Table of Contents

Table of Figures

BANKING APP SYSTEM

Introduction

Banking or finance related tasks has always been considered as one of the most sensitive tasks to perform for people, as it involves the transaction of the most valuable asset required to run daily lives, "money". Ever since the introduction of multiple banking apps in the market, anything related to banking has become one less problem to worry about.

For past few years, banking apps has been a boon for daily human life. With the invention of such system, all the banking related tasks that's done physically has been made possible without having to show up on the bank. By making the use of such system, bank can streamline their operations, improve customer service and ensure accurate and secure financial transactions.

Overview of Application

The Banking App System encompasses key features essential for managing banking operations efficiently:

Account Management: Users can create, update, and manage their accounts, including details such as account type and balance.

Transaction Processing: The system facilitates secure and efficient processing of transactions, including deposits, withdrawals, and fund transfers.

Customer Interaction: The BMS enables banks to interact with customers, providing account information, handling inquiries, and ensuring a seamless banking experience

Problem Statement

The Banking App System addresses various challenges faced by banks, including:

Manual Transaction Processes: Many banks still rely on manual processes for handling transactions, leading to errors, discrepancies, and increased processing times.

Cybersecurity Threats: Traditional banks are prime targets for cyberattacks due to their large data stores and financial assets. Weak data security infrastructure and human error can increase vulnerability to breaches and fraud, eroding customer trust and reputation.

Inefficiency and High Costs: Branch networks and outdated processes contribute to high operational costs for traditional banks. Manual tasks, paper-based transactions, and complex approval processes slow down operations and increase expenses. Streamlining processes and leveraging automation can improve efficiency and profitability.

These challenges can impact banking operations, resulting in errors, security vulnerabilities, and a suboptimal customer experience. The introduction of a comprehensive system is necessary to automate banking processes, provide real-time information, and enhance overall operational efficiency.

Tools Used

The development of the Bank App System involves using a range of tools and programming languages, including an efficient technology stack known as the MERN stack (MongoDB, Express.js, React.js, Node.js) that can optimize development and enhance system performance:

User Interface (UI): React.js can be used to build a dynamic and interactive user interface on the frontend side.

Server-Side Logic: Node.js, along with Express.js, can be chosen to handle the server-side logic.

Database Management: MongoDB can be selected as a database management system for the BMS project.

Methodology

For the development of the Banking App System, a structured approach using the Waterfall methodology is recommended. This method involves sequential phases, ensuring a systematic and organized development process:

Requirement Analysis:

Requirement analysis is a critical phase in the development of the Banking App System. It involves a systematic process of gathering, documenting, and analyzing the needs and expectations of stakeholders, including bank employees, managers, and customers. The primary objectives of requirement analysis in this context are as follows:

Identify Stakeholder Needs:

The first step is to identify and engage with all relevant stakeholders, including end-users, managers, and IT personnel, to understand their needs and expectations from the system. This may involve conducting interviews, surveys, and workshops.

Document Requirements:

All identified requirements are documented in detail. This includes functional requirements (what the system should do) and non-functional requirements (performance, security, usability, etc.).

Prioritize Requirements:

Not all requirements are of equal importance. Prioritization helps in determining which features should be developed first and which can be deferred to later phases.

Resolve Conflicts:

Sometimes, stakeholders may have conflicting requirements. It's essential to resolve these conflicts through negotiation and consensus-building.

Validation:

The gathered requirements are validated with stakeholders to ensure that they accurately represent their needs and expectations. This iterative process helps in refining and improving the requirements.

Feasibility Study

A feasibility study is conducted to assess the viability and practicality of the Bank App System project. It evaluates various aspects, including technical, operational, and economic feasibility, to determine whether the project should proceed. The following aspects are considered in the feasibility study:

Technical Feasibility Assess whether the chosen technologies and tools are suitable for the project's requirements. Ensure that the development team has the necessary expertise in these technologies. Analyze the compatibility of the new system with existing systems and infrastructure within the bank.

Operational Feasibility Assess the willingness of bank employees and customers to adapt to the new system. Identify potential resistance to change and plan change management strategies. Analyze the training needs of bank staff to ensure they can effectively use the new system.

Economic Feasibility Cost-Benefit Analysis: Conduct a cost-benefit analysis to compare the project's costs (development, implementation, maintenance) against the expected benefits (cost savings, increased revenue, improved customer satisfaction).

