UNIVERSITY OF BRAHMANBARIA

PROJECT REPORT

ON

BANKING
SYSTEM

SUBMITTED TO

SYED TAWHID IMAM

SENIOR LECTURER

UNIVERSITY OF BRAHMANBARIA

SUBMITTED BY

NAIMA AKTER MITHILA

JAID HASSAN

M.R.AMIT HASAN

1. Introduction

1.1 Project Overview

The Bank of UOB Banking Management System is a comprehensive desktop application developed using Java Swing for the frontend and MySQL for backend database management. The system provides secure banking operations for both customers and administrators with a user-friendly graphical interface.

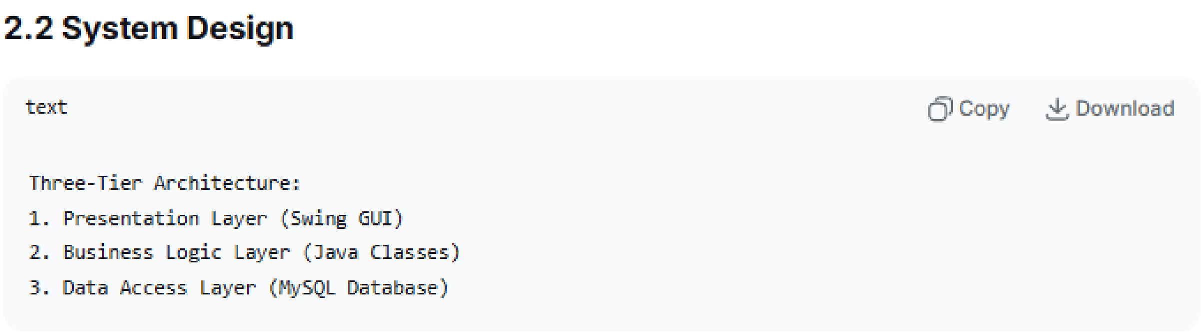
1.2 Objectives

- Develop a secure banking application with user authentication
- Implement core banking operations (deposit, withdrawal, transfer)
- Provide administrative controls for user management
- Ensure data integrity and transaction security
- Create an intuitive user interface for seamless banking experience

