**Project Design Phase-II**

**Technical Architecture**

|  |  |
| --- | --- |
| **Date** | **02 November 2023** |
| **Github name** | **https://github.com/Nareshkumar2001/Naan-Mudhalvan-Salesforce** |
| **Project Name** | A CRM Application For Managing Loan App and Fraudulent Prevention |

### Technical Architecture :

The technical architecture of a CRM application for managing loan applications and preventing fraudulent activities is a critical component in its design and development. It should be designed to support the application's functional and non-functional requirements, ensuring scalability, security, and performance. Here's a high-level technical architecture for such an application:

1. Client Interface :

- The client interface is the front-end of the CRM application, where users interact with the system. This could be a web application or a mobile app. It includes user interfaces for loan application submission, customer data management, and reporting.

2. Web Server :

- A web server handles user requests from the client interface. It serves web pages, processes user input, and communicates with the application server.

3. Application Server :

- The application server is the core of the CRM system, responsible for business logic, data processing, and fraud prevention. It should be designed to handle the following functions:

- Loan application processing and tracking.

- Customer information management.

- Document management, including file storage and retrieval.

- Fraud detection and prevention mechanisms, which may involve machine learning models, data analytics, and integration with external fraud detection services.

- User authentication and authorization.

- Integration with external systems, like credit bureaus and payment processors.

4. Database :

- The database stores and manages all relevant data for the CRM application. Key data includes customer profiles, loan application details, historical data for fraud detection, and system configurations. It's essential to choose a robust and scalable database management system (DBMS) for this purpose.

5. External Services :

- Integration with external services is crucial. This can include:

- Credit bureau services for credit checks and risk assessment.

- Payment processors for loan disbursement and repayment.

- Regulatory reporting services for compliance.

- Third-party fraud detection and prevention services for additional security layers.

6. Fraud Prevention Engine :

- This component is responsible for detecting and preventing fraudulent activities. It may use machine learning models, behavior analysis, and rules-based systems to identify anomalies and suspicious activities.

7. Authentication and Security :

- Implement robust security measures, including encryption, authentication, and authorization, to protect sensitive customer data and prevent unauthorized access.

8. Scalability and Load Balancing :

- To handle a potentially high volume of loan applications, the system should be designed with scalability in mind. Load balancing techniques can distribute traffic across multiple servers to ensure performance and availability.

9. Monitoring and Logging :

- Implement monitoring and logging mechanisms to track system performance, user activity, and potential security threats. This is crucial for both operational insights and compliance purposes.

10. Reporting and Analytics :

- A reporting and analytics component allows users to generate reports on loan application status, fraud detection, and system performance.

11. Backup and Disaster Recovery :

- Implement a robust backup and disaster recovery strategy to ensure data integrity and system availability in case of unexpected events.

12. Compliance and Audit Trail :

- Ensure the CRM application maintains an audit trail of all user interactions and changes to sensitive data to meet compliance and regulatory requirements.

13. Cloud Hosting :

- Consider hosting the application in a cloud environment to provide flexibility, scalability, and high availability. Cloud providers like AWS, Azure, and GCP offer suitable infrastructure for such applications.

14. APIs :

- Develop APIs to allow third-party applications to integrate with the CRM system, facilitating data exchange and extending the application's capabilities.

15. Mobile Application (Optional):

- If a mobile application is part of the architecture, it should interact with the same backend components through APIs and ensure a consistent user experience.

16. DevOps and Continuous Deployment :

- Implement DevOps practices for continuous integration and deployment to facilitate updates and improvements to the system.