Project Proposal

On

Bank Management System with Real-Time Fraud Detection

Guided By:-

Anuj Kumar

Created By:-

Name	AF_id
Tushar Sharma	AF04991839
Yashark Vikal	AF04990815
Uttkarsh Mavi	AF04991809

Batch code :- ANP-D2405
Course code :- ITPR

Table Of Contents

Index

1. Title of the Project

Bank Management System with Real-Time Fraud Detection

2. Introduction

The Bank Management System with Real-Time Fraud Detection is a modern Java-based console (CLI) application developed using Java, JDBC, MySQL, and Maven. This system is designed to manage digital banking operations efficiently, such as customer onboarding, account management, transactions, loan processing, and security monitoring.

In today's banking environment, millions of digital transactions occur every second. To handle this efficiently, banks require fast, secure, and automated systems. This project integrates core banking functionalities with an advanced fraud detection module that continuously analyzes transactions in real time and identifies suspicious behavior using predefined rules.

The goal of this system is to provide:

- **Fast and secure transaction processing**
- **Centralized customer and account management**
- **Automated loan and EMI handling**
- **User authentication and role-based access**
- **Real-time fraud alerts for high-risk transactions**
- **A reliable backend system built with Java and MySQL**

This project demonstrates how modern banks internally handle accounts, customers, security, fraud detection, and data storage using clean programming architecture and stable database connectivity

3. Objective

1. **To make daily banking tasks simple and automatic, such as adding customers, creating accounts, depositing, withdrawing, and transferring money.**

2. **To store all banking information in one place, so customer, account, loan, and transaction details can be accessed easily.**

3. **To make the system secure, by using login, roles (admin/employee), and by keeping track of user activities.**

4. **To detect fraud in real time, so whenever a suspicious transaction happens, the system can immediately generate an alert.**

5. **To make loan and EMI management easy, including loan application, EMI calculation, and EMI payment records.**

6. **To make banking operations fast and accurate, reducing manual work and avoiding common mistakes.**

7. **To build a system using Java, JDBC, and MySQL, which can be expanded in the future by adding more features.**

8. **To maintain a well-structured database, so that all records (customer, accounts, loans, transactions, and alerts) remain consistent, accurate, and easy to manage.**

4. Project Category

This project falls under the category of Software Development (Java Application).
It is a Console-Based Banking System built using the following technologies:

- **Core Java for logic implementation**
- **JDBC (Java Database Connectivity) for connecting Java with MySQL**
- **MySQL Database for storing and managing all banking data**
- **Maven for project management and dependency handling**

5. Analysis

→ Modules and Description :-

⭐ 1. Customer & Account Management Module

This module handles all customer-related and account-related operations.

Functions:

- **Add new customer**
- **Update customer details**
- **Store customer KYC**
- **Create account (Savings / Current / Salary)**
- **View account details**
- **Link customer with account**

📌 ***Purpose:*** **To maintain complete records of customers and their accounts.**

⭐ 2. Transaction Management Module

This module manages all types of money transactions inside the bank.

Functions:

- **Deposit amount**
- **Withdraw amount**
- **Transfer money between accounts**
- **Show mini statement**

- **Show full transaction history**
- **Record each transaction in the database**

📌 ***Purpose:*** **To process daily banking transactions accurately and securely.**

⭐ 3. Loan & EMI Management Module

This module handles loan-related operations.

Functions:

- **Apply for a loan**
- **Store loan details**
- **Calculate EMI**
- **Pay EMI**
- **Track overdue EMIs**
- **Store loan types (Home, Car, Personal, etc.)**

📌 ***Purpose:*** **To manage long-term loan and EMI records of customers.**

⭐ 4. Fraud Detection Module

This module detects suspicious activities in real time.

