A PROJECT REPORT ON

ATM MANAGEMENT SYSTEM DATABASE PROJECT

Submitted by

M.S. LEELAPRASAD (192210150)
MADANAMBETI SUMANTH KUMAR (192210052)

Under the guidance of

Dr. Carmel Mary Belinda

(Professor, Department of Applied Machine Learning)

IN PARTIAL FULFILLMENT FOR THE COMPLETION OF COURSE

CSA0533- DATABASE MANAGEMENT SYSTEM FOR DATA ANALYTICS

SIMATS ENGINEERING

THANDALAM

APRIL 2024

BONAFIDE CERTIFICATE

Certified that this project report titled "***ATM MANAGEMENT SYSTEM DATABASE"*** is the bonafide work of **M.S. LEELAPRASAD (192210150) MADANAMBETI SUMANTH KUMAR (192210052**) who carried out the project work under my supervision as a batch. Certified further, that to the best of my knowledge the work reported herein does not form any other project report.

Date: Project Supervisor: Head of the Department:

INTRODUCTION:

A comprehensive strategy for enhancing the operation and administration of Automated Teller Machines (ATMs) inside a financial network is represented by the ATM Management System Database Project. The system is primarily composed on a carefully designed database structure with multiple linked tables, each of which performs a specialized job that is essential to the system's functionality and security. Forming the basis, the Users Table contains vital user data such IDs (username and password), account information like account number, account type, and balance, and login credentials like username and password. Processes for user management and authentication are made efficient by this central repository. The Transactions Table, which methodically records all financial transactions, including deposits, withdrawals, and transfers, facilitates transaction logging. Together with specific transaction data like TransactionType, Amount, and TransactionDate, every transaction is given a unique identifier (TransactionID) and associated with a foreign key reference (UserID). By keeping important information about every ATM in the network, such as ATMID, Location, Status (Online/Offline), and CashBalance, the ATMs Table offers real-time monitoring capabilities. This makes it possible for banks to maintain adequate cash availability at all times, optimize ATM availability, and respond quickly to maintenance issues.

All user interactions with the system, including login attempts, logouts, and transaction activity, are thoroughly recorded in the Logs Table. For security and compliance reasons, timestamps and the corresponding user identity (UserID) produce a strong audit trail. Critical ATM card information such as unique identifiers (CardID), CardNumber, ExpiryDate, and CVV are kept track of by the Card Table, while related PINs are kept track of by the Pin Table. These security protocols guard against unwanted transactions and guarantee safe access to user accounts.

Users Table: Users are the cornerstone of any banking system. The Users Table in the database contains all of the pertinent account holder data. A UserID acts as the primary key, uniquely identifying each user. In-depth information about user profiles is provided by additional attributes including Username, Password, AccountNumber, AccountType, and Balance, which enable safe transactions and personalized services.

Table of Transactions: Withdrawals, deposits, and transfers are all included in the transactions that make up the foundation of banking operations. Every financial transaction that users initiate is documented in the Transactions Table. This table carefully records transaction details such as TransactionType, Amount, and TransactionDate. It has a primary key called TransactionID and a foreign key called UserID that references the Users Table. Accuracy, accountability, and transparency in financial activities are ensured by such careful documentation.