Planning

In the planning phase of the Banking App System project, tools like the Gantt Chart and Network Diagram are crucial for effective project management and scheduling.

Gantt Chart

In the Banking Management System project, Gantt Chart outlines project tasks and their dependencies, providing a timeline view of the project's milestones.

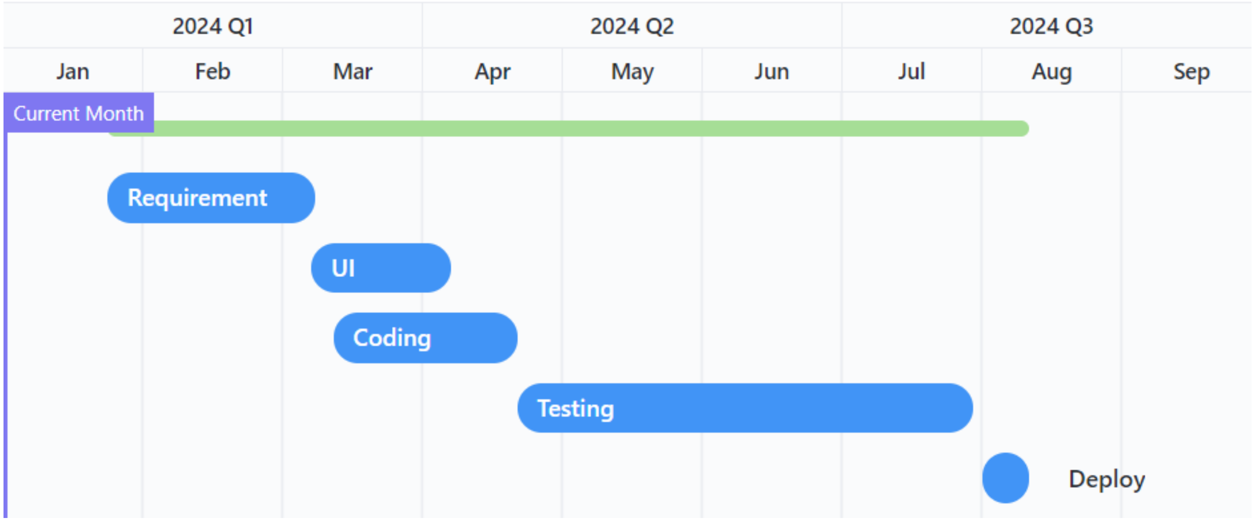

Figure 1 Gantt Chart

Network Diagram

A Network Diagram visually represents project tasks and their relations, aiding project managers in understanding activity flow for the Banking App System.

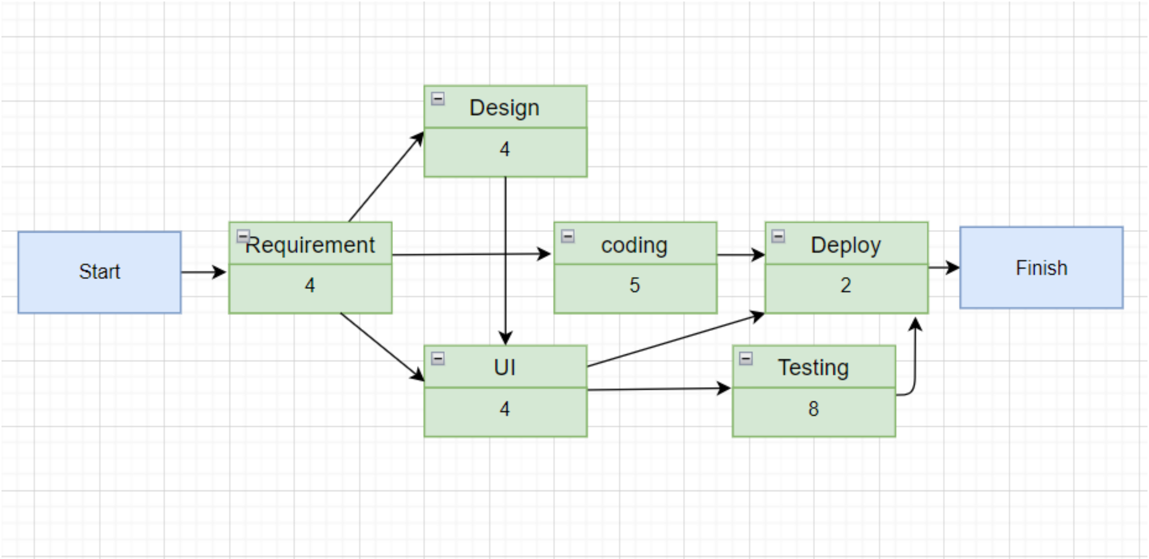

Figure 2 Network Diagram

Design (class diagram, use case, activity, DFD, ER)

In the design phase of the Banking App System project, various diagrams are created to articulate the system's architecture, components, and interactions.

