User Management and Financial Transaction System

Jiani Jin

**Abstract**

The User Management and Financial Transaction System is a comprehensive, user-friendly platform that enables users to manage their personal information, conduct financial transactions, and review their transaction history. Simultaneously, it provides administrators with tools for overseeing user accounts, viewing financial statistics, and ensuring smooth system operations. This project employs a modular design using Java Swing for its graphical user interface (GUI) and MySQL for secure data storage and retrieval.

The system consists of four key components:

RegisterFrame: A registration form that allows new users to create an account.

LoginFrame: A unified login interface for users and administrators.

MainFrame: The primary dashboard where users can manage their personal information, deposit, withdraw, and transfer funds.

AdminMainFrame: The administrative dashboard, offering access to system-wide statistics and user information.

This report provides a detailed explanation of the system's purpose, database design, core functionalities, and user workflows. The system prioritizes security, usability, and efficiency, ensuring a seamless experience for users and administrators. The system's modular nature facilitates future updates, making it suitable for expansion and further customization.

**Introduction**

In today’s fast-paced world, financial management systems have become essential for handling personal accounts and transactions. The demand for a reliable, secure, and user-friendly system has grown significantly, especially as users seek greater control over their financial activities. This project, the User Management and Financial Transaction System, addresses this need by providing users and administrators with a robust platform to manage user accounts, track transactions, and maintain oversight of financial operations.

The system offers a dual-role approach, supporting both regular users and administrators. Users can register for an account, log in, view personal information, deposit and withdraw funds, and transfer money to other users. Each user action is recorded in a history log, ensuring transaction transparency. On the other hand, administrators are granted higher privileges through a dedicated Admin Dashboard, where they can view financial statistics, access user information, and analyze system-wide trends.

The system's user interface is designed using Java Swing, offering a clean, simple, and user-friendly experience. The MySQL relational database ensures that all data, including user details, account balances, and transaction history, is stored securely and efficiently. The system follows a modular architecture, allowing for scalability and future enhancements. This project report outlines the system's purpose, database design, and implementation details, providing a comprehensive view of how the system works.

The key objectives of the system are as follows:

User Registration and Authentication: Allow new users to register and log in using secure login credentials.

User Account Management: Enable users to view and update personal information such as name, password, and birthday.

Transaction Features: Facilitate deposits, withdrawals, and transfers of funds between users.

Admin Dashboard: Provide administrators with system control features, including user search, system statistics, and financial summaries.

This system is designed to be secure, scalable, and user-friendly, ensuring that users can manage their finances efficiently while providing administrators with essential tools for system oversight. The following sections provide a comprehensive breakdown of the system's database design, core functionalities, and implementation details.

**Database Design**

The database design is a critical component of the User Management and Financial Transaction System, as it ensures data integrity, consistency, and efficient management of user, administrator, and transaction-related data. The system employs a MySQL relational database to store and manage key information about users, administrators, departments, and transaction history. Each table is designed to handle a specific aspect of the system, with appropriate constraints to maintain data accuracy.

The database design is represented by an Entity-Relationship Diagram (ERD), which illustrates the relationships between the main tables. The key entities in the system include:

Admin: Tracks system administrators who manage users and view system statistics.

User: Represents individual users who register, log in, and perform financial transactions.

Dept: Represents departments to which users can be linked.

History: Logs every financial transaction (deposits, withdrawals, and transfers) made by users.

The relationship between these entities is shown below:

[Admin] 1 --- N [History]

[User] 1 --- N [History]

[Dept] 1 --- N [User]

The system is composed of four key database tables: Admin, User, Dept, and History. Each table is uniquely designed to handle a specific part of the system. Below is a description of the tables, their columns, and their purpose.

1. Admin Table

This table stores details of system administrators. Admins have higher privileges, allowing them to manage users and view system statistics.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Constraint | Description |
| id | Int | Primary Key | Unique identifier for the admin |
| Username | Varchar (255) | Not NULL | Admin's login username |
| Password | Varchar (225) | Not NULL | Admin's login password |
| Real\_name | Varchar (225) | NULL | Admin's real name |
| Sex | Char (2) | NULL | Admin's gender |
| birthday | Datetime (6) | NULL | Admin's date of birth |

1. User Table

The User Table stores user-related information, including personal details, login credentials, and account balance.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Constraint | Description |
| Id | Int | Primary Key | Unique identifier for the user |
| Card\_no | Int | Unique | Unique card number for login |
| Password | Varchar (225) | NULL | User's login password |
| Real\_name | Varchar (225) | NULL | User's full name |
| Balance | Double | NULL | User's current balance |
| Birthday | Datetime (6) | NULL | User's date of birth |
| did | int | NULL | Foreign key linking to Dept |