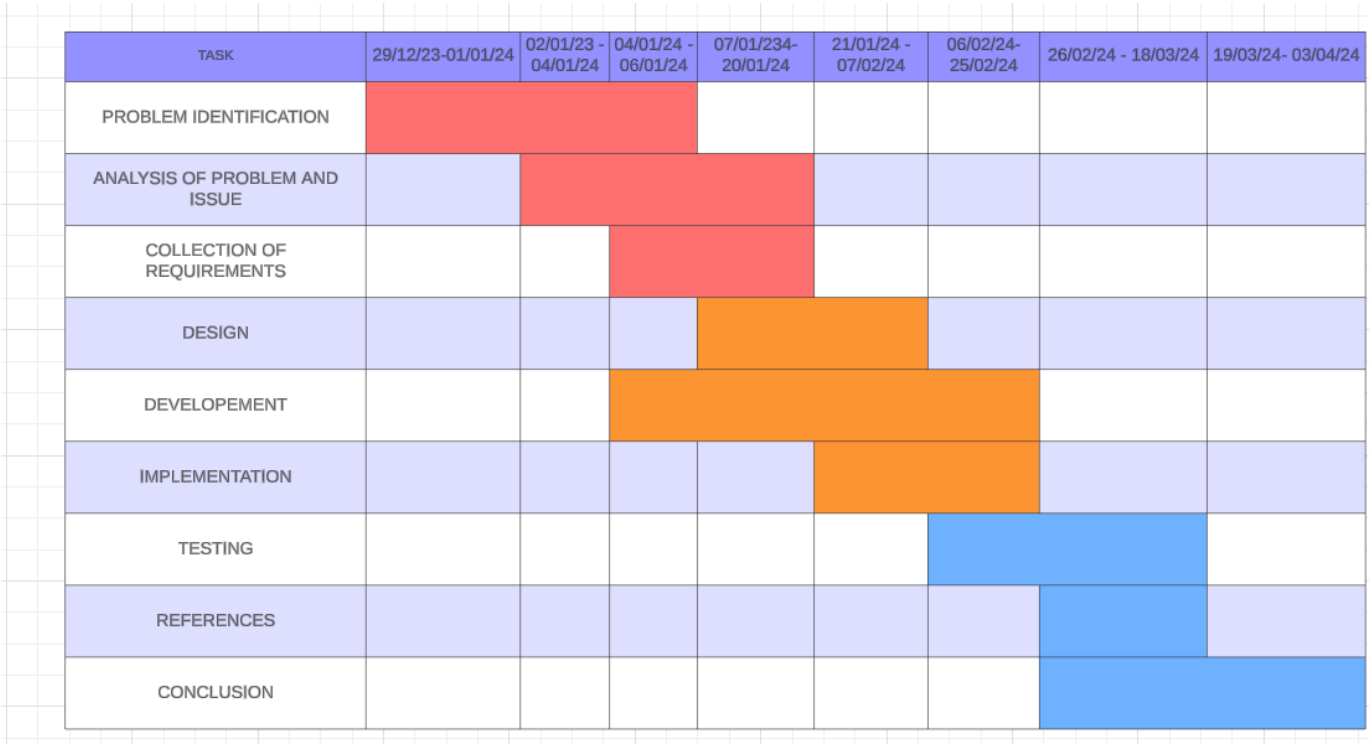
ATM Table: ATMs serve as a physical link between consumers and their accounts, offering convenience and accessibility around-the-clock. Important information regarding ATMs can be found in a table in the database project named ATMs. Every ATM is assigned a unique identification (ATMID), which is connected to a variety of details like its location, its status (online or offline), and its cash level. This real-time data helps administrators monitor ATM operations, ensure availability, and efficiently handle cash logistics.

Logs Table: For security and audit trail reasons, user activity logs are essential. A chronological log of all user operations within the ATM system is captured by the Logs Table. This table logs activities like Login, Logout, and Transaction along with relevant timestamps. It has LogID as the primary key and UserID as a foreign key referencing the Users Table. Thorough logging like this supports security protocols and allows for in-depth analysis to optimize efficiency.

Card Table: To safely use ATM services, bank cards are essential instruments. Important information on issued bank cards is stored in the Card Table of the database project. Every card has a distinct CardID that is connected to its corresponding UserID, guaranteeing that it is associated with the right account. CardNumber, ExpiryDate, and CVV are examples of attributes that help with card authentication and validation during transactions, protecting against fraudulent activity.

Pin Table: PINs, or personal identification numbers, give card-based transactions an extra degree of protection. PIN data related to bank cards is managed by the Pin Table in the database project. This table securely stores PINs with PinID as the primary key and CardID as a foreign key referencing the Card Table. This reduces the possibility of fraudulent transactions and identity theft by guaranteeing that only authorized users can access ATM services.

