Software Requirements Specification

for

Imtiaz Go

Version 1.0 approved

Prepared by Laiba 21F-9061 Habiba 21F-9099 Haiqa 21F-9107

FAST - NU

3-16-2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Laiba Irfan	3-16-2024	Initial Version	1.0
Haiqa	16-3-2024	Revised System Requirements and External Interface Requirements	2.0
Habiba Saleem	16-03-2024	Final Draft	3.0

1.Introduction

1.1.Purpose

The purpose of this document is to outline the detailed software requirements for the Imtiaz Go system, version 1.0. It outlines the scope of the product and details the features that will be developed as part of this project. It aims to guide the development team in implementing the system by providing functional and non-functional requirements and other necessary information.

1.2.Document Conventions

The following standards and typographical conventions were followed when writing this SRS document for the Imtiaz Go project:

Priority Inheritance: Higher-level requirements' priorities are assumed to be inherited by detailed requirements unless explicitly stated otherwise. This means that detailed requirements will inherit the priority of the higher-level requirement they are associated with.

Font Conventions:

Bold Text: Used for headings, section titles, and emphasized points.
Italic Text: Used for terms or concepts that need to be highlighted or defined.
Normal Text: Used for general content, descriptions, and explanations.

Highlighting:

Yellow Highlight: Used to draw attention to important notes, considerations, or key requirements.
Red Highlight: Used to indicate critical or high-priority requirements that require immediate attention.

1.3.Intended Audience and Reading Suggestions

This document is intended for developers, project managers, marketing staff, users, testers, and documentation writers involved in the Imtiaz Go project. The document is organized into sections that cater to different reader types, starting with overview sections and progressing to more detailed requirements. It is recommended to read the document thoroughly to understand the system requirements.

1.4.Product Scope

The Imtiaz Go software is a revolutionary retail solution aimed at transforming the traditional shopping experience by eliminating checkout lines and streamlining the entire shopping process. Its primary purpose is to provide customers with a seamless and convenient shopping experience while optimizing operational efficiency for retail businesses.

Key Objectives and Goals:

Eliminating Checkout Lines: The software aims to remove the need for customers to wait in long checkout lines, saving their time and enhancing overall satisfaction.

***Streamlining Shopping Process*:** By leveraging advanced technologies such as computer vision and automated payment systems, Imtiaz Go simplifies the shopping process, allowing customers to pick up items and leave without manual checkout procedures.

Enhancing Customer Experience: The software focuses on improving the overall shopping experience by reducing friction points, increasing convenience, and promoting a hassle-free environment.

***Operational Efficiency*:** For retail businesses, Imtiaz Go enhances operational efficiency by reducing staffing needs at checkout counters, minimizing errors in transaction processing, and optimizing inventory management.

Integration with Corporate Goals: The Imtiaz Go software aligns with the corporate goal of embracing innovation and leveraging technology to create value for customers and stakeholders. It supports the business strategy of staying ahead in the competitive retail market by offering cutting-edge solutions that enhance customer loyalty and attract new customers.

Benefits:

***Time-Saving*:** Customers save valuable time by bypassing traditional checkout processes, leading to a more efficient shopping experience.

Convenience: The software offers unparalleled convenience by automating payment procedures and eliminating the need for manual scanning or processing of items.

Cost Reduction: Retail businesses benefit from reduced operational costs related to staffing, checkout infrastructure, and transaction processing.

1.5.References

The following documents and web addresses are referenced in this SRS document for the Imtiaz Go project:

Imtiaz Go Vision and Scope Document

Title: Imtiaz Go Vision and Scope Document
Author: Imtiaz Go Project Team
Version Number: 1.0
Date: 3-16-2024
Source/Location:
https://docs.google.com/document/d/17qbJmrXj-GWQWPRkYk2G7j9ES2n79Np3b32FEHy7TIw/edit?usp=sharing

Imtiaz Go User Interface Style Guide

Title: Imtiaz Go User Interface Style Guide
Author: Imtiaz Go Design Team
Version Number: 1.2
Date: 3-16-2024
Source/Location:
https://www.figma.com/file/2fno8UooTxOlAE0Itp6wwC/Imtiaz-Go-Prototype?type=design&node-id=0%3A1&mode=design&t=LUlh6HUzRhNXN9fg-1

Imtiaz Go System Requirements Specifications

Title: Imtiaz Go System Requirements Specifications
Author: Imtiaz Go Development Team
Version Number: 1.1
Date: 3-16-2024
Source/Location:https://docs.google.com/document/d/17qbJmrXj-GWQWPRkYk2G7j9ES2n79Np3b32FEHy7TIw/edit?usp=sharing