Class Diagram

A Class Diagram is a structural diagram that can visualize the static structure of the Banking App System by depicting classes, their attributes, methods, and relationships.

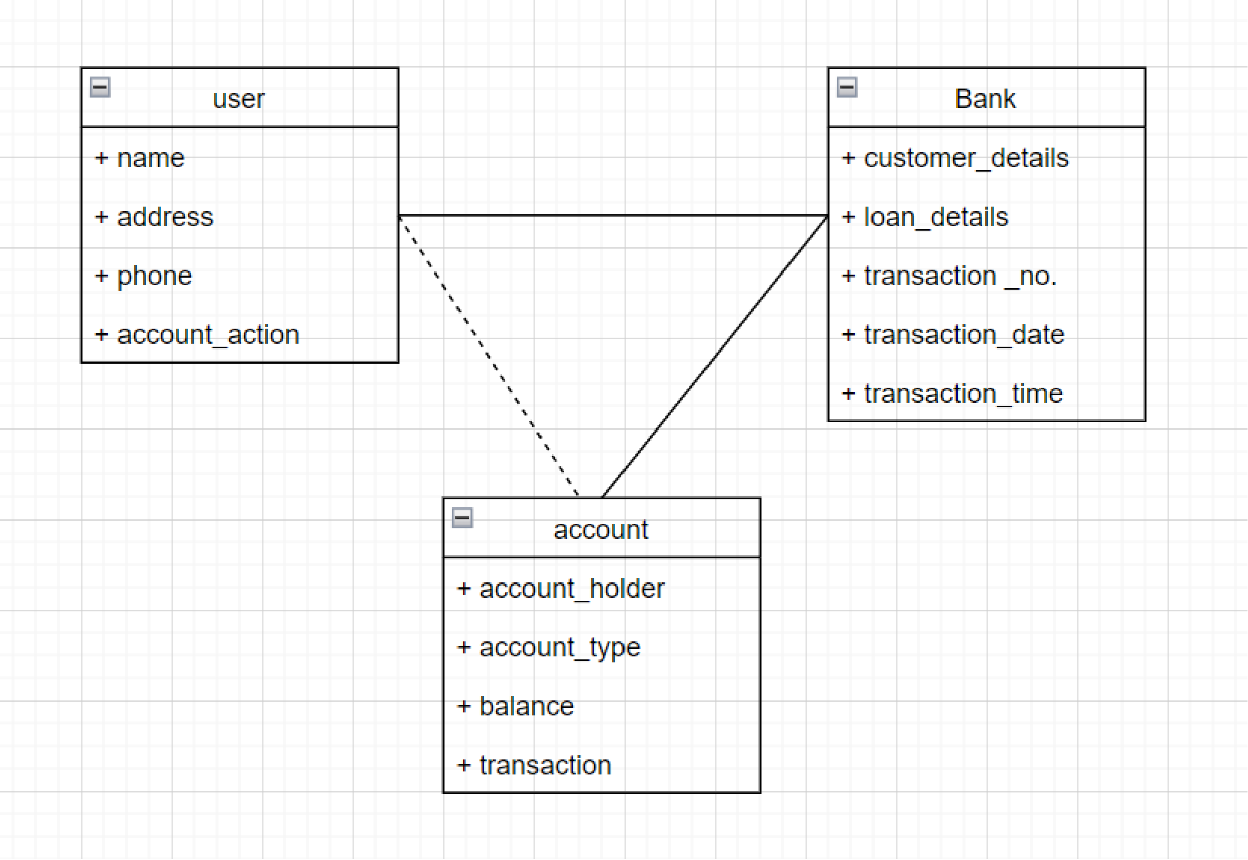

Figure 3 Class Diagram

Use Case Diagram

A Use Case Diagram is a UML diagram that visualizes the interactions between actors (users or external systems) and the functionalities (use cases) within a system. It helps in identifying and organizing system requirements.