GHANTT CHART:



Objectives:

The ATM Management System Database Project has a strong design with multiple linked tables to effectively handle different ATM functions. The Users Table, which houses crucial user data including usernames, passwords, account numbers, kinds, and current balances, is at the center of the system. The functionalities of account management and user authentication are built upon this table. Withdrawals, deposits, and transfers made by users are all recorded in the Transactions Table, which is an addition to the Users Table. Each transaction is linked to a specific user by use of the UserID foreign key. To guarantee a thorough transaction history, transaction details such type, amount, and date are carefully recorded. Simultaneously, the ATMs Table serves as a central database for all things linked to ATMs, such as their distinct ATMIDs, locations, online and offline statuses, and cash balances. The smooth operation of ATMs is ensured by this table, which allows for real-time monitoring and management of ATM status and cash availability. Because it records all user actions and system events, the Logs Table is essential to preserving the integrity and security of the system. Timestamped actions like login, logout, and transaction activities create a thorough audit trail that can be used for accountability and traceability.

By keeping track of card-related data, such as distinct CardIDs, linked UserIDs, card numbers, expiration dates, and CVVs, the Card Table improves security even more. This table strengthens the system's overall security by guaranteeing the safe linking of cards to their corresponding user accounts. Furthermore, the Pin Table stores PIN-related data, such as PIN IDs, linked UserIDs, and PIN values, to enable secure authentication during ATM transactions. These linked tables work together to create a strong database system that efficiently controls ATM operations, guaranteeing data availability, confidentiality, and integrity while meeting regulatory compliance requirements. The ATM Management System Database Project hopes to preserve operational efficiency and security in all aspects of ATM operations while offering users a seamless and secure experience by utilizing this extensive architecture. Using the Transactions Table to keep an extensive transaction history is another important goal. The system makes sure that all user transactions—withdrawals, deposits, and transfers—are accurately tracked and accountable for financial matters. In order to facilitate the efficient retrieval and analysis of transaction data, this purpose also involves assigning each transaction to the corresponding user through the use of foreign keys.

The ATMs Table also tackles another important goal, which is guaranteeing the dependability and accessibility of ATMs. Through the storage of data, including ATM locations, online and offline statuses, and cash levels, the system is able to efficiently monitor and control ATM operations. By minimizing downtime and optimizing cash availability, this goal seeks to improve customer experience as a whole.Moreover, keeping an extensive record of user actions and system events via the Logs Table furthers the goal of improving auditability and security. The system creates a strong audit trail for tracking and monitoring system interactions by timestamping operations like login, logout, and transaction activities. This helps to prevent fraudulent activity and ensures regulatory compliance. All things considered, these goals work together to provide a safe, effective, and trustworthy ATM Management System Database. By achieving these goals, the initiative hopes to improve customer experience, preserve system integrity and security, and expedite ATM operations.

