



TRIBHUVAN UNIVERSITY

Faculty of Management

Shanker Dev Campus

IT 242: SOFTWARE DESIGN AND DEVELOPMENT

BANK MANAGEMENT SYSTEM

A LAB REPORT

Submitted To

Department of BIM
Shanker Dev Campus
Putalisadak, Kathmandu

In partial fulfillment of the requirements for the Fifth Semester of the Bachelor of Information Management Degree

Submitted By

ASHESH NEUPANE
(TU Exam Roll No: 16295 / 23)

Under Supervision: **Mr. Laxman Bhandari**

April 2026

Supervisor's Signature: _____

Date: _____

Acknowledgement

Firstly, I would like to express my sincere gratitude to *Shanker Dev Campus* for providing me with the opportunity to conduct this project. The institution has offered a well-equipped and supportive environment that greatly facilitated the successful completion of this project.

I would like to extend my heartfelt thanks to Mr. Laxman Bhandari, Lecturer at *Shanker Dev Campus*, for his invaluable guidance and supervision throughout this laboratory work. His clear instructions, practical insights, and continuous support helped me to understand the experimental procedures and underlying concepts more effectively.

I am especially grateful for his patience, encouragement, and constructive feedback, which played a significant role in improving the accuracy and quality of this project. His willingness to provide assistance whenever needed made this learning experience both meaningful and productive.

Finally, I would like to acknowledge all those who directly or indirectly supported me during the completion of this project.

List Of Figures

Figure 1: Iterative Agile Methodology	2
Figure 2: Use case Diagram for Bank Management System.....	5
Figure 3: Activity diagram of BMS	11
Figure 4: Sequence diagram of BMS	12
Figure 5: E-R / Schema Diagram of BMS	13
Figure 6: Context Diagram of BMS	14
Figure 7: Level 0 DFD Diagram of BMS	15

Table of Contents

Chapter One: Introduction	1
1.1 Introduction	1
1.2 Objectives	1
1.3 Methodology	2
Chapter Two: Tasks and Activities Performed	3
2.1 Requirement Analysis	3
2.2 Feasibility Study	8
2.3 Solution Design	10
2.3.1 Activity Diagram	11
2.3.2 Sequence Diagram	12
2.3.3 E-R/Schema Diagram	13
2.3.4 Data Flow Diagram	14
2.4 System Development	16
2.4.1 Tools Used	16
2.4.2 Module Description	17
2.4.3 Techniques used in Testing	19
Chapter Three: Discussion and Conclusion	23
3.1 Discussion	23
3.2 Conclusion	24
3.3 Recommendations	25
References	26

Chapter One: Introduction

1.1 Introduction

A well-structured banking system is fundamental to ensuring financial stability, secure transactions, and efficient customer service in modern financial institutions. However, many banks still face challenges such as fragmented data management, delays in transaction processing and inadequate integration between departments. These issues can lead to operational inefficiencies, increased risk of errors, and reduced customer satisfaction. To address these challenges, this project proposes the development of a Bank Management System (BMS).

The proposed Bank Management System is structured around two primary roles: administrator and user. (The administrator oversees the entire system, including managing customer accounts, authorizing transactions, monitoring system activities, and maintaining data integrity. On the other hand, users interact with the system to perform operations such as opening accounts, depositing and withdrawing funds, transferring money, and accessing account statements [1].

This study focuses on the design, development, and implementation of the Bank Management System as a reliable Management Information System tailored for financial institutions. Ultimately, the Bank Management System serves as a reliable solution for modern banking needs, supporting both organizational performance and customer satisfaction.

1.2 Objectives

To develop a comprehensive Bank Management System (BMS) that enables administrators to efficiently manage customer accounts, oversee financial transactions, regulate loans and interest rates, and monitor overall banking operations, while also allowing users to perform essential activities such as account creation, deposits, withdrawals, fund transfers, and access to real-time account information [2]. By integrating these functionalities into a unified platform, the system enhances operational efficiency, strengthens data security, improves transparency, and supports faster decision-making, ultimately leading to improved customer satisfaction and optimized banking performance.

1.3 Methodology

The development of the Bank Management System (BMS) follows a systematic and structured approach to ensure the delivery of a secure, reliable, and efficient Management Information System tailored for financial operations. The methodology is outlined as follows:

- **System Development Approach**

- **Agile Development Methodology:** The system is built using an iterative, incremental process that continuously refines features based on stakeholder feedback, delivering stable banking functions with ongoing testing to ensure security, data integrity, and reliability [1].

