

BANK MANAGEMENT SYSTEM

PROJECT DESCRIPTION

In this project, we need to design a database system for a banking application. The database system is intended to provide a centralized data management solution for the banking system to effectively manage the customers, their transactions and the network involved.

SCOPE:

The scope of the Bank Management system includes branch management, customer management, account and transaction management, card association with accounts, loan type management, loan administration and transaction tracking. The system allows for creating, updating, deleting branch, customer information, assigning customers information and transactions. It enables the association of bank cards with accounts and supports the creation and management of loan types and loans. The system facilitates secure financial operations by recording and tracking transactions.

PURPOSE:

The Bank Management System is to provide a robust software application that effectively manages the daily operations of a bank. The system aims to streamline various tasks such as branch management, account and transaction management. And the system facilitates secure financial operations by ensuring accurate recording and tracking of transactions. The Bank Management System aims to optimize banking processes, improve customer service and enhance overall productivity within the bank.

FUNCTIONAL REQUIREMENTS:

1. Customer Management:

- Capture customer details such as first name, last name, date of birth and gender.
- Maintain a unique identifier (ID) for each customer.
- Allow the creation, retrieval, update, and deletion of customer records.

2. Account Management:

- Store account details including account number, balance and customer ID.
- Enable the creation, retrieval, update and deletion of account records.
- Store account details including account number, balance, and customer ID.
- Link a card to an account.

3. Transaction Management:

- Record transaction details such as description, amount, and date.
- Associate each transaction with a specific account.
- Enable transaction retrieval and reporting.

4. Branch Management:

- Manage branch details, including branch ID and name.
- Associate customers with a specific branch.

5. Card Management:

- Store card details such as card number, expiration date and blocked status.
- Associate cards with one or multiple accounts.
- Enable card retrieval and update.

6. Loan Management:

- Store Loan details including Loan ID, Amount paid, Start date, Due date, Account ID and Loan type ID.
- Associate each loan with a specific account and loan type.
- Allow Loan creation, retrieval, update and deletion.

NON-FUNCTIONAL REQUIREMENTS:

1. Security:

- Ensure data privacy and protection.
- Implement secure authentication and access control mechanisms.
- Encrypt sensitive data such as customer information and transaction details.

2. Performance:

- Handle a high volume of transactions efficiently.
- Provide quick response times for account and transaction retrieval.
- Optimize database queries for faster data retrieval.

3. Scalability:

- Accommodate an increasing number of customers, accounts and transactions.
- Handle concurrent user access and data processing.

4. Reliability:

- Maintain data integrity and consistency.
- Implement backup and recovery mechanisms to prevent data loss.
- Minimize system downtime and ensure continuous availability.

RISK:

1. Data Breaches:

- Implement robust security measures to prevent unauthorized access to customer data.
- Regularly monitor and update security protocols to address potential vulnerabilities.

2. System Failures:

- Develop a backup and recovery strategy to restore data in case of system failures.
- Conduct regular system maintenance to minimize the risk of hardware or software failures.

3. Compliance:

- Ensure compliance with regulatory requirements such as data protection and privacy laws.
- Stay updated with industry regulations and adapt the system accordingly.

ASSUMPTIONS:

- The system assumes the availability of a reliable and secure network infrastructure.
- Customers will provide accurate and valid information during the registration process.
- The system assumes that the bank has already established necessary partnerships with payment networks for card transactions.
- The system assumes that necessary hardware and software resources are available to support the systems functionality and performance requirements.

REQUIRED ENTITIES WITH THEIR ATTRIBUTES

1. **BRANCH:** represents a branch of the bank with attributes id(numeric), name(string), and address(string).
2. **CUSTOMER:** represents a bank customer with attributes id(numeric), first_name(string), last_name(string), dob(date), gender(string) and a foreign key branch_id(numeric).
3. **ACCOUNT:** represents a bank account with attributes id(numeric), balance(numeric) and foreign key customer_id(numeric).
4. **ACCOUNT_TYPE:** represents type of the account with id(numeric), type(string) and a foreign key account_id(numeric).
5. **CARD:** represents a bank card with attributes id(numeric), card_number(string), expiration_date(date), is_blocked(numeric) and a foreign key account_id(numeric).

6. **LOAN**: represents a loan with attribute id(numeric), amount_paid(numeric), start_date(date), due_date(date) and foreign keys account_id(numeric), loan_type_id(numeric) and customer_id(numeric).
7. **LOAN_TYPE**: represents a loan type with attributes id(numeric), type(string), description(string), base_amount(numeric), base_interest_rate(numeric).
8. **TRANSACTION** represents the transaction with attributes id(numeric), description(string), amount(numeric), t_date(date) and a foreign key account_id(numeric).

RELATIONSHIPS AND CARDINALITIES

1. A branch can have many customers. Thus, the cardinality is 1:N.
2. One customer can have many number of loans and one loan can be availed by two persons (like the joint personal loan). Thus, the cardinality is N:N.
3. A customer can have many accounts and a single account can be held by two or more customers (like the joint account). Thus, the cardinality is N:N.
4. One account has one account_type. Thus, the cardinality is 1:1.
5. A distinct card is connected to one account. Thus, the cardinality is 1:1.
6. Each loan belongs to one account. Thus, the cardinality is 1:N.
7. Loan_type has many loans. Thus, the cardinality is 1:N.
8. An account can have many transactions but a transaction can belong to one account. Thus, the cardinality is 1:N

Number of 1:1 relationships = 2

Number of N:N relationships = 2

Number of 1:N relationships = 4

Total number of relationships = 8

CONCLUSION:

Overall, the Bank Management System provides a robust platform for the bank to streamline its operations, improve customer service, ensure efficient management of customer accounts, transactions, loans and cards. It helps to maintain data integrity, enables effective reporting, analysis and enhances the overall Bank Management System.