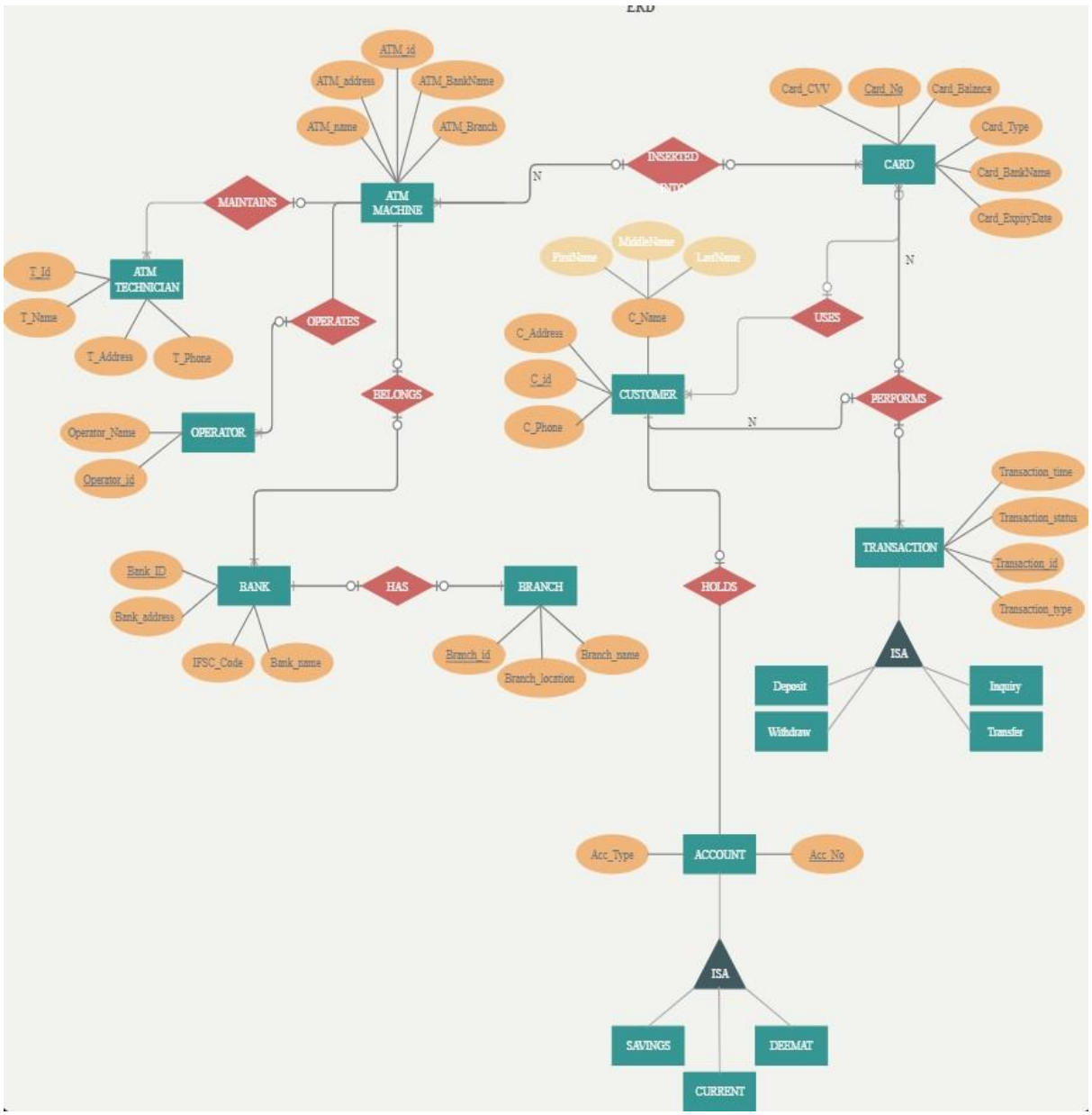
Literature Survey:

Several research works have examined the design and features of current ATM systems, providing insight into their database architectures and functioning mechanisms. These studies shed light on how different ATM systems handle database management, transaction processing, and user identification. Researchers have discovered important design decisions including data standardization, security precautions, and transaction logging procedures by analyzing these systems. The literature on database design principles provides helpful advice on how to organize data pertaining to ATMs. There is a full discussion of subjects including transaction management, normalization approaches, and entity-relationship modeling. The structure of databases that hold user accounts, transaction histories, ATM statuses, and activity logs has been studied by researchers. Moreover, defined best practices have been established for guaranteeing data security, integrity, and performance in database management systems. Because ATM security is so important, a lot of research has been done on access control, encryption, and authentication strategies. Research has examined the security measures taken to protect sensitive information, such as transaction details and user credentials, from intrusions and security breaches. To avoid data compromise and guarantee regulatory compliance, researchers have also looked at secure communication protocols between ATMs, bank computers, and other systems. ATM interface design is heavily influenced by user experience (UX) and human-computer interaction (HCI) design principles in order to achieve the best possible usability and accessibility. The literature in this field examines a variety of design strategies intended to improve user engagement with ATMs, including voice commands, touchscreen interfaces, and personalized preferences.