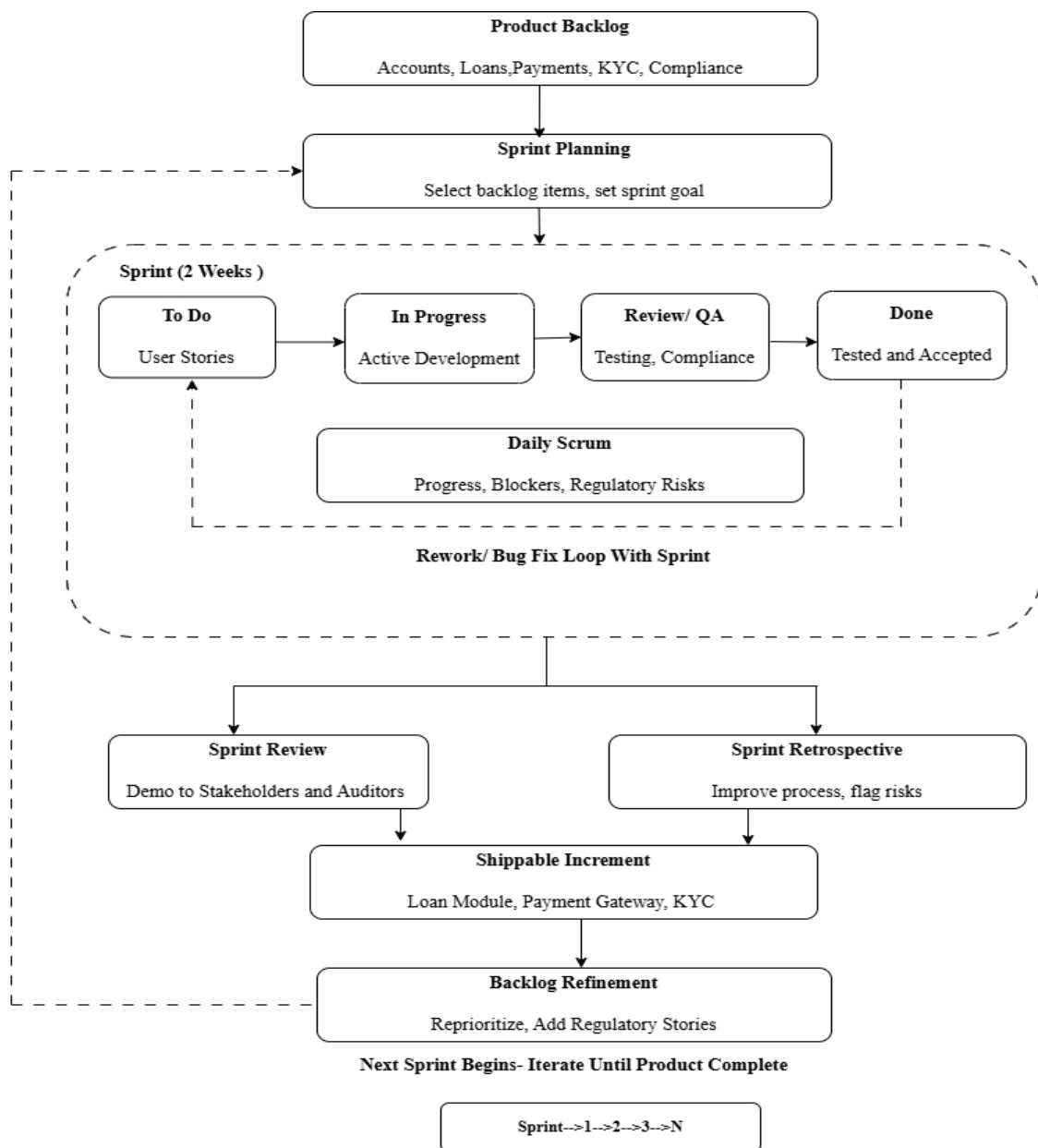


Figure 1: Iterative Agile Methodology

Chapter Two: Tasks and Activities Performed

2.1 Requirement Analysis

Requirement analysis is the process of identifying and specifying the expectations and needs for a new or enhanced product. Similarly, in this phase for the Bank Management System (BMS), the main requirements were identified to address both end-user (customers and bank staff) needs and organizational challenges such as operational inefficiencies, transaction delays, and data management issues [2]. This phase defines what the system should achieve while setting constraints for its implementation and operation. The requirements are categorized into functional and non-functional types. Functional requirements define the core services and capabilities the system must provide, whereas non-functional requirements cover additional qualities such as reliability, security, performance, and compliance with banking regulations [1].

a) Functional Requirements:

Some of the functional requirements of this banking-level MIS include:

- **User Authentication and Role Management**
 - **Admin Login:** The system must allow bank administrators to securely log in using unique credentials.
 - **Staff Login:** Bank staff should be able to log in with their assigned usernames and passwords.
 - **Customer Access:** Customers must be able to access their accounts securely through online or mobile interfaces.
 - **Role-Based Access Control:** The system must enforce access restrictions based on user roles, ensuring that admins, staff, and customers only access appropriate functionalities.
- **Account Management**
 - **Create Account:** Staff or customers (depending on permissions) should be able to create new bank accounts with required details stored in the system.

- **View Account:** Users must be able to view account details, balances, and transaction history.
- **Update Account:** Staff should be able to update customer account information securely.
- **Transaction Management**
 - **Deposit and Withdrawal:** The system should allow deposits and withdrawals, updating account balances in real time.
 - **Funds Transfer:** Both internal (same bank) and external transfers must be supported, with transaction validation and confirmation.
 - **Transaction History:** Users should be able to view detailed records of all completed transactions.
- **Loan and Interest Management**
 - **Loan Application:** Customers should be able to submit loan requests, which admins can review, approve, or reject.
 - **Interest Calculation:** The system must automatically calculate interest for loans and deposits based on predefined rules.
- **Audit and History Logging**
 - **Activity Log:** The system must maintain detailed logs of all critical actions, including account creation, transactions, loan processing, and staff operations, for auditing and regulatory compliance purposes.

This structured approach ensures the Bank Management System meets operational, regulatory, and user-focused requirements, providing a secure, efficient, and transparent banking environment.

