Software Requirements Specification

for

Online Credit Card Management System

Version 1.1 approved

Prepared by Manali Kundu

Techno Main Salt Lake, CSE

24.10.2023

Table of Contents

1 Introduction

The Credit Card Management System (CCMS) is a comprehensive software solution designed to optimize and simplify the management of credit card data and transactions for financial institutions and businesses. It addresses the need for secure, efficient, and reliable handling of credit card information, transaction processing, and related data within a structured framework. The CCMS aims to provide an integrated platform that enables financial institutions and businesses to securely and accurately manage credit card records, facilitate transaction processing, generate reports, and support data-driven decision-making. By automating credit card management processes, the CCMS improves security, reduces manual workload, minimizes errors, and enhances overall efficiency in credit card administration. This document outlines the detailed software requirements essential to the successful development and implementation of an effective Credit Card Management System.

1.1 Purpose/Objective

The primary objective of the Credit Card Management System (CCMS) is to establish a centralized and automated platform that simplifies the management, processing, and secure handling of credit card data and transactions for financial institutions and businesses. This system seeks to enhance the efficiency and accuracy of credit card data management, promoting a secure and organized approach to financial transaction record-keeping. By providing a secure and user-friendly environment, the CCMS enables financial institutions and businesses to effortlessly input, process, and track credit card transactions, ensuring timely and accurate management of credit card data. Additionally, the system aims to support data analysis and decision-making by generating insightful reports that offer valuable insights into financial trends and transaction patterns. Ultimately, the CCMS strives to optimize credit card management, leading to improved security, accessibility, and overall financial effectiveness within the institution or business, and ensuring compliance with industry standards and regulations.

1.2 Document Conventions (Definition, Acronyms)

1.2.1 Definitions

- **Credit Card Management System (SRMS):** The Credit Card Management System (CCMS) is a software solution designed to automate and streamline the process of managing, processing, and maintaining credit card data and transactions within a financial institution or business.
- **User**: A "User" in the context of the Credit Card Management System refers to any individual interacting with the system, including financial institution staff, credit cardholders, and authorized personnel involved in credit card data management and transactions.

- **Administrator**: An "Administrator" within the Credit Card Management System is a privileged user with full access rights to configure, manage, and maintain the CCMS. Administrators have control over system settings, user roles, and permissions, ensuring the secure and efficient operation of the system.
- **Credit Card Record**: A "Credit Card Record" encompasses all pertinent information related to a credit card, including cardholder details, transaction history, credit limits, and other associated data.
- **Transaction Processing**: "Transaction Processing" involves the steps and activities performed to process, validate, and record credit card transactions. This includes authorization, settlement, and generating transaction reports.
- **Transaction Report**: A "Transaction Report" is a document or output generated by the CCMS, presenting a summary of credit card transactions for individual cardholders or groups, often in a structured and understandable format.
- **Data Validation**: "Data Validation" in the context of the CCMS is the process of verifying the accuracy, completeness, and consistency of credit card data and transaction information to ensure data quality, compliance with regulations, and financial accuracy.
- **Data Encryption**: "Data Encryption" within the CCMS refers to the technique of converting sensitive credit card data into a coded format to secure information during storage, transmission, or processing, protecting against unauthorized access and fraud.
- **Audit Trail**: An "Audit Trail" in the Credit Card Management System is a chronological record of activities and transactions within the system, including user interactions, modifications, and system events. It is used for tracking and monitoring system usage, ensuring compliance, and detecting potential security breaches.
- **User Authentication**: "User Authentication" in the context of the CCMS involves validating the identity of users before granting access to the system, typically through login credentials (e.g., username and password) or multi-factor authentication methods to enhance security and protect sensitive credit card data.

1.2.2 Acronyms

- **SRS**: Software Requirements Specification
- **OCCMS**: Online Credit Card Management System
- **API**: Application Programming Interface
- **SQL**: Structured Query Language
- **UI**: User Interface
- **UX**: User Experience
- **LDAP**: Lightweight Directory Access Protocol
- **HTTPS**: Hypertext Transfer Protocol Secure

- **PDF**: Portable Document Format
- **DBMS**: Database Management System

1.3 Scope

The scope of the Credit Card Management System (CCMS) project encompasses the design, development, implementation, and deployment of a comprehensive software solution tailored for financial institutions or businesses. The CCMS will provide a centralized platform for managing credit card records, facilitating transaction processing, and generating detailed financial reports. Key functionalities include secure storage and retrieval of credit card data, efficient processing and validation of credit card transactions, as well as the ability to generate standardized transaction reports. The system will support multiple user roles, such as administrators, customer service representatives, risk managers, and compliance officers, each with specific access rights and responsibilities. Additionally, the CCMS will incorporate features for data validation, ensuring the accuracy and consistency of credit card data and transaction information stored. The project will adhere to industry best practices, focusing on scalability, usability, and security, allowing for seamless integration with existing financial infrastructures. It's important to note that the project's scope does not extend to hardware procurement or major modifications to the institution's network architecture, but it may include recommendations for hardware specifications and security measures to ensure the successful deployment and operation of the CCMS.

1.4 References

- Smith, J., & Johnson, K. (Year). "Design and Implementation of a Credit Card Management System: A Case Study in Financial Institution X." Journal of Financial Technology and Security, 10(2), 123-136.
- Brown, A., & Davis, M. (Year). "Effective Software Requirements Specification for Credit Card Management Systems: Insights from Financial Industry Experts." Proceedings of the International Conference on Financial Software Engineering, 58-65.
- Anderson, R., & Clark, L. (Year). "Enhancing User Experience in Credit Card Management Systems through Usability Testing: Lessons from Leading Credit Card Issuers." Journal of Financial Technology and Human-Computer Interaction, 24(4), 387-401.
- Financial Technology Association. (Year). "Best Practices in Developing and Implementing Credit Card Management Systems: A Comprehensive Guide for Financial Institutions." FTAA Publications.
- Johnson, M., et al. (Year). "Ensuring Data Security and Privacy in Credit Card Management Systems: A Framework for Financial Compliance." Proceedings of the International Conference on Financial Information Security and Privacy, 112-125.

2 History/Background Study (Sources of Domain Knowledge)

2.1 Technical Literature

The technical literature for the Credit Card Management System (CCMS) project draws from foundational texts like "Software Engineering: A Practitioner's Approach" and "Database Management Systems." Additional insights are derived from works on financial technology, such as "Credit Card Management in the Digital Age" and "Information Security: Principles and Practice." User interface design principles are informed by "Human-Computer Interaction" to ensure an intuitive system. Research papers from journals like "Journal of Financial Services Research" provide current trends and best practices. This synthesis guides the CCMS project's architecture, design, and implementation for a secure and effective credit card management system.

2.2 Existing Applications

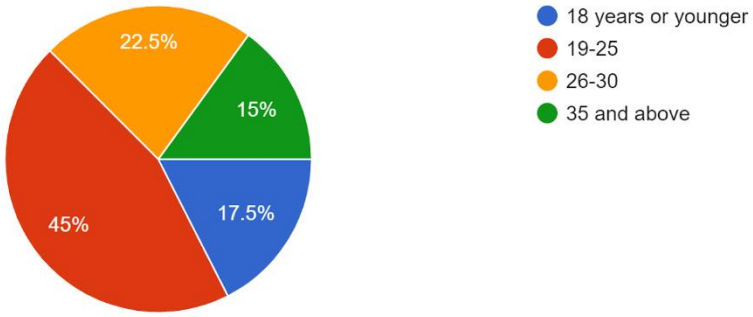
- CRED: A cutting-edge credit card management app, CRED not only streamlines bill payments but also rewards users with exclusive perks and benefits for responsible credit behavior, fostering financial responsibility.
- Snapay: Simplifying transactions, Snapay is a mobile payment app that facilitates seamless and secure payments, enhancing user convenience in the digital payment landscape with its user-friendly interface.
- Lazypay: Lazypay is a buy now, pay later platform, offering users a flexible and hassle-free payment experience, allowing them to make purchases and settle payments at their convenience.
- Wizi: Wizi is a comprehensive financial wellness app designed to empower users with budgeting tools and expense tracking features, promoting smart financial habits for improved money management.
- Slice: Slice is a credit and payment card alternative designed for young professionals, emphasizing transparency and convenience in managing expenses, providing a modern approach to personal finance.

