory data updates should be completed within 10 seconds per operation.

### *Rationale*: In a retail environment, it is vital for data to be updated in real-time to ensure that inventory recommendations are accurate and reflect the current sales and inventory status.

### User Interface Performance. User interface actions, such as menu navigation, data uploads, or report downloads, should be completed within a maximum of 6 seconds. *Rationale*: A fast and responsive user interface enhances the user experience and ensures smooth and efficient operation, minimizing frustration or unnecessary delays.

## Safety and Security Requirements

The following safety and security requirements address potential risks, data protection concerns, and measures that must be taken to prevent any loss, damage, or misuse of the system and the data it handles. These safeguards are designed to ensure secure use of the Smart Inventory Purchases system and compliance with relevant standards and regulations.

### Safety Requirements.

#### Data Backup and Recovery. The system shall automatically back up all critical data (such as sales history and user settings) daily. In the event of data corruption or hardware failure, the system must allow for data recovery without any loss. Rationale: A robust backup and recovery system ensures that critical business data is not permanently lost due to unexpected system failures or accidents.

#### Fail-Safe Mechanism. The system shall include fail-safe mechanisms to prevent data loss or corruption during critical processes such as data upload, recommendation generation, or API communication. If a failure occurs, the system must roll back any incomplete transactions and notify the user.

#### Rationale: Fail-safe mechanisms ensure data integrity and prevent incorrect or partial updates to the system in case of unforeseen errors.

### Security Requirements.

#### User Authentication. The system shall require all users to authenticate using secure credentials (username and password) before accessing any features. Passwords must be stored using encryption to prevent unauthorized access.

#### Rationale: Ensuring only authorized personnel can access the system helps protect sensitive business data, such as sales figures and inventory levels.

#### Data Encryption in Transit and at Rest. All sensitive data, including historical sales data, inventory levels, and user information, shall be encrypted using industry-standard protocols during transmission (via HTTPS) and when stored in the database. Rationale: Encrypting data both in transit and at rest ensures that sensitive information is protected from eavesdropping, interception, or unauthorized access.

#### Access Control and Role-Based Permissions. The system shall implement role-based access control (RBAC) to limit access to specific features based on user roles (e.g., administrator, manager, employee). Only authorized users shall have access to sensitive operations such as changing system configurations or accessing sales data.

#### Rationale: RBAC helps prevent unauthorized users from performing sensitive operations, reducing the risk of accidental or intentional misuse of the system.

#### API Security. All API connections with external systems (such as point-of-sale systems or inventory databases) shall use secure communication protocols (e.g., HTTPS) and require API keys or tokens for authentication.

#### Rationale: Secure API communication ensures that data exchanged between systems is protected from unauthorized access and tampering during transmission.

#### Logging and Monitoring. The system shall log all user actions (such as logins, data uploads, and recommendation downloads) and system events for security auditing purposes. Logs shall be stored securely and be accessible only to system administrators.

#### Rationale: Logging and monitoring user activities help detect and prevent unauthorized actions, providing a trail for security audits and investigations.

#### Data Privacy and Compliance. The system shall comply with applicable data protection regulations, such as GDPR or local privacy laws, to ensure the confidentiality and integrity of customer and business data. Personal information shall not be stored longer than necessary and must be anonymized or deleted when no longer required.

#### Rationale: Ensuring compliance with privacy regulations minimizes legal risks and builds trust with users by protecting their personal and business data.

## Software Quality Attributes

The software quality attributes ensure that the SIP system meets the necessary standards for efficient, reliable, and adaptable performance. Below are key quality attributes along with the mechanisms to achieve them.

### Reliability

Reliability refers to the system’s ability to operate without failure for a specific period.

* The system shall guarantee an availability of 99.5%, ensuring that users can access it without interruptions, except during scheduled maintenance.
* Any critical operation, such as data upload or recommendation generation, shall include automatic recovery mechanisms in case of failure, preventing data loss or corruption.

How it will be achieved: Regular automated tests will be implemented along with continuous system monitoring to identify and fix failures before they impact the end user. Additionally, automatic backups will be configured to prevent data loss.

### Maintainability

Maintainability refers to the system’s ability to be modified easily, whether for bug fixes, performance improvements, or adapting to new requirements.

* The system shall have a modular design, allowing components to be updated or replaced independently without affecting other parts of the system.
* The system code shall be properly documented and follow defined coding standards. Each code module shall contain clear comments explaining its function and dependencies.
* Updates or modifications to the system shall be applied without interrupting the service for more than 10 minutes, achieved by using continuous deployment methodologies.

