SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT (SRS)

1. Introduction

1.1 Purpose

The purpose of this document is to describe the functional and non-functional requirements of the **Smart Product Verification and Fraud Prevention System**. This system provides a camera-based product verification mechanism to ensure authenticity and prevent fraud in online shopping's.

1.2 Scope

The system verifies products at three stages: dispatch, delivery, and return. Using 360-degree scanning, the system compares product images and generates verification reports. It helps customers and sellers avoid disputes and builds trust.

1.3 Definitions, Acronyms, and Abbreviations

FR	Functional Requirement
NFR	Non-Functional Requirement
UI	User Interface
DB	Database

1.4 Product Overview

1.4.1 Product Perspective

The system is a web-based application integrated with a camera module for capturing product scans and a backend database for storing verification records.

1.4.2 Product Functions

The main functions of the system include:

- The system allows users to register and log in securely based on their roles.
- The system enables sellers to add and manage product details.
- The system stores dispatch scan data for future comparison.
- The system allows customers to scan products at the time of delivery.
- The system compares delivery scan data with dispatch scan data.
- The system displays verification results as matched or mismatched.
- The system allows customers to request product returns.
- The system performs product scanning during the return process.
- The system detects fraudulent activities such as product swapping or damage.
- The system generates detailed verification and fraud reports.

- The system stores verification history for auditing purposes.
- The system sends alerts to users when fraud is detected.
- The system allows administrators to monitor all system activities.
- The system provides secure data storage and access control.

1.4.3 User Characteristics

Delivery Agent

- Delivery Agent are registered users responsible for listing products.
- Delivery Agent have basic technical knowledge to operate scanning devices.
- Delivery Agent initiate product scanning at the dispatch stage.
- Delivery Agent can view verification and fraud reports related to their products.

Customer

- Customers are end users purchasing products online.
- Customers have minimal technical knowledge.
- Customers perform product scanning at delivery and return stages.
- Customers can view verification results and request product returns.

Administrator

- Administrators have advanced technical knowledge.
- Administrators manage user accounts and system configurations.
- Administrators monitor verification reports and fraud alerts.
- Administrators ensure system security and proper functioning.

1.4.4 LIMITATIONS

The Smart Product Verification and Fraud Prevention System has the following limitations:

1. The system depends on internet connectivity for real-time verification.
2. QR code verification can be affected if the QR code is damaged or blurred.
3. The system cannot detect fraud if a fake product copies the original QR code perfectly.
4. Performance may reduce when handling very large datasets without additional hardware support.
5. The accuracy of fraud detection depends on the quality of scanned images.
6. The system does not prevent fraud during offline transactions.
7. Initial setup requires product registration by seller**s**, which may take time.
8. Advanced fraud detection using AI requires high computational resources.

1.5 DEFINITIONS

The following terms are used throughout this document:

- **Admin:** A system user responsible for managing users, reports, and fraud alerts.
- **Seller:** A registered user who adds and manages product information.
- **Customer:** A user who verifies product authenticity by scanning QR codes.
- **Product ID**: A unique identifier assigned to each product.
- **QR Code:** A machine-readable code used to store product identification data.
- **Verification:** The process of checking product authenticity using the system.
- **Fraud:** Any attempt to sell or return a counterfeit or unauthorized product.
- **Database:** A structured collection of stored product and user data.
- **Scan Log:** A record of each QR code scan including time and user details.
- **Authentication:** The process of validating user credentials.
- **Authorization**: Permission granted to users based on their roles.

2. REFERENCES

Journals & Research Papers

1. Lowe, D. G. (2004). *Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision.
2. Bay, H., Tuytelaars, T., & Van Gool, L. (2008). *SURF: Speeded Up Robust Features*. Computer Vision and Image Understanding.
3. Zhang, C., & Zhang, Z. (2010). *A Survey of Recent Advances in Face Detection*. Microsoft Research (used for image comparison concepts).
4. Chandola, V., Banerjee, A., & Kumar, V. (2009). *Anomaly Detection: A Survey*. ACM Computing Surveys.
5. Dal Pozzolo, A., et al. (2015). *Adversarial Drift Detection in Fraud Prevention*. IEEE Transactions on Neural Networks.

