Given the complex nature of the application with different roles, database interactions, and web-based functionality, consolidating all parts of the User Access Management System into a single Java program would not follow best practices in web development. Typically, each servlet, JSP page, and database configuration would be modular to maintain clarity and functionality.

However, I can provide a simplified, consolidated version for demonstration purposes that simulates each component in a single Java class with console-based interaction (rather than a full web-based system). This approach will enable you to understand the workflow and logic of a User Access Management System.

Here is a basic, single-file Java program that mimics some of the key functionalities (registration, login, role-based access control, and access request management) in a console application.

Here Java code:

import java.sql.\*;

import java.util.Scanner;

public class UserAccessManagement {

// Establish connection to the PostgreSQL database

public static Connection getConnection() throws SQLException {

String url = "jdbc:postgresql://localhost:5432/dbname";

String user = "username";

String password = "password";

return DriverManager.getConnection(url, user, password);

}

// Main program loop

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.println("Welcome to the User Access Management System");

System.out.println("1. Sign Up");

System.out.println("2. Log In");

System.out.println("3. Exit");

int choice = scanner.nextInt();

scanner.nextLine(); // consume newline

switch (choice) {

case 1:

signUp(scanner);

break;

case 2:

logIn(scanner);

break;

case 3:

System.out.println("Exiting the system.");

return;

default:

System.out.println("Invalid choice. Try again.");

}

}

}

// Sign-up method

public static void signUp(Scanner scanner) {

System.out.print("Enter username: ");

String username = scanner.nextLine();

System.out.print("Enter password: ");

String password = scanner.nextLine();

try (Connection conn = getConnection()) {

String query = "INSERT INTO users (username, password, role) VALUES (?, ?, 'Employee')";

PreparedStatement stmt = conn.prepareStatement(query);

stmt.setString(1, username);

stmt.setString(2, password);

stmt.executeUpdate();

System.out.println("Registration successful! Please log in.");

} catch (SQLException e) {

System.out.println("Error during registration: " + e.getMessage());

}

}

// Log-in method

public static void logIn(Scanner scanner) {

System.out.print("Enter username: ");

String username = scanner.nextLine();

System.out.print("Enter password: ");

String password = scanner.nextLine();

try (Connection conn = getConnection()) {

String query = "SELECT id, role FROM users WHERE username = ? AND password = ?";

PreparedStatement stmt = conn.prepareStatement(query);

stmt.setString(1, username);

stmt.setString(2, password);

ResultSet rs = stmt.executeQuery();

if (rs.next()) {

int userId = rs.getInt("id");

String role = rs.getString("role");

System.out.println("Login successful! Welcome, " + role + ".");

switch (role) {

case "Employee":

requestAccess(scanner, userId);

break;

case "Manager":

approveRequests(scanner);

break;

case "Admin":

createSoftware(scanner);

break;

default:

System.out.println("Role not recognized.");

}

} else {

System.out.println("Invalid username or password.");

}

} catch (SQLException e) {

System.out.println("Error during login: " + e.getMessage());

}

}

// Access request method for Employees

public static void requestAccess(Scanner scanner, int userId) {

System.out.println("Enter the software name to request access:");

String softwareName = scanner.nextLine();

System.out.println("Enter access type (Read, Write, Admin):");

String accessType = scanner.nextLine();

System.out.println("Enter reason for access request:");

String reason = scanner.nextLine();

try (Connection conn = getConnection()) {

String query = "INSERT INTO requests (user\_id, software\_id, access\_type, reason, status) " +

"VALUES (?, (SELECT id FROM software WHERE name = ?), ?, ?, 'Pending')";

PreparedStatement stmt = conn.prepareStatement(query);

stmt.setInt(1, userId);

stmt.setString(2, softwareName);

stmt.setString(3, accessType);

stmt.setString(4, reason);

stmt.executeUpdate();

System.out.println("Access request submitted and pending approval.");

} catch (SQLException e) {

System.out.println("Error during access request: " + e.getMessage());

}

}

// Approval method for Managers

public static void approveRequests(Scanner scanner) {

System.out.println("Pending requests:");

try (Connection conn = getConnection()) {

String query = "SELECT r.id, u.username, s.name, r.access\_type, r.reason " +

"FROM requests r " +

"JOIN users u ON r.user\_id = u.id " +

"JOIN software s ON r.software\_id = s.id " +

"WHERE r.status = 'Pending'";

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(query);

while (rs.next()) {

int requestId = rs.getInt("id");

String username = rs.getString("username");

String softwareName = rs.getString("name");

String accessType = rs.getString("access\_type");

String reason = rs.getString("reason");

System.out.println("Request ID: " + requestId + ", User: " + username +

", Software: " + softwareName + ", Access Type: " + accessType +

", Reason: " + reason);

System.out.println("Approve or Reject? (A/R):");

String decision = scanner.nextLine();

String status = decision.equalsIgnoreCase("A") ? "Approved" : "Rejected";

String updateQuery = "UPDATE requests SET status = ? WHERE id = ?";

PreparedStatement updateStmt = conn.prepareStatement(updateQuery);

updateStmt.setString(1, status);

updateStmt.setInt(2, requestId);

updateStmt.executeUpdate();

System.out.println("Request " + (status.equals("Approved") ? "approved" : "rejected") + ".");

}

} catch (SQLException e) {

System.out.println("Error during request approval: " + e.getMessage());

}

}

// Software creation method for Admins

public static void createSoftware(Scanner scanner) {

System.out.print("Enter software name: ");

String name = scanner.nextLine();

System.out.print("Enter software description: ");

String description = scanner.nextLine();

System.out.print("Enter access levels (e.g., Read, Write, Admin): ");

String accessLevels = scanner.nextLine();

try (Connection conn = getConnection()) {

String query = "INSERT INTO software (name, description, access\_levels) VALUES (?, ?, ?)";

PreparedStatement stmt = conn.prepareStatement(query);

stmt.setString(1, name);

stmt.setString(2, description);

stmt.setString(3, accessLevels);

stmt.executeUpdate();

System.out.println("Software added successfully.");

} catch (SQLException e) {

System.out.println("Error during software creation: " + e.getMessage());

}

}

}

Explanation of the Code:

1. **Database Connection:** Establishes a connection to PostgreSQL using JDBC.

2. **Main Menu:** Displays options for signing up, logging in, and exiting.

3. **Sign-Up Function:** Adds a new user with the default role "Employee" to the database.

4. **Log-In Function:** Authenticates a user and redirects based on their role.

5. **Employee Functionality:** Allows employees to request access to software.

6. **Manager Functionality:** Allows managers to approve or reject access requests.

7. **Admin Functionality:** Allows admins to create new software applications.