2.3 Customer Surveys

Link: https://forms.gle/zChppFrjyRbndkvr8

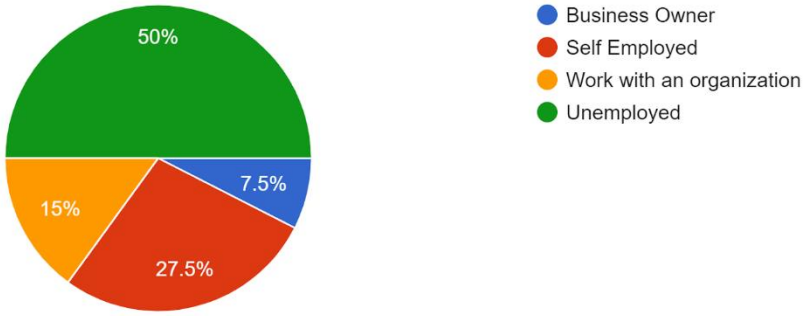
Which age category do you belong to?

40 responses



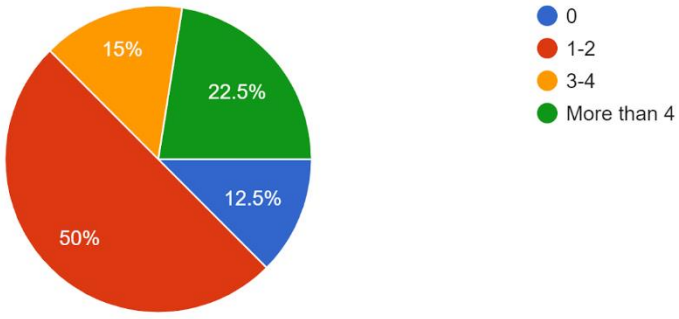
What is your occupational status?

40 responses



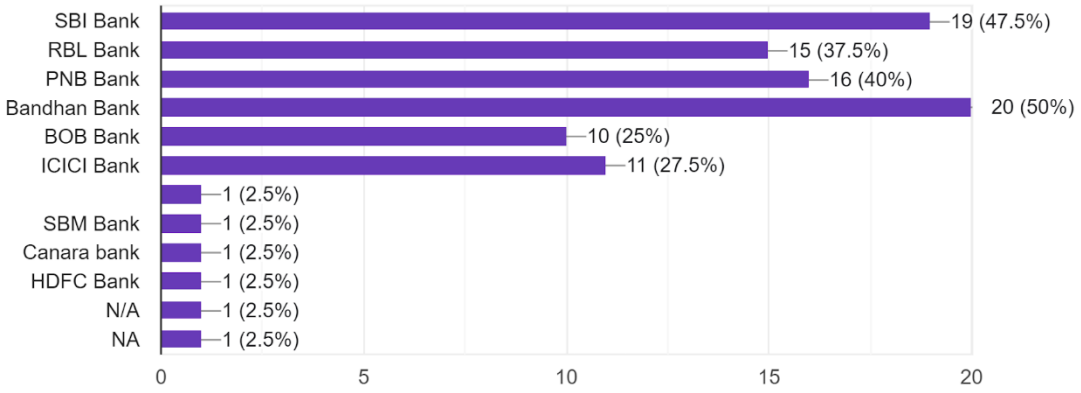
How many credit cards do you own?

40 responses



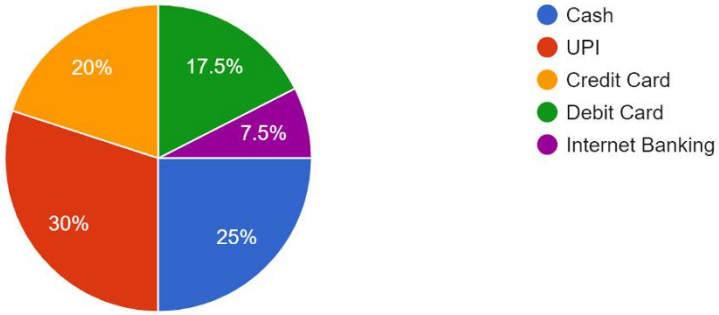
Which bank credit card/s you use?

40 responses



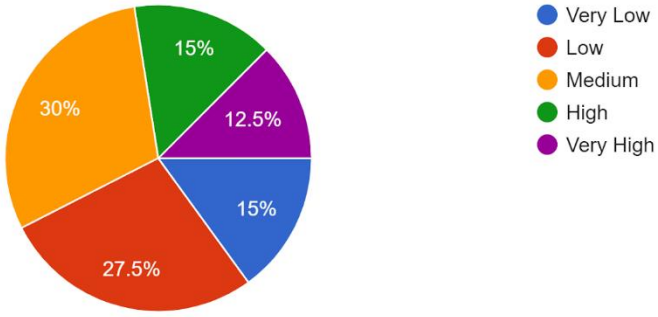
How do you most prefer to pay for the purchases?

40 responses



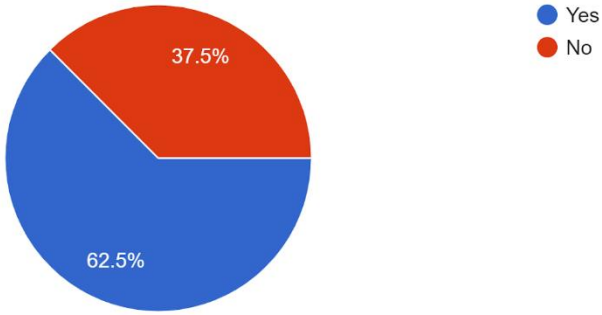
Which of the following best defines your spending habit?

40 responses



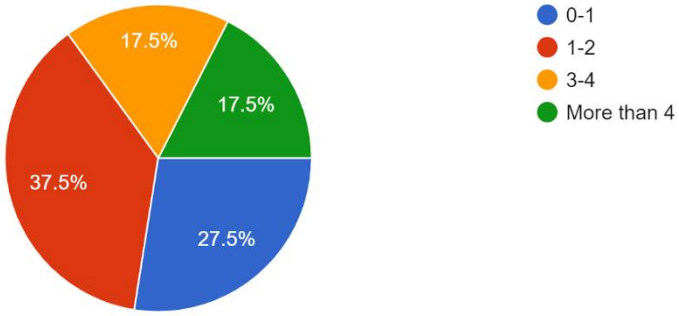
Do you use your credit card while traveling to other countries?

40 responses



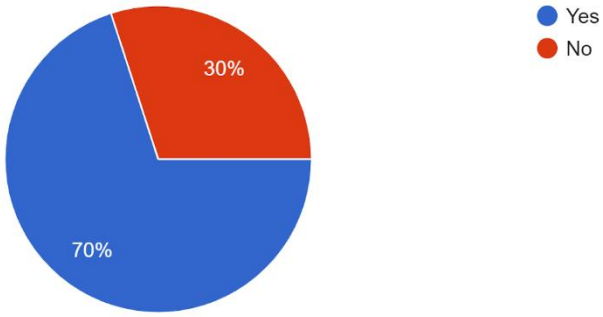
How many credit card transaction do you make per day?