Use Case Diagram

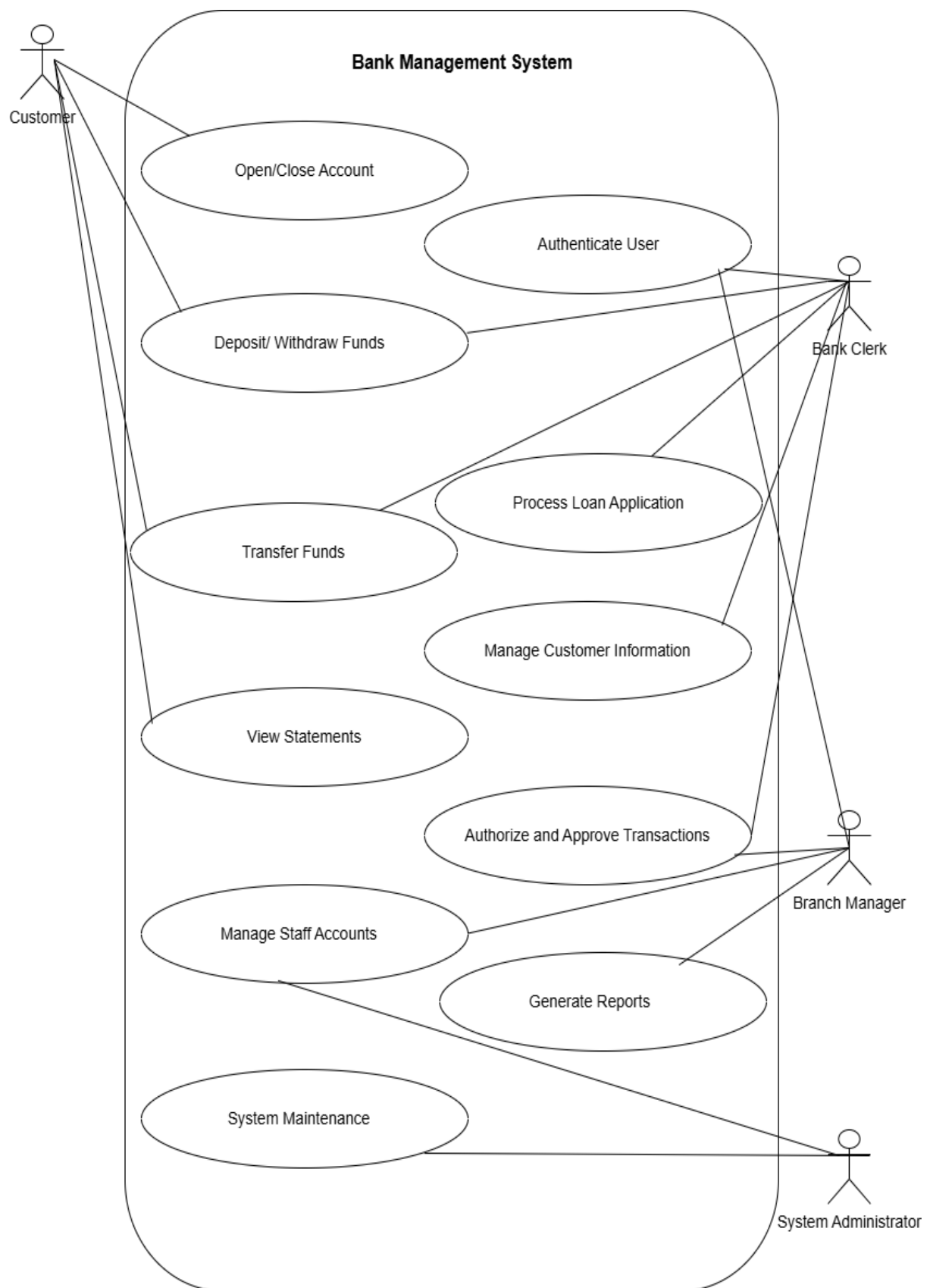


Figure 2: Use case Diagram for Bank Management System

b) Non-functional Requirements:

Non-functional requirements (NFRs) define the quality attributes and operational standards that the Bank Management System (BMS) must satisfy to ensure efficient, secure, and reliable performance. Unlike functional requirements, which describe specific system behaviors, non-functional requirements focus on how the system performs under various conditions and constraints [3]. The key non-functional requirements of the system are as follows:

- **Performance**
 - The system must process banking operations such as deposits, withdrawals, balance inquiries, and fund transfers with high speed and minimal latency, even during peak transaction periods.
 - Real-time transaction updates should be reflected instantly across all user interfaces to ensure accurate account information at all times.
- **Scalability**
 - The system should be capable of handling a growing number of customers, accounts, and transactions without degradation in performance.
 - It must support horizontal and vertical scaling to accommodate increased workloads as the bank expands its services and user base.
- **Security**
 - Sensitive data such as user credentials, financial records, and transaction details must be protected using strong encryption techniques and secure authentication mechanisms.
 - The system should enforce strict access control policies, ensuring that only authorized users can access or modify specific data.
 - Advanced security measures such as multi-factor authentication, fraud detection, and secure communication protocols should be implemented to prevent cyber threats and unauthorized access.

- **Usability**
 - The system should provide an intuitive and user-friendly interface for both bank staff and customers, enabling easy navigation and efficient execution of tasks.
 - Clear instructions, responsive design, and accessible features should be incorporated to enhance the overall user experience across different devices.
- **Reliability and Availability**
 - The system must maintain high reliability with minimal downtime, ensuring continuous access to banking services.
 - It should be available 24/7, with mechanisms to handle unexpected failures gracefully through redundancy and failover strategies.
- **Data Integrity**
 - The system must ensure accuracy and consistency of all financial data, preventing issues such as duplicate transactions or incorrect balances.
 - Concurrency control mechanisms should be in place to maintain data consistency when multiple users perform transactions simultaneously.
- **Auditability**
 - All critical operations, including account activities, transactions, and administrative changes, must be logged with timestamps and user details.
 - These logs should support auditing, regulatory compliance, and troubleshooting processes.
- **Backup and Recovery**
 - The system should perform regular automated backups of all critical data to prevent loss due to system failures or cyber incidents.
 - Efficient recovery mechanisms must be available to restore data quickly and ensure business continuity.

- **Compliance**

- The system must adhere to financial regulations, data protection laws, and industry standards related to banking operations.
- It should ensure customer data privacy, secure transaction handling, and compliance with applicable legal and regulatory frameworks.

These non-functional requirements ensure that the Bank Management System operates efficiently, securely, and reliably while meeting the demands of modern banking environments and maintaining user trust [3].

2.2 Feasibility Study

A feasibility study is a critical step in evaluating the practicality and potential success of a project before allocating significant resources. It determines whether the proposed Bank Management System (BMS) is viable from technical, economic, legal, and operational perspectives. Below is a detailed assessment of the feasibility of implementing the proposed banking system:

a) Technical Feasibility:

This aspect evaluates whether the organization has the necessary technological resources and capabilities to implement and maintain the system effectively.

- **Resources:** The bank must ensure the availability of reliable hardware, secure servers, database systems, and network infrastructure to support continuous and secure banking operations.
- **Capacity:** The existing infrastructure should be capable of handling large volumes of transactions, data storage, and multiple users accessing the system simultaneously without performance issues.
- **Expertise:** The technical team must have the required skills to develop, deploy, and maintain the system. If not available internally, external experts or training programs may be required. The Bank Management System is technically feasible if proper infrastructure and skilled personnel are available or can be developed.

b) Economic Feasibility:

This assessment focuses on analyzing the cost-effectiveness of the system:

- **Development Costs:** These include expenses related to hardware acquisition, software development, licensing, and salaries of technical staff.
- **Operational Costs:** These involve ongoing expenses such as system maintenance, updates, training, and technical support.
- **Benefits:** Expected benefits include improved efficiency, reduced manual errors, faster transaction processing, and enhanced customer satisfaction, which can lead to increased profitability.

The system is economically feasible if the long-term benefits outweigh the initial and operational costs.

c) Legal Feasibility:

This aspect ensures that the system complies with all relevant laws and regulations:

- **Data Protection Laws:** The system must protect sensitive customer and financial data in accordance with data privacy regulations.
- **Financial Regulations:** It must comply with banking laws, anti-money laundering (AML) policies, and other regulatory requirements.
- **Licensing:** All software and third-party tools used in the system must be properly licensed and legally compliant. The Bank Management System is legally feasible as long as it adheres to all applicable legal and regulatory standards.

d) Operational Feasibility:

Operational feasibility evaluates how well the system fits within the organization's operations:

- **Alignment with Goals:** The system supports key banking functions such as account management, transaction processing, and reporting, aligning with organizational objectives.
- **User Acceptance:** With proper training and an intuitive interface, both staff and customers can easily adopt and use the system.
- **Process Compatibility:** The system should integrate smoothly with existing workflows, improving efficiency without causing disruptions. The Bank Management System is operationally feasible as it meets organizational needs and enhances the overall efficiency and effectiveness of banking operations.