.. ATM interfaces that are easy to use and intuitive have been developed with the help of usability studies and user input. For ATM operations and data security, adherence to industry standards and legal regulations is essential. Scholars have scrutinized the legislative environment that governs financial operations, ATM transactions, and data privacy, encompassing standards such as PCI DSS (Payment Card Industry Data Security Standard). ATM management systems can efficiently eliminate associated risks and assure adherence to regulatory requirements by aligning with established standards and recommendations. This well-organized review of the literature covers important topics related to the ATM Management System Database Project and offers insightful information about current studies and industry best practices. The user interface and general user interaction of ATM systems are greatly influenced by human-computer interaction (HCI) and user experience (UX) design principles. Scholars have discovered essential design principles that improve user experience and happiness, such as voice-command functions, touchscreen interfaces, and personalized user preferences, through empirical investigations and usability analyses. Developers can improve customer happiness and loyalty by optimizing user interactions, decreasing transaction times, and optimizing overall usability by incorporating these design ideas into ATM interfaces.

A key component of ATM system development is conformity to industry standards and regulations. Scholars have conducted a thorough analysis of the regulatory environment around ATM operations, financial transactions, and data privacy. They have highlighted the significance of compliance frameworks like the Payment Card Industry Data Security Standard (PCI DSS). ATM management systems may guarantee strong compliance processes by adhering to defined rules and guidelines. This builds trust among stakeholders and effectively mitigates associated risks.

ER DIAGRAM:


ERD
ATM_id
ATM_address
ATM_BankName
ATM_name
ATM_Branch
ATM MACHINE
MAINTAINS
INSERTED
CARD
Card_CVV
Card_No
Card_Balance
Card_Type
Card_BankName
Card_ExpiryDate
T_Id
T_Name
T_Address
T_Phone
ATM TECHNICIAN
OPERATES
BELONGS
FirstName
MiddleName
LastName
C_Name
C_Address
C_id
C_Phone
CUSTOMER
USES
PERFORMS
Operator_Name
Operator_id
OPERATOR
Bank_ID
Bank_address
IFSC_Code
Bank_name
BANK
HAS
BRANCH
Branch_id
Branch_location
Branch_name
HOLDS
TRANSACTION
Transaction_time
Transaction_status
Transaction_id
Transaction_type
ISA
Deposit
Withdraw
Inquiry
Transfer
Acc_Type
ACCOUNT
Acc_No
ISA
SAVINGS
DEMAT
CURRENT
N
N
N

METHODOLOGY:

1. REQUIREMENT ANALYSIS:

- Gather requirements from stakeholders including users, administrators, and management.
- Identify functional and non-functional requirements.
- Define use cases and scenarios to understand system behavior.

2. CONCEPTUAL DESIGN :

- Define the entities involved such as users, ATMs, transactions, logs, etc.
- Identify relationships between entities (e.g., one-to-many, many-to-many).
- Create an Entity-Relationship Diagram (ERD) to visually represent the database schema.

3. LOGICAL DESIGN :

- Translate the conceptual design into a logical schema using a normalized relational model.
- Define tables, primary keys, foreign keys, attributes, and data types.
- Ensure data integrity through normalization to minimize redundancy and anomalies.

4. DATABASE IMPLEMENTATION :

- Select a suitable database management system (DBMS) such as MySQL, PostgreSQL, or SQLite.
- Write SQL scripts to create tables, define relationships, and enforce constraints.
- Execute the SQL scripts to create the database schema.

5. TESTING AND VALIDATION :

- Perform unit testing to ensure each component of the database functions correctly.
- Conduct integration testing to verify interactions between different modules.
- Validate data integrity, constraints, and transactions handling.