Imtiaz Go Standards or Regulations

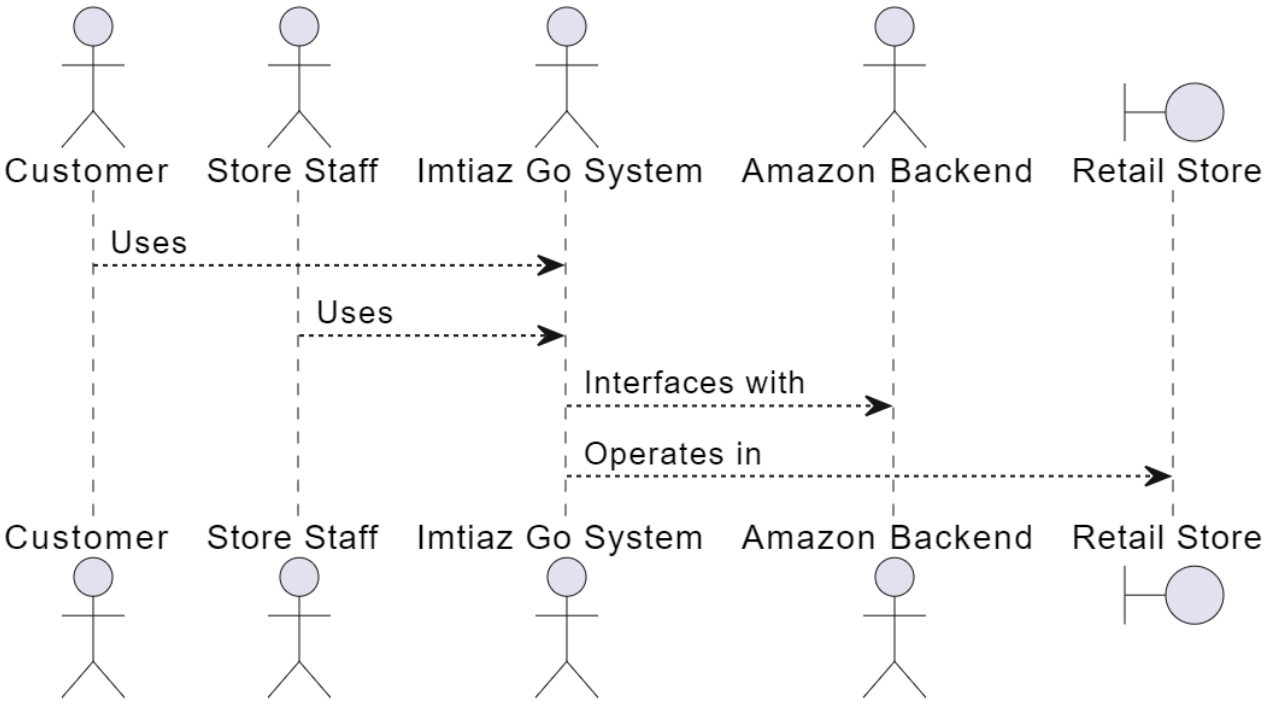
Title: Imtiaz Go Standard and Regulation
Author: Imtiaz Go Compliance Team
Version Number: 1.0
Date: 3-16-2024
Source/Location:
https://docs.google.com/document/d/17qbJmrXj-GWQWPRkYk2G7j9ES2n79Np3b32FEHy7TIw/edit?usp=sharing

2.Overall Description

2.1.Product Perspective

The Imtiaz Go system is a cutting-edge retail solution designed to transform the traditional shopping experience. It operates independently as a self-contained product within retail environments, offering a novel approach to checkout processes and customer interactions. Unlike existing systems that may require integration or replacement, Imtiaz Go introduces a completely new paradigm by leveraging advanced technologies to enhance the efficiency and convenience of retail operations.

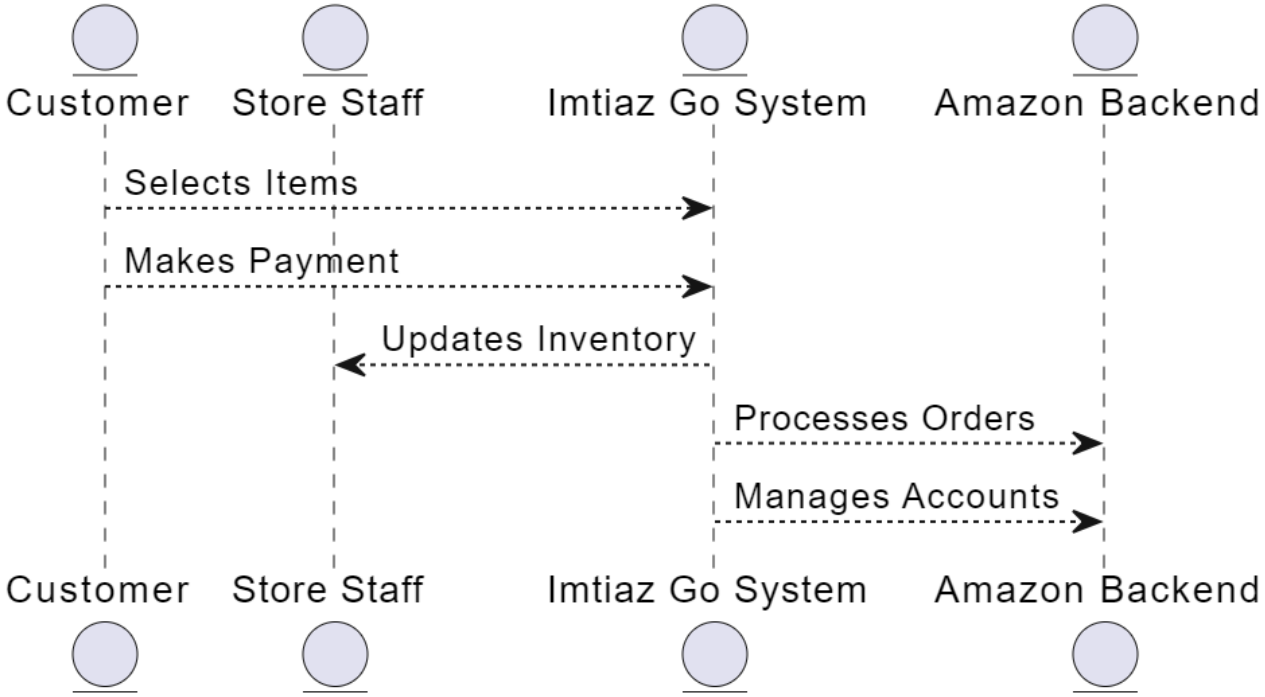
Imtiaz Go aims to enhance the overall retail experience by reducing wait times, simplifying payment processes, and improving customer satisfaction.
By streamlining the shopping process and eliminating checkout lines, Imtiaz Go empowers customers to enjoy a more convenient and frictionless shopping experience.



2.2.Product Functions

- Automated item tracking using special cameras and computer vision technology.
- Seamless payment processing through Amazon's secure payment gateway.

- User-friendly interface for customers to browse products and complete transactions effortlessly.
- Real-time inventory management and monitoring for store staff.
- Integration with Amazon's backend systems for order processing and customer account management.



