Banking APP

With JDBC

CONTENT

Java

- ❖ Java is a high-level, class-based, object-oriented programming language. It is a compiled language, which means that the Java source code is converted into bytecode that can be executed by the Java Virtual Machine (JVM).
- ❖ Java is a popular programming language for a variety of applications, including web development, mobile development, and enterprise software development.
- ❖ Some of the Key features of JAVA are:
  - ➢ Object Oriented
  - ➢ Platform Independent
  - ➢ Secure
  - ➢ Robust

JDBC

- JDBC allows JAVA programs to connect to DB’s
- Provides Connectivity and data access across relational databases from different vendors
- Classes and Interfaces allow users to access the database in a standard way.
- JDBC makes it possible to do:
  - Establish a connection with a database
  - Send SQL statements
  - Process the results

SCOPE OF PROJECT

- The Bank App aims to provide a comprehensive solution for handling and processing financial transactions within a banking environment.
- The Scope of the Project includes the key components :
  - Transaction Processing
    - Facilitate the processing of Deposit(D) and Withdrawal(W) transactions.
    - Validate transactions against predefined rules.
  - Transaction Classification
    - Classify transactions either “VALID” or “InValid”.
  - DataBase Integration
    - Interact with an Oracle database to store and retrieve transactions data.
    - Maintain separate table for Valid and Invalid transactions for auditing purposes.
  - User Interface
    - Display a summary of Valid and Invalid Transactions for easy monitoring.

Project CONFIGs

- ❖ Database Connection Configuration
- ❖ Connection String i.e JDBC URL
- ❖ Project Specific Configurations

DEPENDENCIES

- **Ojdbc14.jar**
  - It serves as the oracle JDBC driver for the java applications and facilitates java database connectivity between our application and Oracle databases.
  - Ensure Ojdbc14.jar is included in the project's classpath.

- **Oracle Driver**
  - It acts as a bridge between Java applications and Oracle database.
  - Specify the JDBC URL in the DriverManager.getconnection(); along the Oracle DB details.

- **Tnsnames.ora**
  - It is a configuration file defining database connection descriptors i.e port no and service name.
  - Ensure Tnsnames.ora is correctly located in the "network/admin" directory.

DEPENDENCIES

- ❖ Ojdbc14.jar, Oracle driver and Tnsnames.ora are pivotal for our project
- ❖ For uninterrupted connectivity carefully include and configure the given files and drivers.

COMPONENTS

DisplayTables Class

- This class is responsible for displaying transactions from a specified table (ValidTrans or InvalidTrans).
- Depends on a valid database connection (Connection con)
- Used by BankTransaction class to display Valid and Invalid transaction tables.

BankTransaction Class

- This class is the central class for handling bank transactions.
- It contains the main logic for processing transactions, updating balances, and determining validity.
- Uses SQL queries to retrieve, update, and insert data into the database.
- Utilizes DisplayTables class to display transaction tables.

Test Class

- The Test class contains the main method, serving as the entry point for the application.
- Invokes BankTransaction.processTransaction() to simulate the processing of bank transactions.
- Used for testing purposes.

Utility Methods

DisplayTable()

- DisplayTable(Connection con, String table_name) displays transactions from the specified table.
- Displays transactions from the specified table.
- Used by BankTransaction class to display Valid and Invalid transaction tables.

processTransaction()

- Processes bank transactions and updates the database accordingly.
- Depends on the Oracle database and JDBC for database operations
- Manages to decide whether a transaction is Valid or InValid.

main(String[] args)

- The main method, serves as the entry point for the application.
- main(String[] args) invokes the processTransaction() method to simulate the processing of bank transactions.

Method CHAINING

- **con.prepareStatement(selectQuery):**
  - con is a Connection object.
  - prepareStatement(selectQuery) is called on the Connection object, returning a PreparedStatement.
  - This PreparedStatement is assigned to the variable selectstatement.
- **selectstatement.executeQuery():**
  - selectstatement is a PreparedStatement object.
  - executeQuery() is called on the PreparedStatement, returning a ResultSet.
  - This ResultSet is assigned to the variable rs.
- **updatestatement.setDouble(1, mnewBal)**
  - Chains multiple setter methods on the updatestatement (PreparedStatement) object.