6. DEPLOYMENT:

- Deploy the database to a suitable environment such as a local server or cloud platform.
- Configure access controls and permissions for users and administrators.
- Ensure proper backup and recovery mechanisms are in place.

7. DOCUMENTATION :

- Document the database schema including tables, relationships, and constraints.
- Provide user manuals and guides for administrators and end-users.
- Document any assumptions made during the design and implementation process.

CODE :

```
-- Create Users Table
CREATE TABLE Users (
    UserID INT PRIMARY KEY,
    Username VARCHAR(50) NOT NULL,
    Password VARCHAR(50) NOT NULL,
    AccountNumber VARCHAR(20) UNIQUE NOT NULL,
    AccountType VARCHAR(20),
    Balance DECIMAL(18, 2)
);

-- Create Transactions Table
CREATE TABLE Transactions (
    TransactionID INT PRIMARY KEY,
    UserID INT,
    FOREIGN KEY (UserID) REFERENCES Users(UserID),
    TransactionType VARCHAR(20),
    Amount DECIMAL(18, 2),
    TransactionDate DATETIME
);

-- Create ATMs Table
CREATE TABLE ATMs (
    ATMID INT PRIMARY KEY,
    Location VARCHAR(100),
    Status VARCHAR(10),
    CashBalance DECIMAL(18, 2)
);
```

```
-- Create Logs Table
CREATE TABLE Logs (
    LogID INT PRIMARY KEY,
    UserID INT,
    FOREIGN KEY (UserID) REFERENCES Users(UserID),
    Action VARCHAR(20),
    Timestamp DATETIME
);

-- Create Card Table
CREATE TABLE Card (
    CardID INT PRIMARY KEY,
    UserID INT,
    FOREIGN KEY (UserID) REFERENCES Users(UserID),
    CardNumber VARCHAR(16) UNIQUE,
    ExpiryDate DATE,
    CVV INT
);

-- Create Pin Table
CREATE TABLE Pin (
    PinID INT PRIMARY KEY,
    CardID INT,
    FOREIGN KEY (CardID) REFERENCES Card(CardID),
    PIN INT
);
```

IMPLEMENTATION:

1. **USERS TABLE :** This table stores information about the users of the ATM system. Each user is identified by a unique UserID. The table includes fields for Username, Password, AccountNumber, AccountType, and Balance.
2. **TRANSACTION TABLE** : This table records all transactions made by users. Each transaction is identified by a unique TransactionID. It includes fields for UserID (which references the Users table), TransactionType (Withdrawal, Deposit, Transfer), Amount, and TransactionDate.

3. ATM'S TABLE : This table holds information about the ATMs in the system. Each ATM is identified by a unique ATMID. The table includes fields for Location, Status (Online/Offline), and CashBalance.

4. LOGS TABLE : This table logs actions performed by users within the system. Each log entry is identified by a unique LogID. It includes fields for UserID (referencing the Users table), Action (e.g., Login, Logout, Transaction), and Timestamp.

5. CARD TABLE : This table stores information about the cards associated with user accounts. Each card is identified by a unique CardID. It includes fields for UserID (referencing the Users table), CardNumber, ExpiryDate, and CVV.

6. PIN TABLE: This table stores PIN information associated with cards. Each PIN is identified by a unique PinID. It includes fields for CardID (referencing the Card table) and the PIN.