Functions:

- **Check high-amount transactions**
- **Detect too many transactions in a short time**
- **Identify repeated login failures**

- **Generate fraud alerts**
- **Store all suspicious activities in fraud database**

📌 ***Purpose:*** **To improve banking security and protect accounts from fraud.**

● ⭐ 5. Authentication & User Management Module

This module manages system users (admin, manager, employee).

Functions:

- **Admin login**
- **Role-based login**
- **Password verification**
- **Track failed login attempts**
- **Manage user roles**

📌 ***Purpose:*** **To keep the system secure and allow only authorized users.**

⭐ 6. Internal Bank Management Module

This module manages internal bank data.

Functions:

- **Manage branch details**
- **Manage employee details**
- **Assign roles and departments**

- **Track employee activity**

📌 ***Purpose:*** **To maintain internal bank-level operations.**

⭐ 7. Charges & Billing Module

This module handles different types of bank charges.

Functions:

- **Maintenance charges**
- **Loan processing fee**
- **Penalty charges**
- **GST / Service charges**

📌 ***Purpose:*** **To automatically deduct charges and store billing records.**

• Database Design (MySQL)

⭐ Table No. 1: customer

field	datatype	properties	description
customer_id	int	primary key,not null	Unique id for each customer
customer_name	varchar(100)	not null	Customer full name

email	varchar(100)	unique , not null	Customer email address
mobile	varchar(15)	not null	Customer mobile number
address	varchar(255)	not null	Customer residential address

Relation: One customer can have many accounts

★ Table No. 2: customer_kyc

field	**datatype**	**properties**	**description**
kyc_id	int	Primary key, not null	Unique id for KYC record
customer_id	int	Not null, foreign key	Linked customer id
aadhaar_no	varchar(20)	Not null	Customer Aadhaar number
pan_no	varchar(20)	Not null	Customer PAN number
address_proof	varchar(255)	Not null	Address proof document
kyc_status	vbarchar(20)	Not null	KYC verification status

Relation: One customer has one KYC record

⭐ Table No. 3: account

field	datatype	properties	description
account_no	int	Primary key, not null	Unique account number
customer_id	int	Not null, foreign key	Linked customer id
account_type_id	int	Not null, foreign key	Linked account type
balance	double	Not null	Current account balance
status	varchar(20)	Not null	Account status (active/closed/frozen)

Relation: One customer can have multiple accounts

⭐ Table No. 4: account_type

field	datatype	properties	description
account_type_id	int	Primary key, not null	Unique account type id
type_name	varchar(50)	Not null	Name of account type (Savings/Current/Salary)

Relation: One account_type is assigned to many accounts

★ Table No. 5: transaction

field	datatype	properties	description
txn_id	int	Primary key, not null	Unique transaction id
account_no	int	Not null, foreign key	Linked account number
amount	double	Not null	Transaction amount
txn_type_id	int	Not null, foreign key	Transaction type
txn_time	timestamp	Not null	Date and time of transaction

Relation: One account can have many transactions

★ Table No. 6: transaction_type

field	datatype	properties	description
txn_type_id	int	primary key, not null	Unique transaction type id
type_name	varchar(50)	not null	Name of transaction type (Deposit/Withdraw/Transfer)

Relation: One transaction type is used by many transactions

⭐ Table No. 7: bank_branch

field	datatype	properties	description
branch_id	int	primary key, not null	Unique branch id
branch_name	varchar(100)	not null	Name of the bank branch
branch_address	varchar(255)	not null	Address of the branch

Relation: One transaction type is used by many transactions

⭐ Table No. 8: loan_details

field	datatype	properties	description
loan_id	int	primary key, not null	Unique loan id
customer_id	int	Not null, foreign key	Linked customer id
loan_type_id	int	Not null, foreign key	Linked loan type
loan_amount	double	not null	Principal loan amount
interest_rate	double	not null	Loan interest rate
tenure_months	int	not null	Loan tenure in months

