TANMAYI S BALIJA (1BM22CS359)

# CREDIT CARD PROCESSING

## Problem Statement:

Credit card processing offline involves the merchant collecting order information (including credit card numbers), 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 the 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 the current client. There is a higher risk of customer chargebacks with no signature · Higher risk of fraud for using stolen credit cards · Many discerning online shoppers will not give their credit card to an "untrusted" online merchant So there is a need for online and trusted credit card processing.

## Software Requirement Specification(SRS) of CREDIT CARD PROCESSING

- **Introduction**:

  Software requirement specification of credit card having the following sub topics introduction purpose of this document scope of this document overview and general description This Software Requirement Specification (SRS) document outlines the requirements for the development of a Credit Card Management System. This system will provide a centralized platform for managing credit card transactions, including processing payments, generating statements, managing credit limits, and monitoring fraud.

  **Purpose of this Document:** The purpose of this document is to provide a detailed outline of the requirements for the Credit Card Management System. This document will serve as a guide for the development team, ensuring that the final product meets the needs of both the business and the end-users. Scope of this Document: This document covers the functional and non-functional requirements for the Credit Card Management System. It outlines the features, constraints, and limitations of the system.

**General Description:** The Credit Card Management System is designed to provide a streamlined and centralized platform for managing credit card transactions. The system will be used by credit card issuers, banks, and financial institutions to process payments, generate statements, manage credit limits, and monitor fraud. The system will consist of several modules, including user management, transaction management, statement generation, credit limit management, and fraud detection. The system will be accessible through a web interface, which will allow users to access their account information, manage their credit card transactions, and make payments. The system will integrate with third-party payment gateways and financial institutions to ensure secure and efficient transaction processing. The system will also include advanced fraud detection algorithms to prevent fraudulent activities and ensure the security of user data. The Credit Card Management System will be developed using Java programming language and will be based on the Model-View-Controller (MVC) architectural pattern. The system will be designed to be scalable, secure, and reliable, ensuring that it can handle a large volume of transactions and support multiple users simultaneously. Overall, the Credit Card Management System will provide a robust and efficient platform for managing credit card transactions, ensuring that users can easily access and manage their accounts while providing financial institutions with the tools they need to manage credit card activities effectively.

**Purpose of this Document:** An SRS forms the basis of an organization's entire project. It sets out the framework that all the development teams will follow. It provides critical information to all the teams, including development, operations, quality assurance (QA) and maintenance, ensuring the teams are in agreement.

**Scope of this document** – The system is mainly for easy management of all the processes and operations regarding banking and transactions. The system will consist of an online framework like otp generation, remote withdrawal system, automation of bill payments, customer service. Some of the other important features of the system are review system, feedback system , enquiry and help features, security and record maintenance.

**Overview -** Credit card is a small plastic card issued to users as a system of payment. It allows its holder to buy goods and services based on the holder's promise to pay for these goods and services. The issuer of the card creates a revolving account and grants a line of credit to the consumer (or the user) from which the user can borrow money for payment to a merchant or as a cash advance to the user.When a purchase is made the merchant swipes the card. The information goes to a gateway processor, which either accepts or rejects the transaction. If it is accepted, the transaction is held until the end of the business day.

## Functional Requirements of Credit Card Processing System:

- **User Registration and Login:** The system shall allow merchants to register for a merchant account and login to the system using their registered credentials.
- **Payment Gateway Integration:** The system shall integrate with various payment gateways and processors to process credit card payments securely and efficiently.
- **Payment Processing:** The system shall allow merchants to process credit card payments from their customers, supporting multiple payment types, including one-time payments, recurring payments, and installment payments.
- **Payment Reporting:** The system shall generate reports on payment transactions, including payment amounts, payment status, and payment gateway information.
- **Fraud Detection:** The system shall include advanced fraud detection algorithms to prevent fraudulent activities and ensure the security of user data.
- **Customer Management:** The system shall enable merchants to manage their customer information, including customer details, billing information, and transaction history.
- **Transaction Processing:** The system shall allow merchants to process transactions, including authorizations, captures, voids, and refunds.
- **Payment Notifications:** The system shall notify customers and merchants of successful payments, failed payments, and payment disputes.
- **User Roles and Permissions:** The system shall have different user roles, including merchant, administrator, and customer. Each role shall have different permissions and access levels.
- **Security:** The system shall be designed to ensure the security and confidentiality of user data and shall comply with industry security standards.
- **Error Handling:** The system shall handle errors and exceptions gracefully and provide appropriate error messages to users.
- **System Administration:** The system shall enable administrators to manage system settings, including payment gateway configurations, fraud detection settings, and user roles and permissions.

Overall, the Credit Card Processing System's functional requirements include user registration and login, payment gateway integration, payment processing, payment reporting, fraud detection, customer management, transaction processing, payment notifications, user roles and permissions, security, error handling, and system administration.

- **Interface Requirements:** The software developed is very easy to navigate. The person has to register and login. Upon clicking on a tab, it traverses to the destined page and lets the user to avail the functionalities of the app. The app is designed using Flutter. There are many buttons via which the user can set his records, select an exercise/ asana, etc. The home page is the landing page of the app. The users are able to view the diets required for their health condition and follow it.