2.3.User Classes and Characteristics

Customers:

General Shoppers: These are the *primary users* of the Imtiaz Go system who visit the retail store to purchase items. They include:

- **Regular Shoppers**: Customers who frequently visit the store for their shopping needs.
- **Occasional Shoppers**: Customers who visit the store on a less frequent basis.

Characteristics:

- Varied levels of technical expertise: Some customers may be tech-savvy and comfortable using digital interfaces, while others may prefer simpler interactions.
- Diverse demographics: Customers from different age groups, backgrounds, and preferences utilize the system.
- Expectation of convenience: Customers expect a seamless and efficient shopping experience without the hassle of traditional checkout processes.

- Preference for self-service: Many customers prefer self-service options that allow them to browse, select items, and make payments independently.

Store Staff:

Employees: These are the employees of the retail store who are responsible for various tasks related to the Imtiaz Go system. They include:
Cashiers*:* Staff members who handle exceptions, assist customers with questions or issues, and provide support as needed during the shopping process.
Inventory Managers: Staff members responsible for monitoring inventory levels, restocking shelves, and ensuring accurate product availability.

Characteristics:

- Technical proficiency: Store staff are trained to use the Imtiaz Go system effectively and efficiently.
- Attention to detail: They are responsible for ensuring accurate tracking of items, resolving any discrepancies, and maintaining inventory accuracy.
- Multitasking abilities: Store staff may need to manage multiple tasks simultaneously, such as assisting customers, monitoring the system, and managing inventory.
- Customer service-oriented: They focus on providing a positive and seamless experience for customers using the Imtiaz Go system, addressing any issues promptly and professionally.

2.4.Operating Environment

The software will operate in the following environment:

2.5.***Hardware Platform***: Compatible with standard retail hardware such as cameras, sensors, and payment terminals.
2.6.***Operating System:*** Compatible with major operating systems such as Windows, macOS, and Linux.
2.7.***Software Components***: Interfaces with Amazon's backend systems for payment processing and order management.

2.8.Design and Implementation Constraints

Regulatory Policies:
Imtiaz Go must adhere to stringent data privacy regulations and security standards to ensure the protection of customer information and compliance with legal requirements. This includes:

Data Privacy: The system must implement robust data privacy measures to safeguard customer data, such as personally identifiable information (PII), transaction records, and

payment details. Compliance with regulations such as GDPR (General Data Protection Regulation) and CCPA (California Consumer Privacy Act) is essential.

.

Hardware Limitations:

Imtiaz Go's software design must consider compatibility with existing retail hardware and infrastructure to ensure seamless integration and functionality. This includes:

Camera and Sensor Integration: Compatibility with a wide range of cameras and sensors commonly used in retail environments, ensuring accurate item tracking and data collection.

Payment Terminal Integration: Seamless integration with standard payment terminals to support multiple payment methods and ensure secure payment processing.

Interfaces:

Integration with Amazon's APIs and backend systems is a critical aspect of Imtiaz Go's functionality. This involves:

API Integration: Imtiaz Go must integrate with Amazon's APIs for payment processing, order management, inventory synchronization, and customer account management. This integration ensures real-time data exchange and synchronization between the Imtiaz Go system and Amazon's backend systems.
Backend Systems Integration: Seamless integration with Amazon's backend systems, including databases, servers, and services, to facilitate efficient communication and data flow between the systems.

Security Considerations:

Security is a top priority for Imtiaz Go, especially concerning payment processing and data handling. Key considerations include:

Secure Payment Processing: Implementation of secure payment processing mechanisms, including tokenization, SSL/TLS encryption, and PCI DSS (Payment Card Industry Data Security Standard) compliance.

Data Encryption Protocols: Implementation of strong encryption protocols for data at rest and data in transit, ensuring the confidentiality and integrity of sensitive information..

2.9. User Documentation

User Manuals:

User manuals are comprehensive documents that provide detailed instructions and guidance on how to use the Imtiaz Go system effectively. They cover various aspects of the system, including:

System Overview: An introduction to the Imtiaz Go system, its features, and functionalities.
Getting Started: Step-by-step instructions on how to set up and configure the system for use.

User Interface Navigation: Guidance on navigating the user interface, accessing different features, and performing common tasks.

Transaction Processes: Instructions on how customers can browse products, select items, make payments, and complete transactions using the Imtiaz Go system.

FAQs: Frequently asked questions and answers to help users quickly find solutions to common queries.

Online Help Resources:

Online help resources complement user manuals by providing additional support and information in a digital format. These resources may include:

Contextual Help: In-context help within the Imtiaz Go system, providing on-screen guidance and tips as users navigate different screens and features.

Knowledge Base: A searchable knowledge base containing articles, tutorials, and troubleshooting guides to address user queries and issues.

Video Tutorials: Video tutorials demonstrating key features and functionalities of the Imtiaz Go system, making it easier for users to understand and learn how to use the system effectively.

Community Forums: Online forums or discussion boards where users can interact, share tips, and seek assistance from other users or support staff
.

Tutorials for Customers and Store Staff:

Tutorials are instructional guides designed to help both customers and store staff understand and use the Imtiaz Go system efficiently. These tutorials may include:

Customer Tutorials: Step-by-step guides for customers on how to use the Imtiaz Go system for shopping, making payments, and navigating the store interface.

Store Staff Tutorials: Training materials and guides for store staff on how to monitor the system, handle exceptions, manage inventory, and provide support to customers using the Imtiaz Go system.

2.10.Assumptions and Dependencies

Assumptions:

Availability of Reliable Internet Connectivity:

- The assumption is that there will be a consistent and reliable internet connection available in the retail environment where Imtiaz Go is implemented. This connectivity is crucial for real-time data synchronization between the Imtiaz Go system and external systems, such as Amazon's backend.

Compliance with Amazon's API Usage Policies:

- It is assumed that Imtiaz Go will comply with Amazon's API usage policies and guidelines. This includes adhering to rate limits, data usage policies, authentication requirements, and any other guidelines set forth by Amazon for interacting with their APIs.