2.3 Solution Design

System design is the process of defining the key elements of a system, including its modules, architecture, components, interfaces, and data structures, based on the specified requirements. It involves carefully planning how different parts of the system will interact and function together to achieve the desired objectives. This phase focuses on translating requirements into a structured blueprint that guides the development process, ensuring that the system is efficient, scalable, and reliable [3]. Additionally, system design includes decisions related to technology selection, database design, user interface layout, and system integration. Overall, it is a critical stage in the development lifecycle that ensures the final system effectively meets the specific needs and operational goals of a business or organization.

2.3.1 Activity Diagram

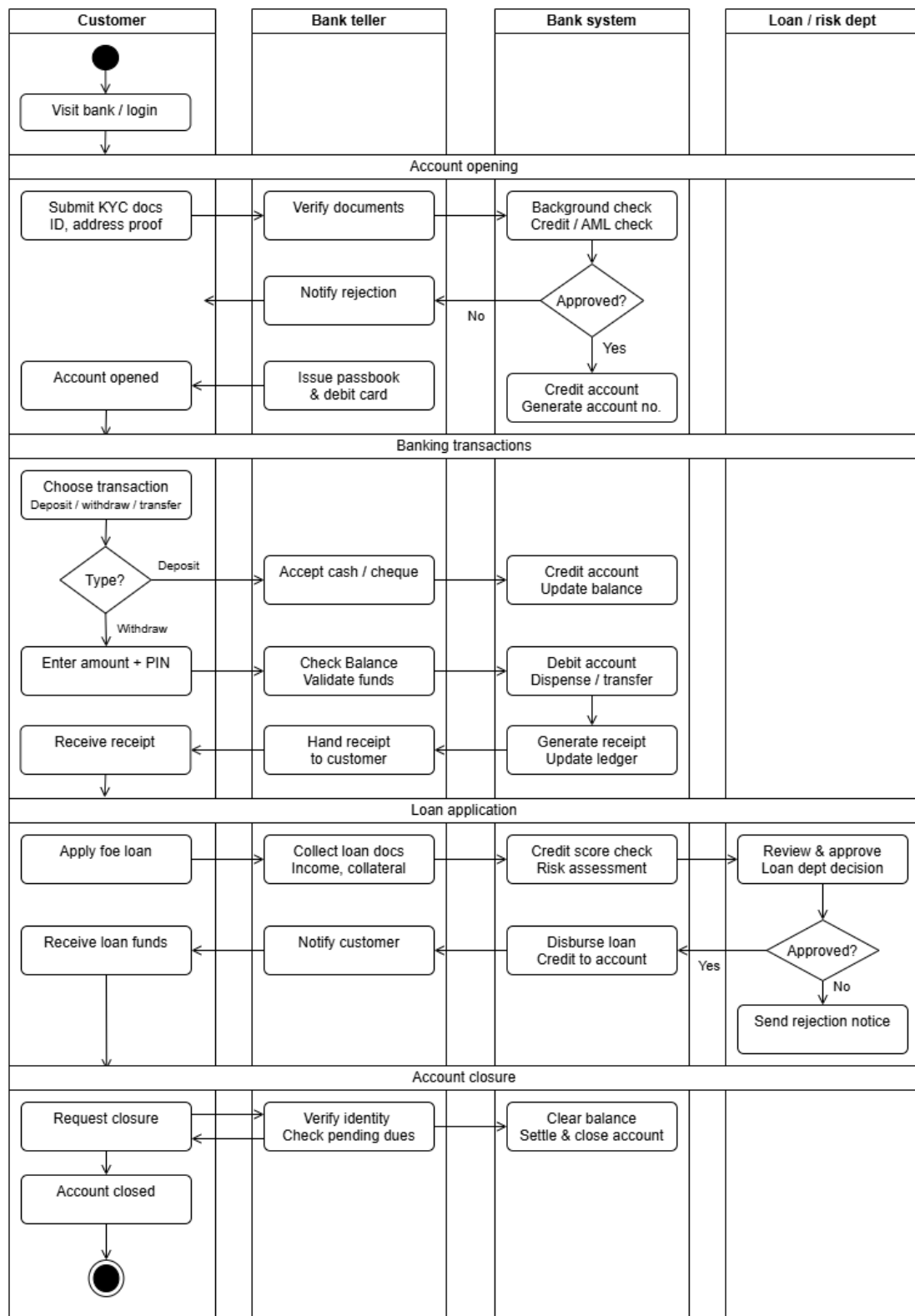


Figure 3: Activity diagram of BMS

2.3.2 Sequence Diagram

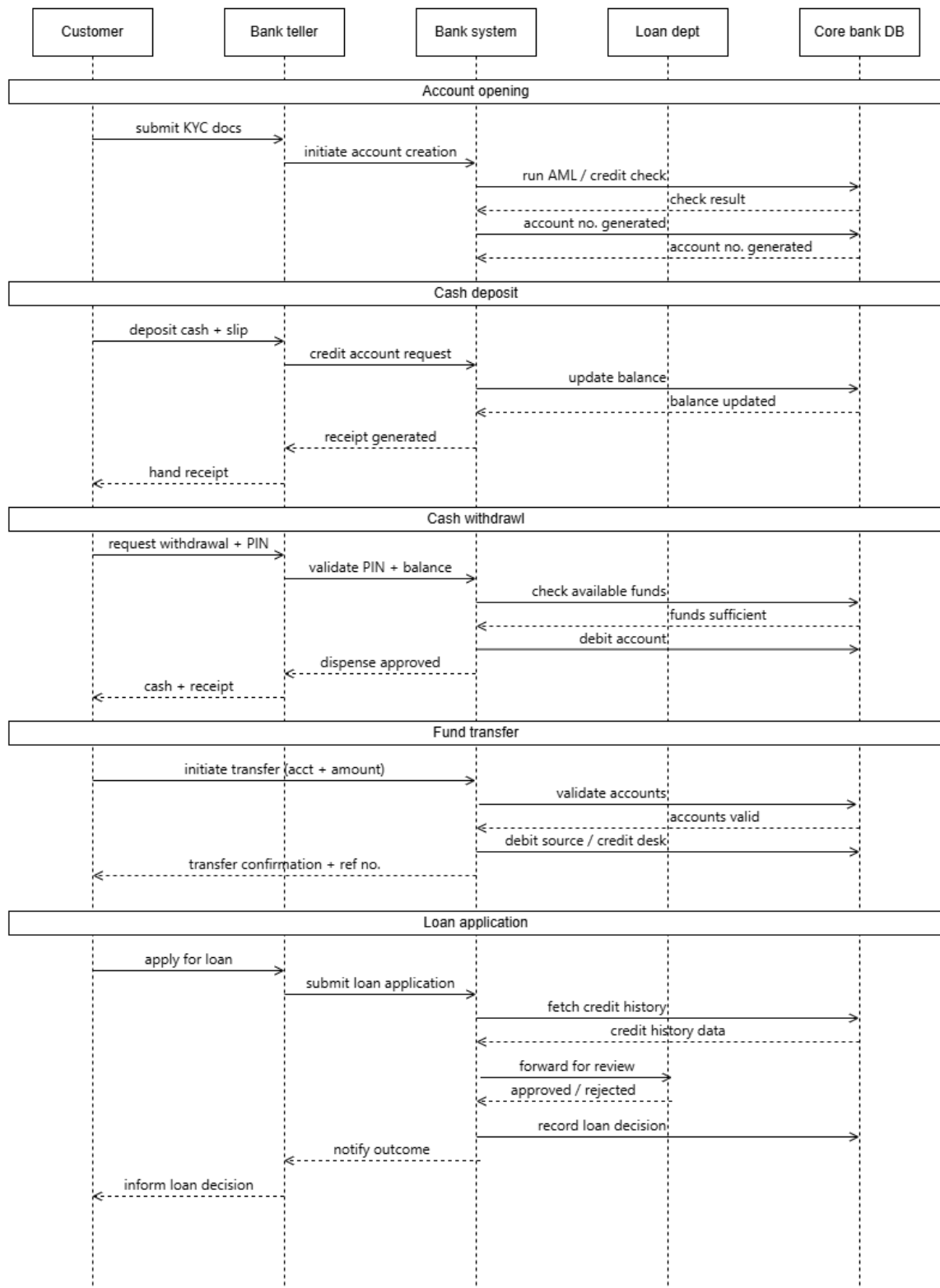


Figure 4: Sequence diagram of BMS

2.3.3 E-R/Schema Diagram

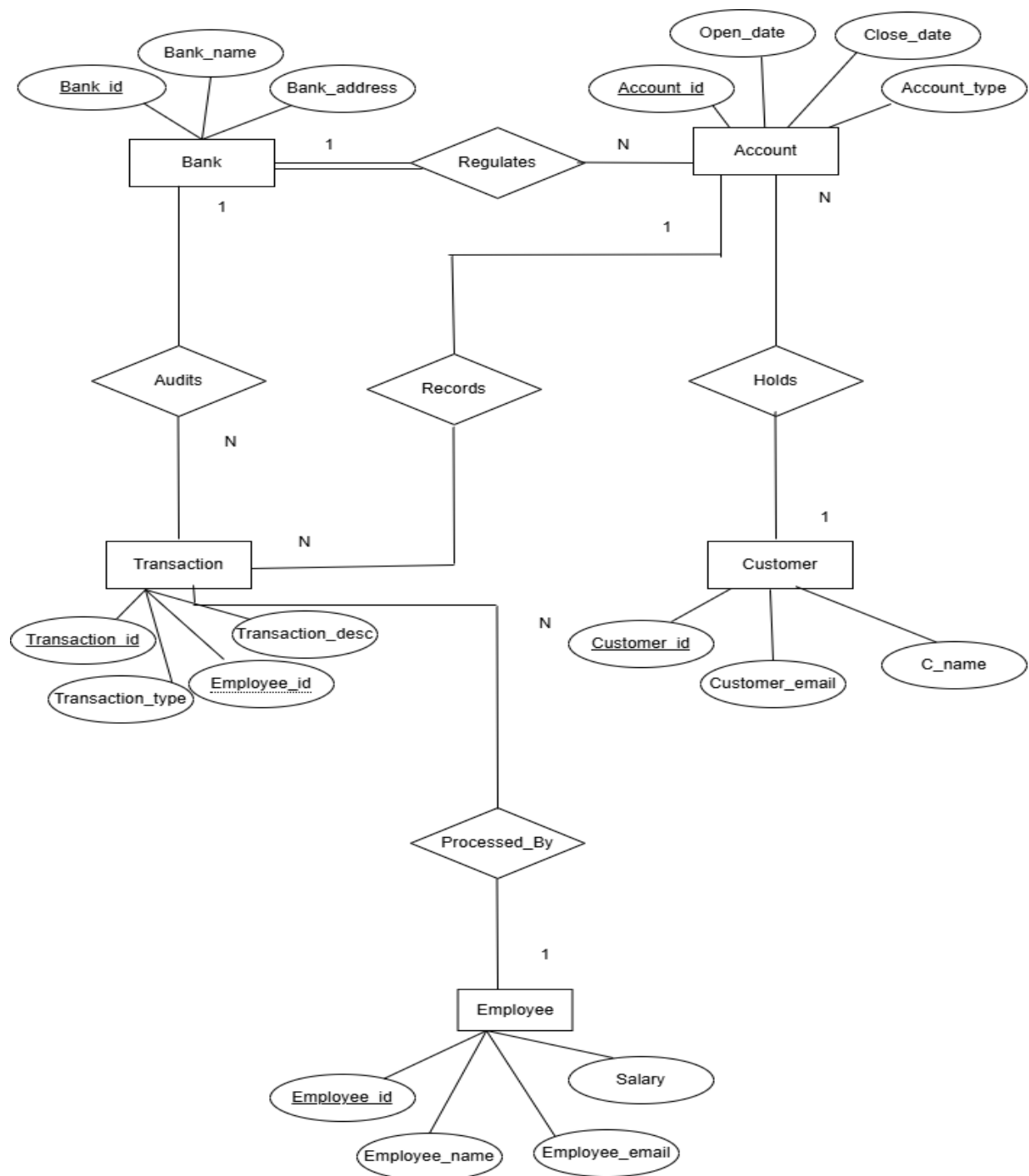


Figure 5: E-R / Schema Diagram of BMS

2.3.4 Data Flow Diagram

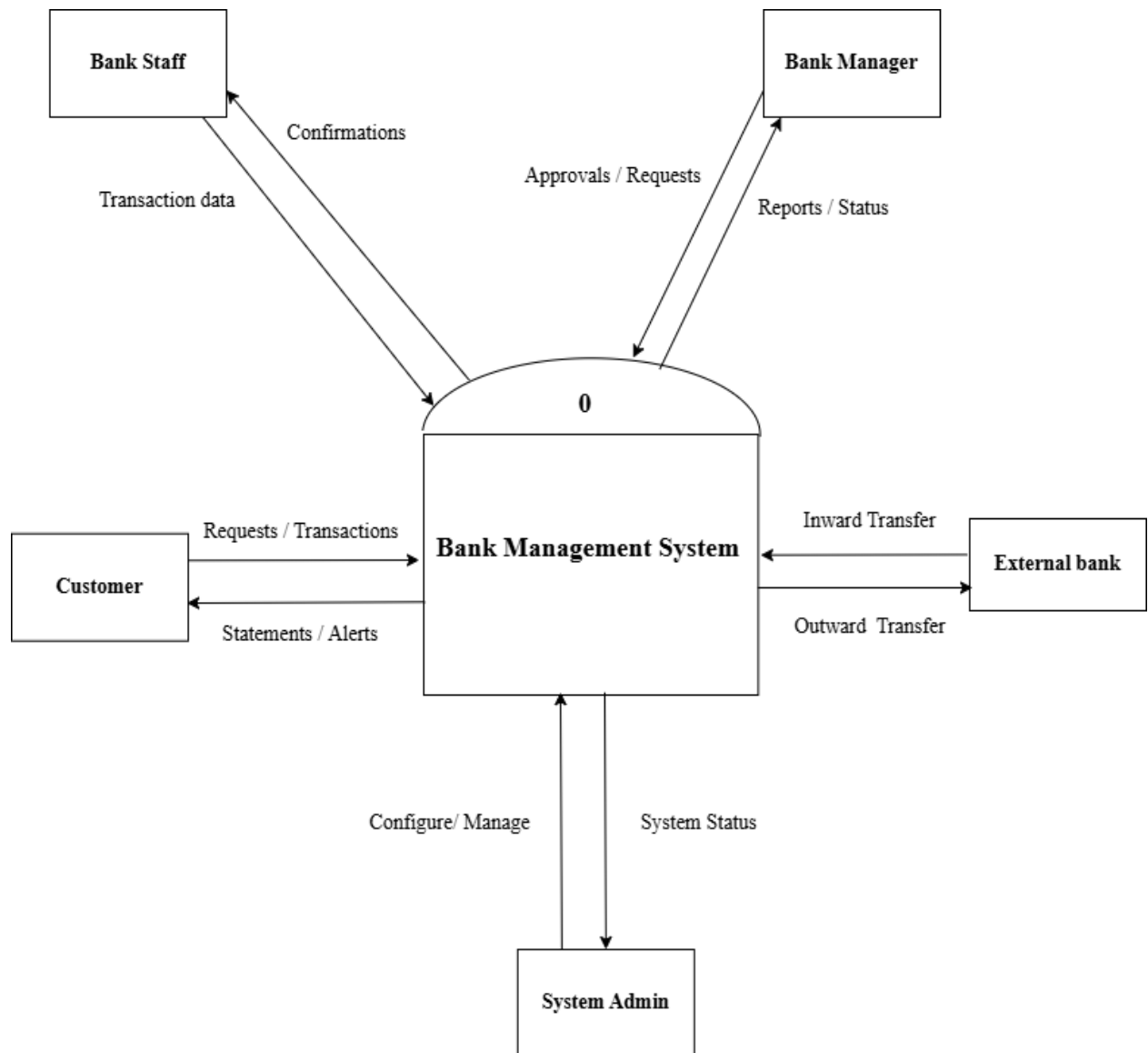


Figure 6: Context Diagram of BMS

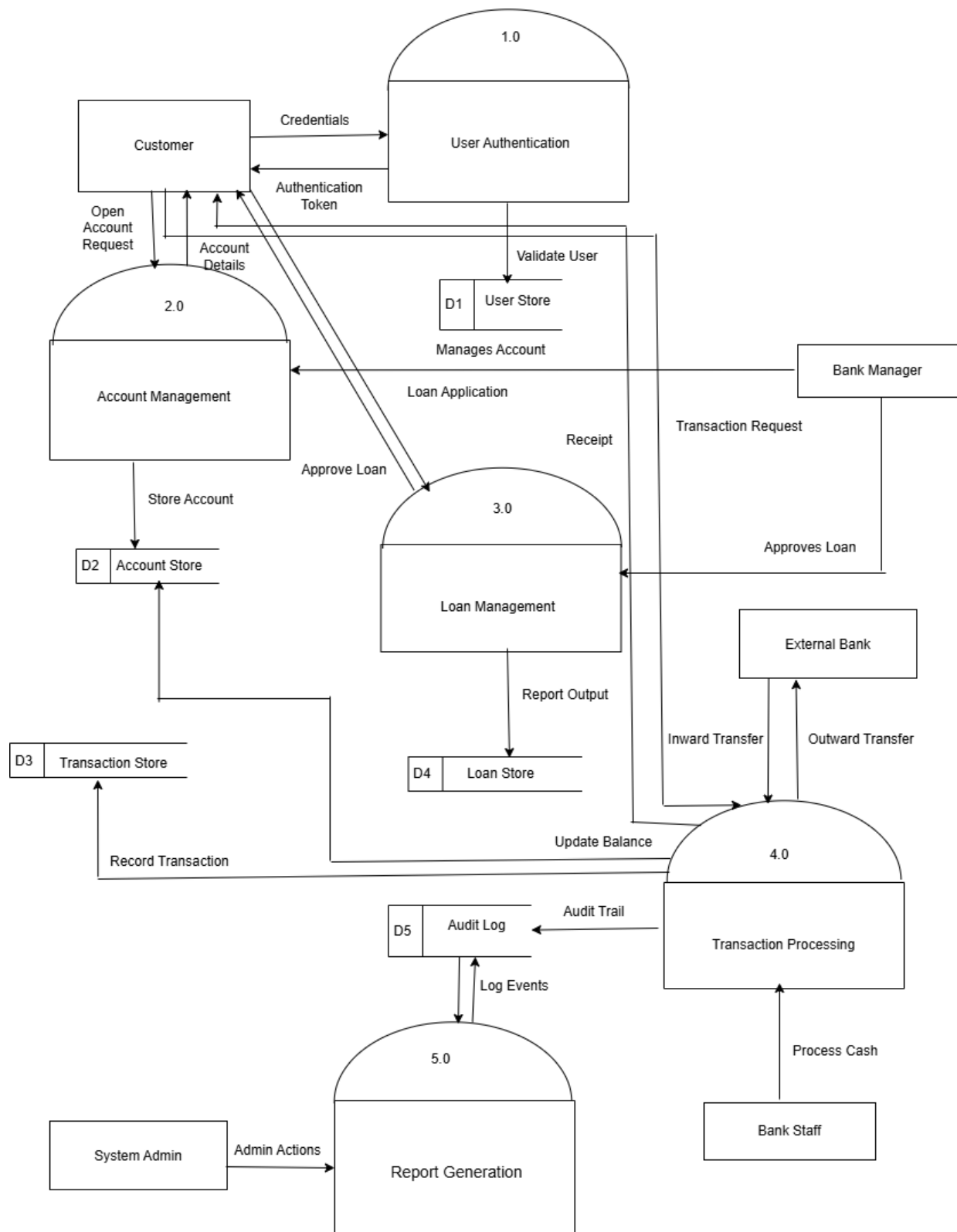


Figure 7: Level 0 DFD Diagram of BMS

2.4 System Development

In this stage, the executable software, i.e., the Bank Management System (BMS), is developed using a combination of modern web development tools and technologies. The development process focuses on creating a secure, scalable, and efficient system that supports core banking operations such as account management, transaction processing, and customer services. [5] The tools and technologies used in this system are outlined below:

2.4.1 Tools Used

The tools used for the development of the system are categorized as follows:

- **Frontend (React):**

Frontend tools are used to design and develop the user interface of the system.

- React components are used to build dynamic and responsive user interfaces for customers and bank staff.
- State management tools such as Redux or Context API are used to manage application states, including user sessions, account details, and transaction data.
- Axios is used to enable communication between the frontend and backend systems for seamless data exchange.

- **Backend (Django):**

Backend tools are responsible for handling business logic, data processing, and database management.

- The Django framework is used to manage core banking functionalities such as account creation, transaction processing, loan management, and user authentication.
- Django REST Framework (DRF) is used to develop secure API endpoints for communication between the React frontend and Django backend.
- PostgreSQL or a similar relational database is used to store customer information, account details, and transaction records [2].

- **Design Tools:**

Design tools are used to create system diagrams and database models.

- draw.io is used to design UML diagrams, flowcharts, and system architecture representations.
- ERD Plus is used to create entity-relationship diagrams, relational schemas, and database structures for the system.

2.4.2 Module Description

The Bank Management System is designed to manage banking operations efficiently by defining clear roles for administrators (bank staff) and customers [3]. The system consists of the following key modules and workflows:

- **Admin Login Module:**

- **Functionality:** Allows administrators or bank staff to securely log into the system.
- **Actions:**
 - Admin enters login credentials.
 - The system authenticates the credentials and grants access upon successful verification.