Relation: One customer can have multiple loans

★ Table No. 9: loan_type

field	datatype	properties	description
loan_type_id	int	primary key, not null	Unique loan type id
type_name	varchar(50)	Not null	Name of loan type (Home/Car/Personal)

Relation: One loan type is used by many loans

★ Table No. 10: emi_payment

field	datatype	properties	description
emi_id	int	primary key, not null	Unique EMI id
loan_id	int	not null, foreign key	Linked loan id
emi_amount	double	not null	EMI amount to be paid
payment_date	date	not null	EMI payment date
status	varchar(20)	not null	EMI status (paid/pending/overdue)

Relation: One loan has many EMI payments

★ Table No. 11: fraud_alert

field	datatype	properties	description
alert_id	int	primary key, not null	Unique alert id
txn_id	int	not null, foreign key	Linked transaction id
reason	varchar(255)	not null	Reason for fraud alert
alert_time	timestamp	not null	Time of alert generation

Relation: One transaction can generate multiple fraud alerts

★ Table No. 12: employees

field	datatype	properties	description
employee_id	int	primary key, not null	Unique employee id
branch_id	int	not null, foreign key	Linked branch id
name	varchar(100)	not null	Employee full name
role	varchar(50)	not null	Employee role (Manager/Clerk/etc.)
departement	varchar(50)	not null	Employee department
email	varchar(100)	unique, not null	Employee email

mobile	varchar(15)	not null	Employee mobile number

Relation: One branch can have many employees

★ Table No. 13: users

field	datatype	properties	description
user_id	int	primary key, not null	Unique user id
employee_id	int	not null, foreign key	Linked employee id
username	varchar(50)	unique, not null	Login username
password	varchar(255)	not null	Login password
role	varchar(50)	not null	User role (Admin/Manager/Employee)
status	varchar(20)	not null	Account status (active/inactive)

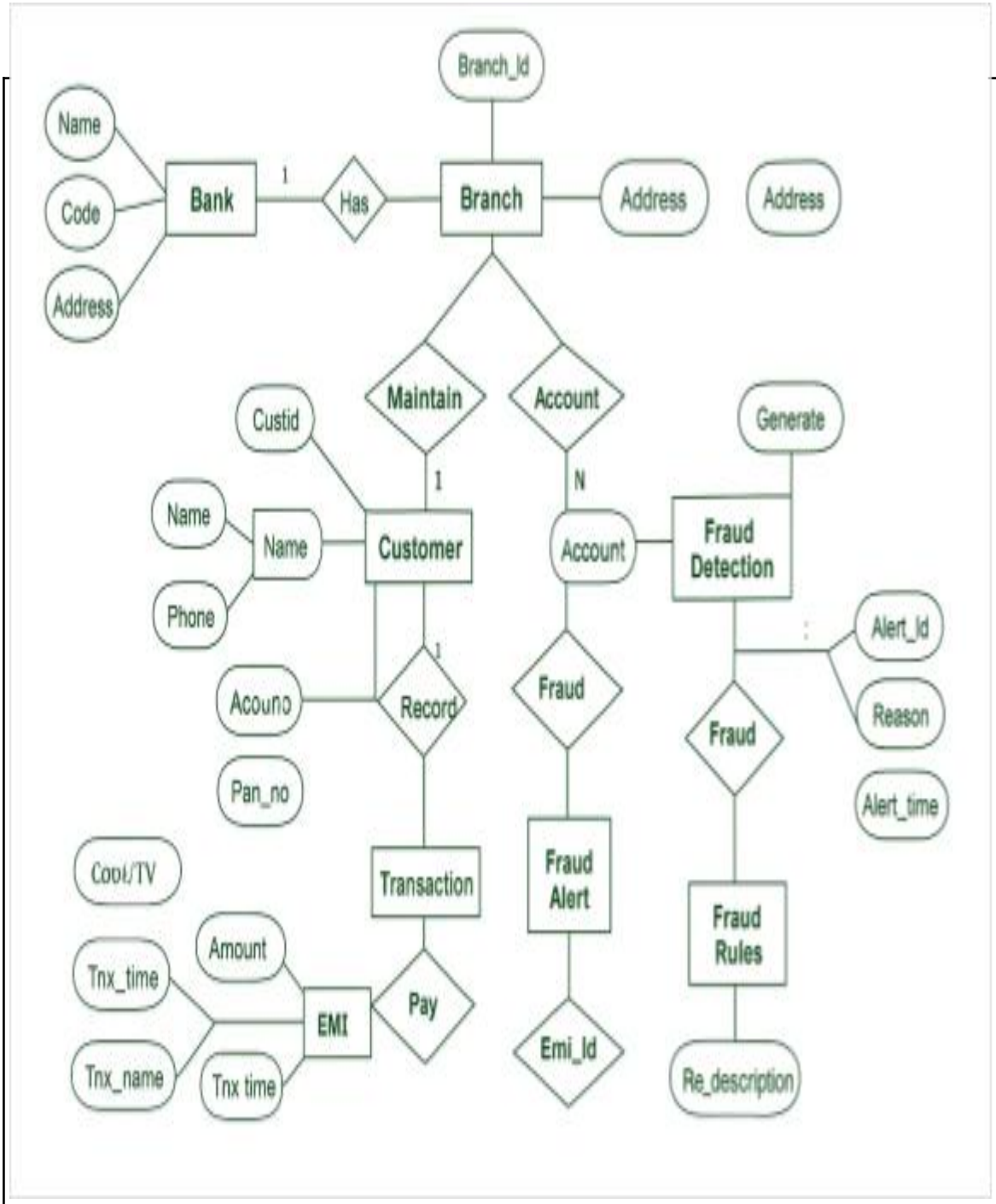
Relation: One employee can have one user login