2. System Architecture

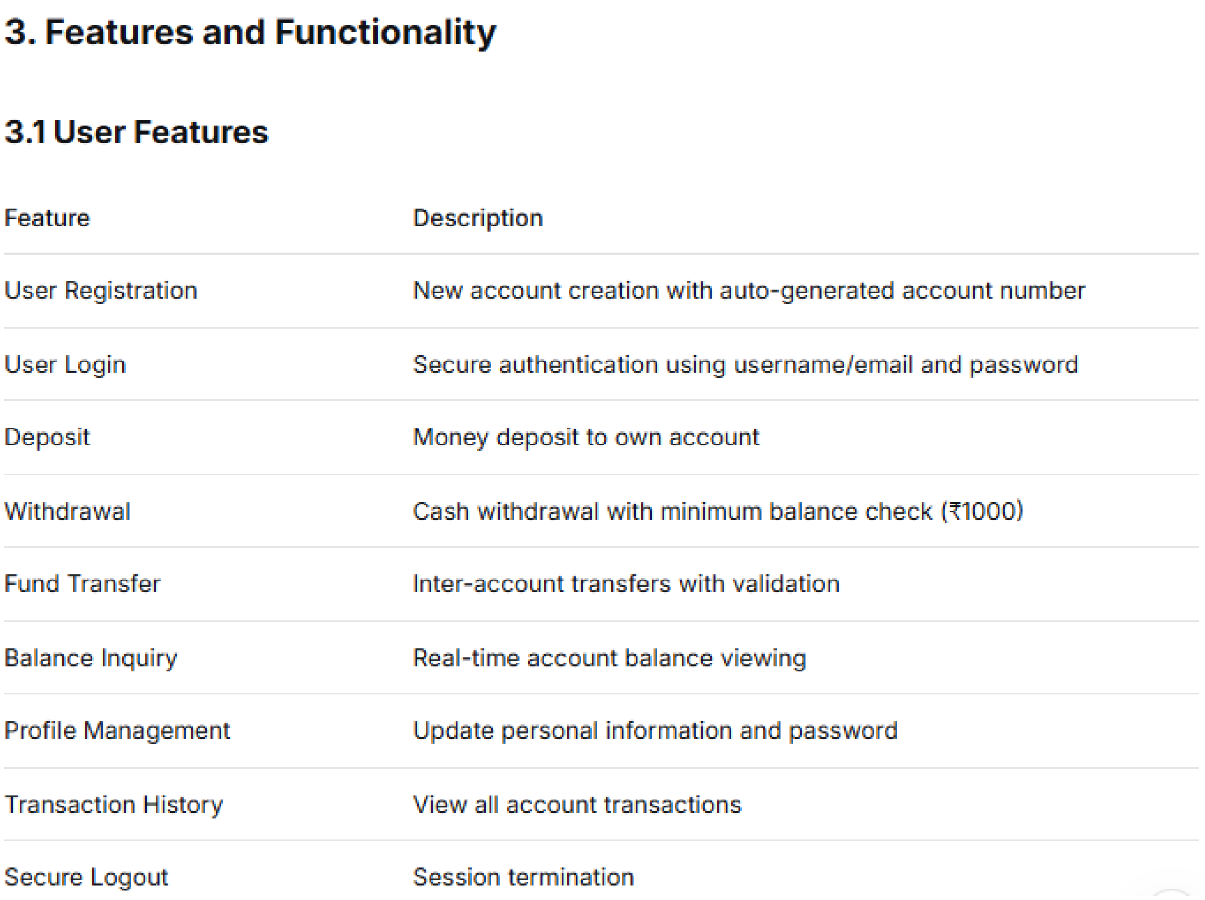
2.1 Technology Stack

Component	Technology Used
Frontend	Java Swing (GUI)
Backend	Java (Business Logic)
Database	MySQL 8.0+
JDBC Driver	MySQL Connector/J
IDE	VS Code



2.2 System Design

```text
Three-Tier Architecture:
1. Presentation Layer (Swing GUI)
2. Business Logic Layer (Java Classes)
3. Data Access Layer (MySQL Database)
```



3. Features and Functionality

3.1 User Features

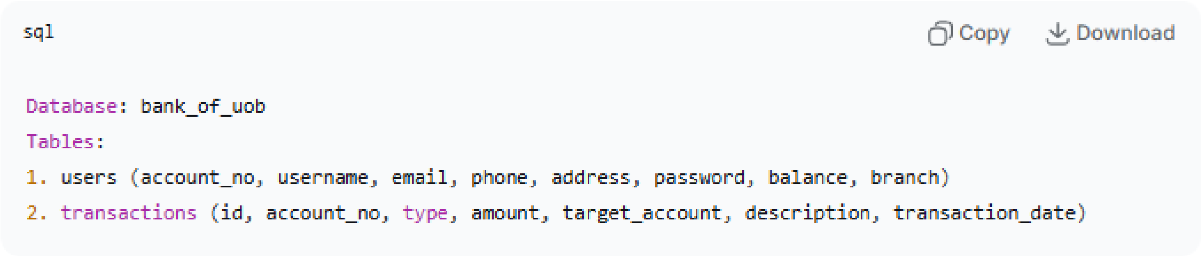
Feature	Description
User Registration	New account creation with auto-generated account number
User Login	Secure authentication using username/email and password
Deposit	Money deposit to own account
Withdrawal	Cash withdrawal with minimum balance check (₹1000)
Fund Transfer	Inter-account transfers with validation
Balance Inquiry	Real-time account balance viewing
Profile Management	Update personal information and password
Transaction History	View all account transactions
Secure Logout	Session termination

3.2 Admin Features

Feature	Description
Admin Login	Dedicated admin credentials (admin/admin123)
User Management	View and remove user accounts
Transaction Monitoring	View all system transactions
System Oversight	Complete banking system supervision

4. Database Design

4.1 Database Schema



```sql
Database: bank_of_uob
Tables:
1. users (account_no, username, email, phone, address, password, balance, branch)
2. transactions (id, account_no, type, amount, target_account, description, transaction_date)
```