- **Account Management Module:**

- **Create Account:**
 - **Admin Action:** Staff creates new customer accounts by entering required details.
 - **System Response:** The system validates and stores the account information, confirming successful creation.
- **View Account:**
 - Staff and customers can view account details, including balance and account status.

- **Transaction Management Module:**
 - **Deposit/Withdraw Funds:**
 - **User Action:** Customers or staff initiate deposit or withdrawal requests.
 - **System Response:** The system processes the transaction and updates the account balance in real time.
 - **Fund Transfer:**
 - **User Action:** Customers transfer funds between accounts.
 - **System Response:** The system verifies and completes the transaction securely.
- **Customer Login Module:**
 - **Functionality:** Enables customers to access their accounts and perform banking operations.
 - **Actions:**
 - Customers log in using secure credentials.
 - The system authenticates and provides access to account features.
- **Viewing Account and Transactions:**
 - **View Account Details:**
 - Customers can view their account balance and profile information.
 - **View Transaction History:**
 - Users can access detailed transaction records for transparency.
- **Transaction Verification and Processing:**
 - **Verify Transactions:**
 - The system validates each transaction for accuracy and security before processing.
 - **Process Transactions:**
 - Once verified, transactions are executed and recorded in the database.

- **Admin Monitoring and Reporting Module:**

- **Monitor Activities:**

- Admin can track all banking operations, including transactions and account activities.

- **Generate Reports:**

- The system generates financial reports, summaries, and analytics for decision-making.