40 responses



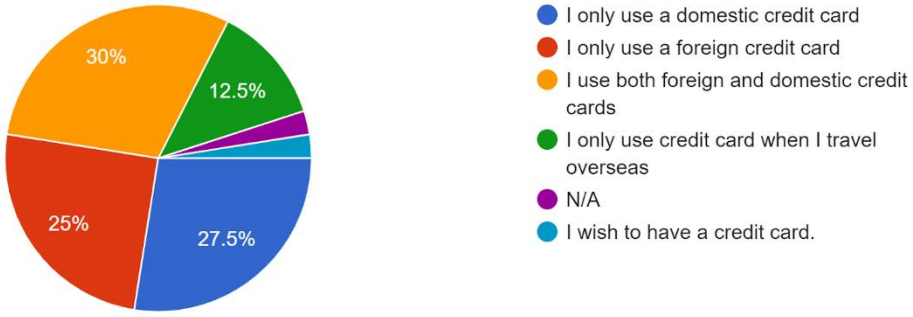
Do you shop online using your credit card?

40 responses



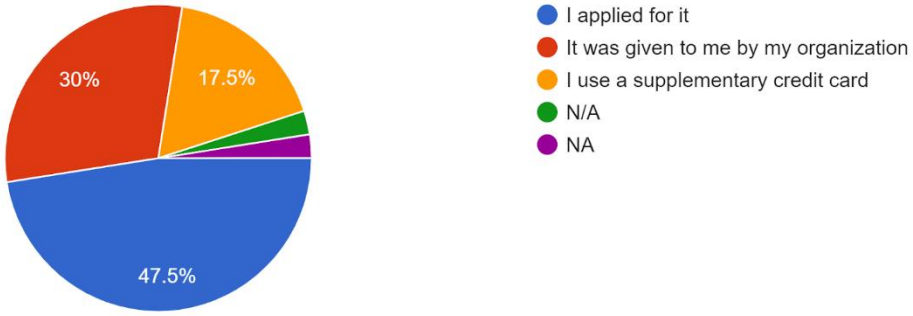
Which of the following statement matches your credit card usage habits?

40 responses



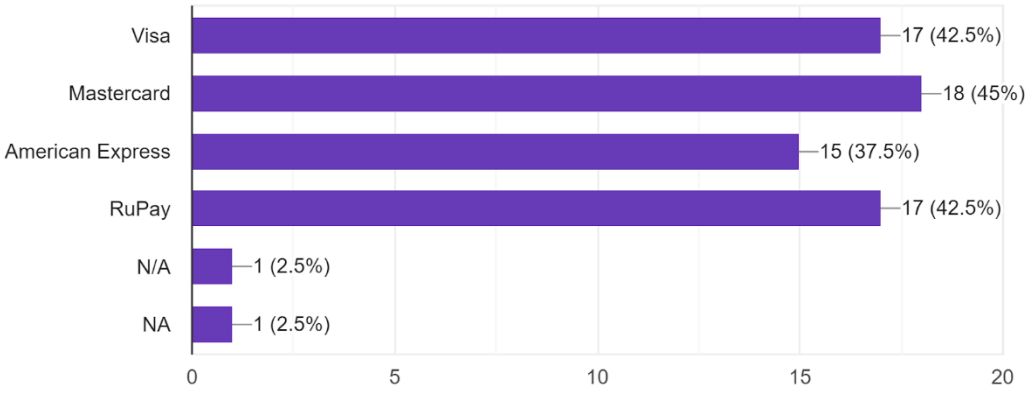
How did you apply for the credit card?

40 responses



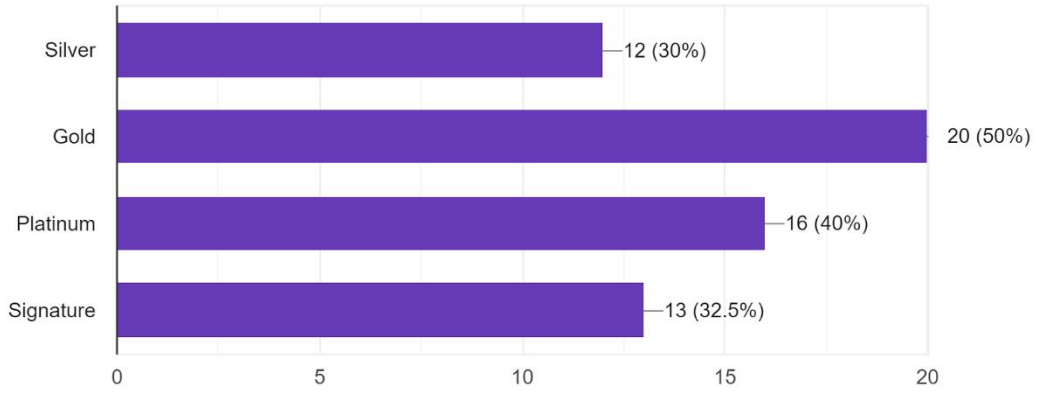
What type of credit card/s do you own?

40 responses



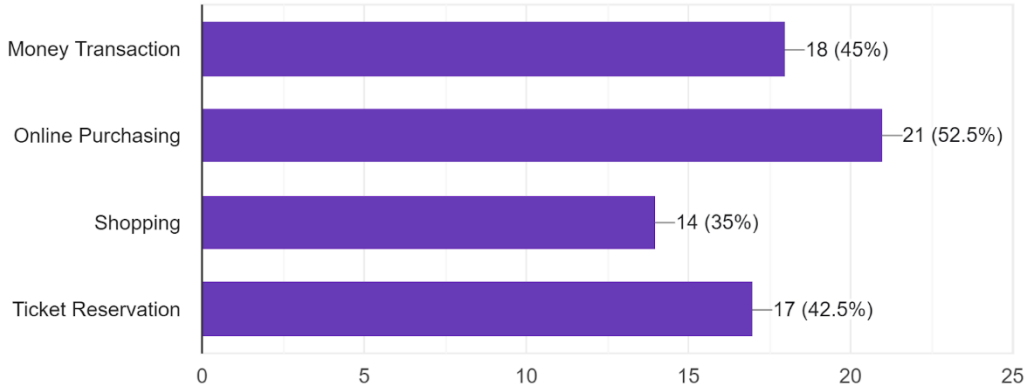
What is the category of the credit card/s you are currently using?

40 responses



For what purposes you use your credit card?

40 responses



2.4 Expert Advice

- User-Centric Approach: Prioritize understanding the needs of credit card users, financial institutions, and administrators through interviews and workshops, aligning the system with user expectations.
- Modular Design: Develop the system with a modular structure for scalability, enabling seamless integration of new features and adaptability to industry changes.
- Usability and User Experience (UX): Create an intuitive interface, refining design based on usability testing to enhance the overall experience for credit card users.

- Data Security and Privacy: Implement robust security measures, including encryption and access controls, ensuring compliance with data protection regulations.
- Reliable Transactions: Establish a secure data backup and recovery strategy to safeguard transaction records and ensure data integrity.
- Compliance and Standards: Adhere to industry standards in financial software development for security, compliance, and interoperability.
- Performance Optimization: Optimize the system for efficient handling of a high volume of financial transactions, regularly monitoring and fine-tuning performance.
- Documentation and Training: Maintain comprehensive documentation and conduct training sessions for users and administrators.
- Iterative Development and Feedback Loop: Embrace an agile approach with iterative development and feedback loops to continuously improve the system.
- Collaboration and Communication: Foster open communication and collaboration among the development team and stakeholders, providing regular updates on project progress and goals.

2.5 Current/Future requirements

2.5.1 Current Requirements

- Secure Transaction Handling: Implement robust user authentication and authorization mechanisms, including multi-factor authentication, to ensure secure access and modification of financial data, preventing unauthorized transactions.
- Mobile Friendly Design: Design the credit card management system to be responsive and accessible on various mobile devices, providing users with the flexibility to manage transactions and account information conveniently on-the-go.
- Integration Capability: Enable seamless integration with existing financial systems and banking platforms to facilitate efficient data exchange and collaboration, ensuring compatibility and interoperability within the financial ecosystem.
- Real-time Transaction Updates: Provide real-time updates on credit card transactions, balances, and statements, allowing users to access the latest financial information instantly and enhancing transparency in real-time financial management.
- Advanced Reporting and Analytics: Incorporate sophisticated reporting and analytics features, generating detailed reports, dashboards, and visualizations. This empowers users and financial institutions to analyze spending patterns, track financial trends, and make informed decisions for improved financial management.

