

Software Requirements Specifications

Project Name	Debt Management System
Prepared By	The 5 idjuts
Date	March 7, 2025
Version	1.0

Introduction

Purpose
The purpose of this document is to define the function and the non-functional requirements for a Debt Management System that allows the customers to manage the debt balance and debt due date.
Scope
This system will allow users to :
<ul style="list-style-type: none">• Manage debt balances• Check the debt due date• Manage personal profile and security settings
The system will support the desktop platform and will be available only for 24/7.
Constraints and Assumptions
<ul style="list-style-type: none">• User must have a valid ID• The system requires an internet connection for the transaction

Functional Requirements

1.	Function	User Authentication
	Description	Users must log in securely.
	Actors	Seller
	Steps	<ol style="list-style-type: none">1. Seller enters their email, username, and password.2. The system requires a confirmation password3. If correct, the user gains access. Otherwise, an error message is displayed.
	Exceptions	<ul style="list-style-type: none">• If the seller enters the wrong password 3 times, the account is locked for 30 minutes.
2.	Function	Customer Profiles

	Description	The system allows the seller to create and manage customer profiles, storing essential details for tracking credit transactions.
	Actors	The Seller
	Preconditions	<ul style="list-style-type: none"> • The seller must be logged into the system. • The customer profile must be created before recording debts.
	Steps	<ol style="list-style-type: none"> 1. The seller selects "Add New Customer." 2. The seller enters the following details: <ul style="list-style-type: none"> • Full Name • Mobile Number (for notifications) • Address (optional) • Credit Limit (maximum borrowable amount) 3. The seller saves the profile. 4. The system generates a unique Customer ID and stores the data.
	Exceptions	<ul style="list-style-type: none"> • If the mobile number is invalid, the system prompts the seller to correct it. • If the customer already exists, the system prevents duplicate entries. • If required fields are empty, the system alerts the seller before proceeding

3.	Function	Debt Recording
	Description	The seller will record and manage customer debts, including purchases on credit, payments, and outstanding balances.
	Actors	The seller
	Preconditions	<ul style="list-style-type: none"> • The seller must be logged into the system. • The customer must be registered in the system.
	Steps	<ol style="list-style-type: none"> 1. The seller selects "Add Debt." 2. The seller enters the customer's name or selects an existing profile. 3. The seller inputs the purchased items, quantity, and price. 4. The system calculates the total debt amount. 5. The seller sets the due date (default or custom). 6. The seller confirms and saves the transaction. 7. The system updates the customer's outstanding balance.
	Exceptions	<ul style="list-style-type: none"> • If the customer has exceeded their credit limit, the system alerts the seller and prevents the transaction. • If there is missing information (e.g., no items or price entered), the system prompts the seller to complete the details.

3.	Function	Payment Tracking
	Description	The system allows the seller to record customer payments, update outstanding balances, and track partial or full repayments.
	Actors	The seller
	Preconditions	<ul style="list-style-type: none"> • The seller must be logged into the system. • The customer must have an existing debt record.
	Steps	<ol style="list-style-type: none"> 1. The seller selects "Record Payment." 2. The seller searches for the customer using their name or customer ID. 3. The system displays the customer's total outstanding balance. 4. The seller enters the payment details: 5. Amount Paid 6. Payment Date 7. Payment Method (Cash) 8. The system updates the remaining balance. 9. If the payment is partial, the system keeps the debt status as "Active." 10. If the payment is full, the system marks the debt as "Paid." 11. The seller confirms and saves the transaction.
	Exceptions	<ul style="list-style-type: none"> • If the payment amount exceeds the outstanding balance, the system alerts the seller. • If the seller enters an invalid payment method, the system prompts for correction. • If no debt is found for the selected customer, the system prevents the transaction.