This structured system development ensures that the Bank Management System operates efficiently, securely, and reliably while meeting the needs of both the bank and its customers.

2.4.3 Techniques used in Testing

a) Unit Testing

1. Unit Testing for Customer Login Module

This tests the authentication process for a customer trying to access their online banking dashboard.

Test Case	Preconditions	Steps to be executed	Expected Results	Actual Results	Pass/Fail
Login with valid credentials	Customer account must be registered and active	1) Enter valid username and password 2) Click "Login"	Customer should log in successfully and be redirected to the account dashboard.	Customer logged in and redirected to dashboard.	Pass
Login with invalid credentials	Customer account must be registered	1) Enter valid username and password 2) Click "Login"	Customer should not log in. System must display an "Invalid username or password" error message	Error message displayed	Pass

2. Unit Testing for Open Account Module (Bank Clerk)

This tests the form validation and creation process when a bank clerk creates a new account for a customer.

Test Case	Preconditions	Steps to be executed	Expected Results	Actual Results	Pass/Fail
Create account with valid complete data.	Bank clerk is logged into the system.	1) Fill mandatory fields (Name, Gov ID, DOB, Deposit) 2)Click“Submit”	Account created successfully in the database. A unique Account ID is generated and displayed.	Account ID generated successfully.	Pass
Create account with missing mandatory data.	Bank clerk is logged into the system.	1) Fill Name and DOB but leave Gov ID blank 2)Click“Submit”	System blocks submission and highlights the missing field with a "This field is required" error.	Error message prompts for missing Gov ID.	Pass

3. Unit Testing for Deposit Funds Module

This tests the logic of adding money to an existing account balance.