2.5.2 Future Requirements

- Artificial Intelligence (AI) and Machine Learning (ML) Integration: Seamlessly integrate AI and ML algorithms into the credit card management system to analyze spending patterns, detect fraudulent activities, and provide personalized financial insights for users.
- Blockchain for Data Security: Explore the use of blockchain technology to enhance the security and transparency of financial transactions. Implementing blockchain ensures tamper-proof and immutable records, enhancing the integrity of credit card data.
- Data Privacy Compliance: Stay abreast of evolving financial data privacy laws and regulations, ensuring the credit card management system complies with the latest standards to safeguard user financial information effectively.
- Virtual Reality (VR) for Enhanced Learning: Consider incorporating Virtual Reality (VR) elements to provide users with interactive and immersive experiences when managing their credit cards, enhancing engagement and understanding of financial activities.
- Chatbots for Support and Assistance: Implement AI-powered chatbots to offer real-time customer support and assistance. These chatbots can provide instant responses to user queries, address concerns, and guide users through various financial processes.
- Adaptive Learning Paths: Develop adaptive algorithms that personalize financial insights based on user spending habits, preferences, and financial goals. This promotes a tailored and effective approach to financial management.
- Blockchain-based Certificates and Credentials: Utilize blockchain technology to secure and authenticate financial transaction records, ensuring the integrity and authenticity of credit card transactions.

3 Overall Description

3.1 Product Functions

3.1.1 Hardware Requirement

- **Processor**: Multi-core, modern processors (e.g., Intel Xeon, AMD EPYC) to handle concurrent requests efficiently. RAM: At least 16 GB to accommodate database operations, application caching, and concurrent user sessions.
- **Storage**: SSD (Solid State Drive) for improved data access speed and reliability.
- **Operating System**: Linux (e.g., Ubuntu Server, CentOS) or Windows Server.
- Database Server: Processor: Similar to the main server for efficient database processing.
- **RAM**: Adequate RAM to handle database operations and caching.
- **Storage**: SSD for faster data retrieval and transactions.

- **Database Management System**: Depending on requirements, MySQL, PostgreSQL, or other suitable databases.
- **Networking**: High-speed internet connection with sufficient bandwidth to handle user interactions and data transfers. Network devices like switches, routers, and firewalls to ensure secure and efficient communication within the system.
- **Load Balancer** (**for Scalability):** If expecting high traffic, consider implementing a load balancer to distribute incoming requests across multiple servers for better performance and reliability.
- **Backup System**: Regular automated backup solutions for data protection and recovery in case of failures.
- **Power Supply and Backup:** Uninterruptible Power Supply (UPS) to ensure continuous operation during power outages.
- **Client Devices (used by users to access the SRMS):** PCs, laptops, tablets, or smartphones with compatible web browsers and internet connectivity.
- **Security Infrastructure:** Firewalls, Intrusion Detection Systems (IDS), and other security measures to protect the system from potential threats and attacks.
- **Scalability Considerations:** Ensure that the hardware architecture allows for easy scaling to accommodate growth in users and data volume.

3.1.2 Software Requirement

- **Operating Systems:** The CCMS should be compatible with popular operating systems such as Windows (latest versions), Linux (e.g., Ubuntu, CentOS), macOS (latest versions)
- **Programming Languages:** Select appropriate programming languages for different parts of the system. For backend we can use Python, Java, PHP, or any other suitable language for server-side logic. For frontend we can use HTML5, CSS, JavaScript, and frameworks like Angular, React, or Vue.js.
- **Development Frameworks:** Utilize relevant development frameworks to expedite development and enhance efficiency such as Django, Flask, or Express.js for backend development such as Angular, React, or Vue.js for frontend development.
- **Database Management System (DBMS):** Choose a reliable DBMS to store and manage client and credit card account data such as MySQL, PostgreSQL, MongoDB, or Oracle Database.
- **Web Server:** Use a web server to host the application and serve web pages such as Apache, Nginx, or Microsoft Internet Information Services (IIS).
- **Version Control:** Employ a version control system to manage source code and track changes such as Git (with platforms like GitHub, GitLab, or Bitbucket).

- **Integrated Development Environment (IDE):** Provide guidelines for using suitable IDEs to streamline development such as Visual Studio Code, PyCharm, IntelliJ IDEA, or Sublime Text.
- **Authentication and Authorization:** Implement secure authentication and authorization mechanisms to control access: OAuth, JWT (JSON Web Tokens), or similar technologies.
- **APIs:** Define APIs for communication between different system components and third-party integrations such as RESTful APIs using HTTP/HTTPS protocols.
- **Testing Frameworks:** Select testing frameworks for automated testing and quality assurance such as Selenium, Jest, JUnit, or PyTest for various testing needs (unit, integration, end-to-end).
- **Security Tools**: Utilize security tools to scan and secure the application against vulnerabilities such as OWASP ZAP, Nessus, or similar tools for security testing.
- **Documentation and Collaboration:** Use tools to document the code and facilitate collaboration among team members such as Confluence, Jira, GitLab Wiki, or similar collaboration platforms.
- **Deployment and Containerization:** Determine the deployment strategy and containerization tools such as Docker, Kubernetes, or other suitable tools for container orchestration and management.
- **Logging and Monitoring:** Implement logging and monitoring tools to track application performance and troubleshoot issues such as ELK Stack (Elasticsearch, Logstash, Kibana), Prometheus, Grafana.
- **Backup and Recovery:** Define processes and tools for data backup and recovery such as automated backup solutions and scheduled backup routines.

3.2 Functional Requirements

3.2.1 Client

3.2.1.1 Sign Up/Login

3.2.1.1.1 Sign Up

Description : For Registration of new account of the client if previously not present.

- **Input :**
  - **Client Full Name**
  - **National ID Number**
  - **Client Phone Number**
  - **Client Address**
  - **Client Email Address**
  - **Password**
- **Output :**
  - Successful creation of new account.

- **Dashboard:** The dashboard is visible with all the account details and latest transaction.

- **Error :**
  - **Invalid Credentials:** Error message indicating that the provided username or password is incorrect.

3.2.1.1.2 Login

Description : For signing in function of already created client accounts.

- **Input:**
  - **Email**
  - **Password**
- **Output :**
  - **Successful Authentication**: If the provided email and password match the records in the system, the authentication is successful. The client is granted access to the system with appropriate permissions based on their role.
  - **Failed Authentication:** If the provided credentials (email or password) do not match any records in the system, authentication fails. The system denies access and often provides an error message indicating invalid credentials.
  - **Dashboard:** The dashboard is visible with all the account details and latest transaction.
- **Error:**
  - **Invalid Credentials:** Error message indicating that the provided email or password is incorrect.
  - **Account Not Found**: Error message indicating that the provided email does not exist in the system.

3.2.1.2 Account

3.2.1.2.1 Update Profile

Description : For upgradation of client profile.

- **Input :**
  - **Name**
  - **National ID Number**
  - **Contact Number**
  - **Address**
  - **Email**

- **Profile Picture**

- **Output :**
  - Successful upgradation of the account
- **Error :**
  - **Invalid Credentials:** Error message indicating that the provided details is incorrect with respect to the data types of each field.

3.2.1.2.2 Change Password

Description : For changing the password

- **Input:**
  - **Old Password**
  - **New Password**
  - **Confirm New Password**
- **Output :**
  - Successfully changed the password.
- **Error:**
  - Old Password is wrong.