Machine Learning & Image Processing

6. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
7. OpenCV Documentation. *Image Processing and Computer Vision Library*.
8. TensorFlow Documentation. *End-to-End Machine Learning Framework*.
9. Scikit-learn Documentation. *Machine Learning in Python*.

Web & Backend Technologies

10. Flask Documentation. *Flask: A Lightweight Web Framework for Python*.
11. Python Software Foundation. *Python Programming Language Documentation*.
12. Mozilla Developer Network (MDN). *HTML, CSS, and JavaScript Documentation*.

E-Commerce & Fraud Prevention

13. Bhattacharyya, S., et al. (2011). *Data Mining for Credit Card Fraud: A Comparative Study*. Decision Support Systems.
14. Whitrow, C., et al. (2009). *Transaction Aggregation as a Strategy for Credit Card Fraud Detection*. Data Mining and Knowledge Discovery.

Project & Internal Documentation

15. Project README.md – *Smart Product Verification and Fraud Prevention System*.
16. QUICKSTART.md – *System Setup and Execution Guide*.
17. requirements.txt – *Project Dependencies and Libraries*.

3. SPECIFIC REQUIREMENTS

3.1 Functional Requirements

The system shall provide the following core functions:

3.1.1 User Management

- The system shall allow users to register and log in as Seller, Customer, or Admin.
- The system shall authenticate users using valid credentials.
- The system shall allow the admin to activate, deactivate, or manage user accounts.

3.1.2 Product Registration (Seller)

- The system shall allow Sellers to register products with unique product IDs.
- The system shall generate a unique QR code for each registered product.
- The system shall store product details such as name, category, manufacturer, and batch number.

3.1.3 Product Verification

- The system shall allow Customers to scan QR codes to verify product authenticity.
- The system shall validate whether the scanned product exists in the database.
- The system shall display verification results as Genuine / Duplicate / Fraudulent.

3.1.4 Fraud Detection

- The system shall detect multiple scans of the same product ID.
- The system shall flag suspicious activities and generate fraud alerts.
- The system shall notify Admins about potential fraud cases.

3.1.5 Return Management

- The system shall allow Customers to request product returns.
- The system shall verify whether the product is eligible for return.
- The system shall update product status after successful return.

3.1.6 Reporting and Monitoring

- The system shall generate reports on verified products, fraud attempts, and user activities.
- The admin shall be able to view and analyse system-wide reports.

3.2 Usability Requirements

- The system shall provide a simple and intuitive user interface.
- The system shall be accessible through web browsers without complex installation.
- The system shall provide clear instructions and feedback messages.
- The system shall be usable by non-technical users.
- The system shall support mobile-friendly access for QR code scanning.

3.3 Performance Requirements

- The system shall respond to QR code verification requests within 2–3 seconds.
- The system shall support multiple concurrent users without performance degradation.
- The system shall handle large volumes of product records efficiently.
- The fraud detection mechanism shall execute in real time during verification.

3.4 Logical Database Requirements

- The system shall maintain a centralized database.
- The database shall store:
    - User information (User ID, Role, Credentials)
    - Product information (Product ID, QR Code, Status)
    - Scan logs (Scan time, Location, User)

- Fraud alerts and reports

- The database shall enforce unique constraints on product IDs.
- The database shall maintain referential integrity between users and products.
- Database Tables:

3.4.1 Users Table

Field Name	Data Type	Description
user_id (PK)	INT	Unique user ID
name	VARCHAR	User name
email	VARCHAR	Email address
password	VARCHAR	Encrypted password
role	VARCHAR	Seller / Customer / Admin
phone	VARCHAR	Phone number
created_at	VARCHAR	Account creation time

3.4.2 Products Table

Field Name	Data Type	Description
product_id (PK)	INT	Unique identifier assigned to each product.
seller_id (FK)	INT	Identifier of the seller who registered the product.
product_name	VARCHAR (100)	Name of the product provided by the seller.
category	VARCHAR (50)	Category to which the product belongs (e.g., Electronics, Fashion).
price	DECIMAL (10,2)	Selling price of the product.
description	TEXT	Detailed information about the product including features, specifications, brand details, and physical characteristics used for

		verification and fraud detection.
created_at	DATETIME	Date and time when the product was registered in the system.

