24/08/25

(2) Credit card processing system

Problem statement:
Modern businesses and banks face the challenge of processing credit card transactions quickly, securly and accurately. Manual handling / outdated systems can lead to failed payments, delayed settlements, fraudulent activities and poor customer service experience.

SRS:

1. Introduction

1.1 Purpose of the document
The document outlines the soft. requirements for a credit card processing statements. Its purpose is to clearly define functional + non functional requirements for all stakeholders - developers, testers, business analysts, project managers and the client.

1.2 Scope of the document
The CCPS is a web-based and API-enabled platform that processes ~~creid~~ credit card transactions securely and efficiently. It will handle card authorizations, payment settlements, refunds, transactions tracking and reporting.

1.3 Overview

The CCPS is a secured and scalable platform for handling card transactions. It supports real-time payments, fraud checks, settlements, refunds and reporting, ensuring fast, reliable and compliant processing for customers, merchants and administrators.

2. General Description

2.1 Product functions

- Authorize and process credit card transactions
- Validate card details and detect fraudulent attempts
- Handle settlements and refunds
- Generate transaction and account statements
- Manage user accounts
- Provide real time notifications and alert
- Generate reports for compliance.

2.2 User Characteristics.

- Customers: Make payments securly, check transaction history.
- Merchants: Accept payments, view settlements, manage refunds.
- Bank/Admin staff: Monitor transactions, detect fraud.
- System Admins: Manage users, configure system parameters.

3. Functional Requirements

FR 3.1 Transaction Processing : The system shall authorize credit card payments in real-time.

FR 3.2 Fraud detection and security : The system shall detect suspicious transactions using rules and patterns.

FR 3.3 Account Management : Customers can view transaction history and statements.

FR 3.4 Notifications : Send email/SMS alerts for succeful transactions, fail attempts / refunds.

FR 3.5 Reporting : Generate daily / weekly / monthly transaction reports.

4. Interface Requirements.

4.1 User interfaces

A responsive web dashboard for merchants / admins to manage transaction and reports.

4.2 Hardware interfaces

Supports POS terminals for in-store payments Integrates with EMV and contactless card readers.

5. Performance Requirements

- Support up to 10,000 concurrent transaction/min
- Authorization time/~~per trans~~ transaction < 3seconds.

- 99.99% uptime for critical transaction services.
- Automatic daily backups of transaction data.

6. Design Constraints

- Must use secure, open source techs
- Must comply with PCI DSS, GDPR and others.
- Must ensure end-to-end encryption.
- Support multiple currencies.

7. Non-Functional Attributes

7.1 Security
All sensitive data encrypted (AES-256 / higher)

7.2 Usability
Simple and intuitive UI for customers and merchants.

7.3 Maintainability
Modular code design for easy updates.

7.4 Reliability
Failover and disaster recovery mechanisms in place.

8. Preliminary Schedule and Budget

8.1

Schedule (phase)	Timeline.
Requirements analysis	2 weeks
UI/UX Design	3 weeks
Development	12 weeks
Testing & QA	4 weeks

8.2 Budget → ~~23 weeks~~ $47,000.