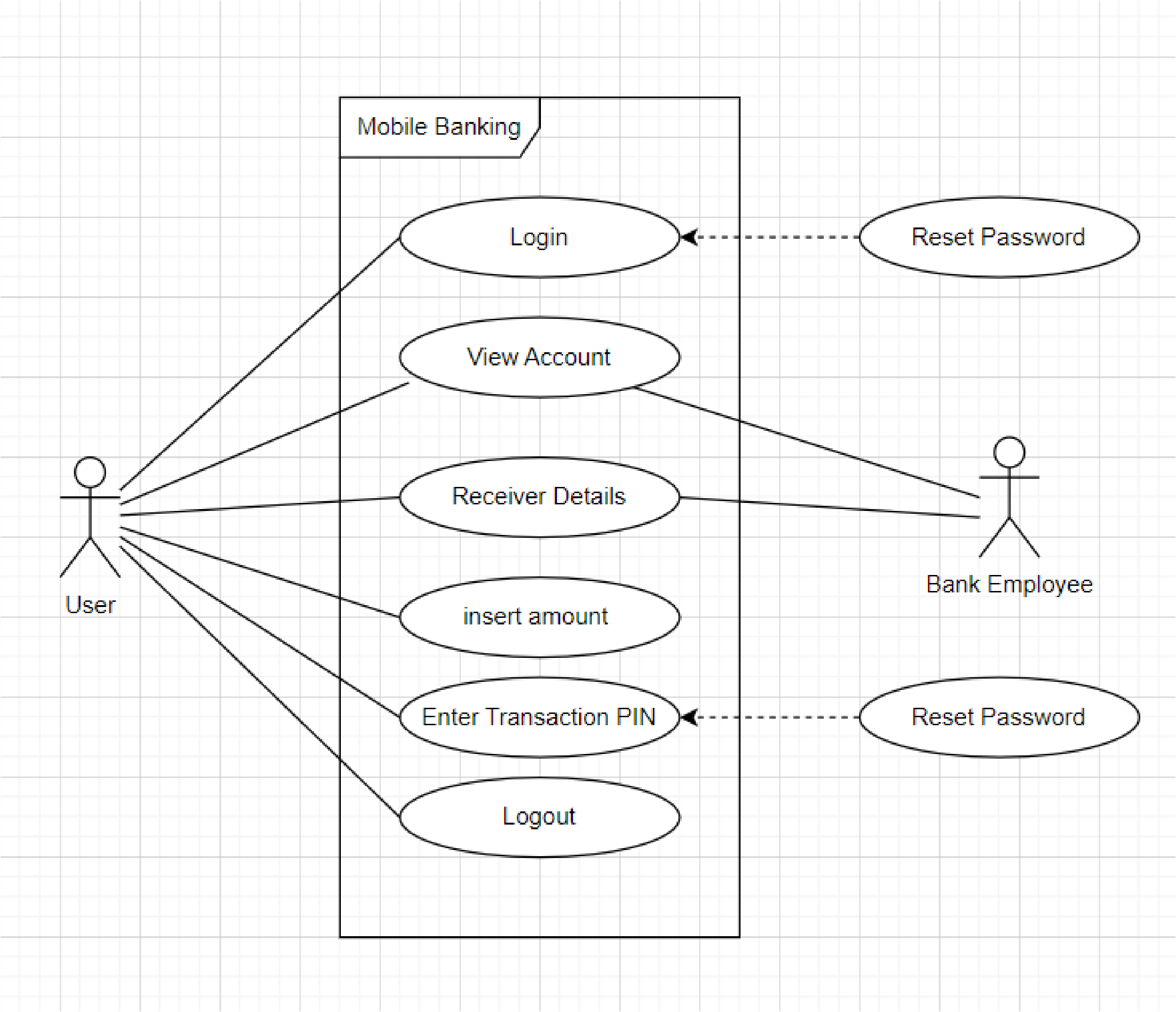

Figure 4 Use Case Diagram

Activity Diagram

An Activity Diagram is a UML diagram that illustrates the flow of activities or processes within the Banking App System.