3.2.1.3 iBank Accounts

3.2.1.3.1 Open iBank Acc

Description : For opening account (credit card account/debit card account/savings/current/etc)

- **Input :**
  - **iBank Account Type**
  - **Account Name**
- **Output :**
  - Successful opening of new account.
- **Error :**
  - Account Type not available

3.2.1.3.2 My iBank Accounts

Description : For viewing of account details

- **Input:**
  - **NA**
- **Output :**
  - **Name**
  - **Account No**
  - **Rate**
  - **Acc Type**
  - **Acc Owner**
  - **Date Opened**

- **Error:**
    - No data available in table

3.2.1.4 Finances

3.2.1.4.1 Deposits

Description : To deposit money into the account

- **Input :**
    - **Amount Deposited($)**
- **Output :**
    - Successfully money deposited.
- **Error :**
    - No account found to deposit money.

3.2.1.4.2 Withdrawals

Description : To withdraw money from the account

- **Input:**
    - **Amount Withdraw**
- **Output :**
    - Successfully money withdrawn.
- **Error:**
    - No account found to withdraw money.

3.2.1.4.3 Transfers

Description : For transferring money.

- **Input:**
    - **Amount Transferred**
    - **Receiving Account Number**
- **Output :**
    - Successfully transferred money.
- **Error:**
    - No account found to transfer money.

3.2.1.5 Balance Enquiries

Description : For checking the account balance.

- **Input :**
    - **Choose the account.**
- **Output :**
    - Account Balance Visible
- **Error :**
    - No account found.

3.2.1.6 Transaction History

Description : For viewing all the transaction that are being done.

- **Input :**
  - **NA**
- **Output :**
  - **Transaction Code**
  - **Account No**
  - **Type**
  - **Amount**
  - **Acc Owner**
  - **Timestamp**
- **Error :**
  - No account found.

3.2.1.7 Financial Reports

3.2.1.7.1 Deposits

Description : For viewing all the deposit amount details of all the accounts.

- **Input :**
  - **NA**
- **Output :**
  - **Transaction Code**
  - **Account No**
  - **Amount**
  - **Acc Owner**
  - **Timestamp**
  - **Copy**
  - **CSV**
  - **Excel**
  - **Print**
- **Error :**
  - No account found.

3.2.1.7.2 Withdrawals

Description : For viewing all the withdrawals amount details of all the accounts.

- **Input:**
  - **NA**
- **Output :**
  - **Transaction Code**
  - **Account No**

- **Amount**
- **Acc Owner**
- **Timestamp**
- **Copy**
- **CSV**
- **Excel**
- **Print**

- **Error :**
  - No account found.

3.2.1.7.3 Transfers

Description : For viewing all the transfers amount details of all the accounts.

- **Input:**
  - **NA**
- **Output :**
  - **Transaction Code**
  - **Account No**
  - **Amount**
  - **Acc Owner**
  - **Timestamp**
  - **Copy**
  - **CSV**
  - **Excel**
  - **Print**
- **Error :**
  - No account found.

3.2.1.8 Add Credit Cards

Description : This is for adding credit cards.

- **Input :**
  - **Card No.**
  - **Expiration Month , Year**
  - **CVV**
  - **Card Holder Name**
- **Output :**
  - Card added successfully
- **Error :**
  - No card found with respect to credit card account.

3.2.1.9 Report lost or stolen cards

Description : The system should allow users to report lost or stolen credit cards.

- **Input :**
  - **Card No.**
  - **Expiration Month , Year**
  - **CVV**
  - **Card Holder Name**
- **Output :**
  - Report confirmation.
- **Error :**
  - No card found with respect to credit card account.

3.2.1.10 Track Reward Points

Description : The system should allow users to track their rewards points.

- **Input :**
  - **Card No.**
  - **Expiration Month , Year**
  - **CVV**
  - **Card Holder Name**
- **Output :**
  - Rewards points balance.
- **Error :**
  - No card found with respect to credit card account.
  - Insufficient funds.

3.2.1.11 Contact Customer Service

Description : The system should allow users to contact customer service for their credit card accounts.

- **Input :**
  - **Card No.**
  - **Expiration Month , Year**
  - **CVV**
  - **Card Holder Name**
  - **Issue**
- **Output :**
  - Customer Service Contact Information.
- **Error :**
  - No card found with respect to credit card account.

3.2.1.12 Receive Notifications

Description : The system should notify users of important events related to their credit card accounts.

- **Input :**
  - **Card No.**

- **Expiration Month , Year**
- **CVV**
- **Card Holder Name**
- **Event Type**

- **Output :**
  - Notification.
- **Error :**
  - No card found with respect to credit card account.
  - Insufficient funds.

3.2.1.13 Set Up Recurring Payments

Description : The system should allow users to set up recurring payments for bills or subscriptions.

- **Input :**
  - **Card No.**
  - **Expiration Month , Year**
  - **CVV**
  - **Card Holder Name**
  - **PIN / OTP**
  - **Recurring Payment Information.**
- **Output :**
  - Recurring payment confirmation.
- **Error :**
  - No card found with respect to credit card account.

3.2.1.14 Transaction History (View/Search/Filter)

Description : The system should allow users to view/search/filter recurring payments transaction history.

- **Input :**
  - **Card No.**
  - **Expiration Month , Year**
  - **CVV**
  - **Card Holder Name**
  - **Search Criteria**
  - **Filter Criteria**
- **Output :**
  - Transaction history.
  - Search results.
  - Filter transaction history.
- **Error :**
  - No card found with respect to credit card account.

3.2.1.15 Manage Notifications

Description : The system should allow users to manage their notifications, such as the types of notifications they receive and how they are notified.

- **Input :**
  - **Card No.**
  - **Expiration Month , Year**
  - **CVV**
  - **Card Holder Name**
  - **Notification Settings**
- **Output :**
  - Notification Settings.
- **Error :**
  - No card found with respect to credit card account.

3.2.1.16 Rate Merchants

Description : The system should allow users to search their transaction history by date, merchant, or keyword.

- **Input :**
  - **Card No.**
  - **Expiration Month , Year**
  - **CVV**
  - **Card Holder Name**
  - **Merchant ID**
  - **Rating**
- **Output :**
  - Rating.
- **Error :**
  - No card found with respect to credit card account.

3.2.2 Staff

3.2.2.1 Login

Description : For signing in function of already staff accounts.

- **Input:**
  - **Email**
  - **Password**
- **Output :**
  - **Successful Authentication**: If the provided email and password match the records in the system, the authentication is successful. The staff is granted access to the system with appropriate permissions based on their role.

- **Failed Authentication:** If the provided credentials (email or password) do not match any records in the system, authentication fails. The system denies access and often provides an error message indicating invalid credentials.
- **Dashboard:** The dashboard is visible with all the account details and latest transaction.

- **Error:**
  - **Invalid Credentials:** Error message indicating that the provided email or password is incorrect.
  - **Account Not Found**: Error message indicating that the provided email does not exist in the system.

3.2.2.2 Account

3.2.2.2.1 Update Profile

Description : For upgradation of staff profile.

- **Input :**
  - **Name**
  - **Contact**
  - **Gender**
  - **Email**
  - **Profile Picture**
- **Output :**
  - Successful upgradation of the account
- **Error :**
  - **Invalid Credentials:** Error message indicating that the provided details is incorrect with respect to the data types of each field.

3.2.2.2.2 Change Password

Description : For changing the password

- **Input:**
  - **Old Password**
  - **New Password**
  - **Confirm New Password**
- **Output :**
  - Successfully changed the password.
- **Error:**
  - Old Password is wrong.

3.2.2.3 Clients

3.2.2.3.1 Add Client

Description : For adding more clients.