3.4.3 Orders Table

Field Name	Data Type
order_id (PK)	INT
product_id (FK)	INT
customer_id (FK)	INT
order_date	DATETIME
status	VARCHAR

3.4.4 Scans Table

Filed Name	Data Type
scan_id (PK)	INT
order_id (FK)	INT
scan_type	VARCHAR (Dispatch/Delivery/Return)
scan_image_path	TEXT
scan_time	DATETIME

3.4.5 Verification Reports Table

Field Name	Data Type
report_id (PK)	INT
order_id (FK)	INT
dispatch_scan_id (FK)	INT
delivery_scan_id (FK)	INT
return_scan_id (FK)	INT
result	VARCHAR (Matched/Mismatch)
generated_time	DATETIME

3.4.6 Returns Table

Field Name	Data Type
return_id (PK)	INT
order_id (FK)	INT
Reason	VARCHAR
Status	VARCHAR

3.4.7 Fraud Alerts Table

Field Name	Data Type
alert_id (PK)	INT
order_id (FK)	INT
description	TEXT
alert_time	DATETIME

3.5 Design Requirements

- The system shall follow a modular architecture.
- The backend shall be implemented using Python (Flask).
- The frontend shall use HTML, CSS, and JavaScript.
- The system shall use QR code technology for product verification.
- The UI design shall follow a dashboard-based layout for each user role.

3.6 Design Constraints

- The system must run on standard web browsers.
- The system shall use open-source technologies only.
- The system shall comply with limited hardware and network resources.
- The database design must support scalability and data consistency.
- The project must be completed within academic time constraints.

3.7 Software System Attributes

Security

- The system shall protect user data using authentication and authorization.
- The system shall prevent unauthorized access to sensitive information.
- QR codes shall be tamper-resistant.

Reliability

- The system shall maintain consistent operation under normal conditions.
- Backup mechanisms shall be provided for data recovery.

Maintainability

- The system shall be easy to modify and extend.
- Code shall follow proper documentation and naming conventions.

Scalability

- The system shall support an increasing number of users and products.
- The database shall scale to handle future expansion.

3.8 Supporting Information

- QR Code technology is used for product identification and verification.
- Machine Learning models may be integrated for advanced fraud detection.

- The system follows standard software engineering and SRS guidelines.
- Documentation includes:
    - README file
    - User manuals
    - Installation and configuration guides

4.Operating Environment

4.1 Hardware Requirements:

- **Processor:** Intel Core i5 or higher
- **RAM:** Minimum 8 GB (4 GB for end users)
- **Storage:** Minimum 20 GB free disk space
- **Network:** Stable broadband internet connection.