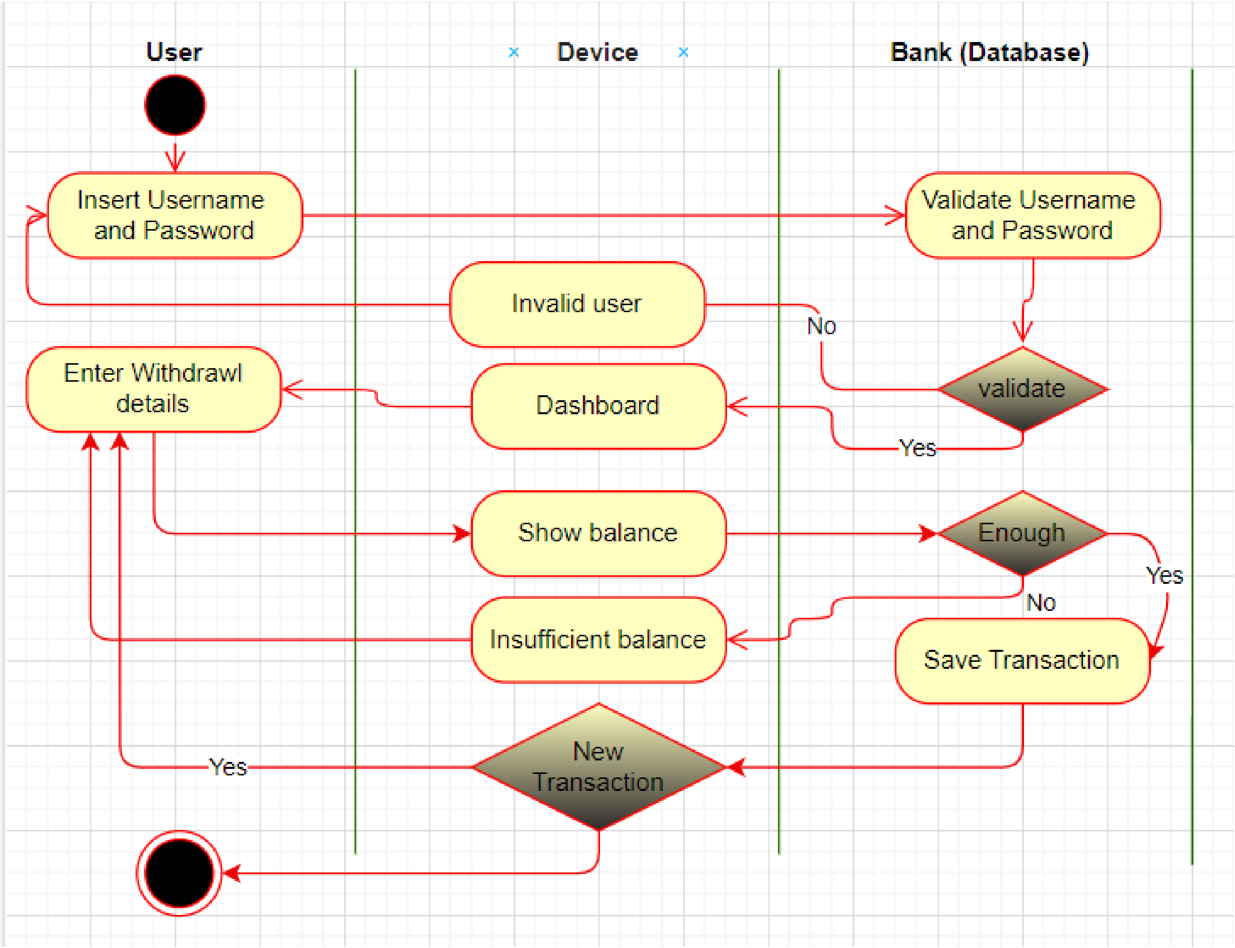

Figure 5 Activity Diagram

DFD

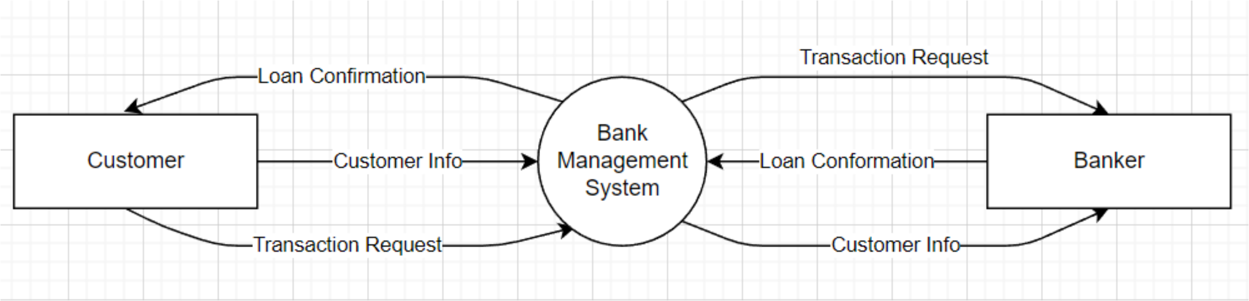

Figure 6 Level 1 DFD of Banking App System

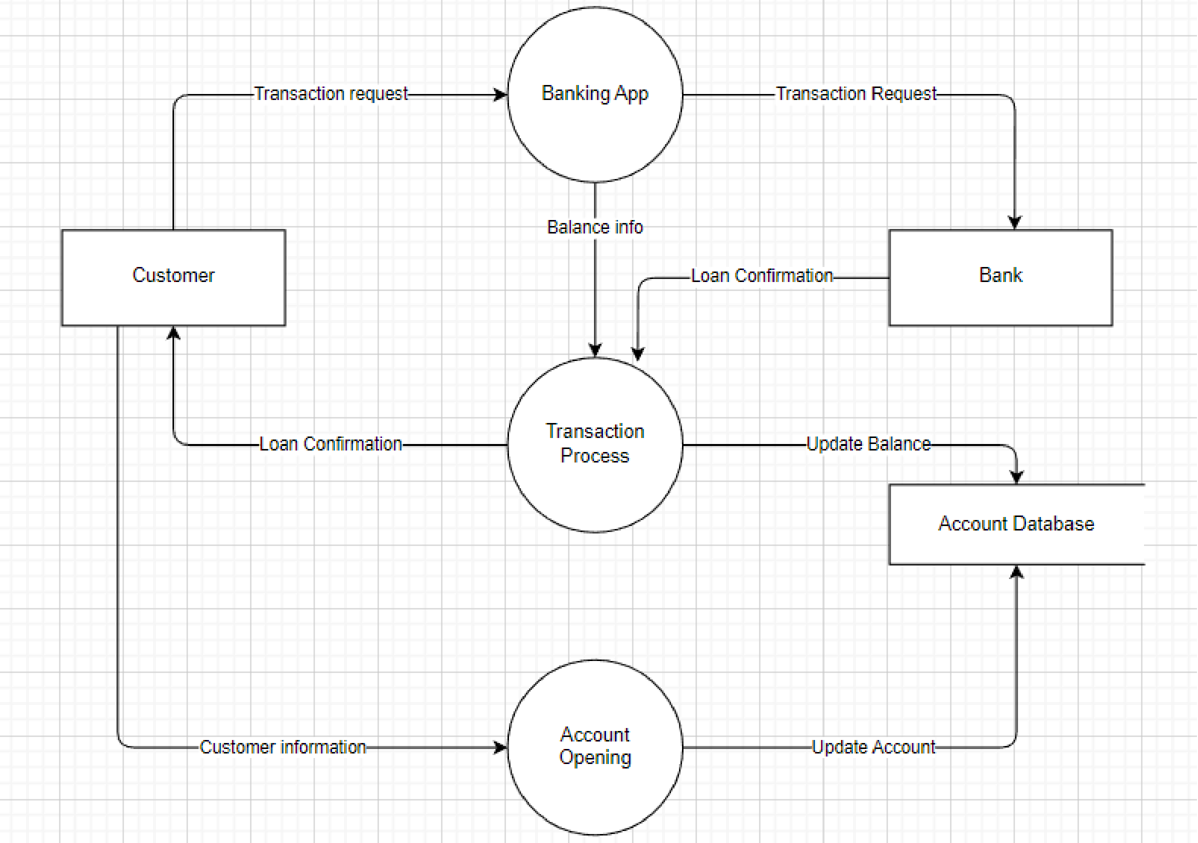

Figure 7 Level 1 DFD of Banking App System

ER Diagram

An Entity-Relationship Diagram is a visual representation of the relationships among entities in a database.