4.2 Key Constraints

- Minimum account balance: ₹1000
- Unique username and email constraints
- Auto-generated account numbers (UOB000001 format)
- Transaction history maintenance
- Referential integrity between tables

5. Class Structure and Design

5.1 Core Classes

Class Name	Purpose
`LoginFrame`	User authentication interface
`RegistrationFrame`	New user registration
`UserDashboard`	Main user operations dashboard
`AdminDashboard`	Administrative controls
`BankDAO`	Database operations handler
`DBConnection`	Database connectivity management
`User`	User data model

5.2 Transaction Classes

- `DepositFrame` - Handle money deposits
- `WithdrawFrame` - Process withdrawals
- `TransferFrame` - Manage fund transfers
- `BalanceFrame` - Display account information
- `EditProfileFrame` - User profile updates
- `TransactionHistoryFrame` - Transaction records

5.3 Admin Classes

- `UserManagementFrame` - User account management
- `AllTransactionsFrame` - System-wide transaction view

6. Security Implementation

6.1 Authentication System

- Dual-layer authentication (username/email + password)
- Admin-specific login credentials
- Session management
- Input validation and sanitization

6.2 Data Protection

- Password validation rules
- Transaction authorization checks
- Minimum balance enforcement
- SQL injection prevention using PreparedStatements

---

7. User Interface Design

7.1 Design Principles

- **Consistency:** Uniform color scheme and layout
- **Usability**: Intuitive navigation and clear labels
- **Accessibility**: Responsive design and error messages
- **Professionalism:** Banking-appropriate visual elements

7.2 Color Scheme

- Primary: Dark Blue (#00008B) - Trust and professionalism
- Secondary: Green (#006400) - Success and confirmation
- Accent: Steel Blue (#4682B4) - Action buttons
- Warning: Fire Brick (#B22222) - Caution and errors

---

8. Implementation Details

8.1 Key Algorithms

1. **Account Number Generation:** Sequential UOB-prefixed numbers
2. **Balance Validation:** Minimum ₹1000 balance check
3. **Transaction Processing:** Atomic operations for fund transfers
4. **User Authentication:** Credential verification against database

8.2 Database Operations

- CRUD operations for user management
- Transaction logging and history
- Balance updates with integrity checks
- Admin reporting and monitoring

9. Testing and Validation

9.1 Test Scenarios

Test Case	Expected Result
User Registration	Account creation with unique validation
Login Authentication	Successful access with valid credentials
Deposit Operation	Balance increase with transaction record
Withdrawal Operation	Balance decrease with minimum balance check
Fund Transfer	Successful transfer between accounts
Admin Functions	Proper user and transaction management

9.2 Validation Rules

- Empty field validation
- Password confirmation
- Numeric amount validation
- Account existence checks
- Balance sufficiency verification

---

10. Challenges and Solutions

10.1 Technical Challenges

Challenge	Solution
Database Connection	Implemented connection pooling and error handling
Transaction Integrity	Used database transactions for fund transfers
UI Responsiveness	Employed SwingUtilities.invokeLater()
Data Validation	Comprehensive input validation at multiple levels

10.2 Business Logic Challenges

- Minimum balance enforcement
- Concurrent access handling
- Transaction rollback mechanisms
- User session management

---

11. Future Enhancements

11.1 Planned Improvements

- Password encryption and hashing
- Email verification system
- Mobile application integration
- Advanced reporting and analytics
- Multi-branch support
- Interest calculation system
- Loan management module
- Cheque processing system

11.2 Security Upgrades

- Two-factor authentication
- Session timeout implementation
- Audit trail enhancement
- Role-based access control

---

12. Installation and Setup Guide

12.1 Prerequisites

- Java JDK 8 or higher
- MySQL Server 8.0+
- MySQL Connector/J
- VS Code with Java extensions

12.2 Setup Steps

1. Create MySQL database using provided SQL script
2. Update database credentials in `DBConnection.java`
3. Compile all Java files: `javac *.java`
4. Run application: `java LoginFrame`

12.3 Default Credentials

- **Admin**: username: `admin`, password: `admin123`
- **Initial User Balance**: ₹1000.00