- **Input :**
  - **Client Name**
  - **Client Phone Number**
  - **Client National ID No**
  - **Client Email**
  - **Client Password**
  - **Client Address**
  - **Client Profile Picture**
- **Output :**
  - Successful added new client.
- **Error :**
  - **Invalid Credentials:** Error message indicating that the provided details is incorrect with respect to the data types of each field.

3.2.2.3.2 Manage Clients

Description : For managing existing clients.

- **Input:**
  - **Name**
  - **Email**
  - **Contact**
  - **Address**
  - **Profile Picture**
  - **Old Password**
  - **New Password**
  - **Confirm New Password**
- **Output :**
  - **Manage Client :** For upgradation of client information and changing password.
  - **Delete Client :** For deletion of client information.
- **Error:**
  - Client not found.

3.2.2.4 iBank Accounts

3.2.2.4.1 Add Acc Type

Description : For addition of any particular type of account (credit card account/debit card account/savings/current/etc)

- **Input :**
  - **Account Category Name**

- **Account Category Rates Per Year**
- **Account Category Description**

- **Output :**
  - Successful addition of particular type of account.
- **Error :**
  - Account Type not available

3.2.2.4.2 Manage Acc Types

Description : For managing of any particular type of account (credit card account/debit card account/savings/current/etc)

- **Input:**
  - **Account Category Name**
  - **Account Category Rates Per Year**
  - **Account Category Description**
- **Output :**
  - Successfully updated the account type details.
- **Error:**
  - Account type not found.

3.2.2.4.3 Open iBank Acc

Description : For opening account (credit card account/debit card account/savings/current/etc)

- **Input :**
  - **iBank Account Type**
  - **Account Name**
- **Output :**
  - Successful opening of new account.
- **Error :**
  - Account Type not available

3.2.2.4.4 My iBank Accounts

Description : For viewing of account details

- **Input:**
  - **NA**
- **Output :**
  - Name
  - Account No
  - Rate
  - Acc Type
  - Acc Owner

- Date Opened
- **Error:**
  - No data available in table

3.2.2.5 Finances

3.2.2.5.1 Deposits

Description : To deposit money into the account

- **Input :**
  - **Amount Deposited($)**
- **Output :**
  - Successfully money deposited.
- **Error :**
  - No account found to deposit money.

3.2.2.5.2 Withdrawals

Description : To withdraw money from the account

- **Input:**
  - **Amount Withdraw**
- **Output :**
  - Successfully money withdrawn.
- **Error:**
  - No account found to withdraw money.

3.2.2.5.3 Transfers

Description : For transferring money.

- **Input:**
  - **Amount Transferred**
  - **Receiving Account Number**
- **Output :**
  - Successfully transferred money.
- **Error:**
  - No account found to transfer money.

3.2.2.6 Balance Enquiries

Description : For checking the account balance.

- **Input :**
  - **Choose the account.**
- **Output :**
  - **Account Balance Visible**
- **Error :**
  - No account found.

3.2.2.7 Transaction History

Description : For viewing all the transaction that are being done.

- **Input :**
  - **NA**
- **Output :**
  - **Transaction Code**
  - **Account No**
  - **Type**
  - **Amount**
  - **Acc Owner**
  - **Timestamp**
  - **Roll Back Transaction :** For cancelling of any particular transaction
- **Error :**
  - No account found.

3.2.2.8 Financial Reports

3.2.2.8.1 Deposits

Description : For viewing all the deposit amount details of all the accounts.

- **Input :**
  - **NA**
- **Output :**
  - **Transaction Code**
  - **Account No**
  - **Amount**
  - **Acc Owner**
  - **Timestamp**
  - **Copy**
  - **CSV**
  - **Excel**
  - **Print**
- **Error :**
  - No account found.

3.2.2.8.2 Withdrawals

Description : For viewing all the withdrawals amount details of all the accounts.

- **Input:**
  - **NA**
- **Output :**

- **Transaction Code**
- **Account No**
- **Amount**
- **Acc Owner**
- **Timestamp**
- **Copy**
- **CSV**
- **Excel**
- **Print**

- **Error :**
  - No account found.

3.2.2.8.3 Transfers

Description : For viewing all the transfers amount details of all the accounts.

- **Input:**
  - **NA**
- **Output :**
  - **Transaction Code**
  - **Account No**
  - **Amount**
  - **Acc Owner**
  - **Timestamp**
  - **Copy**
  - **CSV**
  - **Excel**
  - **Print**
- **Error :**
  - No account found.

3.2.2.9 Notifications

Description : Notification oof all the transaction that are being done.

- **Input :**
  - **NA**
- **Output :**
  - Notification about the particular transaction
- **Error :**
  - No transaction details found.

3.2.2.10 Set Spending Limit

Description : The system should allow staff to set spending limits on client credit card accounts based on balance available on credit card accounts.

- **Input :**
  - **Card No.**
  - **Expiration Month , Year**
  - **CVV**
  - **Card Holder Name**
  - **Spending Limit**
- **Output :**
  - Spending Limit confirmation.
- **Error :**
  - No card found with respect to credit card account.

3.2.2.11 Generate Fraud Alert

Description : The system should generate fraud alerts when it detects suspicious activity, such as a large number of unauthorized transactions.

- **Input :**
  - **Card No.**
  - **Expiration Month , Year**
  - **CVV**
  - **Card Holder Name**
  - **Suspicious Activity**
- **Output :**
  - Fraud Alert
- **Error :**
  - No card found with respect to credit card account.

3.2.3 Admin

3.2.3.1 Login

Description : For signing in function of already admin accounts.

- **Input:**
  - **Email**
  - **Password**
- **Output :**
  - **Successful Authentication**: If the provided email and password match the records in the system, the authentication is successful. The admin is granted access to the system with appropriate permissions based on their role.
  - **Failed Authentication:** If the provided credentials (email or password) do not match any records in the system, authentication fails. The system denies access and often provides an error message indicating invalid credentials.
  - **Dashboard:** The dashboard is visible with all the account details and latest transaction.

- **Error:**
    - **Invalid Credentials:** Error message indicating that the provided email or password is incorrect.
    - **Account Not Found**: Error message indicating that the provided email does not exist in the system.

3.2.3.2 Account

3.2.3.2.1 Update Profile

Description : For upgradation of staff profile.

- **Input :**
    - **Name**
    - **Contact**
    - **Email**
- **Output :**
    - Successful upgradation of the account.
- **Error :**
    - **Invalid Credentials:** Error message indicating that the provided details is incorrect with respect to the data types of each field.

3.2.3.2.2 Change Password

Description : For changing the password

- **Input:**
    - **Old Password**
    - **New Password**
    - **Confirm New Password**
- **Output :**
    - Successfully changed the password.
- **Error:**
    - Old Password is wrong.

3.2.3.3 Staff

3.2.3.3.1 Add Staff

Description : For adding more clients.

- **Input :**
    - **Staff Name**
    - **Staff Phone Number**
    - **Staff Gender**
    - **Staff Email**
    - **Staff Password**
    - **Staff Profile Picture**
- **Output :**

- Successful added new staff.

- **Error :**
  - **Invalid Credentials:** Error message indicating that the provided details is incorrect with respect to the data types of each field.

3.2.3.3.2 Manage Clients

Description : For managing existing staffs.

- **Input:**
  - **Name**
  - **Email**
  - **Contact**
  - **Profile Picture**
  - **Old Password**
  - **New Password**
  - **Confirm New Password**
- **Output :**
  - **Manage Client :** For upgradation of staff information and changing password.
  - **Delete Client :** For deletion of staff information.
- **Error:**
  - Staff not found.

3.2.3.4 Clients

3.2.3.4.1 Add Client

Description : For adding more clients.

- **Input :**
  - **Client Name**
  - **Client Phone Number**
  - **Client National ID No**
  - **Client Email**
  - **Client Password**
  - **Client Address**
  - **Client Profile Picture**
- **Output :**
  - Successful added new client.
- **Error :**
  - **Invalid Credentials:** Error message indicating that the provided details is incorrect with respect to the data types of each field.