Dependencies:

Integration with Amazon's Backend Systems:

- Imtiaz Go is dependent on seamless integration with Amazon's backend systems for payment processing, order management, inventory synchronization, and customer account management. This integration ensures that Imtiaz Go can leverage Amazon's infrastructure and services to provide a seamless and efficient retail experience.

Availability of Necessary Hardware Components:

- The successful implementation of Imtiaz Go depends on the availability and compatibility of necessary hardware components, such as cameras, sensors, payment terminals, and other retail hardware. These components are essential for the functionality and operation of the Imtiaz Go system within the retail environment.

3. External Interface Requirements

The user interface of the Imtiaz Go system is meticulously crafted to offer an intuitive and visually appealing experience for users, enhancing their interaction with the platform. Here are the logical characteristics of the interface between the software product and its users:

3.1. User Interfaces

GUI Standards:

In line with modern web design principles and inspired by the ChicVogue UI Style Guide, the user interface of Imtiaz Go boasts a clean and minimalist design. It features intuitive navigation elements and maintains visually consistent layouts across all pages. Every aspect of the interface is meticulously designed to ensure a seamless and engaging user experience.

Sample Screen Images:

To provide users with a comprehensive understanding of the system's capabilities, the interface includes sample screen images showcasing various features. Users can explore clothing items, product details, and different aspects of the platform, including the homepage, product listing page, product details page, shopping cart, and checkout process. These images serve as visual guides, offering users a glimpse into the functionality and aesthetics of Imtiaz Go.

Screen Layout Constraints:

Imtiaz Go's screen layouts are designed to be responsive, adapting effortlessly to different screen sizes and resolutions. This ensures optimal viewing and usability across a wide range of devices, including desktops, laptops, tablets, and smartphones. Regardless of the device used, users can expect a consistent and enjoyable browsing experience.

Standard Buttons and Functions:

Imtiaz Go prioritizes accessibility by prominently displaying common buttons and functions throughout the interface. Users can easily access actions like "Add to Cart," "Checkout," "Sign In/Sign Up," and "Search" from any page. Additionally, supplementary functions such as "Help" or "FAQs" are readily available to assist users in navigating the platform and resolving any queries they may encounter.

Keyboard Shortcuts:

While the primary mode of interaction with Imtiaz Go is through mouse or touch input, the platform does not implement keyboard shortcuts extensively. Instead, it focuses on providing a seamless browsing experience that caters to users' preferences and habits.

Error Message Display Standards:

In the event of encountering an error, Imtiaz Go ensures that users are promptly notified with clear and informative error messages. These messages succinctly indicate the nature of the error and provide actionable instructions or suggestions for resolution. By following a consistent format and design, Imtiaz Go enables users to easily identify and address any issues they may encounter during their interaction with the platform.

Software Components Requiring User Interface:

The user interface is indispensable for various software components within Imtiaz Go, including product catalog management, user account management, shopping cart and checkout processes, order management, search functionality, and navigation menus and links. These components rely on the interface to facilitate seamless interactions between users and the platform, ensuring a smooth and efficient user experience.

3.2.Hardware Interfaces

The Imtiaz Go system interfaces with various hardware components to ensure seamless functionality within retail environments. Below are the logical and physical characteristics of each interface between the software product and the hardware components of the system:

Cameras:

- Supported Device Types: Imtiaz Go is compatible with a range of cameras equipped with computer vision technology.
- Nature of Data and Control Interactions: Cameras capture visual data of the retail environment, enabling automated item tracking and monitoring.
- Communication Protocols: Communication with cameras typically involves standard protocols for data transmission, ensuring real-time updates and accurate tracking.

Sensors:

- Supported Device Types: Various sensors, such as proximity sensors and motion sensors, are compatible with Imtiaz Go.
- Nature of Data and Control Interactions: Sensors detect movement, proximity, and other environmental factors within the store.
- Communication Protocols: Interaction with sensors often involves protocols for data transmission and control signals to trigger specific actions or responses within the system.

Payment Terminals:

- Supported Device Types: Imtiaz Go integrates with standard payment terminals commonly used in retail environments.
- Nature of Data and Control Interactions: Payment terminals facilitate secure payment processing for transactions conducted through the system.
- Communication Protocols: Communication with payment terminals typically adheres to industry-standard protocols for secure data transmission, ensuring compliance with payment regulations.

Other Retail Hardware:

- Supported Device Types: Imtiaz Go interfaces with various other retail hardware components, such as barcode scanners, RFID readers, and display screens.
- Nature of Data and Control Interactions: These hardware components contribute to functions like inventory management, product identification, and customer interaction.
- Communication Protocols: Communication protocols vary based on the specific hardware components, ensuring seamless integration and data exchange between Imtiaz Go and the devices.
-

3.3.Software Interfaces

Connections with Other Software Components:

- **Database Management System (DBMS):**
    - Name and Version: PostgreSQL 13.2
    - Purpose: ImtiazGo stores and retrieves crucial data such as product information, user data, order details, and transaction records from the PostgreSQL database.
    - Services Needed: CRUD operations for managing data entities, SQL queries for data retrieval and manipulation, and transaction management.
    - Communication Nature: ImtiazGo communicates with the PostgreSQL database through SQL queries over TCP/IP protocol, ensuring secure data transmission using database connection libraries like JDBC or ORM frameworks.
- **Operating System:**
    - Name and Version: Linux Ubuntu 20.04 LTS
    - Purpose: ImtiazGo operates within the Linux Ubuntu environment, leveraging system resources and services for hosting the web application.
    - Services Needed: File system access for reading/writing application files, process management for executing server-side scripts, and network services for communication with clients.
    - Communication Nature: Interaction with the operating system occurs through system calls and APIs provided by the OS kernel, facilitating resource allocation, process scheduling, and file operations.
- **Web Server:**
    - Name and Version: Apache HTTP Server 2.4
    - Purpose: Apache HTTP Server hosts and serves the ImtiazGo web application, handling HTTP requests and responses.
    - Services Needed: HTTP request processing, static and dynamic content delivery, SSL/TLS encryption for secure communication.
    - Communication Nature: ImtiazGo communicates with the Apache HTTP Server using HTTP/HTTPS protocols over TCP/IP, with server configuration files defining behavior and routing requests to appropriate application components.
- **Payment Gateway Integration:**
    - Name and Version: Amazon Pay API (Version 2.0)
    - Purpose: Facilitates secure payment processing for transactions conducted through ImtiazGo, handling payment authorization, capture, and settlement.
    - Services Needed: Payment card processing, tokenization, fraud prevention, and transaction logging.