13. Conclusion

13.1 Project Achievements

✅ Successfully developed a fully functional banking system
✅ Implemented secure user authentication and authorization
✅ Created intuitive GUI for both users and administrators
✅ Established robust database architecture
✅ Ensured transaction integrity and data consistency

13.2 Business Impact

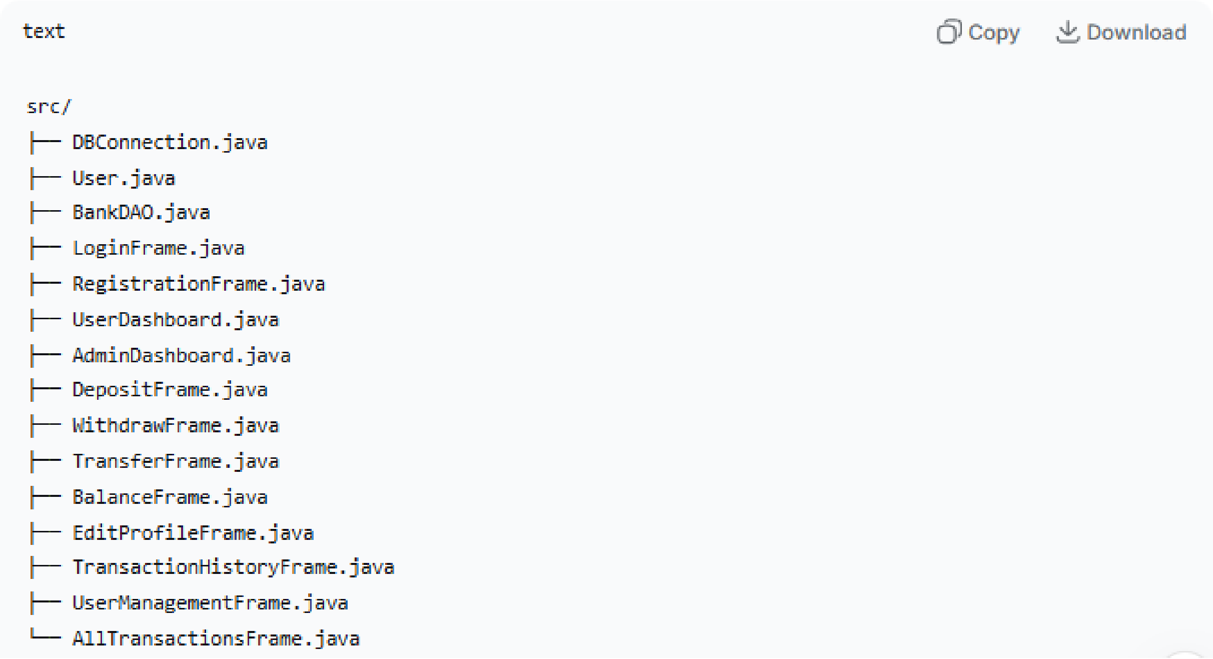
- **For Customers**: Convenient banking operations with 24/7 access
- **For Bank**: Efficient user management and transaction monitoring
- **Operational**: Reduced manual processing and human errors

13.3 Learning Outcomes

- Enhanced Java Swing GUI development skills
- Improved database design and management capabilities
- Strengthened understanding of banking system workflows
- Gained experience in full-stack application development

