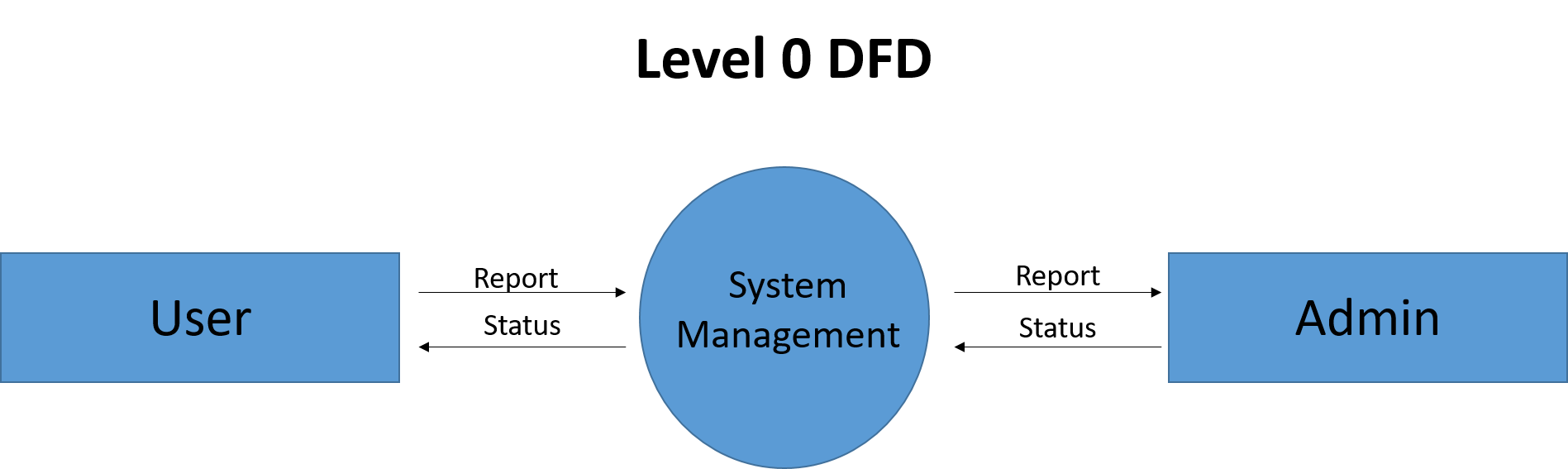
**Project name:** **System Management**

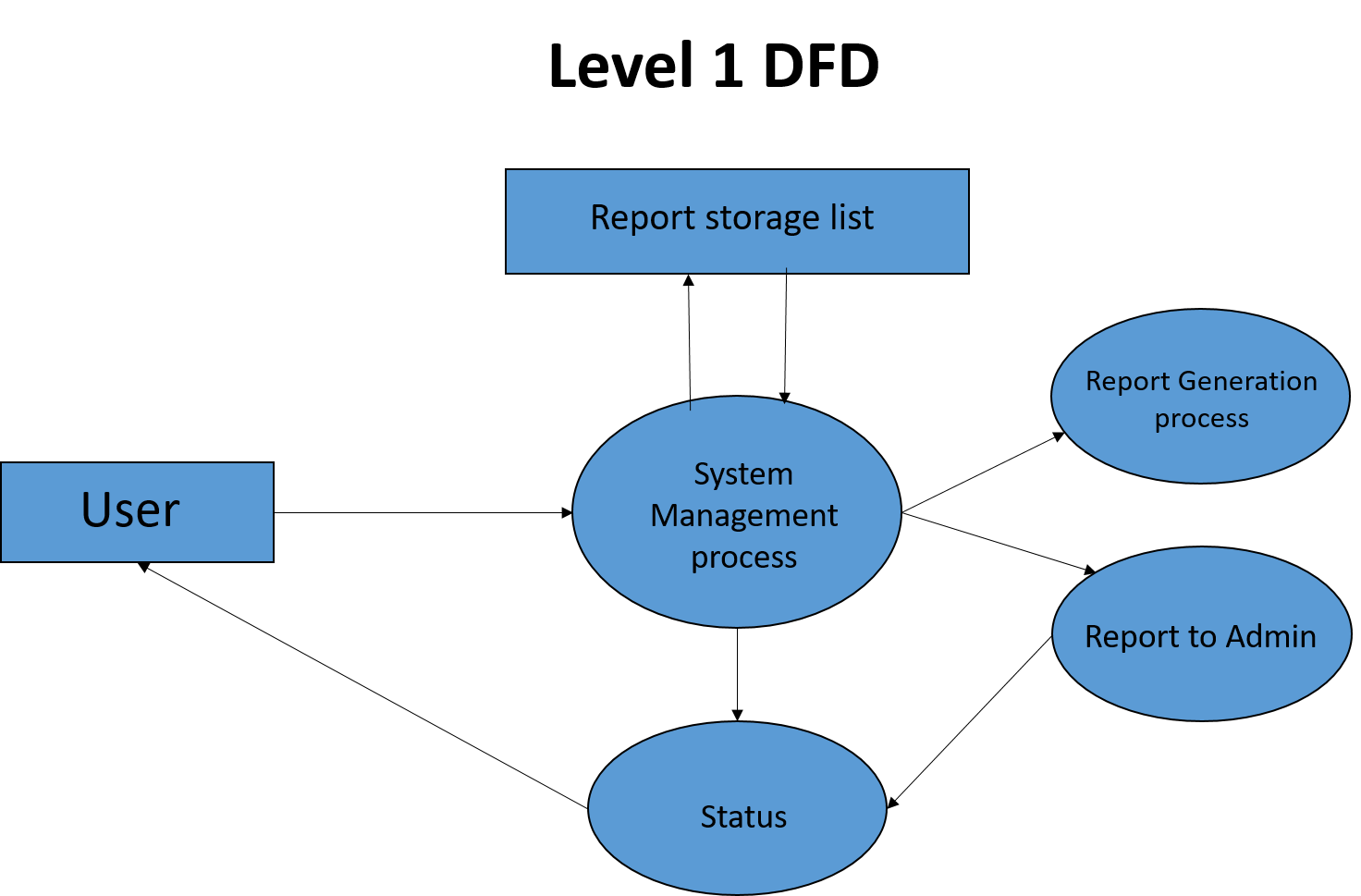
**Project Overview:**

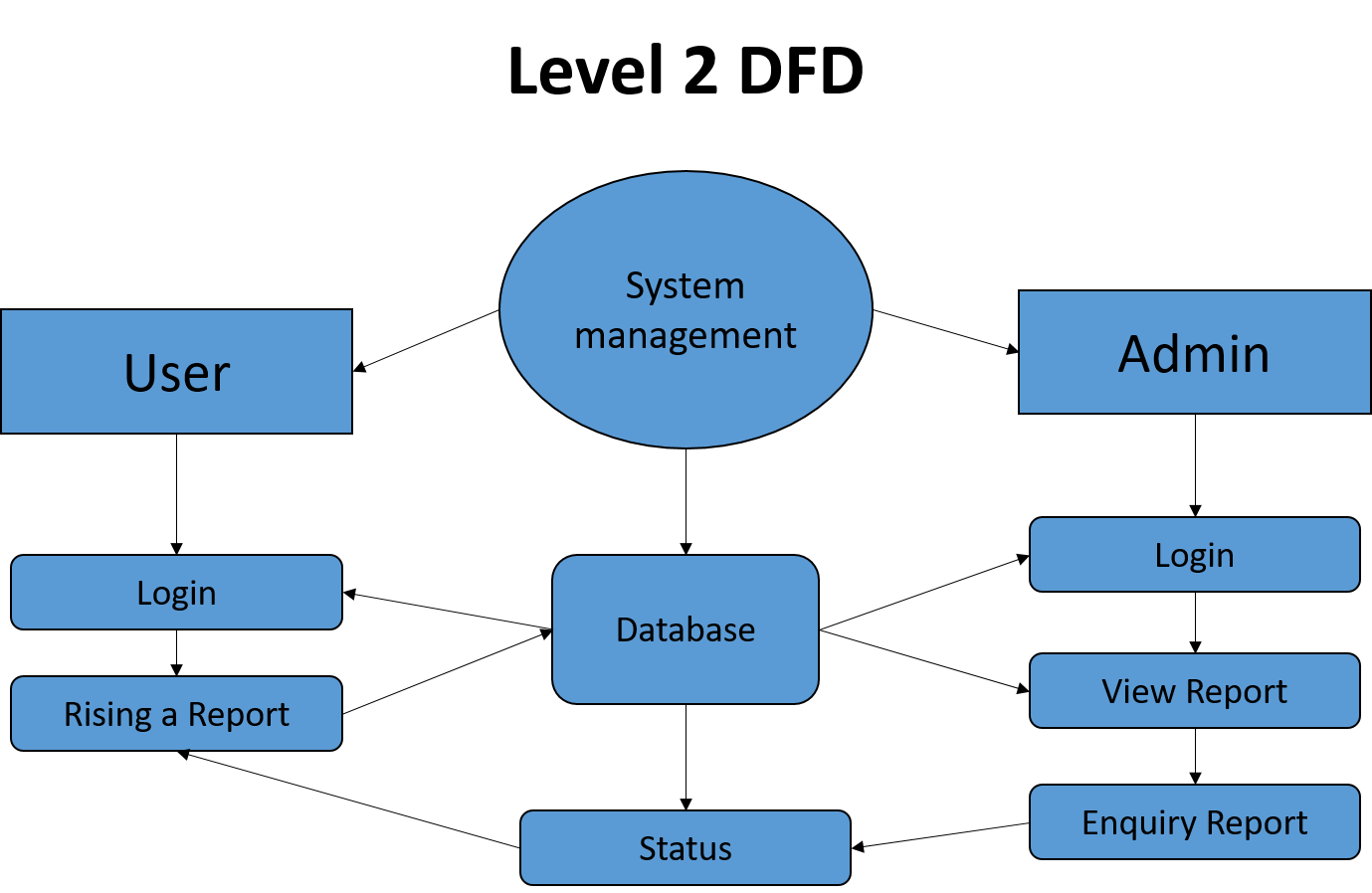
**Objective:** The System Management project aims to develop a web-based system for *managing issues* and providing status updates within a group of systems. It will allow users to report problems they encounter, while administrators can track these reports and provide status updates.

