02/04/25

Date
Page

LAB-2

2) Credit Card Processing

Problem Statement

A credit card processing system automates the secure handling of card transactions between users, merchants, and banks. It should support authorization, capture, settlement, and reporting. Key features include fraud detection, encrypted data transfer, and transaction logs. The system must ensure real-time processing and high reliability. Regulatory compliance is critical.

Software Requirements Specification

1. Introduction

1.1. Purpose of the document

The purpose of the document is to outline the functionality, scope and technical requirements of a credit card processing application that enables fast, secure and efficient transactions

1.2. Scope of the Document

This system will support real-time transaction processing, fraud detection, user account management, and transaction history. It is intended for merchants, customers, and financial institutions

1.3. Overview

This document provides functional, non-functional, and interface requirements, as well as constraints and expected timeline and budget for system development.

2. General Description

The system involves three main parties: customers, merchants, and payment gateways. It processes transactions through a series of secure validation steps.

It must handle high concurrency with minimal latency.

3. Functional Requirements
User authentication and account management
Transaction authorization and settlement
Fraud detection and alert system
Transaction ~~logs~~ logging and reporting.

~~4. Interference R~~

4. Interface Requirements
System must integrate with both web and mobile frontends
Must provide merchant POS systems ~~and~~ and bank APIs
Must support HTTPS and REST protocols

5. Performance Requirements
System should support 1000+ transactions per second.
Transaction confirmation must be generated within 2-3 seconds under normal load.

6. Design Constraints
Must comply with PCI - DSS standard for data security
Use TLS encryption for data in transit
Must support high availability and failover architecture
User interface must be responsive (web + mobile compatable)

7. Non-functional Requirements
99.99% uptime required for transaction availability
All sensitive data to be encrypted at rest and in transit
System should scale upto 10000 concurrent users
Full audit logs must be maintained and tamper-proof.

8. Preliminary Schedule and Budget

Schedule (6 months)

Month 1 – requirements analysis and architecture planning
Month 2 – user authentication module & initial UI mockups
Month 3 – Transaction engine & API integration with banks
Month 4 – Fraud detection, logging, and alert systems
Month 5 – Security testing, compliance verification & load testing
Month 6 – Final QA, deployment & documentation

Budget Breakdown

Developer Salaries	–	$180000
Security & Compliance Tools	–	$40000
Hosting & Infrastructure	–	$30000
Testing and QA	–	$20000
Project management & documentation	–	$30000

Total Cost – $300000