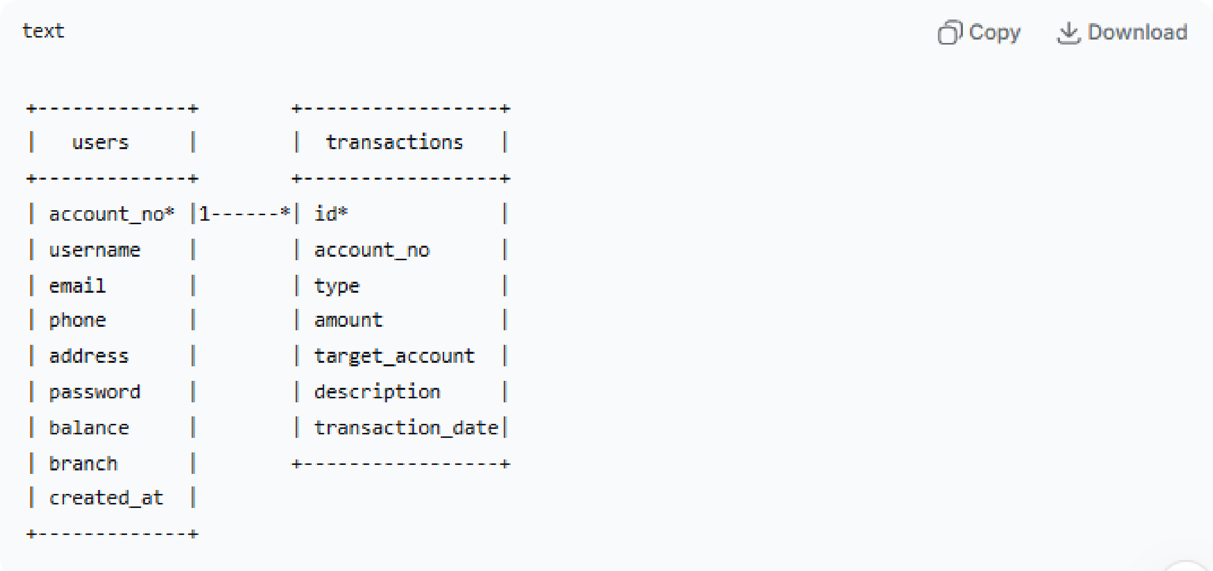
14. Appendix

14.1 File Structure



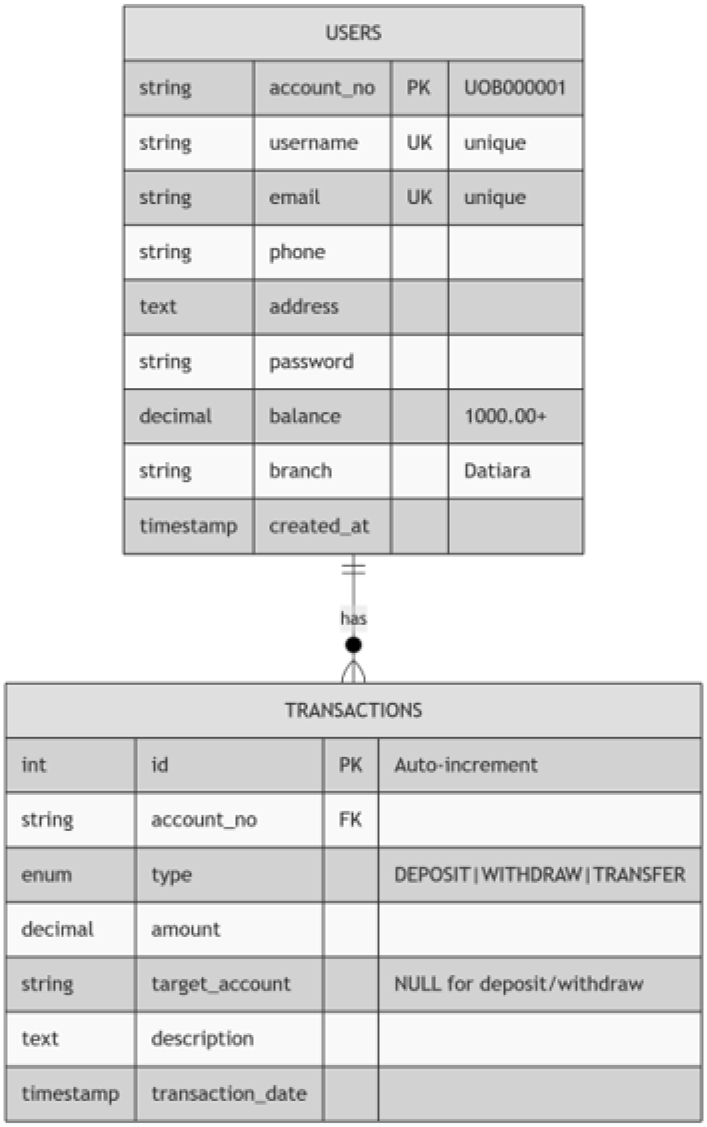
```
src/
├── DBConnection.java
├── User.java
├── BankDAO.java
├── LoginFrame.java
├── RegistrationFrame.java
├── UserDashboard.java
├── AdminDashboard.java
├── DepositFrame.java
├── WithdrawFrame.java
├── TransferFrame.java
├── BalanceFrame.java
├── EditProfileFrame.java
├── TransactionHistoryFrame.java
├── UserManagementFrame.java
└── AllTransactionsFrame.java
```

