Project Proposal

On

POLLUTION MANAGEMENT SYSTEM

Guided By:-

Mr. Anuj Kumar

Created By:-

(Lucky Kumar, Manshi, Rakhi)

(AF04990650, AF04991266, AF04991724)

ANP-D2405

ITPR

Table Of Contents

PROJECT TITLE:

Pollution Management System (Terminal-Based)

Using Java, JDBC, MySQL

OBJECTIVE

- To provide a system that stores pollution data reliably in a database.
- To allow users to insert, update, view and manage pollution readings.
- To calculate pollution levels in percentage.
- To improve efficiency in environmental data handling.
- To maintain proper records of pollution sources and readings.
- To contribute towards environmental awareness and cleaner surroundings.
- To build a scalable system for future enhancement.
- To make pollution monitoring easier and faster.

Project Category

- This project belongs to the Database Management System category.
- It is a terminal - based software application.
- Core Java is used as the primary level programming language.
- Java Database Connectivity is used to connect Database with Java.
- MySQL is used as the Relational Database Management System.
- It demonstrates CRUD (i.e., CREATE, READ, UPDATE, DELETE) operations.

Analysis

➢ Modules and Description:

- **Module-1: User Authentication**
  - 1.1 User Registration
  - 1.2 User Login
  - 1.3 User Update
  - 1.4 User List
  - 1.5 Delete User
  - 1.6 Role based access (Admin/User)
- **Module-2: Pollution Source Management**
  - 2.1 Add Pollution Source
  - 2.2 View Source List
  - 2.3 Source Details
  - 2.4 Assign Standard Limit Value
  - 2.5 Remove Pollution Source
- **Module-3: Pollution Reading Management**
  - 3.1 Add Pollution Reading
  - 3.2 Update Reading
  - 3.3 View Reading List
  - 3.4 Delete Reading
  - 3.5 Reading Details (Source-wise/User-wise)
- **Module-4: Pollution Calculation and Result Management**
  - 4.1 Fetch Standard Limit from Source
  - 4.2 Calculate Pollution Percentage
  - 4.3 Pollution Category (Safe/Moderate/Dangerous/Critical)
  - 4.4 Store Result in Database
  - 4.5 View Result List
  - 4.6 Compare Results
- **Module-5: Activity Log & Reporting**
  - 5.1 Record User Actions
  - 5.2 View Log History
  - 5.3 Filter Logs(User/Action/Date)
  - 5.4 Generate Report
  - 5.5 Final Pollution Report
  - 5.6 Export

➢ Database Design:

- **Table-1: users**

Fields	DataType	Properties
user_id	varchar(30)	primary key
username	varchar(50)	not null
password	varchar(30)	not null
full_name	varchar(100)	not null
created_at	timestamp	auto generated

Relationship:

- One user can have many pollution readings.
- One user can have many activity logs

Table-2: pollution_sources

Fields	DataType	Properties
source_id	varchar(30)	primary key
source_name	varchar(100)	not null
description	varchar(400)	not null
standard_limit	double	not null

Relationship

- One pollution source can have many readings

- **Table-3: pollution_readings**

Fields	DataType	Properties
reading_id	varchar(30)	primary key
source_id	varchar(30)	not null,foreign key
user_id	varchar(30)	not null,foreign key

emission_value	double	not null
reading_time	timestamp	auto generated

Relationship:

- Many readings belongs to one pollution source
- Many readings belong to one user
- One reading has one calculated result

➢ User_pollutionReading table:

Fields	datatype	properties
user_id	varchar(30)	foreign key
reading_id	varchar(30)	foreign key

➢ pollutionSource_pollutionReading table:

Fields	datatype	properties
source_id	varchar(30)	foreign key
reading_id	varchar(30)	foreign key

- **Table-4: calculated_results**

Fields	DataType	Properties
result_id	varchar(30)	primary key
reading_id	varchar(30)	not null,foreign key
pollution_percent	float	not null
pollution_status	varchar(50)	not null
calculated_on	timestamp	auto generated

Relationship:

- One calculated result belongs to one reading(one-to-one).