4.2 Software Requirements:

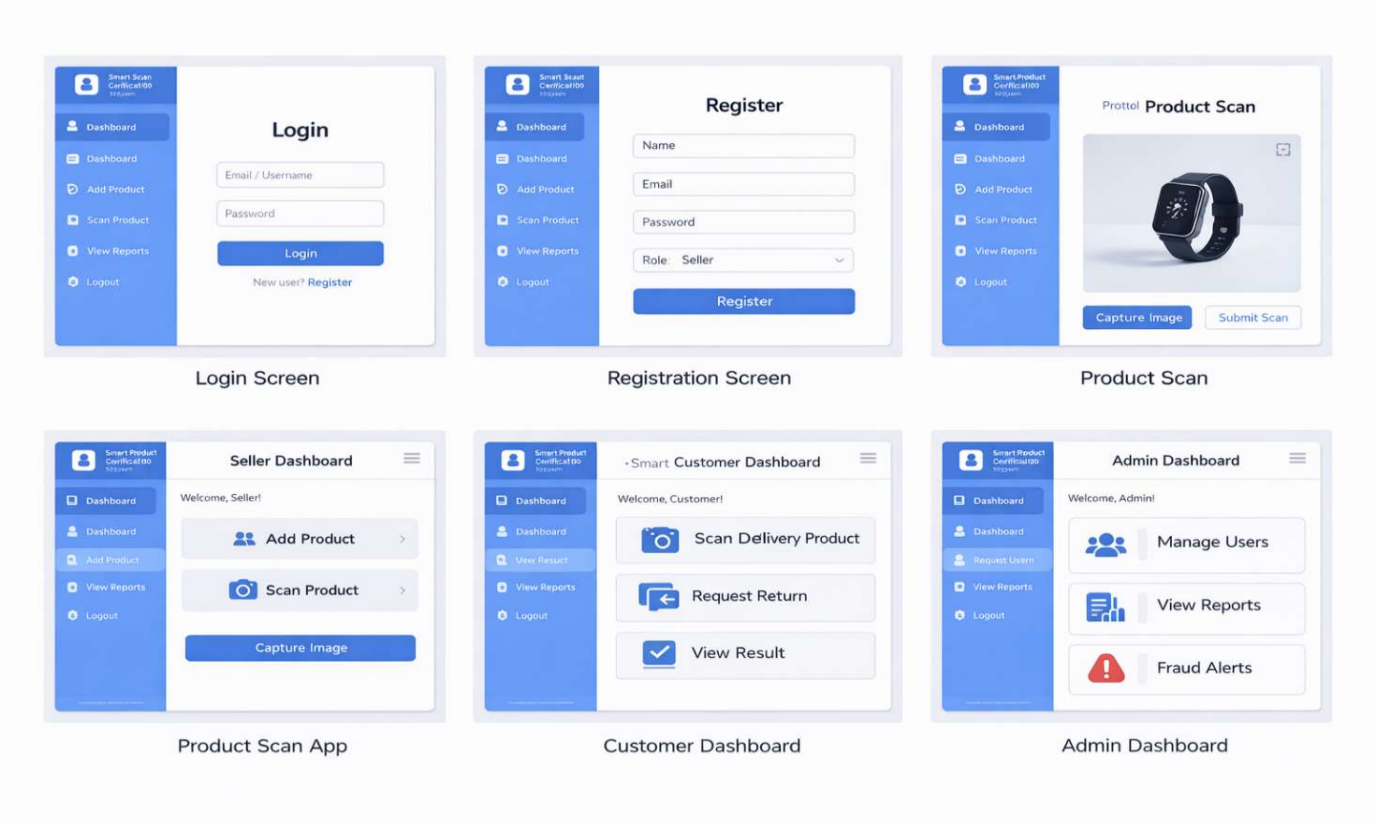
- **Operating System:** Windows 10/11 or Linux (Ubuntu), Android/iOS for client access.
- **Browser:** Google Chrome, Mozilla Firefox, Microsoft Edge (latest versions)
- **Database:** MongoDB
- **Framework:** ReactjS