14.2 Database Schema Diagram



```
+-------------+        +-----------------+
|   users     |        |  transactions   |
+-------------+        +-----------------+
| account_no* |1------*| id*             |
| username    |        | account_no      |
| email       |        | type            |
| phone       |        | amount          |
| address     |        | target_account  |
| password    |        | description     |
| balance     |        | transaction_date|
| branch      |        +-----------------+
| created_at  |
+-------------+
```

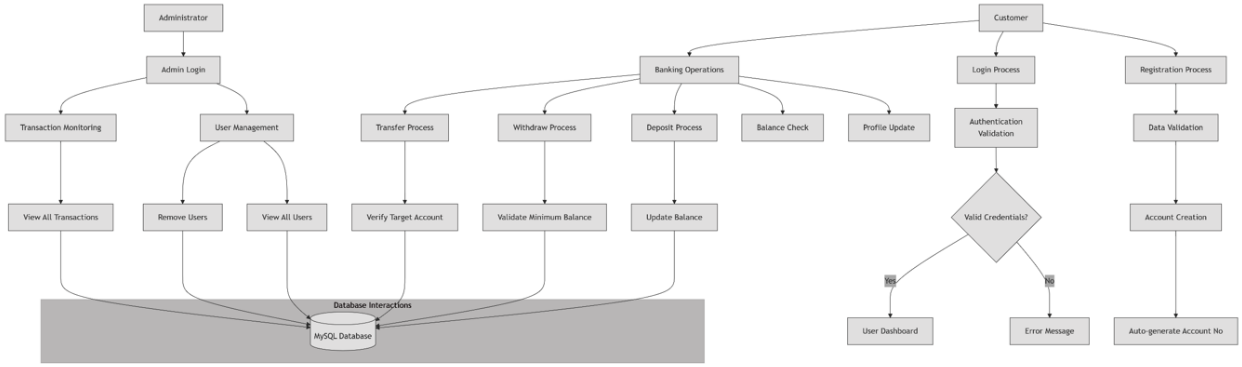
14.3 ER Diagram



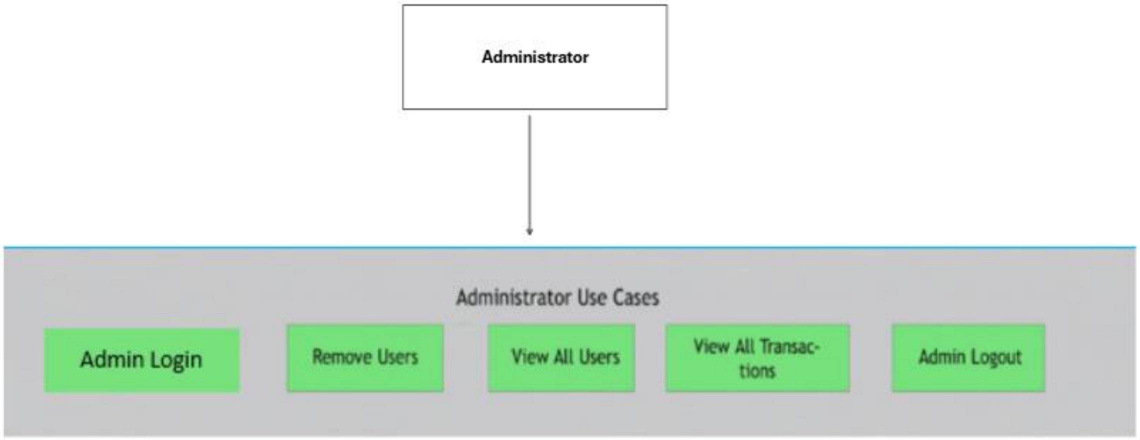
USERS			
string	account_no	PK	UOB000001
string	username	UK	unique
string	email	UK	unique
string	phone		
text	address		
string	password		
decimal	balance		1000.00+
string	branch		Datiara
timestamp	created_at		