3.2.3.4.2 Manage Clients

Description : For managing existing clients.

- **Input:**
  - **Name**
  - **Email**
  - **Contact**
  - **Address**
  - **Profile Picture**
  - **Old Password**
  - **New Password**
  - **Confirm New Password**
- **Output :**
  - **Manage Client :** For upgradation of client information and changing password.
  - **Delete Client :** For deletion of client information.
- **Error:**
  - Client not found.

3.2.3.5 iBank Accounts

3.2.3.5.1 Add Acc Type

Description : For addition of any particular type of account (credit card account/debit card account/savings/current/etc)

- **Input :**
  - **Account Category Name**
  - **Account Category Rates Per Year**
  - **Account Category Description**
- **Output :**
  - Successful addition of particular type of account.
- **Error :**
  - Account Type not available

3.2.3.5.2 Manage Acc Types

Description : For managing of any particular type of account (credit card account/debit card account/savings/current/etc)

- **Input:**
  - **Account Category Name**
  - **Account Category Rates Per Year**

- **Account Category Description**

- **Output :**
  - Successfully updated the account type details.
- **Error:**
  - Account type not found.

3.2.3.5.3 Open iBank Acc

Description : For opening account (credit card account/debit card account/savings/current/etc)

- **Input :**
  - **iBank Account Type**
  - **Account Name**
- **Output :**
  - Successful opening of new account.
- **Error :**
  - Account Type not available

3.2.3.5.4 My iBank Accounts

Description : For viewing of account details

- **Input:**
  - **NA**
- **Output :**
  - **Name**
  - **Account No**
  - **Rate**
  - **Acc Type**
  - **Acc Owner**
  - **Date Opened**
- **Error:**
  - No data available in table

3.2.3.6 Finances

3.2.3.6.1 Deposits

Description : To deposit money into the account

- **Input :**
  - **Amount Deposited($)**
- **Output :**
  - Successfully money deposited.
- **Error :**
  - No account found to deposit money.

3.2.3.6.2 Withdrawals

Description : To withdraw money from the account

- **Input:**
  - **Amount Withdraw**
- **Output :**
  - Successfully money withdrawn.
- **Error:**
  - No account found to withdraw money.

3.2.3.6.3 Transfers

Description : For transferring money.

- **Input:**
  - **Amount Transferred**
  - **Receiving Account Number**
- **Output :**
  - Successfully transferred money.
- **Error:**
  - No account found to transfer money.

3.2.3.7 Balance Enquiries

Description : For checking the account balance.

- **Input :**
  - **Choose the account.**
- **Output :**
  - **Account Balance Visible**
- **Error :**
  - No account found.

3.2.3.8 Transaction History

Description : For viewing all the transaction that are being done.

- **Input :**
  - **NA**
- **Output :**
  - **Transaction Code**
  - **Account No**
  - **Type**
  - **Amount**
  - **Acc Owner**
  - **Timestamp**
  - **Roll Back Transaction :** For cancelling of any particular transaction

- **Error :**
  - No account found.

3.2.3.9 Financial Reports

3.2.3.9.1 Deposits

Description : For viewing all the deposit amount details of all the accounts.

- **Input :**
  - **NA**
- **Output :**
  - **Transaction Code**
  - **Account No**
  - **Amount**
  - **Acc Owner**
  - **Timestamp**
  - **Copy**
  - **CSV**
  - **Excel**
  - **Print**
- **Error :**
  - No account found.

3.2.3.9.2 Withdrawals

Description : For viewing all the withdrawals amount details of all the accounts.

- **Input:**
  - **NA**
- **Output :**
  - **Transaction Code**
  - **Account No**
  - **Amount**
  - **Acc Owner**
  - **Timestamp**
  - **Copy**
  - **CSV**
  - **Excel**
  - **Print**
- **Error :**
  - No account found.

3.2.3.9.3 Transfers

Description : For viewing all the transfers amount details of all the accounts.

- **Input:**
  - **NA**
- **Output :**
  - **Transaction Code**
  - **Account No**
  - **Amount**
  - **Acc Owner**
  - **Timestamp**
  - **Copy**
  - **CSV**
  - **Excel**
  - **Print**
- **Error :**
  - No account found.

3.2.3.10 System Settings

Description : For reconfiguring the company name, company tagline and system logo.

- **Input :**
  - **Company Name**
  - **Company Tagline**
  - **System Logo**
- **Output :**
  - Successfully reconfigured system.
- **Error :**
  - **Invalid Credentials:** Error message indicating that the provided details is incorrect with respect to the data types of each field.

3.2.3.11 Notifications

Description : Notification of all the transaction that are being done.

- **Input :**
  - **NA**
- **Output :**
  - Notification about the particular transaction
- **Error :**
  - No transaction details found.

3.2.3.12 Set Spending Limit

Description : The system should allow admin to set spending limits on client credit card accounts based on balance available on credit card accounts.

- **Input :**
  - **Card No.**
  - **Expiration Month , Year**
  - **CVV**
  - **Card Holder Name**
  - **Spending Limit**
- **Output :**
  - Spending Limit confirmation.
- **Error :**
  - No card found with respect to credit card account.

3.2.3.13 Generate Fraud Alert

Description : The system should generate fraud alerts when it detects suspicious activity, such as a large number of unauthorized transactions.

- **Input :**
  - **Card No.**
  - **Expiration Month , Year**
  - **CVV**
  - **Card Holder Name**
  - **Suspicious Activity**
- **Output :**
  - Fraud Alert
- **Error :**
  - No card found with respect to credit card account.

3.3 Non-Functional Requirements

3.3.1 Correctness Requirement

- **Algorithmic Precision:** The system must execute algorithms and calculations with precision to ensure accurate accounting information.
- **Adherence to Account Standards**: The system must adhere to established accounting standards and policies to generate account information consistently and correctly.
- **Error-Free Data Handling**: The system must handle and process data accurately, preventing errors in data input, computation, and storage to maintain data integrity and correctness.

3.3.2 Portability requirement

- **Cross-Platform Compatibility:** The software must be compatible with various platforms (e.g., Windows, macOS, Linux) to ensure users can access and use the application seamlessly across different operating systems.

- **Ease of Deployment:** The system should be easy to deploy, allowing for straightforward installation and configuration on different hardware and software environments.
- **Data Transferability:** The software must facilitate easy transfer of data between instances or installations, ensuring data portability and accessibility when migrating to different systems or versions.

3.3.3 Efficiency Requirement

- **Response Time Optimization:** The system must respond to user interactions promptly, ensuring low latency and efficient performance to enhance user experience.
- **Resource Utilization Efficiency**: The software should utilize system resources (CPU, memory, storage) efficiently, aiming for optimal usage to minimize resource wastage and maintain high performance.
- **Scalability and Performance Scaling**: The system must scale efficiently to handle an increasing number of users or growing data volumes, maintaining consistent performance levels and responsiveness even under increased load.

3.3.4 Usability Requirement

- **Intuitive User Interface:** The system must feature an intuitive and user-friendly interface, allowing users to easily navigate and interact with the application without extensive training.
- **Accessibility and Inclusivity**: The software should adhere to accessibility standards, ensuring that individuals with disabilities can access and use the application effectively, promoting inclusivity and usability for all users.
- **Help and Documentation:** The system should provide comprehensive and easily accessible help resources, user guides, and documentation to assist users in understanding and utilizing the software's functionalities efficiently.

3.3.5 Reusability Requirement

- **Modularity and Component Reusability**: The system should be designed with clear, modular components, allowing for easy identification and reuse of specific functionalities across the application or in other projects.
- **Code Reusability:** The software codebase should be well-organized and documented to encourage reuse of code snippets, functions, or modules in other parts of the system or in future projects, enhancing efficiency and maintainability.
- **Flexibility for Integration**: The system architecture should support seamless integration with other systems or software, enabling reusability of components or services in different contexts or environments, promoting interoperability and extensibility.