    - Communication Nature: ImtiazGo communicates with the Amazon Pay API via HTTPS requests using RESTful APIs, exchanging payment details, authorization tokens, and transaction status updates securely.
- **Third-Party Libraries and Frameworks:**
    - Name and Version: React 17.0, Redux 4.0, Node.js 14.15
    - Purpose: Provides frontend UI components (React), state management (Redux), and backend application runtime (Node.js) for building and enhancing ImtiazGo's functionality.
    - Services Needed: UI rendering, state management, server-side request handling, dependency management, and transaction management.
    - Communication Nature: ImtiazGo integrates with these libraries and frameworks through code dependencies and API invocations, utilizing their functionalities to deliver a seamless and efficient user experience.
    -

3.4.Communications Interfaces

HTTP/HTTPS Protocol:

- Message Formatting: Standard HTTP request and response messages with headers and body content.
- Communication Standards: ImtiazGo follows HTTP/1.1 or HTTP/2 protocol standards for communication between clients and servers.
- Communication Security: HTTPS encryption ensures secure data transmission, protecting sensitive information exchanged between the client and server.
- Data Transfer Rates: Data transfer rates vary based on network bandwidth and server performance, with optimization techniques such as caching and compression employed to enhance performance.
- Synchronization Mechanisms: Asynchronous communication mechanisms like AJAX or WebSocket protocols may be utilized for real-time updates and interactive features.

Email Communication:

- Message Formatting: Email messages adhere to standard email formatting conventions with headers, subject lines, body content, and attachments.
- Communication Standards: SMTP standards are followed for sending outgoing emails from the web server to email servers.
- Communication Security: Email transmission may be encrypted using SMTPS or STARTTLS protocols for secure communication between mail servers.
- Data Transfer Rates: Email delivery speed depends on factors such as mail server configuration and network latency.

Integration with External APIs:

- Message Formatting: API requests and responses adhere to predefined message formats specified by the respective API documentation, typically in JSON or XML format.
- Communication Standards: RESTful principles or SOAP standards are followed for API communication, depending on the API specifications.
- Communication Security: Secure communication with external APIs is ensured using HTTPS and authentication mechanisms such as API keys or OAuth tokens.
- Data Transfer Rates: Data transfer rates for API requests and responses vary depending on network latency, API server performance, and data payload size.

4.System Features

4.1.Automated Item Tracking

4.1.1 Description and Priority:

This feature enables the automated tracking of items within the retail environment using specialized cameras and computer vision technology. It is of high priority as it forms the foundation for the Imtiaz Go system's functionality, facilitating seamless and accurate item tracking.

4.1.2 Stimulus/Response Sequences:

- **Stimulus**: Customer places items in the designated area for scanning.

  Response: Cameras capture images of the items and process them using computer vision algorithms to identify and track each item.

- **Stimulus**: Customer removes an item from the scanning area.

  Response: The system updates the inventory and removes the item from the virtual cart.

4.1.3 Functional Requirements:

REQ-1: The system must be able to identify and track multiple items simultaneously.

REQ-2: The system should have the capability to handle various types of products, including different shapes, sizes, and packaging.

REQ-3: In case of an item not being detected or misidentified, the system should provide an alert to store staff for manual intervention.

REQ-4: The system should update the virtual cart in real-time as items are added or removed from the scanning area.

REQ-5: If a scanned item is not recognized, the system should prompt the user for manual input or verification.

REQ-6: The system should have built-in error handling mechanisms to handle situations like poor lighting conditions or obstructions in the scanning area.

REQ-7: The system should provide visual feedback to the user, such as highlighting scanned items or displaying a running total of items scanned.

REQ-8: The system must ensure data accuracy and integrity by regularly calibrating cameras and verifying tracking algorithms.

REQ-9: The system should log scanning activities for audit and analysis purposes.

REQ-10: In case of technical issues or system downtime, the system should provide alternative checkout options to customers.
Seamless Payment Processing

4.2.Payment Processing

4.2.1 Description and Priority:

This feature enables seamless payment processing through ImtiazGo's secure payment gateway. It is of high priority as it directly impacts the user experience and is crucial for completing transactions efficiently.

4.2.2 Stimulus/Response Sequences:

Stimulus: User selects items for purchase and proceeds to checkout.

- Response: The system redirects the user to Amazon's payment gateway for secure payment processing.

Stimulus: User enters payment information and confirms the transaction.

- Response: The system verifies payment details and updates the order status accordingly.

4.2.3 Functional Requirements:

- REQ-1: The system must securely transmit payment data to Amazon's payment gateway using encryption protocols.
- REQ-2: The system should support various payment methods, including credit/debit cards and digital wallets.
- REQ-3: Upon successful payment authorization, the system should generate a unique transaction ID for tracking purposes.
- REQ-4: In case of payment failure or authorization issues, the system should provide clear error messages and guidance for resolving the issue.
- REQ-5: The system should ensure PCI DSS compliance to protect sensitive payment information.
- REQ-6: Payment processing should be seamless and fast, minimizing wait times for customers.

- REQ-7: The system should provide confirmation messages to users upon successful payment processing.

4.3.Real-Time Inventory Management

4.4.1 Description and Priority:

This feature enables real-time inventory management and monitoring for store staff. It is of high priority as it ensures accurate product availability and minimizes out-of-stock situations.

4.4.2 Stimulus/Response Sequences:

Stimulus: New inventory is added to the system or existing inventory is updated.

- Response: The system updates the inventory database and reflects changes in product availability.