★ Table No. 14: login_failed_attempts

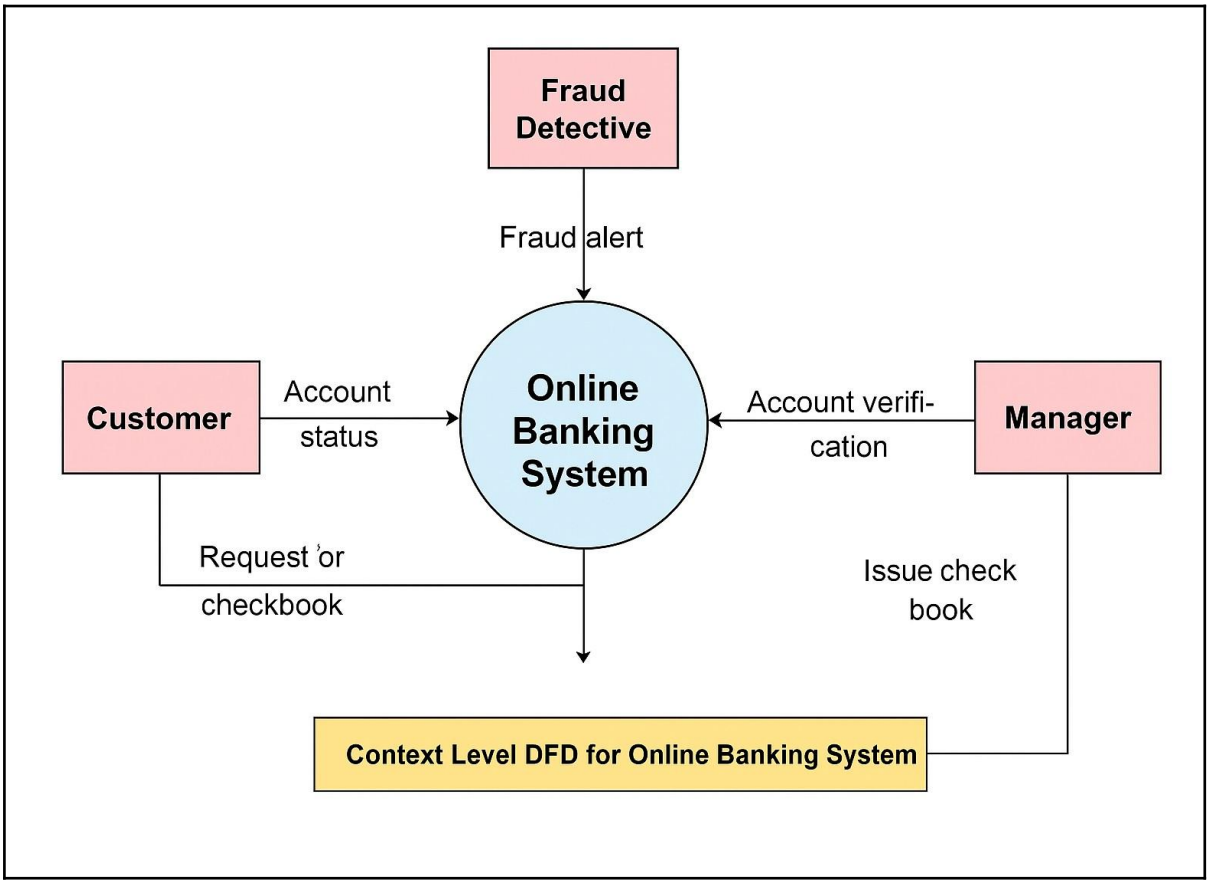
field	datatype	properties	description
attempt_id	int	primary key, not null	Unique attempt id
user_id	int	not null, foreign key	Linked user id
attempt_time	timestamp	not null	Time of failed login
reason	varchar(255)	not null	Reason for login failure