REQUIREMENT ID	REQUIREMENT CATEGORY	REQUIREMENT TYPE	PRIORITY	HIERARCHY	REF
R001	FUNCTIONAL	STATED	HIGH		

REQUIREMENT DESCRIPTION	ESTABLISHING DATABASE CONNECTION
SCOPE	• Define the methods and classes responsible for establishing a connection to the Oracle Database and specify the connection parameters, including URL, username, and password. • Implement mechanisms for handling exceptions related to database connection issues.
REQUIREMENT METHODOLOGICAL DETAILS	• The project requires the Oracle JDBC driver (ojdbc14.jar) to be available in the classpath. • Utilize the DriverManager class for loading the JDBC driver and establishing a connection. • DriverManager.*getConnection*("jdbc:oracle:thin:@localhost:1521:XE", "scott", "tiger")

REQUIREMENT ID	REQUIREMENT CATEGORY	REQUIREMENT TYPE	PRIORITY	HIERARCHY	REF
R002	FUNCTIONAL	STATED	HIGH		

REQUIREMENT DESCRIPTION	USING THE DATA FROM BANKTRANS TABLE IN JAVA CODE AND BUILDING LOGIC
SCOPE	• Data Retrieval: • Data Processing Logic • Transaction Validation • Update Logic and Logging
REQUIREMENT METHODOLOGICAL DETAILS	• Design methods to retrieve data from the BankTrans table using SQL queries. • Specify the columns to be retrieved, such as TransID, AcctNo, OldBal, TransType, and TransAmt. • Define SQL queries to retrieve data from the BankTrans table. • Include necessary WHERE clauses to filter data, if needed. • Use PreparedStatement to execute SQL queries for retrieving data from the BankTrans table.

REQUIREMENT ID	REQUIREMENT CATEGORY	REQUIREMENT TYPE	PRIORITY	HIERARCHY	REF
R003	FUNCTIONAL	STATED	HIGH		

REQUIREMENT DESCRIPTION	UPDATING BANKTRANS TABLE WITH NEWBAL AND TRANSSTAT
SCOPE	• Modify the existing records in the BankTrans table to update the NewBal and TransStat fields. • Log the details of the updates, including the transaction ID, old balance, new balance, and transaction status.
REQUIREMENT METHODOLOGICAL DETAILS	• Define the logic for determining the transaction status based on the calculated new balance. • Categorize transactions as "Valid" or "Invalid" based on the new balance. • Specify the SQL UPDATE statements for modifying NewBal and TransStat in the BankTrans table. • Include placeholders for the new balance, transaction status, and transaction ID.

REQUIREMENT ID	REQUIREMENT CATEGORY	REQUIREMENT TYPE	PRIORITY	HIERARCHY	REF
R004	FUNCTIONAL	STATED	HIGH		

REQUIREMENT DESCRIPTION	INSERT NEW VALUES INTO VALID AND INVALID TRANS TABLE AND DISPLAYING THEM
SCOPE	• Insert new records into the ValidTrans and InvalidTrans tables. • Log the details of the inserted transactions. • Retrieve and display data from the ValidTrans and InvalidTrans tables.
REQUIREMENT METHODOLOGICAL DETAILS	• Specify SQL INSERT statements to add new records to both the ValidTrans and InvalidTrans tables • Specify the columns to be retrieved from ValidTrans and InValidTrans Tables. • Use PreparedStatement to execute SQL queries for retrieving data from the BankTrans table • Specify the format and content of log entries, including transaction ID, transaction type, amount, and validity

