Credit Card SRS

Introduction

1.1 Purpose

The purpose of this document is to specify the requirements for a credit card processing system to be developed by the software development team at XYZ Inc. The system will allow merchants to accept credit card payments from customers in-store, online, and through mobile devices.

1.2 Scope

The credit card processing system will include the following features:

- Ability to authorize and capture credit card transactions in real-time
- Support for multiple payment gateways and acquiring banks
- Integration with point-of-sale systems, e-commerce platforms, and mobile devices
- Compliance with PCI DSS and other security standards
  Support for a variety of credit card types, including Visa, Mastercard, and American Express

1.3 Definitions, Acronyms, and Abbreviations

- PCI DSS: Payment Card Industry Data Security Standard
- POS: Point of Sale
- API: Application Programming Interface

Overall Description

2.1 Product Perspective

The credit card processing system will be a standalone application that integrates with a variety of payment channels. It will be designed to handle large volumes of transactions and ensure fast and reliable processing. The system will communicate with payment gateways and acquiring banks through secure APIs, and will support multiple payment methods, including credit cards, debit cards, and mobile payments

.

2.2 Product Functions

The credit card processing system will include the following functions:

- Authorization of credit card transactions in real-time
- Capture of authorized transactions
- Refund of captured transactions
- Voiding of authorized transactions
- Settlement of captured transactions with acquiring banks
- Reporting of transaction data to merchants and acquiring banks
  Management of merchant accounts and payment methods

2.3 User Characteristics

The credit card processing system will be used by merchants and their employees, who may have varying levels of technical expertise. The system will be designed to be intuitive and easy to use, with clear instructions and guidance provided throughout.

2.4 Operating Environment

The credit card processing system will be designed to operate on a variety of hardware and software platforms, including Windows, MacOS, iOS, and Android. The system will require a reliable internet connection to communicate with payment gateways and acquiring banks.

2.5 Design and Implementation Constraints

- The credit card processing system will be designed to comply with PCI DSS and other security standards. It will use encryption and other security measures to protect cardholder data and prevent fraud. The system will be implemented using a combination of programming languages and development frameworks, with a focus on modularity, scalability, and maintainability.

Specific Requirements

3.1 Functional Requirements

- The system must be able to authorize credit card transactions in real-time.
- The system must be able to capture authorized transactions and initiate settlement with acquiring banks.
- The system must support multiple payment gateways and acquiring banks, and provide failover mechanisms to ensure continuous operation.

- The system must support a variety of payment methods, including credit cards, debit cards, and mobile payments.
- The system must provide reporting capabilities to merchants and acquiring banks, including transaction data, settlement data, and chargeback data.
  The system must support account management and payment method management for merchants.

3.2 Non-Functional Requirements

- The system must comply with PCI DSS and other security standards.
- The system must be highly reliable and able to handle large volumes of transactions.
- The system must be easy to use and provide clear instructions

Interface Requirements

4.1 User Interfaces

- The credit card processing system must provide a user-friendly interface for merchants and their employees to manage transactions, view reports, and manage their accounts.
- The user interface must be intuitive and easy to navigate, with clear instructions and guidance provided throughout.
- The system must provide a consistent user experience across different platforms, including desktop, mobile, and tablet devices.
  The user interface must be responsive and fast, with minimal lag time between actions and system responses.

4.2 Hardware Interfaces

- The credit card processing system must be compatible with a variety of hardware devices, including card readers, barcode scanners, and printers.
- The system must support the integration of hardware devices through standard communication protocols, such as USB and Bluetooth.
  The system must provide clear instructions for hardware installation and configuration.

4.3 Software Interfaces

- The credit card processing system must integrate with a variety of software platforms, including point-of-sale systems, e-commerce platforms, and mobile applications.
- The system must provide APIs and other integration mechanisms to facilitate seamless communication with external systems.
- The system must provide clear documentation and guidance for software integration.

Performance Requirements

5.1 Response Time

- The credit card processing system must provide fast and responsive transaction processing, with an average response time of no more than 2 seconds.
- The system must provide real-time transaction status updates to merchants and acquiring banks.
  The system must be able to handle a large volume of transactions simultaneously, with no degradation in response time or system performance.

5.2 Availability

- The credit card processing system must be available 24/7, with minimal downtime for maintenance or system upgrades.
- The system must provide failover mechanisms to ensure continuous operation in the event of a system failure.
  The system must be capable of handling unexpected spikes in transaction volume without any impact on system availability or performance.

5.3 Scalability

- The credit card processing system must be scalable to handle future growth in transaction volume and user traffic.
- The system must provide a mechanism for adding processing capacity as needed, such as through the addition of servers or cloud-based infrastructure.

- The system must be designed to support horizontal scaling, with the ability to distribute processing across multiple servers or instances.