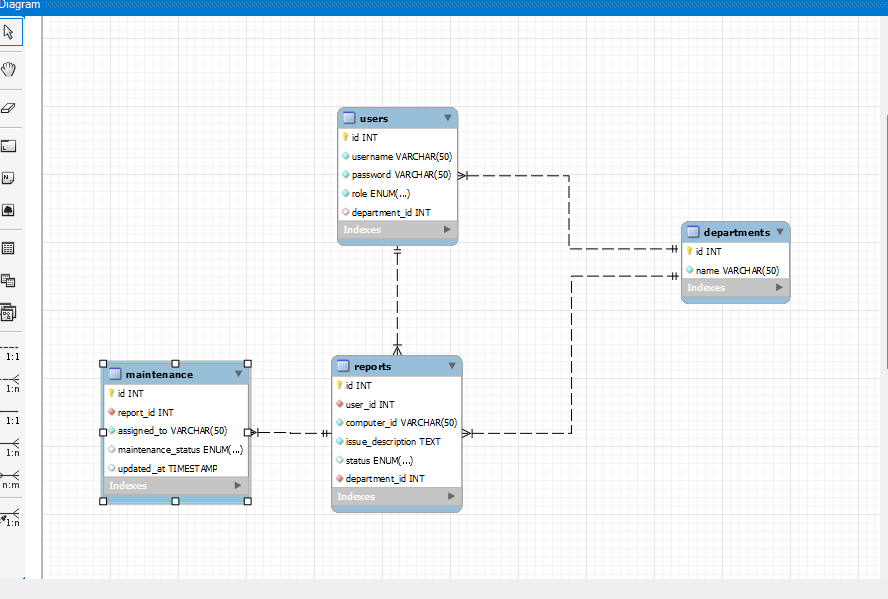
**Scope:** The system will include functionalities for user authentication, problem reporting, status updates, and administration. It will cover the entire lifecycle of a reported problem, from submission to resolution.