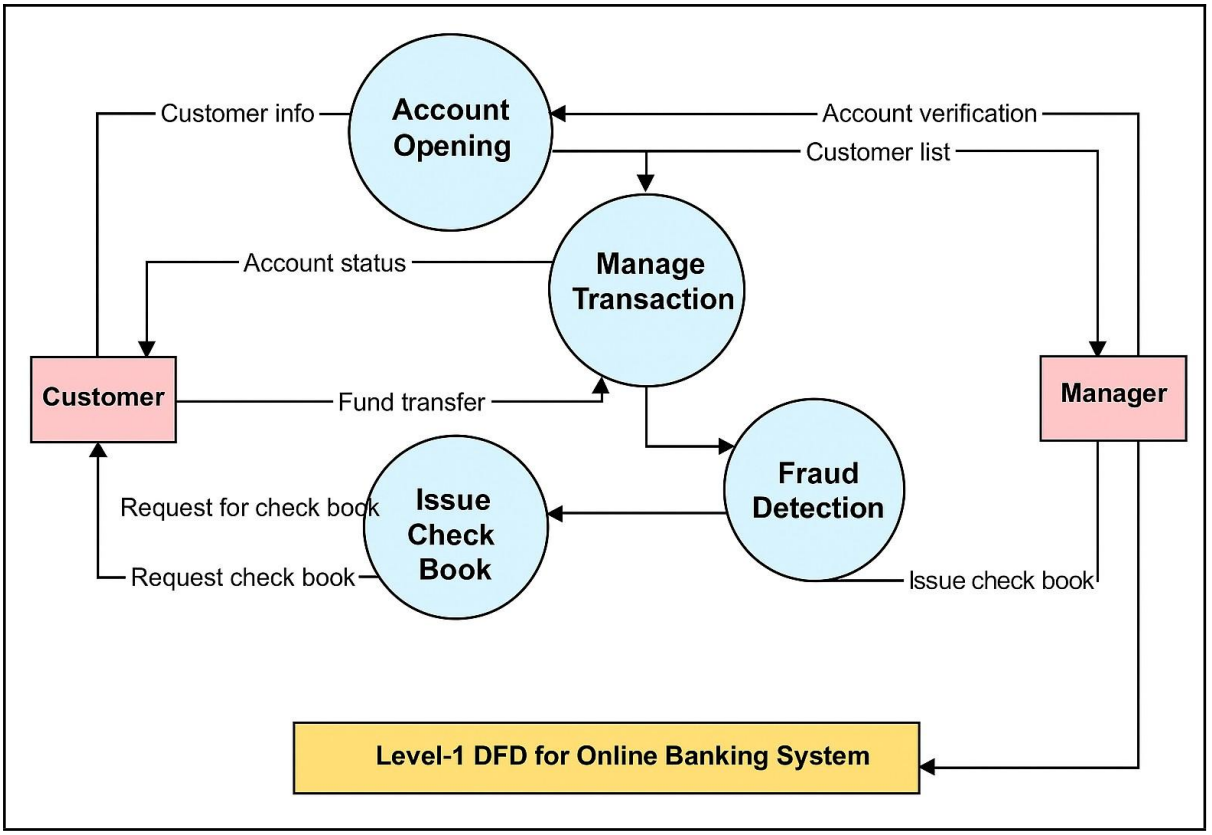
Relation: One user can have multiple failed login attempts