Stimulus: Customer purchases an item, reducing the available quantity.

- Response: The system decrements the available quantity in the inventory database and updates the product status.

4.4.3 Functional Requirements:

- REQ-1: The system should track inventory levels for each product SKU in real-time.
- REQ-2: Automatic alerts should be generated for low stock levels or out-of-stock items.
- REQ-3: Inventory updates should be synchronized across all channels, including online and in-store purchases.
- REQ-4: Store staff should have access to real-time inventory reports and dashboards for monitoring purposes.
- REQ-5: The system should support automated replenishment processes, triggering restocking orders when inventory falls below a predefined threshold.
- REQ-6: Inventory adjustments and corrections should be logged for audit and analysis purposes.
- REQ-7: The system should integrate with barcode scanning technology for efficient inventory management.

4.4.Integration with Loyalty Program

4.5.1 Description and Priority:

This feature enables integration with the store's loyalty program, allowing customers to earn rewards and redeem points for discounts or special offers. It is of high priority as it fosters customer loyalty and encourages repeat purchases.

4.5.2 Stimulus/Response Sequences:

Stimulus: Customer completes a transaction using Imtiaz Go.

- Response: The system updates the customer's loyalty points based on the purchase amount.

Stimulus: Customer accesses their loyalty account through the Imtiaz Go interface.

- Response: The system displays the customer's points balance, transaction history, and available rewards.

4.5.3 Functional Requirements:

- REQ-1: The system should integrate with the store's existing loyalty program database to synchronize points and rewards.
- REQ-2: Customers should be able to link their Imtiaz Go account with their existing loyalty account for seamless points accrual.
- REQ-3: The system should automatically apply discounts or rewards during checkout if the customer chooses to redeem points.
- REQ-4: Store staff should have access to customer loyalty information for personalized assistance and promotions.
- REQ-5: The system should send automated notifications to customers about new rewards, promotions, or points expiry.

4.5.Multi-Language Support

4.6.1 Description and Priority:

This feature provides multi-language support for the Imtiaz Go interface, catering to a diverse customer base with varying language preferences. It is of medium to high priority, depending on the demographic diversity of the store's customer base.

4.6.2 Stimulus/Response Sequences:

Stimulus: User accesses the Imtiaz Go interface.

- Response: The system detects the user's language preferences or allows manual selection of language options.

Stimulus: User changes the language preference during their session.

- Response: The system updates the interface language accordingly for a seamless user experience.

4.6.3 Functional Requirements:

- REQ-1: The system should support multiple languages, including English, Urdu, Arabic, and others commonly spoken by the store's customer base.
- REQ-2: Users should be able to switch between languages easily through a language selector dropdown or menu.
- REQ-3: All interface elements, including text labels, buttons, and notifications, should be localized and translated accurately.
- REQ-4: The system should provide support for right-to-left (RTL) languages, ensuring proper layout and alignment.
- REQ-5: Store staff should also have access to multi-language support for backend administrative tasks and customer assistance.

4.6.Personalized Recommendations

4.7.1 Description and Priority:

This feature utilizes customer data and purchase history to provide personalized product recommendations, enhancing the shopping experience and increasing sales. It is of medium to high priority, depending on the level of sophistication desired for the recommendation engine.

4.7.2 Stimulus/Response Sequences:

Stimulus: User browses product listings or completes a purchase.

- Response: The system analyzes user behavior and preferences to generate relevant product recommendations.

Stimulus: User views recommended products on the Imtiaz Go interface.

- Response: The system displays personalized recommendations based on the user's profile and browsing history.

4.7.3 Functional Requirements:

- REQ-1: The system should collect and store user data, including browsing history, past purchases, and demographic information.
- REQ-2: An advanced recommendation algorithm should be implemented to analyze user data and generate personalized suggestions.
- REQ-3: Recommended products should be displayed prominently on the interface, integrated seamlessly with the browsing experience.
- REQ-4: Users should have the option to provide feedback on recommended products, refining future recommendations.
- REQ-5: The system should continuously update and improve recommendation algorithms based on user interactions and feedback.

5.Other Nonfunctional Requirements

5.1.Performance Requirements

5.1.1 Transaction Processing Time:

The system should process transactions in real-time, with a maximum processing time of X seconds per transaction. This ensures that customers can complete their purchases quickly without experiencing delays.

5.1.2 System Responsiveness:

The system should respond to customer actions (e.g., picking up items, placing them back on shelves) within X milliseconds to maintain a smooth and interactive shopping experience.

5.1.3 Camera Recognition Speed:

The camera and computer vision system should recognize and track items picked up by customers with high accuracy and minimal latency. Aim for a recognition speed of X milliseconds per item to ensure timely and accurate tracking.

5.1.4 Scalability:

The system should be capable of handling a high volume of concurrent transactions, especially during peak hours or promotional events. Specify the maximum number of transactions per hour or per day that the system should support without degradation in performance.

5.1.5 Network Latency:

Minimize network latency between the in-store sensors and the backend systems to ensure real-time data processing and synchronization. Aim for a maximum latency of X milliseconds for data transmission to maintain responsiveness.

5.1.6 Error Handling Time:

Define the maximum allowable time for the system to detect and handle errors or exceptions, such as item misplacements or technical glitches. Ensure that error handling processes do not significantly impact the overall performance of the system.

5.1.7 Availability and Uptime:

Specify the desired level of system availability and uptime to minimize disruptions to the shopping experience. Aim for a minimum of X% uptime over a specified timeframe (e.g., 99.9% uptime per month) to ensure continuous service availability.

5.1.8 Data Processing Speed:

Define the maximum processing time for backend systems to analyze and aggregate transaction data for reporting and analytics purposes. This ensures timely access to valuable insights for business decision-making.

5.1.9 Checkout Speed:

Aim for a checkout process that is significantly faster than traditional checkout methods. Define a target average checkout time per customer (e.g., X seconds) to streamline the shopping experience and minimize wait times.