## Interface Requirements of a Credit Card Processing System:

User Interface: The system shall have a user interface that is easy to use, intuitive, and accessible to all users, including merchants and customers.
Payment Gateway Integration Interface: The system shall integrate with various payment gateways and processors through APIs, SDKs, or other integration methods.
Customer Payment Interface: The system shall provide a payment interface for customers to enter their credit card information and complete the payment process.
Payment Notification Interface: The system shall provide an interface for sending payment notifications to customers and merchants, including successful payment notifications, failed payment notifications, and payment dispute notifications.
Reporting Interface: The system shall provide a reporting interface for generating payment reports, including payment amounts, payment status, and payment gateway information.
Customer Management Interface: The system shall provide a customer management interface for managing customer information, including customer details, billing information, and transaction history.
Transaction Processing Interface: The system shall provide a transaction processing interface for processing authorizations, captures, voids, and refunds.
Error Handling Interface: The system shall provide an error handling interface for handling errors and exceptions gracefully and providing appropriate error messages to users.
Security Interface: The system shall provide a security interface for implementing security measures, including encryption, tokenization, and PCI DSS compliance.
Overall, the Credit Card Processing System's interface requirements include a user interface, payment gateway integration interface, customer payment interface, payment notification interface, reporting interface, customer management interface, transaction processing interface, error handling interface, and security interface. These interfaces must be designed to be user-friendly, efficient, and secure, meeting the needs of merchants and customers.

- **Performance Requirements:** The software consists of multiple pages which provide various features to the user. The basic features include:
  - **Login and sign-up page :**The login page allows a registered user to login to the app with the registered email and verified password. The sign-up page helps a user to create an account to login to the app.
  - **Home page**:The home page is the navigation page from which a user can navigate to any category. The user can logout, share the app over any known platform, provide ratings and reviews/ recommendations for the app.
  - **Yoga page :**The yoga page displays a series of yoga asana, which the user can do by clicking on the start button. Once the timer starts, the screen displays GIFs of the asana to be done, and also the timer for the particular asana.
  - **Exercise page :**The exercise page displays a series of exercises, done with equipments, which the user can do by clicking in the start button. Once the exercise has started, the

screen displays GIFs of the exercises to be done, and also the timer for the particular exercise.

- **Home-workout page** :The home-workout page displays a series of exercises, which the user can do by clicking in the start button. The exercise page shows exercises with equipments and the home-workout page shows exercises without equipments.
- **Diet page** :The diet page helps the user to know what to eat according to his health condition. There are many health diets available: general diet, diet for high blood pressure, diet for diabetes, and also diet for acidity. The user can be educated about what to eat when they are experiencing a health condition.
- **Tracker page** :The tracker page maps the route travelled by the user. It shows how much the user has travelled with his phone. It also provides a route when a source and destination is selected. The view of the map can be changed, according to the users' convenience.
- **Settings page** : The settings page helps a user to provide his details. He can make changes in his account. There is a provision for logging out of the current account and even deleting the account. The user can even send feedback about the app.
- **Form page** :The form page is used to collect all required details of a user. All details include name, email, height, weight, underlying medical issues (if any) and preferred level of training.
- **Timer functionality** :Each exercise has a designated time slot and the user has to perform the exercise for the entire time. There will be a break time between each exercise and the user will be directed to the finish page where they can opt to redo the exercise or go to the home page.

## Performance Requirements of a Credit Card Processing System:

**Response Time:** The system must provide fast response times for processing credit card payments. The response time for payment processing should be less than 2 seconds.

**Transaction Volume:** The system must be able to handle a high volume of credit card transactions without any performance degradation. The system should be designed to handle at least 10,000 transactions per hour.

**Availability:** The system must have a high level of availability to ensure that merchants can process credit card payments without interruption. The system should have an uptime of at least 99.99%.

**Scalability:** The system must be able to scale up or down to handle changes in transaction volume. The system should be able to handle an increase in transaction volume by at least 50% without any performance degradation.

**Load Testing:** The system must be load tested to ensure that it can handle the expected transaction volume. Load testing should be performed before the system goes live and periodically thereafter.

**Data Storage:** The system must be able to store large amounts of data securely and efficiently. The system should be designed to handle at least 1 terabyte of data.

**Concurrent Users:** The system must be able to handle a large number of concurrent users without any performance degradation. The system should be able to handle at least 1000 concurrent users.

**Network Latency:** The system must be designed to handle network latency and ensure that credit card transactions can be processed quickly even with network delays.

Overall, a Credit Card Processing System's performance requirements include response time, transaction volume, availability, scalability, load testing, data storage, concurrent users, and network latency. These requirements must be designed to ensure that the system can handle high transaction volumes, provide fast response times, and be highly available and scalable.

- ## Design Constraints:
  - Programming language to be used is Flutter.
  - Storage of the data will be done on Firebase.
  - Android systems are compatible with the app.
  - Internet availability is a must.
  - Minimum Processor 1 GHz, 512 MB RAM and 850 MB free HDD for 32-bit or 2 GB for 64-bit.
  - Windows 10 and above, Windows server 2003, Windows server 2008, Windows server 2012, Windows server 2016.
  - Internet connection of 4 MBPS or higher.

- **Non-Functional Attributes:**
  - Availability
  - Scalability
  - Reliability
  - Performance
  - Security
  - Usability
  - Extensibility

- **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.