**Data Flow Diagram**

****







Code:

**package** com.projct.jdbc.systemmanagement;

**import** java.sql.\*;

**public** **class** SystemManagementDAO {

// JDBC URL, username, and password

**private** **static** **final** String ***JDBC\_URL*** = "jdbc:mysql://localhost:3306/system\_management";

**private** **static** **final** String ***USERNAME*** = "root";

**private** **static** **final** String ***PASSWORD*** = "root";

// JDBC connection

**private** Connection connection;

**public** SystemManagementDAO() {

**try** {

// Connect to the database

connection = DriverManager.getConnection(***JDBC\_URL***, ***USERNAME***, ***PASSWORD***);

} **catch** (SQLException e) {

e.printStackTrace();

}

}

// Method to authenticate a user

**public** **boolean** authenticateUser(String username, String password) {

**try** {

PreparedStatement statement = connection.prepareStatement("SELECT \* FROM users WHERE username = ? AND password = ?");

statement.setString(1, username);

statement.setString(2, password);

ResultSet resultSet = statement.executeQuery();

**return** resultSet.next(); // Returns true if user exists, false otherwise

} **catch** (SQLException e) {

e.printStackTrace();

**return** **false**;

}

}

// Method to authenticate an admin

**public** **boolean** authenticateAdmin(String username, String password) {

**try** {

PreparedStatement statement = connection.prepareStatement("SELECT \* FROM admins WHERE username = ? AND password = ?");

statement.setString(1, username);

statement.setString(2, password);

ResultSet resultSet = statement.executeQuery();

**return** resultSet.next(); // Returns true if admin exists, false otherwise

} **catch** (SQLException e) {

e.printStackTrace();

**return** **false**;

}

}

// Method to report a problem

**public** **void** reportProblem(**int** userId, String problemDescription) {

**try** {

PreparedStatement statement = connection.prepareStatement("INSERT INTO problems (user\_id, problem\_description) VALUES (?, ?)");

statement.setInt(1, userId);

statement.setString(2, problemDescription);

statement.executeUpdate();

System.***out***.println("Problem reported successfully!");

} **catch** (SQLException e) {

e.printStackTrace();

}

}

// Method to send status updates to a user

**public** **void** sendStatusUpdate(**int** userId, String updateDescription) {

**try** {

// Get the problem id of the latest reported problem by the user

PreparedStatement statement = connection.prepareStatement("SELECT id FROM problems WHERE user\_id = ? ORDER BY id DESC LIMIT 1");

statement.setInt(1, userId);

ResultSet resultSet = statement.executeQuery();

**if** (resultSet.next()) {

**int** problemId = resultSet.getInt("id");

// Insert status update

statement = connection.prepareStatement("INSERT INTO status\_updates (problem\_id, update\_description) VALUES (?, ?)");

statement.setInt(1, problemId);

statement.setString(2, updateDescription);

statement.executeUpdate();

System.***out***.println("Status update sent successfully!");

} **else** {

System.***out***.println("No problems reported by the user!");

}

} **catch** (SQLException e) {

e.printStackTrace();

}

}

// Main method for testing

**public** **static** **void** main(String[] args) {

SystemManagementDAO systemManagementDAO = **new** SystemManagementDAO();

// Authenticate a user

**boolean** userAuthenticated = systemManagementDAO.authenticateUser("user1", "password123");

System.***out***.println("User Authentication: " + userAuthenticated);

// Authenticate an admin

**boolean** adminAuthenticated = systemManagementDAO.authenticateAdmin("admin1", "adminpassword");

System.***out***.println("Admin Authentication: " + adminAuthenticated);

// Report a problem

systemManagementDAO.reportProblem(1, "Unable to connect to the network.");

// Send a status update to the user

systemManagementDAO.sendStatusUpdate(1, "The network issue has been resolved.");

}

}

Sql code:

CREATE DATABASE IF NOT EXISTS system\_management;

USE system\_management;

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) NOT NULL,

password VARCHAR(50) NOT NULL );

CREATE TABLE admins (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) NOT NULL,

password VARCHAR(50) NOT NULL );

CREATE TABLE problems (

id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

problem\_description TEXT,

status VARCHAR(50) DEFAULT 'Pending',

FOREIGN KEY (user\_id) REFERENCES users(id) );

CREATE TABLE status\_updates (

id INT AUTO\_INCREMENT PRIMARY KEY,

problem\_id INT,

update\_description TEXT,

timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (problem\_id) REFERENCES problems(id) );

desc select \* from users;

select \* from users;