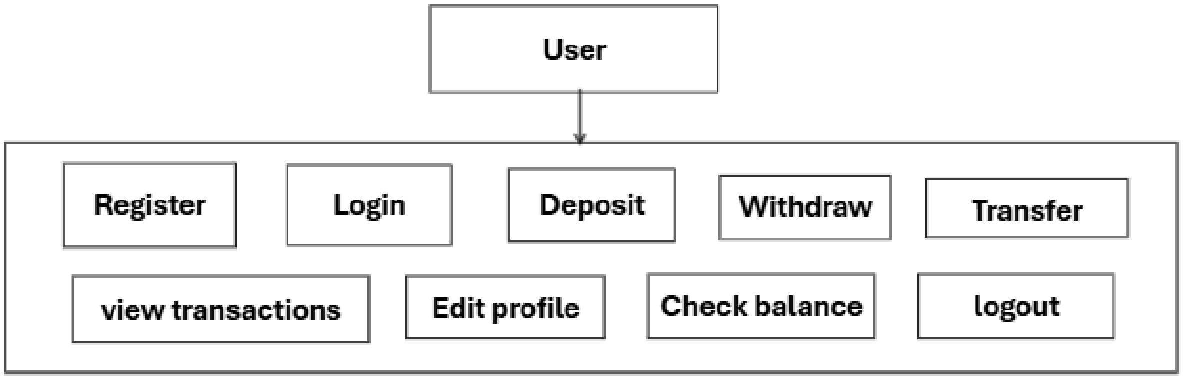
TRANSACTIONS			
int	id	PK	Auto-increment
string	account_no	FK	
enum	type		DEPOSIT|WITHDRAW|TRANSFER
decimal	amount		
string	target_account		NULL for deposit/withdraw
text	description		
timestamp	transaction_date		