1. Department Table

The Department Table stores details of organizational departments to which users can be linked.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Constraint | Description |
| Did | Int | Primary Key | Unique identifier for department |
| Dept\_no | Int | NULL | Department number |
| Dept\_name | Varchar (225) | NULL | Name of the department |

1. History Table

The History Table logs all financial transactions performed by users.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Constraint | Description |
| Id | Int | Primary Key | Unique identifier for transaction |
| User\_id | Int | Foreign Key | User ID who performed the transaction |
| Type | Tinyint (0) | NULL | Transaction type (Deposit, Withdrawal, Transfer) |
| Money | Double (10, 2) | NULL | Amount of money involved |
| time | Datetime (6) | NULL | Timestamp of the transaction |

The History Table allows for full traceability of all transactions, ensuring users and administrators have a complete audit trail of the system's financial activities.

**Project Implementation**

The implementation of the User Management and Financial Transaction System revolves around four key interfaces: RegisterFrame, LoginFrame, MainFrame, and AdminMainFrame. Each interface has specific functionalities that support the overall system's purpose, including user registration, login, financial transactions, and administrative control. The system follows a clear and structured workflow, enabling users to register, log in, manage their accounts, and conduct transactions, while administrators have access to a specialized dashboard with oversight capabilities.

The system employs Java Swing to creatMySQL for data storage and retrieval. The design and implementation ensure data integrity, security, and scalability, while the system’s modular structure allows for easy future updates and enhancements. Below is a comprehensive description of each major part of the system's implementation.

User Registration

The RegisterFrame is the interface where new users can create an account. Registration is the first step in the system's workflow and is essential for enabling new users to gain access to their accounts. The RegisterFrame form prompts users to enter the following details:

Username (Card Number): A unique identifier for each user.

Password: Used for secure login authentication.

Real Name: The user's full name.

Gender: The user's gender (e.g., Male, Female, Other).

Balance: The initial deposit amount for the user's account.

Birthday: The user's date of birth.

The user enters this information into the corresponding input fields of the RegisterFrame. Once all fields are completed, the user clicks the "Save" button to create their account. Upon clicking, the following key processes take place:

1. Input Validation: The system ensures that all fields are filled correctly. For instance, the balance field must be a numeric value, and the card number must be an integer.
2. User Data Insertion: The system calls the UserDao.register(user) method to store the user's information in the User Table in the MySQL database.
3. Redirect to Login: If registration is successful, the RegisterFrame is closed, and the user is redirected to the LoginFrame.

If any of the input fields are incorrect or incomplete, the system will not allow registration, ensuring data integrity. The UserDao class handles all database-related tasks, ensuring that user information is properly inserted into the system. This process ensures that only valid and complete user data is saved in the system.

User & Admin Login

Once a user is registered, they can log in to the system via the LoginFrame. The LoginFrame is the system's entry point and serves as a shared login interface for both users and administrators. The login process requires users to enter a username (card number) and password.

The interface offers two login options:

User Login: If the user selects the "User" option, they are directed to the MainFrame after successful login.

Admin Login: If the user selects the "Administrator" option, they are directed to the AdminMainFrame.

The login process works as follows:

1. Input Validation: The system checks that the username and password are not empty.
2. User Authentication: If the user selects "User", the system checks the User Table via the UserDao.login(username, password) method. If the login is successful, the user is redirected to the MainFrame.
3. Admin Authentication: If the user selects "Administrator", the system checks the Admin Table via the AdminDao.login(username, password) method. If the login is successful, the user is redirected to the AdminMainFrame.
4. Error Handling: If the login fails (e.g., incorrect password), the system displays an error message: "Wrong username or password".

This system ensures that users and administrators have distinct access points and privileges. Regular users can only access their own account information, while administrators have the ability to view system-wide statistics and user data.

Main User Dashboard

Once a user logs in successfully, they are presented with the MainFrame, which serves as the user’s primary dashboard. This interface allows users to manage their personal information, deposit funds, withdraw funds, and transfer funds to other users. The MainFrame is divided into several panels, each with a specific purpose.

Key Features of MainFrame

1. User Information Management

Users can view and edit their personal information, such as:

Real Name

Gender

Login Name (card number)

Balance

Password

Birthday

Users can make changes to their real name and password by clicking the "Update Info" button. This action makes the input fields editable. Once the user clicks "Confirm Update", the UserDao.update() method is called to update the corresponding records in the User Table.

1. Deposit Funds

Users can deposit funds into their account via the Deposit Panel. This process involves:

Entering the amount to be deposited.

Clicking the "Confirm Deposit" button.

The system calls UserDao.cunKuan(amount, userId), which updates the user's balance in the User Table.

The system records the transaction in the History Table using the HistoryDao.savemoney(userId, amount) method.

1. Withdraw Funds

Users can withdraw funds via the Withdrawal Panel. This process involves:

Entering the amount to be withdrawn.