5.1.10 System Load Testing:

Conduct thorough load testing to assess the system's performance under different load conditions, including peak usage scenarios. Specify performance benchmarks and thresholds for key metrics to validate the system's scalability and resilience.

5.2.Safety Requirements

1. Loss Prevention:

- **Shoplifting:**
  - **Action:** Develop algorithms to detect discrepancies between weight sensors and scanned items.
  - **Action:** Integrate with real-time video analytics to identify suspicious activity patterns (unusual item movement, following customers).
  - **Action:** Implement secure zones within the store layout that require staff assistance for specific high-value items.
- **Payment Errors:**
  - **Action:** Utilize secure payment gateways with tokenization to prevent unauthorized access to credit card details.
  - **Action:** Display clear itemized receipts with confirmation prompts for customers to verify purchase details before payment.
  - **Action:** Integrate secure communication channels within the software for customers to report discrepancies directly through the system.

2. Data Security:

- **Customer Information:**
  - **Action:** Encrypt all customer data at rest and in transit using industry-standard algorithms.
  - **Action:** Implement role-based access control (RBAC) within the software to restrict employee access to sensitive data based on their roles.
  - **Action:** Enforce strong password policies with minimum character length, complexity requirements, and regular mandatory password changes.
  - **Action:** Develop and implement data breach protocols for timely detection, containment, and reporting of security incidents.

3. System Reliability:

- **System Downtime:**
  - **Action:** Design the software with redundancy measures like failover servers to minimize downtime during hardware failures.
  - **Action:** Develop automatic system recovery procedures to restore functionality in case of software crashes.
  - **Action:** Integrate an in-app notification system to inform customers about system outages and estimated recovery time.
- **False Alarms:**
  - **Action:** Fine-tune weight discrepancy algorithms to minimize false positives based on product weight variations and packaging materials.
  - **Action:** Implement a review process within the software for staff to verify flagged discrepancies before alerting security.

4. User Interface (UI) Safety:

- **Action:** Design clear and intuitive user interfaces to minimize user errors during product scanning and payment processing.
- **Action:** Implement visual and audio cues to warn users about potential issues like unscanned items or exceeding weight limits for bags.

5. System Access Control:

- **Action:** Implement secure login procedures with multi-factor authentication for authorized staff access to the system.
- **Action:** Log all system activity with timestamps and user identification for audit purposes.

6. Software Testing:

- **Action:** Conduct thorough security testing to identify and address vulnerabilities before deployment.
- **Action:** Perform comprehensive functional testing to ensure accurate item recognition, weight measurements, and payment processing.
- **Action:** Include user acceptance testing (UAT) with real customers to identify usability issues and ensure a smooth shopping experience

5.3.Security Requirements

User Authentication:
Customer Authentication (Optional):

- The system may not require mandatory account creation for basic shopping. However, consider features like:
- Secure login system with strong password hashing for optional functionalities like purchase history, loyalty points, or personalized promotions.
- Integration with mobile wallets or contactless payment methods (e.g., NFC) for faster checkout with appropriate user verification (e.g., PIN or fingerprint).

Staff Authentication (Mandatory):

- Enforce mandatory login with multi-factor authentication (MFA) for all staff accessing the system.
- Implement role-based access control (RBAC) to restrict staff functionalities based on their job roles (e.g., cashiers cannot access system configuration).

Data Security:

- All customer data, including personally identifiable information (PII) like names, addresses, purchase history, and payment details, must be encrypted at rest and in transit using industry-standard algorithms (e.g., AES-256).
- Minimize data collection. Only store data essential for the shopping experience and legal requirements.

- Develop and implement data breach protocols for timely detection, containment, and reporting of security incidents.

System Security:
Secure Coding Practices: Adhere to secure coding practices to minimize software vulnerabilities. This includes regular code reviews and vulnerability assessments.

Network Security: Implement secure network protocols and firewalls to protect the system from unauthorized access and external threats. Regularly update software components and security patches to address known vulnerabilities.

Data Access Control: Implement role-based access control (RBAC) within the software to restrict access to sensitive data based on user roles and permissions. Log all system activity with timestamps and user identification for audit purposes.

Compliance:
- Ensure the system complies with relevant data privacy regulations regarding data collection, storage, usage, and user rights (e.g., access, rectification, erasure). This may include regulations like:
- General Data Protection Regulation (GDPR) or other regional data privacy regulations.
- If processing credit card payments directly, ensure the system adheres to PCI DSS (Payment Card Industry Data Security Standard) compliance standards for secure payment processing.

Security Certifications:
Consider pursuing relevant security certifications for the software development process and data security practices (e.g., SOC 2, ISO 27001). These certifications demonstrate adherence to industry best practices for secure development and data handling.

5.4.Software Quality Attributes

1. Usability:

- **Ease of Use:** Target a System Usability Scale (SUS) score above 80, indicating a highly usable system for customers with varying technical skills.
- **Learning Curve:** Aim for minimal training requirements for customers. The system should be intuitive enough for first-time users to navigate checkout efficiently within 2 minutes of initial interaction.

2. Performance:

- **Speed:** Achieve checkout processing within a timeframe of less than 15 seconds per item on average. This ensures a quick and efficient shopping experience.

- **Response Time:** Maintain a user interface response time of less than 2 seconds for a smooth and responsive experience. This includes product recognition, weight updates, and receipt display.
- **Availability:** Strive for system uptime exceeding 99.5%. This minimizes downtime and ensures consistent service for customers.

3. Reliability:

- **Mean Time Between Failures (MTBF):** Target an MTBF of at least 24 hours. This minimizes system crashes and ensures reliable operation throughout the day.
- **Data Integrity:** Maintain 99.99% accuracy in data capture and processing throughout the checkout process. This ensures accurate transactions, customer information, and inventory data.

4. Maintainability:

- **Modular Design:** Develop the software with modular components to facilitate future updates and modifications. Aim for at least 70% modularity to ensure maintainability.
- **Documentation:** Create clear and comprehensive system documentation, including user manuals, developer guides, and system architecture diagrams. This aids in troubleshooting, maintenance, and future development efforts.