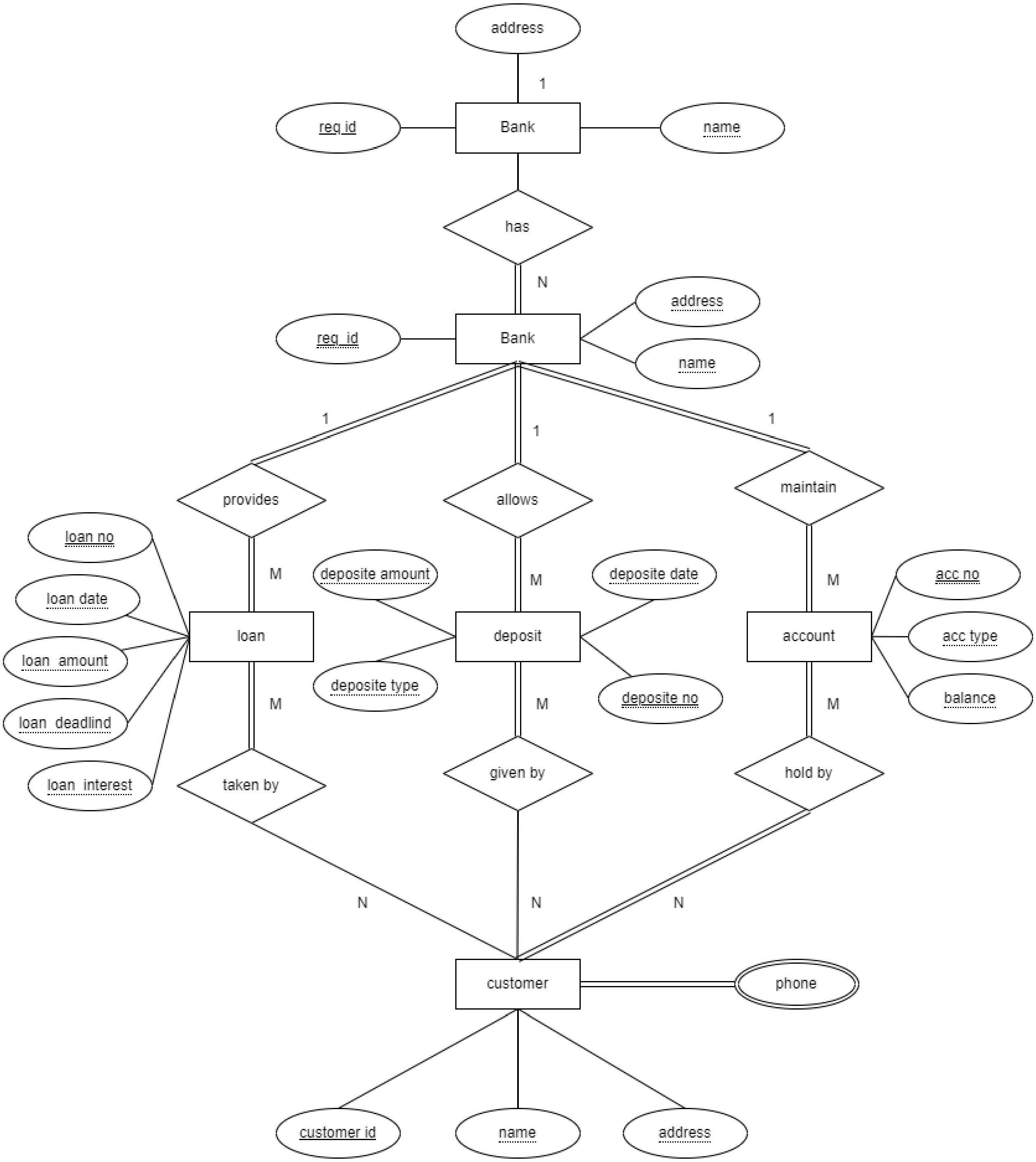

Figure 8 ER Diagram of Banking App System

Conclusion

The online banking app system is not just an app; it's a revolution. It's the dawn of a new era where banking is personalized, convenient, and secure. It's about empowering individuals to take control of their finances and build a brighter financial future. So, embrace the change, download the app, and experience the freedom of banking redefined. As mentioned in the introduction, human lives have one less thing to worry about, thanks to this system.