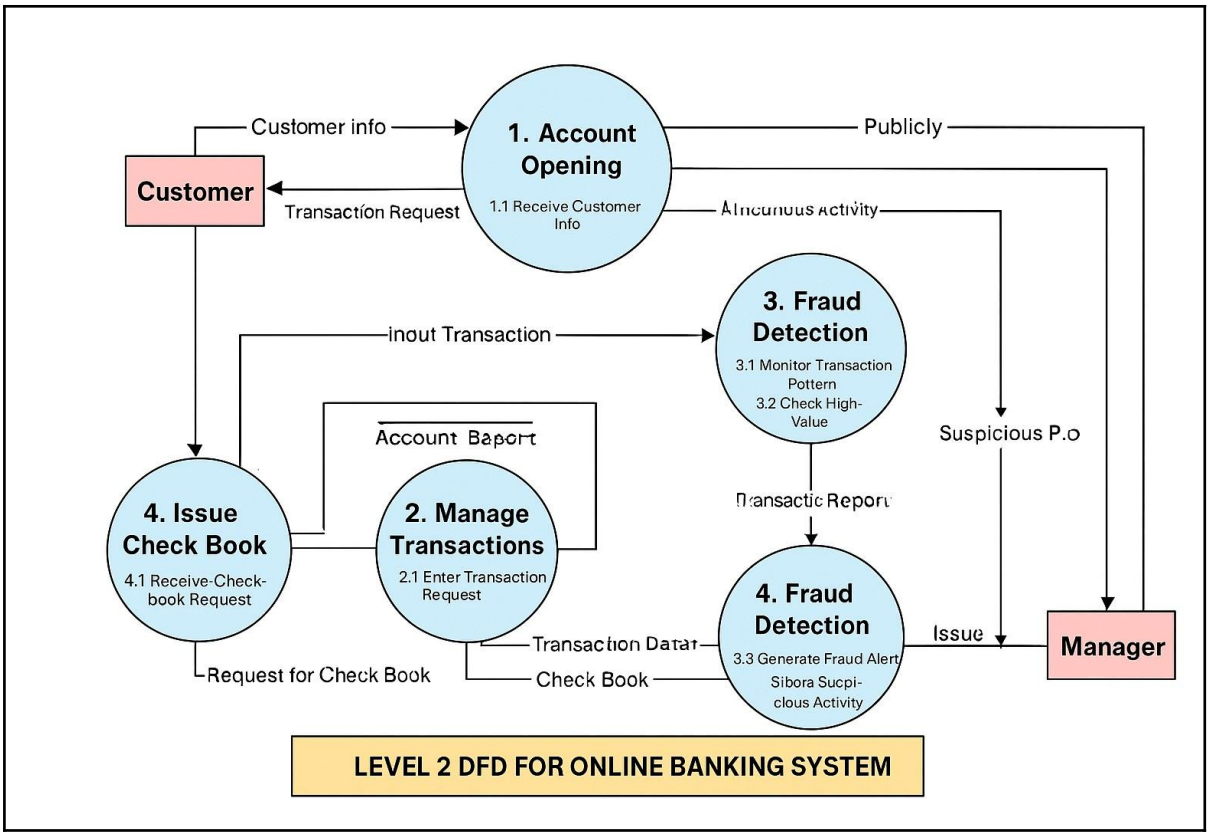
- **ER Diagram**


Branch_Id
Name
Code
Address
Bank
1
Has
Branch
Address
Address
Maintain
Account
Custid
Generate
1
N
Name
Name
Customer
Account
Fraud Detection
Phone
Alert_Id
1
Acouno
Record
Fraud
Reason
Fraud
Pan_no
Alert_time
Cool/TV
Transaction
Fraud Alert
Fraud Rules
Amount
Tnx_time
EMI
Pay
Emi_Id
Tnx_name
Tnx time
Re_description

- **Data Flow Diagram**




Account Opening
Customer info
Account verification
Customer list
Manage Transaction
Account status
Customer
Fund transfer
Manager
Fraud Detection
Issue Check Book
Request for check book
Request check book
Issue check book
Level-1 DFD for Online Banking System


1. Account Opening
1.1 Receive Customer Info
Customer info
Publicly
Customer
Transaction Request
3. Fraud Detection
3.1 Monitor Transaction Pottern
3.2 Check High-Value
Suspicious P.o
4. Issue Check Book
4.1 Receive-Check-book Request
2. Manage Transactions
2.1 Enter Transaction Request
Request for Check Book
Check Book
4. Fraud Detection
3.3 Generate Fraud Alert Sibora Sucpi-clous Activity
Issue
Manager
LEVEL 2 DFD FOR ONLINE BANKING SYSTEM

6. Complete Structure

- **Process Logical Structure**

Login
↓
Dashboard
↓
Customer Management → Account Management → Transaction Management
↓
Loan & EMI → Fraud Detection (Real-Time) → User & Authentication
↓
Branch Management → Employee Management → KYC Verification
↓
Reports & Statements → Logout

7. Platform Used