14.4 Data Flow Diagram



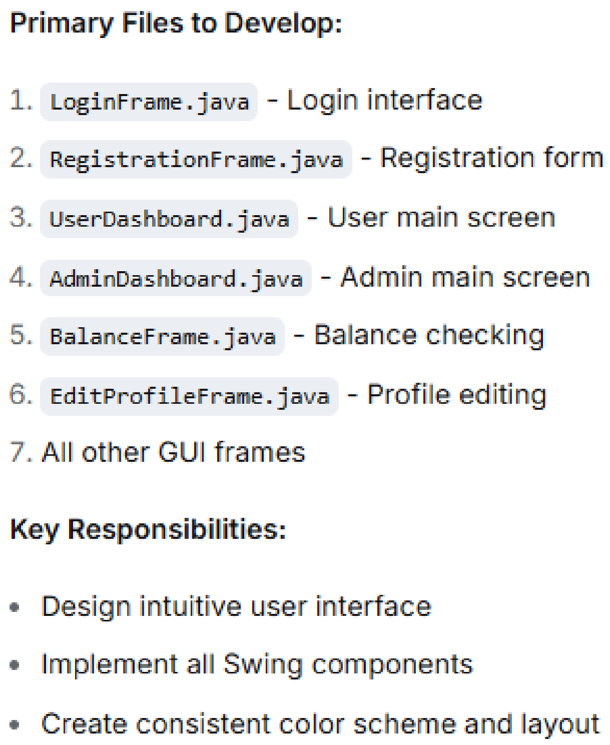
14.5 Use Case Diagram





15.Team Roles

Jaid Hassan: Frontend Developer & UI Designer



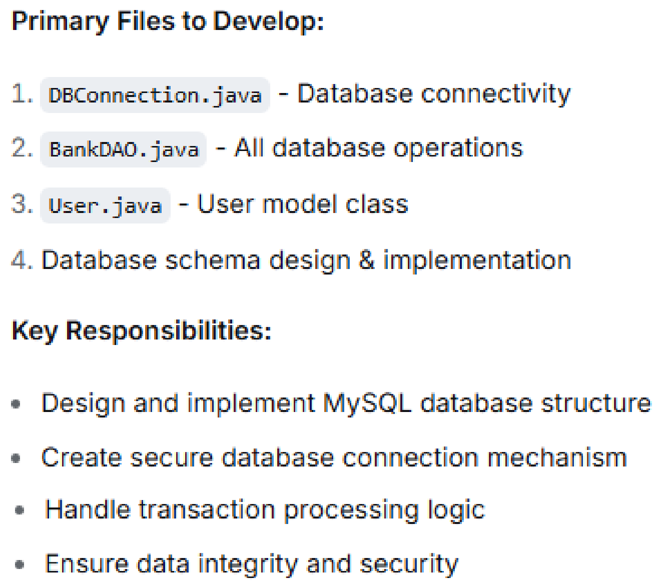
Primary Files to Develop:

1. `LoginFrame.java` - Login interface
2. `RegistrationFrame.java` - Registration form
3. `UserDashboard.java` - User main screen
4. `AdminDashboard.java` - Admin main screen
5. `BalanceFrame.java` - Balance checking
6. `EditProfileFrame.java` - Profile editing
7. All other GUI frames

Key Responsibilities:

- Design intuitive user interface
- Implement all Swing components
- Create consistent color scheme and layout

M.R. Amit Hassan: Team Lead & Backend Developer



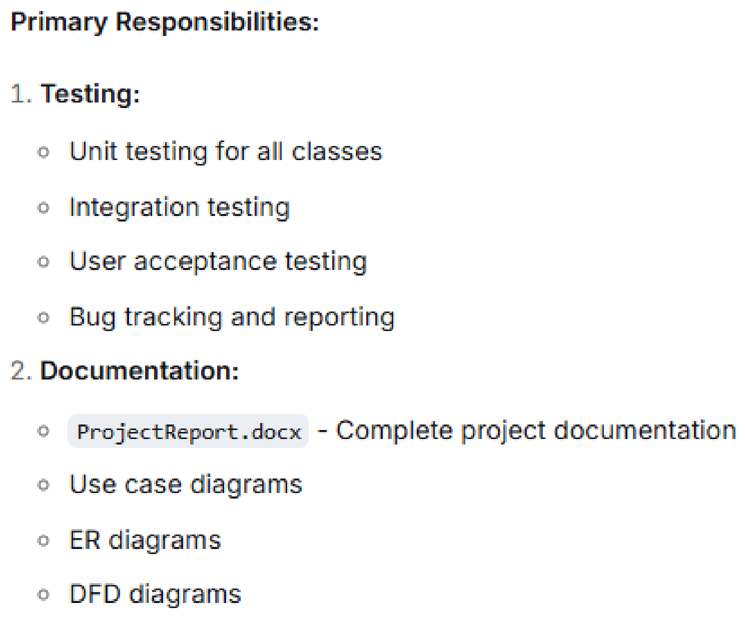
Primary Files to Develop:

1. `DBConnection.java` - Database connectivity
2. `BankDAO.java` - All database operations
3. `User.java` - User model class
4. Database schema design & implementation

Key Responsibilities:

- Design and implement MySQL database structure
- Create secure database connection mechanism
- Handle transaction processing logic
- Ensure data integrity and security

Naima Akter Mithila: Quality Analyst & Documentation



Primary Responsibilities:

1. **Testing:**
   - Unit testing for all classes
   - Integration testing
   - User acceptance testing
   - Bug tracking and reporting
2. **Documentation:**
   - `ProjectReport.docx` - Complete project documentation
   - Use case diagrams
   - ER diagrams
   - DFD diagrams

16. References

https://venngage.com/templates/diagrams/green-banking-system-er-diagram-169efd12-812f-43fa-b123-f195b4aa3d5a

https://online.visual-paradigm.com/diagrams/templates/data-flow-diagram/data-flow-diagram-example-bank-system/;VPSESSIONID=FC1ABEDBCB2E1D6D7DB0CD0F009876F3

https://venngage.com/templates/diagrams/light-banking-business-use-case-ac674046-d4af-43e1-a864-55e48ad61a3b