Non-Functional Requirements

Security
<ul style="list-style-type: none"> • All transactions must be encrypted using AES-256. • Users must log out after 5 minutes of inactivity. • The app must comply with ISO 27001 security standards. • Multi-factor authentication (MFA) must be enforced for all user logins.
Performance
<ul style="list-style-type: none"> • The app should load within 2 seconds under normal conditions. • Transactions should be completed in less than 5 seconds. • Reports should be generated within 10 seconds.
Availability

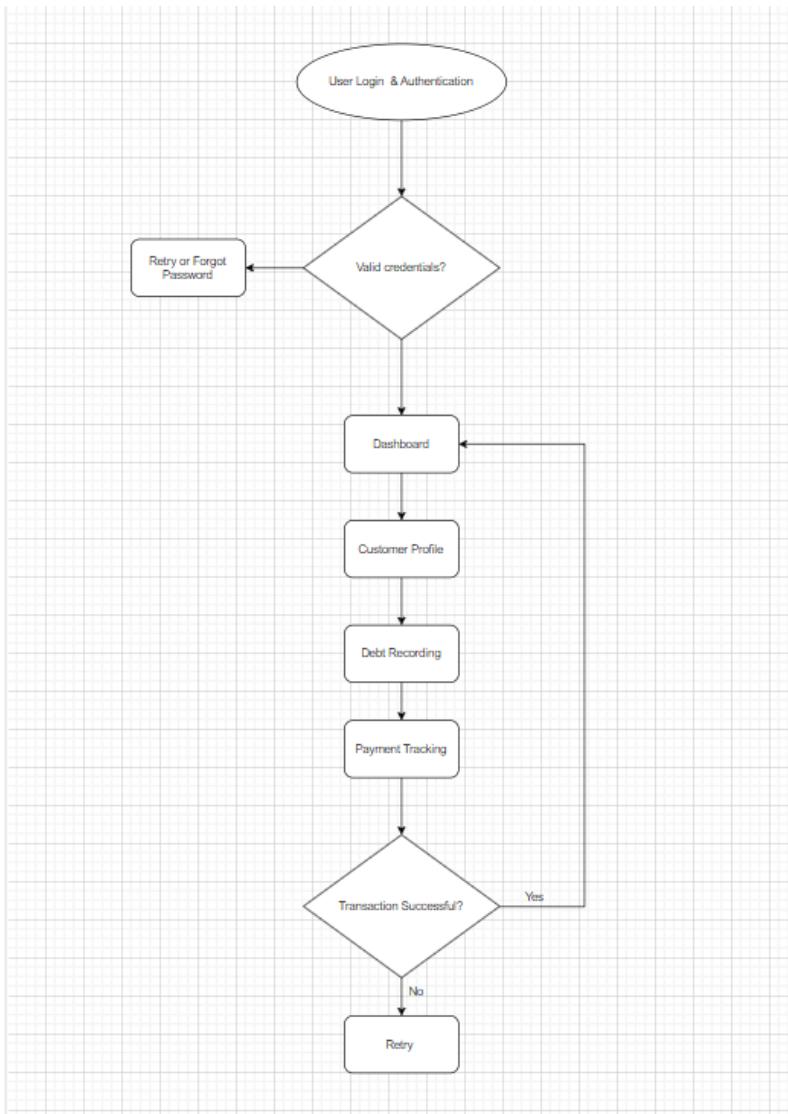
- Scheduled maintenance must be announced at least 48 hours in advance.
- The system should have automatic failover mechanisms to ensure minimal downtime.
- Data backups must be performed daily and stored securely.

Usability

- The app must follow Material Design (Android) and Human Interface Guidelines (iOS) for UI/UX.
- The interface should be intuitive and require no more than 3 steps to complete key actions.
- The system should be accessible to users with disabilities, following WCAG 2.1 standards.
- User feedback should be collected regularly to enhance usability and performance.

Mockups & Diagrams

Entity Relationship Diagram ERD



Graphical User Interface

Still in production

Flow Chart

Still in production

Glossary

- OTP: One-Time Password

- 2FA: Two-Factor Authentication
- AES-256: Advanced Encryption Standard (256-bit)

Appendix

- Compliance Documents (e.g., GDPR, PCI-DSS)
- API Documentation (if applicable)