SOFTWARE REQUIREMENTS SPECIFICATION

for

DIGITAL RECEIPT MANAGEMENT SYSTEM

Version 1.0

Prepared by :
Tejas Joshi (612203078)
Tejas Kolhe (612203097)

Instructor: Prof. Seema Chavan

College Of Engineering Pune

08 March 2025

Contents

Revision History

Name	Date	Reason for Changes	Version
Tejas Joshi	08 March 2025	Initial draft	1.0

1 Introduction

1.1 Purpose

This document specifies the requirements for the **Digital Receipt Management System (DRMS)**, a software solution designed to help users scan, store, categorize, and analyze receipts digitally. The system aims to simplify receipt management for individuals and small businesses, providing features such as OCR-based data extraction, automatic receipt capture, expense analytics, and warranty tracking.

1.2 Document Conventions

- **Bold text** is used for key terms and section headings.
- *Italic text* is used for emphasis.
- Requirements are numbered sequentially (e.g., REQ-1, REQ-2).
- Diagrams are referenced as figures (e.g., Figure 1).

1.3 Intended Audience and Reading Suggestions

This document is intended for:

- **Developers**: To understand the system requirements and design the software.
- **Project Managers**: To plan and track the project's progress.
- **Testers**: To create test cases based on the requirements.
- **Users**: To understand the system's capabilities and features.

Readers should start with the **Introduction** and **Overall Description** sections, followed by the **System Features** and **Nonfunctional Requirements**.

1.4 Product Scope

The DRMS is a web-based application that allows users to manage receipts digitally. It includes features such as receipt scanning, automatic data extraction, categorization, expense tracking, and warranty reminders. The system aims to improve financial tracking, reduce paper waste, and provide a seamless user experience.

1.5 References

- IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications.
- `https://reactjs.org`
- `https://nodejs.org`
- `https://www.mongodb.com`

2 Overall Description

2.1 Product Perspective

The DRMS is a standalone system that integrates with third-party services such as email providers, payment gateways, and cloud storage platforms. It will be accessible via web and mobile interfaces.

2.2 Product Functions

- Receipt scanning and storage using OCR.
- Automatic receipt capture from emails and payment gateways.
- Categorization and tagging of receipts.
- Expense analytics and reporting.
- Warranty tracking with reminders.
- Multi-device access and cloud storage.

2.3 User Classes and Characteristics

- **Individual Users**: Manage personal receipts and track expenses.
- **Small Businesses**: Track business expenses and generate tax-ready reports.
- **Administrators**: Manage system settings and user accounts.

2.4 Operating Environment

- **Frontend**: Web browsers (Chrome, Firefox, Safari) and mobile devices (iOS, Android).
- **Backend**: Node.js server hosted on cloud platforms (AWS, Heroku).
- **Database**: MongoDB hosted on cloud services (MongoDB Atlas).

2.5 Design and Implementation Constraints

- Use of MERN stack (MongoDB, Express.js, React.js, Node.js).
- Integration with third-party OCR APIs (e.g., Google Cloud Vision).
- Compliance with GDPR and other data protection regulations.

2.6 User Documentation

- User manual for receipt scanning and categorization.
- Online help and tutorials for expense tracking and reporting.
- API documentation for developers.

2.7 Assumptions and Dependencies

- Users have access to a smartphone or computer with a camera.
- The system relies on third-party OCR services for data extraction.
- Cloud storage is required for long-term receipt storage.

3 External Interface Requirements

3.1 User Interfaces

- Intuitive dashboard for viewing and managing receipts.
- Receipt scanning interface with camera integration.
- Search and filter functionality for receipts.

3.2 Hardware Interfaces

- Camera for scanning physical receipts.
- Cloud storage for secure data storage.

3.3 Software Interfaces

- Integration with email providers (e.g., Gmail, Outlook) for automatic receipt capture.
- Integration with payment gateways (e.g., PayPal, Amazon) for digital receipts.
- Integration with OCR APIs (e.g., Google Cloud Vision) for data extraction.

3.4 Communications Interfaces

- HTTPS for secure data transmission.
- RESTful APIs for communication between frontend and backend.

4 System Features

4.1 Receipt Scanning and Storage

4.1.1 Description and Priority

This feature allows users to scan physical receipts using a camera and store them digitally. **Priority: High**.

4.1.2 Stimulus/Response Sequences

- User opens the app and selects ”Scan Receipt.”
- System captures the receipt image and processes it using OCR.
- System extracts key data (merchant, amount, date) and stores it in the database.

4.1.3 Functional Requirements

- **REQ-1**: The system must support receipt scanning via camera.
- **REQ-2**: The system must use OCR to extract data from scanned receipts.
- **REQ-3**: The system must store receipts securely in the cloud.

4.2 Expense Analytics

4.2.1 Description and Priority

This feature provides users with insights into their spending habits. **Priority: Medium**.

4.2.2 Stimulus/Response Sequences

- User selects ”View Analytics” from the dashboard.
- System generates a report based on categorized receipts.
- User views spending trends and summaries.

4.2.3 Functional Requirements

- **REQ-4**: The system must categorize receipts by type (e.g., groceries, travel).
- **REQ-5**: The system must generate monthly/weekly spending summaries.

5 Other Nonfunctional Requirements

5.1 Performance Requirements

- The system must process OCR requests within 5 seconds.
- The system must handle up to 10,000 concurrent users.

5.2 Safety Requirements

- The system must ensure data privacy and comply with GDPR.

5.3 Security Requirements

- All data must be encrypted during transmission and storage.
- User authentication must be implemented using JWT or OAuth.

5.4 Software Quality Attributes

- **Usability**: The system must have an intuitive and responsive UI.
- **Scalability**: The system must scale to support increasing numbers of users and receipts.

5.5 Business Rules

- Only authenticated users can access their receipts.
- Administrators can manage user accounts and system settings.

6 Other Requirements

- The system must support multi-language interfaces.
- The system must comply with international data protection laws.

A Glossary

- **OCR**: Optical Character Recognition.
- **DRMS**: Digital Receipt Management System.
- **API**: Application Programming Interface.

B To Be Determined List

- TBD-1: Finalize integration with payment gateways.
- TBD-2: Define warranty reminder frequency.