5. Security:

- **Encryption Strength:** Utilize industry-standard encryption algorithms like AES-256 for all customer data at rest and in transit.
- **Vulnerability Testing:** Conduct regular penetration testing to identify and address potential security vulnerabilities before deployment. Aim for a maximum of 5 identified high-risk vulnerabilities per quarterly security assessment.

6. Testability:

- **Test Coverage:** Achieve at least 85% code coverage through unit and integration testing to ensure thorough software functionality testing.
- **Automated Testing:** Implement a high degree of automated testing (at least 70%) to streamline regression testing and facilitate faster release cycles.

Trade-offs:

6. In some cases, trade-offs between these attributes may be necessary. For example, maximizing security might involve additional processing steps that could slightly impact performance. During development, carefully evaluate these trade-offs to achieve an optimal balance between usability, performance, maintainability, and security for the Imtiaz Go system.

6.1.Business Rules

Customer Interactions:

- **Basic Shopping:** Customers can browse and select items without mandatory account creation.
- **Optional Account Features:** Customers can create accounts for features like:
  - Purchase history tracking
  - Loyalty program participation
  - Personalized promotions
- **Payment Methods:** The system should accept various payment methods including:
  - Cash
  - Credit cards
  - Debit cards
  - Mobile wallets (if integrated)
- **Age Verification:** For age-restricted items (e.g., alcohol, tobacco), the system should prompt for age verification through ID scanning or other approved methods.
- **Unexpected Items:** The system should handle unexpected situations like:
  - Unscanned items (flag for customer review)
  - Weight discrepancies (trigger security alert for review)
  - System errors (provide clear error messages and guide customers to assistance)

Staff Roles and Permissions:

- **Cashiers:**
  - Can process customer payments (with appropriate authorization for cash handling)
  - Can assist customers with troubleshooting and unexpected situations
  - Cannot access system configuration or sensitive data
- **Supervisors:**
  - Can access all checkout system functionalities
  - Can override certain system actions (e.g., weight discrepancies) with proper justification
  - Can manage staff accounts and permissions
- **System Administrators:**
  - Manage overall system configuration and security settings
  - Monitor system performance and troubleshoot technical issues
  - Grant access and permissions to other staff roles

Inventory Management:

- **Product Database:** The system should maintain an accurate and up-to-date product database with:
  - Item codes (linked to barcodes or RFID tags)
  - Descriptions
  - Prices

    - Age restrictions (if applicable)
- **Inventory Updates:** The system should automatically update inventory levels after each successful purchase.
- **Low Stock Alerts:** The system should notify staff of low stock levels for specific items to trigger restocking procedures.

Data Management:

- **Customer Data:** Only collect data essential for the shopping experience and legal requirements.
- **Data Storage:** Customer data should be stored securely with appropriate encryption.
- **Data Retention:** Define clear data retention policies for customer information based on local regulations and business needs.

Compliance:

The system should operate in accordance with all relevant laws and regulations, including:

6.2. Consumer protection laws
6.3. Data privacy regulations (e.g., GDPR)
6.4. Payment card industry standards (e.g., PCI DSS)

7. Other Requirements

Database Requirements:

7.1. Define the type of database management system (DBMS) required for the project (e.g., relational database, NoSQL database).
7.2. Specify data storage needs (e.g., transaction volume, data size).
7.3. Outline data backup and recovery procedures.

Internationalization Requirements (if applicable):

7.4. If Imtiaz Go plans to operate in multiple regions, consider requirements for:
- Multilingual support for user interfaces and receipts
- Currency conversion functionalities
- Adapting to different date and time formats

Legal Requirements:

7.5. List any specific legal requirements that the system must comply with, beyond those mentioned in the Business Rules section.
7.6. This may include regulations on data privacy, accessibility, or product labeling.

8. **Reuse Objectives:**

8.1. Specify if there are any existing software components or libraries that can be reused in the development of Imtiaz Go.
8.2. Define the extent to which code reuse is planned and how it will be implemented.

Non-Functional Requirements - User Interface (UI):

8.3. While UI details might be covered in a separate UI specification document, you can mention high-level UI requirements here.
8.4. For example, specify the need for a touch-friendly interface or requirements for accessibility features for users with disabilities.

Appendix A: Glossary

This section should define all the technical terms, acronyms, and abbreviations used throughout the SRS document to ensure clear understanding.

- **Automated Checkout System:** The self-service system that enables customers to scan, weigh, and pay for their purchases without cashier assistance.
- **SKU (Stock Keeping Unit):** A unique identifier assigned to each product in the inventory database.
- **RFID (Radio-Frequency Identification):** A wireless technology used for product identification through radio waves.
- **PCI DSS (Payment Card Industry Data Security Standard):** A set of security standards for organizations that handle credit card information.
- **GDPR (General Data Protection Regulation):** A regulation in EU law on data privacy and protection for individuals within the European Union.
- **MTBF (Mean Time Between Failures):** The average time between failures of a system.
- **SUS (System Usability Scale):** A standardized questionnaire used to measure the usability of a system.

Appendix B: Analysis Models (Optional)

This section can include various diagrams that visually represent the system's functionality and data flow. Some examples include:

- **Data Flow Diagram (DFD):** Illustrates the flow of data through the system, including external entities, processes, and data stores.
- **Entity-Relationship Diagram (ERD):** Shows the relationships between different entities (data objects) within the system.
- **Use Case Diagram:** Depicts the interactions between actors (users) and the system functionalities.

Appendix C: To Be Determined List (TBD)

This section serves as a record of any unresolved details or missing information within the SRS. It allows for tracking progress and ensures all crucial aspects are addressed before finalizing the document.

- TBD-1: Specific encryption algorithm strength for data at rest (e.g., AES-128, AES-256).
- TBD-2: Confirmation of mobile wallet integration options (e.g., Apple Pay, Google Pay).
- TBD-3: Definition of data retention period for customer purchase history