Hardware Requirements

• Processor: Intel i5
• RAM: 16GB
• Storage: 1GB free

Software Requirements

• Windows
• JDK 17+
• MySQL Server
• MySQL Workbench
• Maven
• IDE: Eclipse

8. Future Scope (For Bank + Fraud Detection System)

- **AI-Based Fraud Prediction**
- **Mobile Banking App Integration**
- **Biometric Login System**
- **Automated Loan Approval Using AI**
- **Cloud-Based Banking System**

- **Blockchain-Based Transaction Security**
- **Real-Time Credit Card Fraud Detection**
- **Chatbot-Based Customer Support**
- **Advanced Risk Scoring Engine**
- **Geo-Location Based Fraud Alerts**

---

9. Bibliography (For This Project)

- **MySQL Documentation**
  Used for designing relational database and SQL queries.

- **JDBC API Documentation**
  Used for connecting Java application with MySQL database.

- **Java Official Documentation (Oracle Docs)**
  Core Java, OOPs, exception handling, collection framework.

- **OWASP Security Guidelines**
  Authentication, authorization, and security rules for fraud prevention.

- **GeeksforGeeks**
  Concepts and examples related to Java, JDBC, and DS/Algo.

- **Maven Project Documentation**
  Used for project dependency management and folder structure.

- **Research Papers on Fraud Detection**
  Understanding AI/ML techniques for detecting financial fraud.

- **Banking Domain Standards (RBI Guidelines)**
  Transaction limits, KYC norms, and security compliance research