**CRM Project – Banking and Financial CRM Implementation**

**Phase 1 : Problem Understanding & Industry Analysis**

**Goal:** To gain a deep understanding of the bank’s current processes, customer relationship challenges.

**. Problem Statement:**

Banks and financial institutions manage a huge number of customers, accounts,

transactions, and financial products. Banks need a smart system to organize customer

info, track interactions, and provide faster, personalized financial services.

**. Understand Business Needs:**

**.** Identify challenges in customer onboarding, service, sales, and compliance.

**.** Capture expectations from departments like Retail Banking, Wealth Management.

**. Explore CRM Capabilities & Tools:**

**.** Research Salesforce Financial Services Cloud or similar platforms.

**.** Look into AppExchange or integrations (e.g., KYC tools, e-signature)

**. Stakeholder Analysis:**

**.** Engage with Relationship Managers, Compliance Officers, Service Teams, and IT.

**.** Gather pain points, feature needs, and current workarounds.

**Phase 2 : Org Setup & Configuration**

**Goal**: Establish a secure, role-based CRM foundation tailored to banking operations and compliance requirements.

**. Define User Roles and Profiles:**

**.** Set up roles for Relationship Managers, Branch Managers, Compliance Officers, etc.

**.** Configure profiles and permission sets based on job responsibilities.

**. Sandbox Usage:**

To test new configurations, customizations, integrations, or apps before deploying them to production.

**.** Developer Sandbox → For coding and unit testing (small data).

**.** Developer Pro → Larger storage than Developer Sandbox.

**. Enable Key CRM Features:**

**.** Activate essential modules (e.g., Chatter, Email-to-Case, Files, Notes).

**.** Prepare system for further configuration in later phases.

**Phase 3 : Data Modeling & Relationships:**

**Goal:** Build a data model for 360° customer view in banking CRM.

**. Record Types:**

. *Account Object* → Retail Banking, Corporate Banking (different record types).

. *Case Object* → Loan Complaint, Credit Card Dispute.

*. Opportunity Object* → Record Types: Loan Application, Investment Application.

**. Establish Data Relationships:**

Create links between entities (e.g., Customer to Account) for seamless data flow.

**. Customize Objects and Fields:**

. Objects are like containers that hold specific types of data — for example, Customers, Accounts, Loans.

**.** Fields are the individual pieces of information stored inside these objects — like Customer Name, Account Number, Loan Amount.

**Phase 4 : Process Automation(Admin)**

**Goal:** Automate banking tasks to save time and improve efficiency.

**. Validation Rules:**

**.** Loan amount must be ≥ 1000.

. KYC document required for high-value customers**.**

**. Email Alerts:**

**.** Loan Application Submitted

Trigger: New Loan Application record is created.

Recipients: Customer + Loan Officer.

Message: “Your loan application has been submitted. Our team will review it shortly.”

**. Workflow Rules:**

Rule Criteria → Condition to check (e.g., Loan Amount > 100000).

Actions → What should happen if criteria are true.

**Phase 5 : Apex Programming(Developer)**

**Goal:** To automate banking tasks and handle customer data efficiently.

**. Batch Apex:**

Handles large data volumes efficiently.

Prevents hitting limits (like SOQL 50,000 rows).

**. Apex Triggers:**

**.** Apex Trigger is a piece of Apex code that runs before or after specific DML operations (Insert, Update, Delete, Undelete) on a Salesforce object.

**.** Purpose: Automate business logic, enforce validations, maintain data consistency.

**. Scheduled Apex:**

Scheduled Apex lets you run Apex code at specific times (daily, weekly, monthly) to automate tasks like sending payment reminders, generating statements, or updating account balances**.**

**Phase 6:User Interface Development**

**Goal:** To make the application easy and pleasant for users to use.

**. Record Pages:**

Record Details – Standard fields or custom fields displayed on the record.

Related Lists – Shows related records like Loans, Transactions, Opportunities.

**. Lightning App Builder:**

Customer Account Page

* Components: Customer details, open accounts, loans, credit risk score, assigned RM.

Loan Application Page

* Components: Loan amount, repayment schedule, approval status, related tasks, compliance checklist.

**Phase 7 : Integration and External Access**

**Goal:** Link Salesforce with bank systems and let users safely access data.

**.** Connect Salesforce with bank systems to share data smoothly.

**.** Let users and apps safely access data from outside Salesforce.

**.** Keep data updated in real-time or in batches.

**.** Ensure security and follow banking rules.

**Phase 8 : Data Management & Deployment**

**Goal:** Ensure accurate, secure, and efficient management of banking data and deploy.

**.** Data Migration: Move existing banking data (accounts, transactions, customers) accurately into Salesforce.

**.** Data Quality: Maintain clean, complete, and consistent data for reliable operations.

**.** Backup & Security: Protect sensitive financial data during migration and deployment.

**Phase 9 : Reporting & Dashboards**

**Goal:** To track and visualize customer and financial data for better banking decisions.

**Dashboards:**

**.**  Loan Dashboard: Track loan applications, approvals, pending tasks, and high-value loans.

**.** Customer Dashboard: Show portfolio distribution, account balances, and risk levels.

**Field level Security:**

. Field-Level Security controls who can view or edit specific fields on an object in Salesforce.

. Ensures sensitive data (like account balances, SSN, or credit scores) is accessible only to authorized users.

**Phase 10 : Final Presentation & Demo Day**

**Goal**: To show how the Banking CRM works and its main features to the stakeholders.

**. Pitch Presentation**:

**Benefits** – Explain time savings, improved efficiency, better decision-making, and compliance.

**Demo / Screenshots** – Show a few live screens or flows to illustrate functionality.

**Future Scope** – Optional improvements or advanced features.