Test Case	Preconditions	Steps to be executed	Expected Results	Actual Results	Pass/Fail
Deposit a valid positive amount	Target account exists and is active.	1. Enter target account number. 2. Enter a deposit amount of \$500. 3. Click "Deposit."	Account balance increases by \$500. A success transaction record is generated.	Balance updated and receipt generated.	Pass

Deposit a valid negative amount	Target account exists and is active.	1. Enter target account number. 2. Enter a deposit amount of -\$100. 3. Click "Deposit."	System rejects the transaction. Displays error: "Amount must be greater than zero."	Error message displayed; balance unchanged	Pass
---------------------------------	--------------------------------------	--	--	--	------

4. Unit Testing for Fund Transfer Module

This tests the critical logic of moving money between accounts, ensuring a user cannot overdraw.

Test Case	Preconditions	Steps to be executed	Expected Results	Actual Results	Pass/Fail
Transfer with sufficient funds	Sender account has \$1,000 balance. Target account exists.	1. Enter target account. 2. Enter transfer amount of \$200. 3. Click "Transfer."	Sender balance becomes \$800. Target balance increases by \$200. Success message shown.	Balance updated and receipt generated.	Pass
Transfer with insufficient funds	Sender account has \$50 balance. Target account exists.	1. Enter target account. 2. Enter transfer amount of \$200. 3. Click "Transfer."	System rejects the transaction. Displays error: "Amount must be greater than zero."	Transaction blocked. System displays error: "Insufficient funds for this transfer."	Pass

5. Unit Testing for Process Loan Application Module

This tests the workflow for loan approvals, handled by a Branch Manager or Loan Officer

Test Case	Preconditions	Steps to be executed	Expected Results	Actual Results	Pass/Fail
Approve valid loan application	Customer has applied for a loan. Manager is logged in.	1. Navigate to pending loans. 2. Review valid application. 3. Click "Approve."	Loan status changes to "Approved". Loan amount is scheduled for disbursement to customer account	Status updated to "Approved".	Pass
Reject loan application	Customer has applied for a loan. Manager is logged in.	1. Navigate to pending loans. 2. Review application. 3. Click "Reject" and add reason. 4. Submit.	Loan status changes to "Rejected". Customer receives notification with the stated reason.	Status updated to "Rejected".	Pass

Chapter Three: Discussion and Conclusion

3.1 Discussion

Throughout the development and implementation of the Bank Management System (BMS), several challenges were identified and effectively addressed to ensure the system met organizational requirements and industry standards.

Requirement Analysis:

One of the most critical phases was identifying both functional and non-functional requirements. In the context of a banking system, this included defining features such as account management, transaction processing, loan management, and customer data handling. This phase ensured that the system could handle real-world banking operations securely and efficiently while complying with financial regulations.

Technical Challenges:

The integration of the frontend (React) with the backend (Django) required careful architectural planning. Ensuring seamless communication between the user interface and server-side logic was achieved through well-designed RESTful APIs. Another significant challenge was maintaining data consistency and integrity, where multiple transactions could occur simultaneously [5].

Testing and Quality Assurance:

Extensive testing was conducted to ensure the system's reliability, accuracy, and security. Unit testing was performed on core banking functionalities such as deposits, withdrawals, transfers, and account updates. Role-based access control was rigorously validated to ensure that sensitive financial data remained protected and accessible only to authorized personnel [4].

User Feedback:

Feedback from potential users, including bank staff and administrative personnel, played a crucial role in refining the system. Their insights helped improve the usability of the interface, simplify workflows, and ensure that the system aligned with daily banking operations. Continuous feedback during development allowed iterative enhancements, resulting in a more practical and user-friendly system. Despite achieving the primary objectives, there is still room for improvement.

3.2 Conclusion

The Bank Management System (BMS) successfully addresses the key challenges faced by financial institutions in managing customer accounts, transactions, and operational workflows [2]. By implementing a structured and efficient Management Information System, the solution enhances the accuracy, security, and efficiency of banking operations while improving communication between staff and management.

Key accomplishments of the system include:

1. **Improved Operational Efficiency:**

Bank administrators and staff can efficiently manage customer accounts, process transactions, and oversee daily banking activities with greater accuracy and reduced manual effort.

2. **Time Optimization:**

Automated processes for transactions, account updates, and reporting have significantly reduced processing time, enabling faster service delivery to customers.

3. **Enhanced Monitoring and Control:**

The system provides real-time tracking of transactions and account activities, allowing administrators to monitor operations effectively and detect anomalies promptly.

4. **Scalability and Security:**

The system is designed to support a growing number of users and transactions while maintaining high security standards. Data encryption, authentication mechanisms, and secure access controls ensure the protection of sensitive financial information [4].

The project successfully met its objectives through the adoption of an Agile development methodology, allowing continuous improvements and incorporation of user feedback. This iterative approach ensured that the final system is robust, secure, and aligned with modern banking requirements.

3.3 Recommendations

To further enhance the performance and capabilities of the Bank Management System, several improvements are recommended:

- **Advanced Analytics and Reporting:**

Implement detailed financial analytics and reporting tools to provide insights into customer behavior, transaction trends, and overall bank performance.

- **AI-Based Features:**

Incorporating artificial intelligence can enable fraud detection, predictive analysis for loan approvals, and intelligent recommendations for financial products.

- **Mobile Banking Integration:**

Developing a mobile application would improve accessibility, allowing customers and staff to perform banking operations anytime and anywhere.

- **Integration with External Systems:**

Linking the system with third-party payment gateways, ATM networks, and digital wallets would expand its functionality and improve customer convenience.

- **Regular Updates and Training:**

Continuous system updates are essential to address security vulnerabilities and technological advancements. Additionally, training programs for staff will ensure effective utilization of the system.

Overall, these enhancements will contribute to making the Bank Management System more efficient, secure, and adaptable to the evolving needs of the banking industry

References

- [1] S., Ritesh; N., Pavitha, "The Design and Implementation of Bank Management System," *Grenze International Of Engineering and Technology*, vol. 8, no. 2, p. 86, 2022.
- [2] B. Neverauskas, "Conceptional Model of Commercial Bank Management," *Engineering Economics*, no. Kauno Technologijos Universitetas, p. 7, 2008.
- [3] Chen, Yehning; Hasan, Iftekhar, "The transparency of the banking system and the efficiency of information-based bank runs," *Journal of Financial Intermedian*, vol. 15, no. 3, p. 24, 2006.
- [4] P. Rai, A. Singh, T. Sharma and N. M. Sreenarayan, "Bank Management System- The Web Application Way," in *Applications of Computational Methods in Manufacturing and Product Design*, Rourkela, India, Springer , 2022, p. 9.
- [5] Shah, Mahmood; Larke, Steve, E-Banking Management, London, UK: Yurchak Printing Inc, 2009.