OPERATING ENVIRONMENT

01 JAVA RUNTIME ENVIRONMENT (JRE)

02 ORACLE SERVER AND DRIVER

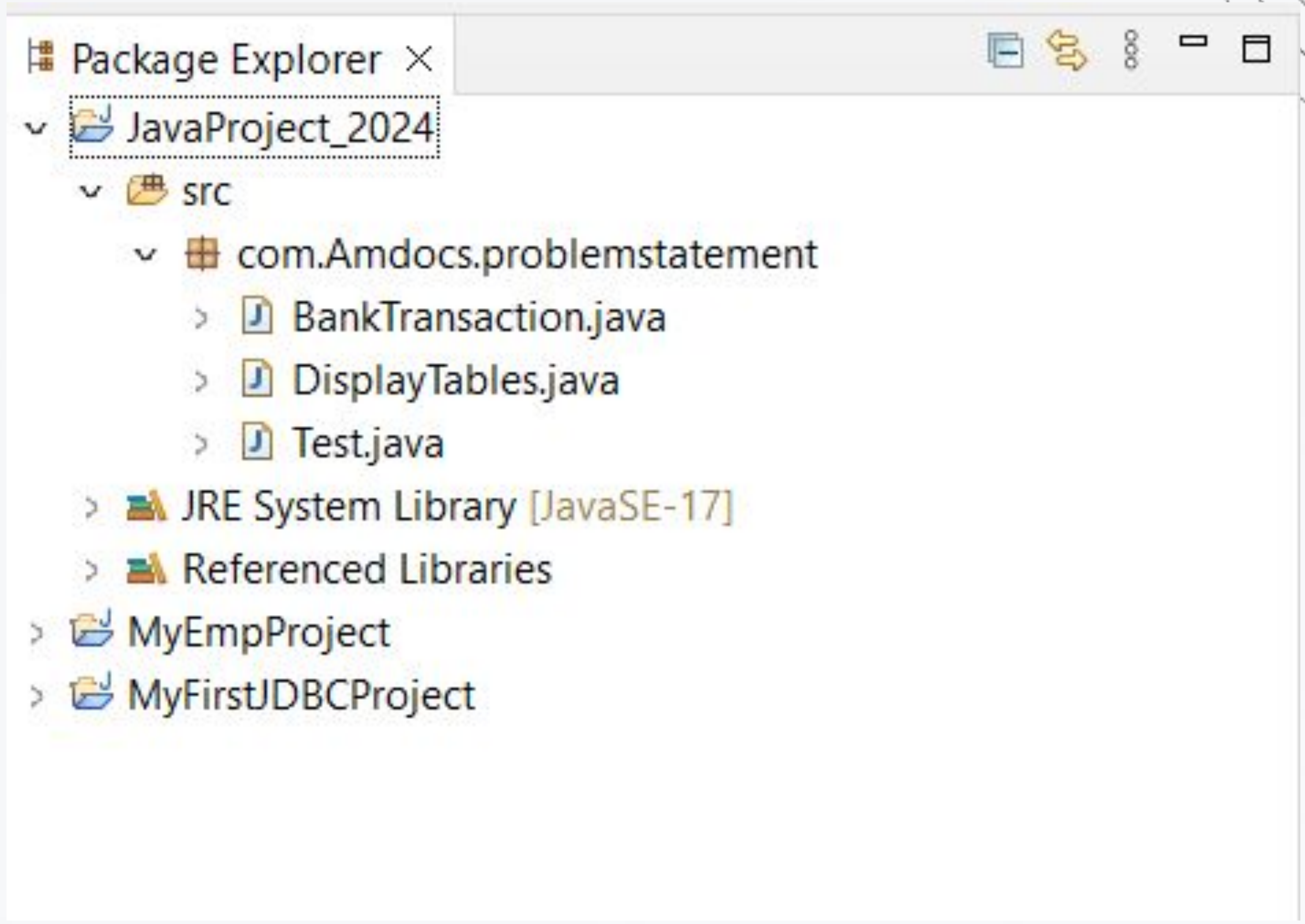
03 DATABASE SCHEMA AND TABLES

04 JDBC AND SQL

Hierarchy of Project Artifacts

- This hierarchy outlines the structure of the project, highlighting the main classes, their relationships, and the organization of source code.

- It's designed to provide clarity, modularity, and maintainability to the overall project structure.



```
Package Explorer
v JavaProject_2024
    v src
        v com.Amdocs.problemstatement
            > BankTransaction.java
            > DisplayTables.java
            > Test.java
    > JRE System Library [JavaSE-17]
    > Referenced Libraries
> MyEmpProject
> MyFirstJDBCProject
```

THANK YOU

By : **JAGJEEVAN SINGH SONI**