Clicking the "Confirm Withdrawal" button.

The system validates the user's balance to ensure they have sufficient funds.

The user's balance is updated, and the transaction is recorded in the History Table.

If the withdrawal amount exceeds the user's available balance, an error message is displayed.

1. Transfer Funds

Users can transfer funds to other users via the Transfer Panel. This process involves:

Entering the recipient's card number.

Entering the amount to be transferred.

Clicking the "Confirm Transfer" button.

If the transfer is successful, the sender’s balance is updated, and the History Table records the transaction as a type 3 (transfer) transaction.

1. Transaction History

Every deposit, withdrawal, and transfer is recorded in the History Table. Each entry includes:

Transaction Type (Deposit, Withdrawal, or Transfer)

Amount

Timestamp

This history allows users and administrators to track all financial activities in the system.

Admin Dashboard

The AdminMainFrame is designed for system administrators. Administrators have enhanced privileges that allow them to view system statistics and access user data. The dashboard displays the following:

View User Information: Admins can search for users by card number and view their account information.

System Statistics: The system shows the total number of users, total deposits, withdrawals, and transfers.

Key Features of AdminMainFrame

1. View User Info

Admins can search for users by entering a card number.

The system queries the User Table to display user information such as name, balance, and birthday.

If the card number is invalid, the system displays an error message.

1. System Financial Statistics

Admins can view system-wide statistics, such as:

Total number of users

Total deposits made

Total withdrawals made

Total amount of funds transferred

These statistics provide administrators with an overview of the system's financial performance.

**Conclusion**

The User Management and Financial Transaction System is a comprehensive, secure, and user-friendly platform designed to streamline personal account management and financial transactions. By integrating Java Swing for a graphical user interface (GUI) and MySQL for data storage, the system provides a modular, scalable, and reliable solution for users and administrators alike.

The system supports two distinct roles -- users and administrators -- each with different levels of access and functionality. Regular users can register an account, log in, view and update personal information, and conduct financial transactions such as deposits, withdrawals, and transfers. The system maintains a detailed transaction history to provide users with a transparent record of their financial activity. On the other hand, administrators can access the Admin Dashboard, where they can view system-wide financial statistics, track user activity, and oversee the overall performance of the system.

One of the system’s most significant achievements is its modular design. By separating core functions into four main interfaces -- RegisterFrame, LoginFrame, MainFrame, and AdminMainFrame -- the system achieves clear separation of concerns. This modular approach ensures that updates, bug fixes, or new features can be implemented with minimal impact on existing functionality.

The database design plays a crucial role in the system's overall performance and data integrity. The four primary database tables -- Admin, User, Dept, and History -- are structured to support seamless data retrieval, storage, and tracking. The History Table, in particular, serves as a robust transaction log, recording every deposit, withdrawal, and transfer made by users. This ensures that users and administrators have complete visibility into all system activities, which is critical for financial transparency and accountability.

The implementation of secure login and authentication further enhances the system's security. By validating usernames and passwords through UserDao and AdminDao, the system ensures that only legitimate users and administrators can access their respective dashboards. Error messages and validation checks are in place to guide users when login credentials are incorrect or incomplete.

From a user experience perspective, the system is simple and intuitive. The interface is easy to navigate, with clearly labeled buttons and input fields for deposits, withdrawals, and transfers. Error messages and confirmation messages ensure that users are guided through each transaction step-by-step, reducing the likelihood of mistakes.

In summary, the User Management and Financial Transaction System achieves its objectives by providing a secure, efficient, and user-friendly platform for financial management. With its clear workflows, modular design, and secure authentication system, it meets the needs of users and administrators alike. Users benefit from transparency, control, and the ability to manage their financial activities with ease, while administrators have access to powerful tools for system oversight.

The key strengths of this system include:

Security: Role-based access control, login validation, and secure transaction processing.

Transparency: A comprehensive transaction history for each user, viewable by both users and administrators.

Modularity: The system is divided into well-structured components, allowing for seamless maintenance and upgrades.

User Experience: The user-friendly GUI provides a clear, simple, and intuitive experience.

Data Integrity: The use of a relational MySQL database ensures that all data is stored consistently, with referential integrity enforced across all tables.

This system can be further enhanced by incorporating data encryption for sensitive information like passwords, multi-factor authentication (MFA) for improved login security, and role-based access control (RBAC) for fine-grained administrative privileges. Additionally, a report generation feature could be added to allow administrators to export system statistics for financial reporting.

In conclusion, the \*\*User Management and Financial Transaction System\*\* provides a solid foundation for personal financial management and administrative oversight. It is \*\*scalable, secure, and efficient\*\*, making it suitable for both small-scale financial institutions and larger organizations. With its modular architecture and robust database design, the system is well-positioned for future enhancements, ensuring continued relevance and adaptability in an ever-evolving financial landscape.