CREDIT CARD MANAGEMENT SYSTEM

Software Requirement Specification(SRS)

1 Introduction:

1.1 **Purpose of this Document:** At first, main aim of why this document is necessary and what's purpose of document is explained and described

1.2 **Scope of this document** – The credit card processing is usually done with a help of the swiper which scans all the details of the card. After each purchase, the details are recorded and document is made by the credit card issuer. Accepting credit cards is an integral part of business today for merchants who want to be competitive in their market and grow their business to its greatest potential. Software and Gateway processing helps reduce fraud losses, saves you time and money, and provides powerful features and performance, including detailed transaction records and reports.Swiping credit cards ensures lower rates, resulting in potential savings of hundreds of dollars each month for your business.

1.3 **Overview** – Credit card processing through offline involves the merchant collecting order information (, storing this in a database on your site, and entering it using their on-site merchant credit card processing system. Takes time to manually enter credit card information for each order. This solution creates following cons: · Insecure – there is a possibility that a skilled hacker could break into the database and steal an entire list of credit card numbers, thereby damaging the merchant's reputation with current client. ·

2 General description:

The system typically includes hardware and software components that work together to facilitate transactions. The hardware may include a point-of-sale (POS) terminal, card reader, or mobile device, while the software may include a payment gateway, merchant account, and fraud detection and prevention tools. The credit card processing system works by securely transmitting customer payment information to the card issuer or the card network, which then approves or declines the transaction based on factors such as available credit, fraud risk, and other security checks. Once approved, the payment is settled and funds are transferred from the customer's account to the merchant's account.

3 Functional Requirements:

- Allows merchants to accept credit and debit card payments from customers. This includes authorizing transactions, verifying cardholder information, and settling payments.
- Integrated with a payment gateway that securely transmits payment information between the merchant and the card issuer.
- Has measures in place to detect and prevent fraudulent transactions, including address verification, card verification, and real-time fraud screening.
- Allows merchants to manage their accounts, view transaction histories, and generate reports on payment activity.
- The system allows merchants to issue refunds and handle chargebacks in a timely and

efficient manner.
- Supports mobile payments, enabling customers to pay using their mobile devices.

4 Interface Requirements:

- Has a user-friendly interface that makes it easy for merchants to access and manage their accounts, view transaction history, and generate reports.
- Provides an API (Application Programming Interface) that allows merchants to integrate the payment processing functionality into their own software applications.
- Integrated with a payment gateway that securely transmits payment information between the merchant and the card issuer.
- Supports mobile payments, enabling customers to pay using their mobile devices.
- The system should support transactions in multiple currencies and provide currency conversion functionality.
- Provides a reporting interface that allows merchants to generate reports on transaction activity, settlement, and reconciliation.
- Provides a security interface that enables merchants to configure security settings and monitor security events.

5 Performance Requirements:

- The system should respond quickly to user requests and actions, with minimal latency and delay.
- The system should be scalable, capable of handling a large number of users and transactions, without compromising performance or reliability.
- The system should be available 24/7, with minimal downtime for maintenance or upgrades.
- The system should be reliable, with minimal errors or failures, and capable of recovering quickly from any disruptions.
- The system should be secure, protecting user data and transactions from unauthorized access or breaches.
- The system should undergo regular load testing, to ensure that it can handle peak loads and heavy traffic without compromising performance or availability.
- The system should be optimized for performance, with efficient algorithms, data structures, and processing techniques, to minimize resource usage and improve response time.

6 Design Constraints:

- The system design may be constrained by the hardware resources available, such as servers, storage devices, and network equipment.
- The system design may be constrained by the software resources available, such as the operating system, database management system, and programming languages.
- The system design may be constrained by the project timeline, which may limit the scope, features, and functionality of the system.
- The system design may be constrained by the available budget, which may limit the investment in hardware, software, and personnel.
- The system design may be constrained by regulatory requirements, such as data privacy laws, security standards, and compliance regulations.

- The system design may be constrained by the need to integrate with existing systems, such as payment gateways, reservation systems, and loyalty programs.
- The system design may be constrained by the need to ensure usability and accessibility, such as designing for users with disabilities, elderly users, or users with limited technical skills.

7 **Non-Functional Attributes:**

- The system should be user-friendly, easy to learn, and intuitive, with a well-designed user interface that enables users to perform tasks quickly and efficiently.
- The system should be reliable, with a low error rate and minimal downtime, ensuring that users can access and use the system at all times.
- The system should be secure, protecting user data and transactions from unauthorized access or breaches, and complying with regulatory requirements.
- The system should be fast and responsive, with minimal latency and delay, enabling users to perform tasks quickly and efficiently.
- The system should be scalable, able to handle a large number of users and transactions, without compromising performance or reliability.
- The system should be easy to maintain, with well-documented code, clear error messages, and easy-to-use debugging tools, enabling developers to identify and fix issues quickly.

8 **Preliminary Schedule and Budget:** The project is scheduled to be completed within three months of the start date. The budget is allotted only for the man-hours and not for different softwares used.