How it will be achieved: Version control (e.g., Git) and code reviews will ensure that any changes to the system are controlled and can be reverted if necessary. Additionally, a microservices-based architecture will be employed, allowing for specific parts of the system to be modified without needing to shut down the entire application.

### Usability

Usability refers to how easily end users can operate the system to achieve their goals.

* The system shall allow a new user to complete basic tasks (such as uploading data and downloading recommendations) in less than 15 minutes after their first login, with minimal training.
* The interface shall be intuitive, with clearly labeled menus and options following a logical flow. Additionally, clear and actionable error messages will be provided if a user makes a mistake.
* The interface design shall follow accessibility principles (e.g., WCAG 2.0 guidelines) to ensure all users, including those with disabilities, can interact with the system smoothly.

How it will be achieved: Usability tests with representative users will be conducted to identify potential interaction issues. Continuous user feedback will be collected to improve the experience with each update.

### Portability

Portability refers to the system's ability to be used in different environments or platforms with minimal adjustments.

* The system shall be compatible with the main operating systems used in retail businesses, including Windows and macOS.
* The codebase shall be designed to be deployable both on local infrastructure (on-premises) and in cloud environments (such as AWS or Azure) without major modifications.
* The use of standard frameworks and technologies (such as REST APIs and SQL databases) will facilitate migration of the system to other environments or platforms.

How it will be achieved: Compatibility tests across multiple platforms will be conducted during development. Additionally, containerization tools like Docker will be employed to ensure consistent system performance across different environments.

### Interoperability

Interoperability is the system’s ability to interact with other external systems efficiently.

* The system shall integrate with external point-of-sale (POS) systems and inventory databases via secure APIs in JSON or XML format.
* Integrations shall be tested to ensure that the exchanged data remains consistent and secure during transfer.

How it will be achieved: The system will include specific integration modules that follow standardized protocols. Interoperability tests with external systems will be conducted to ensure efficient and secure data exchange.

# Other Requirements

## Other Requirements

### Maintenance Requirements

The system must include mechanisms that facilitate its maintenance and updates over time. This includes:

* **Regular updates.** The system must be updated regularly to fix bugs, improve performance, and add new features. Updates should be applied with no more than 10 minutes of system downtime.
* **Technical support.** Users should have access to technical support, including detailed user manuals, troubleshooting guides, and online assistance. Support should be available during business hours, with a maximum response time of 24 hours.
* **System monitoring.** The system must include tools to monitor its performance, detect potential issues, and automatically generate alerts in case of problems.

### Documentation Requirements

The system must be accompanied by complete and up-to-date documentation that covers:

* **User manual:** Clear and detailed instructions to help users operate the system without needing advanced technical knowledge.
* **Technical documentation:** A detailed description of the system's architecture, technologies used, available APIs, and developer guides to facilitate easy integration and customization.
* **Maintenance documentation:** Guidelines for proper system operation and updates, including procedures for applying patches or migrating to new versions.

### Integration Requirements

The system must integrate with other existing systems to ensure smooth operation in enterprise environments. Key integration points include:

* **Integration with external databases:** The system must be able to connect to third-party databases (e.g., inventory databases or point-of-sale systems) to securely import and export data.
* **Standard API compatibility:** The system must offer APIs that allow integration with other platforms, adhering to industry standards like REST or SOAP.
* **Interoperability with enterprise systems:** It should be possible to integrate the system with broader enterprise systems like ERP (Enterprise Resource Planning) and CRM (Customer Relationship Management) to efficiently share inventory and sales data.

Appendix A – Data Dictionary

Users of the system are expected to understand how to input historical sales data, interpret system-generated recommendations, and operate within their assigned roles and responsibilities, whether as Operators or Admins.

Error Handling. The system shall handle errors during user interactions and system operations to ensure data integrity.

Output-Input Relationship. The system shall ensure that generated outputs accurately reflect the input data provided by the user.

Appendix B - Group Log

This document was developed by Group 10 of the TC4016.10 – Software Analysis, Design and Construction subject to solve a need in a retail business.

During the first session, all the known opportunities were listed, assessed by complexity and priority, resulting in the selection of the Smart Inventory Purchases proposal to solve a current problematic in a retail business.

The Group created a communication channel to work through all the items. Database variables description and Uses Cases not reviewed in first sessions.