5. External Interface Requirements

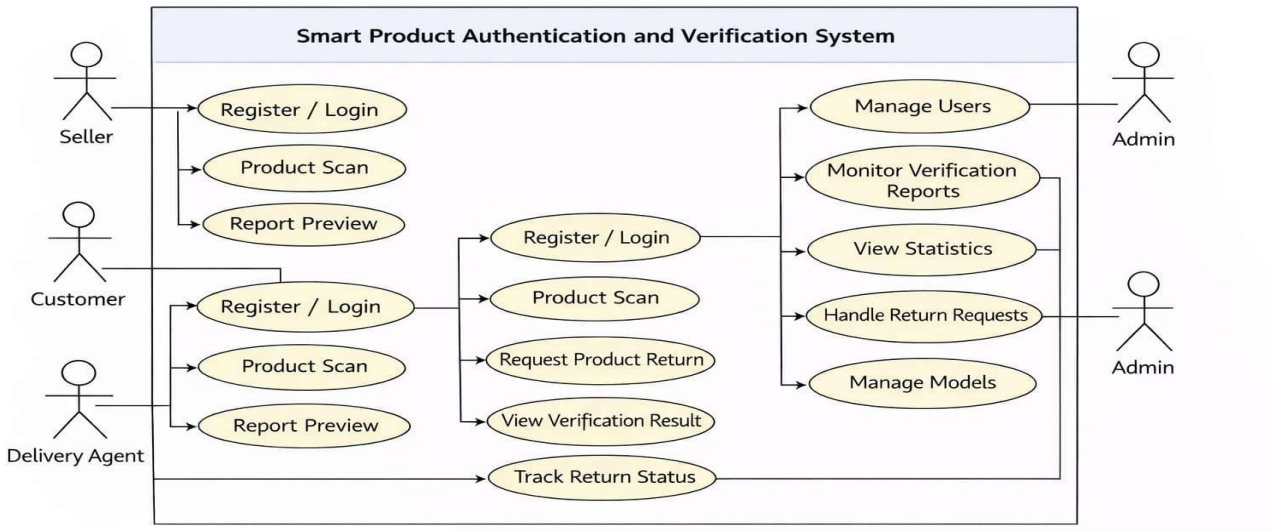
5.1 User Interfaces

• **Dashboard:** Displays user profile, product verification status, scan history, and fraud alerts.
• **Navigation Menu:** Provides access to features such as product upload, scanning, verification reports, and admin controls.
• **Forms:** Used for user registration, login, product details entry, scan submission, and return requests.
• **Reports**: Displays product verification results and fraud detection reports in tabular and graphical formats.

https://www.figma.com/make/dTM7XlPF2tUWQx1SxJ7qa2/Modern-Login-Screen-UI?fullscreen=1&t=orButZPC2K4EYoA



Use case Diagram:



5.2 Hardware Interfaces

The system requires a camera-enabled device such as a smartphone or webcam to capture product images for 360-degree scanning. The server communicates with client devices through standard input/output interfaces without requiring specialized hardware connections.

5.3 Software Interfaces

System	Purpose	Data Format
Web Application (Frontend)	User interaction and data input	JSON
Backend API NumPy, Pandas, TensorFlow	Business logic and verification processing	REST/JSON
Database System MongoDB	Storage of user, product, and scan data	NO-SQL
Email Service (Fire base)	Sending alerts and notifications	SMTP

5.4 Communication Interfaces

• **HTTP/HTTPS:** Used for secure communication between client and server.
• **REST API:** Enables data exchange between frontend and backend services.
• **Email:** Used for user notifications, verification alerts, and password recovery.

6. VERIFICATION

Verification ensures that the system meets the specified requirements.

6.1 Functional Verification

- Verify user registration and login functionality.
- Verify QR code generation for products.
- Verify accurate product verification results.
- Verify fraud detection when duplicate scans occur.
- Verify return verification and status updates.

6.2 Performance Verification

- Verify system response time during QR code scanning.
- Verify system performance under multiple simultaneous users.
- Verify database query efficiency.

6.3 Security Verification

- **Verify user authentication and access control.**

- **Verify protection of sensitive user data.**
- **Verify prevention of unauthorized product access.**

6.4 Usability Verification

- Verify ease of use for customers and sellers.
- Verify clarity of messages and system feedback.
- Verify mobile compatibility for QR scanning.

6.5 System Validation

- Validate the system against real-world product verification scenarios.
- Validate fraud alerts generated by the system.
- Validate reports generated for admin review.