OUTPUT:

```
+-------------------+
|     Users Table   |
+-------------------+
| UserID | Username | Password | AccountNumber | AccountType | Balance  |
+--------+----------+----------+---------------+-------------+----------+
|   1    |  user1   | password1|  1234567890   |  Savings    |  1000.00 |
|   2    |  user2   | password2|  0987654321   |  Checking   |   500.50 |
|   3    |  user3   | password3|  1357924680   |  Savings    |  1500.25 |
+-------------------+


+------------------------+
|    Transactions Table  |
+------------------------+
| TransactionID | UserID | TransactionType | Amount | TransactionDate       |
+---------------+--------+-----------------+--------+-----------------------+
|       1       |   1    |   Withdrawal    |  50.00 | 2024-04-01 10:00:00   |
|       2       |   2    |    Deposit      | 100.00 | 2024-04-02 12:30:00   |
|       3       |   3    |    Transfer     | 200.50 | 2024-04-03 15:45:00   |
+------------------------+
```

ATMs Table

ATMID	Location	Status	CashBalance
1	Main Street	Online	5000.00
2	Park Avenue	Offline	10000.00

Logs Table

LogID	UserID	Action	Timestamp
1	1	Login	2024-04-01 09:55:00
2	2	Transaction	2024-04-02 12:35:00
3	3	Logout	2024-04-03 15:50:00

Card Table

CardID	UserID	CardNumber	ExpiryDate	CVV
1	1	1234567890123456	2025-12-31	123
2	2	9876543210987654	2024-10-31	456

Pin Table

PinID	CardID	PIN
1	1	1234
2	2	5678

CONCLUSION:

In conclusion, the provided SQL code establishes the database schema for an ATM Management System. The database consists of several tables including Users, Transactions, ATMs, Logs, Cards, and Pins, each serving a specific purpose in managing the ATM system effectively.

The Users table stores information about users such as their usernames, passwords, account numbers, account types, and balances. Transactions are recorded in the Transactions table, which tracks transaction details like transaction type, amount, and date, with references to the corresponding users via foreign keys.

ATMs are managed through the ATMs table, which stores ATM location, status, and cash balance. Logs of user activities within the system are maintained in the Logs table, including details like user actions and timestamps.

Additionally, user card information such as card numbers, expiry dates, and CVVs are stored in the Card table, and associated PINs are stored in the Pin table.

The database schema is designed to ensure data integrity, enforce referential integrity through foreign key constraints, and provide a solid foundation for managing user accounts, transactions, and ATM operations within the system.

Overall, this database schema provides a comprehensive structure for an ATM Management System, facilitating efficient and secure management of user transactions and ATM operations.

FUTURE ENHANCEMENT :

1. **Advanced Reporting and Analytics**: Implement features to generate detailed reports and analytics on transaction trends, ATM usage patterns, and user behaviors. This can provide valuable insights for optimizing ATM placements, identifying fraudulent activities, and improving service offerings.
2. **Integration with Mobile Banking**: Develop interfaces or APIs to integrate the ATM Management System with mobile banking applications. This would allow users to perform tasks such as locating nearby ATMs, initiating transactions remotely, and receiving real-time notifications on their mobile devices.
3. **Enhanced Security Measures**: Introduce additional security measures such as biometric authentication (e.g., fingerprint or facial recognition) for user authentication at ATMs. Implement encryption techniques to secure sensitive data stored in the database and during transmission between the ATM and the backend system.
4. **Real-Time Transaction Monitoring**: Implement a real-time monitoring system to track transactions as they occur. This would enable immediate detection of suspicious activities or anomalies, triggering alerts for further investigation or intervention by security personnel.

REFERENCES :

1. Sharma, Risabh. "ATM Management System." *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 12.6 (2021): 5151-5160.
2. Hara, Takahiro, et al. "DB-MAN: a distributed database system based on database migration in ATM networks." *Proceedings 14th International Conference on Data Engineering*. IEEE, 1998.
3. Kawashima, Kazuyuki, et al. "ATM internet management system: ATOMVIEWplus." *Electronics and Communications in Japan (Part I: Communications)* 80.9 (1997): 80-94.
4. Vayias, Evangelos, et al. "A Web-based database system for providing technical information on ATM networking platforms." *Journal of computing and information technology* 11.1 (2003): 1-13.
5. McDysan, David E., and Darren L. Spohn. *ATM theory and applications*. McGraw-Hill Professional, 1998.