3.3.6 Reliability Requirement

- **Fault Tolerance:** The system must maintain its functionality and availability even in the presence of faults or errors, ensuring uninterrupted performance and minimizing downtime.
- **Error Handling and Logging:** The software should effectively handle errors, log relevant information, and notify appropriate parties, enabling prompt identification and resolution of issues to maintain system reliability.
- **Data Integrity and Consistency:** The system must ensure the integrity and consistency of data throughout its lifecycle, preventing data corruption or loss to maintain the reliability and accuracy of information.

3.3.7 Maintainability Requirement

- **Modifiability and Flexibility:** The system should be designed with modifiability in mind, allowing for easy updates, enhancements, or alterations to accommodate changing requirements and technological advancements while maintaining system stability.
- **Readability and Documentation:** The software codebase and design should be well-documented and written in a clear, understandable manner, aiding developers and maintainers in comprehending the system's architecture, logic, and functionalities for efficient maintenance and updates.
- **Standardized Coding Practices:** The system development should adhere to established coding conventions, guidelines, and best practices, ensuring a consistent and uniform codebase that is easier to maintain, debug, and enhance over time.

3.4 User Characteristics

- **Clients**: Credit card holders constitute a significant user group for the Credit Card Management System. This diverse group possesses varying levels of financial literacy and technical proficiency. Some clients may be highly familiar with digital financial tools, while others may have more limited experience. The primary goal for clients is to access and manage their credit card transactions, view account information, and ensure financial security. The system should feature an intuitive and user-friendly interface to facilitate easy navigation, providing clients with prompt access to essential financial information.
- **Staff**: Client services staff, responsible for assisting credit card holders, engage actively with the Credit Card Management System to address client queries, provide support, and ensure a positive customer experience. Their technical proficiency ranges from moderate to advanced, depending on the organization's training and tools provided. The system should prioritize efficient customer support features, accessible tools for issue resolution, and quick access to relevant client financial data. Enhancing efficiency in managing client interactions allows staff to provide timely and effective assistance.

- **Administrators**: Administrators hold a crucial role in managing and maintaining the Credit Card Management System. This group possesses a moderate to advanced level of technical proficiency and a deep understanding of financial and administrative processes. Their responsibilities include configuring the system, managing user access, ensuring security, and customizing the system to align with organizational needs. Key expectations for administrators include an efficient user management interface, robust security features, and compliance with financial regulations to facilitate seamless administrative operations and secure handling of financial data.

3.5 Design & Implementation Constraints

- **Design and Implementation Constraints**: The Credit Card Management System must navigate specific design and implementation constraints, including alignment with existing financial infrastructure and technologies. Compatibility with widely used browsers and devices is crucial for ensuring seamless access across a diverse user base. Moreover, the system design should be scalable to accommodate potential increases in user volume and transaction data.
- **Data Privacy and Compliance:** Design and implementation must strictly adhere to data privacy laws and financial regulations. The incorporation of robust data encryption, secure authentication mechanisms, and stringent access controls is paramount to safeguarding sensitive financial information. Compliance with regulations such as PCI DSS ensures the secure handling of financial data, promoting ethical and legal practices within the Credit Card Management System.
- **Integration with Existing Systems**: Seamless integration with existing financial tools and databases is imperative to establish a unified financial ecosystem. The system must demonstrate API compatibility and adhere to data exchange standards, facilitating interoperability with other financial applications. The design should prioritize efficient data synchronization and updates between the Credit Card Management System and other financial systems, ensuring a cohesive and accurate representation of financial information.

3.6 Assumptions & Dependencies

- **Assumptions**: The Credit Card Management System operates under the assumption of consistent internet connectivity for users to access and manage their financial data securely. It is also assumed that users possess compatible devices, such as computers, tablets, or smartphones, along with standard web browsers for system interaction. Furthermore, the assumption includes the availability of necessary financial transaction data for initial system setup and testing.
- **Dependencies**: The project is dependent on external APIs or services for crucial functionalities like transaction notifications and secure user authentication. Additionally, the system relies on a reliable hosting or cloud infrastructure to ensure continuous accessibility and optimal performance. Integration with existing financial databases and systems serves as a critical dependency to facilitate seamless data flow and maintain accuracy within the Credit Card Management System. The success of

the project is contingent upon the availability and reliability of these external components and services.

4 Interface Requirements

4.1 User Interfaces

The Credit Card Management System will boast user-friendly interfaces tailored to meet the needs of diverse users, including credit cardholders, financial institutions, and administrators. The interfaces will prioritize intuitive navigation, clear presentation of financial data, interactive visualization of transactions, and a consistent layout to ensure a user-friendly experience. Customized interfaces will cater to the specific requirements of each user group, facilitating efficient and engaging interactions with the system.

4.2 Hardware Interfaces

Accessible through standard computing hardware, the Credit Card Management System will support desktops, laptops, tablets, and smartphones. The interfaces will be designed for compatibility with various screen sizes and resolutions, ensuring a seamless user experience across different devices. Users will engage with the system using input devices such as keyboards, mice, touchscreens, and stylus pens. Network connectivity will be essential for data access, utilizing internet capabilities for optimal performance and accessibility.

4.3 Software Interfaces

Designed to operate on standard web browsers, the Credit Card Management System will be accessible across diverse operating systems, including Windows, macOS, and Linux. Internet connectivity will be a prerequisite for system access. The software may depend on specific components such as web servers, databases, and encryption protocols to guarantee secure data storage and transmission. Employing scalable, reliable, and cross-platform-compatible technologies will ensure efficient performance and accessibility for all users.

4.4 Communication Interfaces

The Credit Card Management System will leverage standard communication protocols over the internet to facilitate seamless data exchange between users and the system. Employing HTTP/HTTPS for secure data transmission, users can interact with the application through their preferred web browsers. Communication will be enhanced through email notifications, serving as a key interface for system alerts, updates, and timely information. API endpoints may be utilized to enable integration with external systems or services, ensuring efficient data exchange and interoperability.

5 Conclusion

The Credit Card Management System (CCMS) is a critical tool designed to streamline financial transactions and enhance account management within an integrated financial ecosystem. This Software Requirements Specification (SRS) provides a detailed framework for the successful development and implementation of the CCMS. In terms of functional requirements, the CCMS encompasses essential features such as transaction processing, real-time fraud detection, account management functionalities, balance inquiries, and secure user authentication. These functionalities are pivotal in ensuring the seamless and secure management of credit card-related data. The non-functional requirements focus on usability, performance, security, and reliability. Usability considerations emphasize the need for an intuitive user interface catering to users with varying levels of financial literacy. Performance benchmarks aim at delivering a system that meets high standards of efficiency, while robust security measures safeguard sensitive financial data, ensuring data integrity and compliance with industry regulations. Recognizing the diverse user base of the CCMS, from credit cardholders to financial institutions and administrators, the usability requirements stress the importance of an interface that accommodates different levels of technical proficiency, ensuring accessibility and engagement. Assumptions and dependencies play a crucial role in guiding the development process. Assumptions include factors like consistent internet connectivity, availability of compatible devices, and secure initial data access. Dependencies on external APIs, secure hosting infrastructure, and seamless database integration underscore the collaborative nature of the project. Regarding interfaces, the CCMS ensures accessibility through standard hardware and software, prioritizing compatibility for a diverse user base. Communication interfaces adhere to established protocols, emphasizing secure and efficient data exchange, critical for maintaining the integrity of financial transactions. In conclusion, this comprehensive Software Requirements Specification serves as a foundational guide for the development and implementation of the Credit Card Management System. It directs the development team towards achieving project objectives, exceeding the expectations of financial institutions and users. The well-defined requirements and considerations within this document ensure the successful delivery of a robust, secure, and user-centric Credit Card Management System.