➢ pollutionReading_calculatedResults table:

Fields	datatype	properties
reading_id	varchar(30)	foreign key
result_id	varchar(30)	foreign key

- **Table-5: activity_log**

Fields	DataType	properties
log_id	varchar(30)	primary key
user_id	varchar(30)	not null, foreign key
action	varchar(500)	not null
action_time	timestamp	auto generated

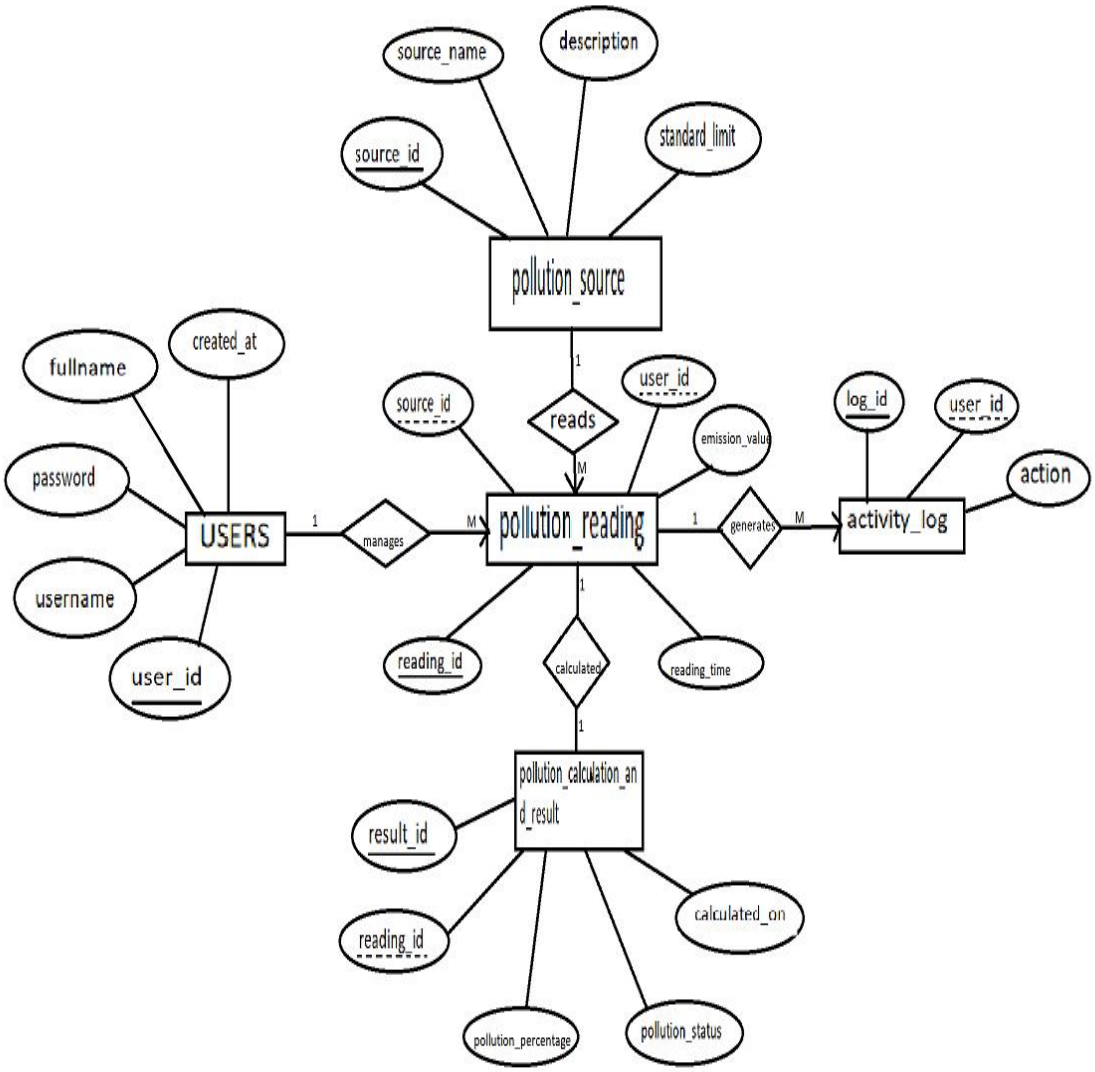
Relationship

- Many logs belong to one user.

➢ User_activityLog table:

Fields	datatype	properties
user_id	varchar(30)	foreign key
log_id	varchar(30)	foreign key

ENTITY RELATIONSHIP(ER) DIAGRAM

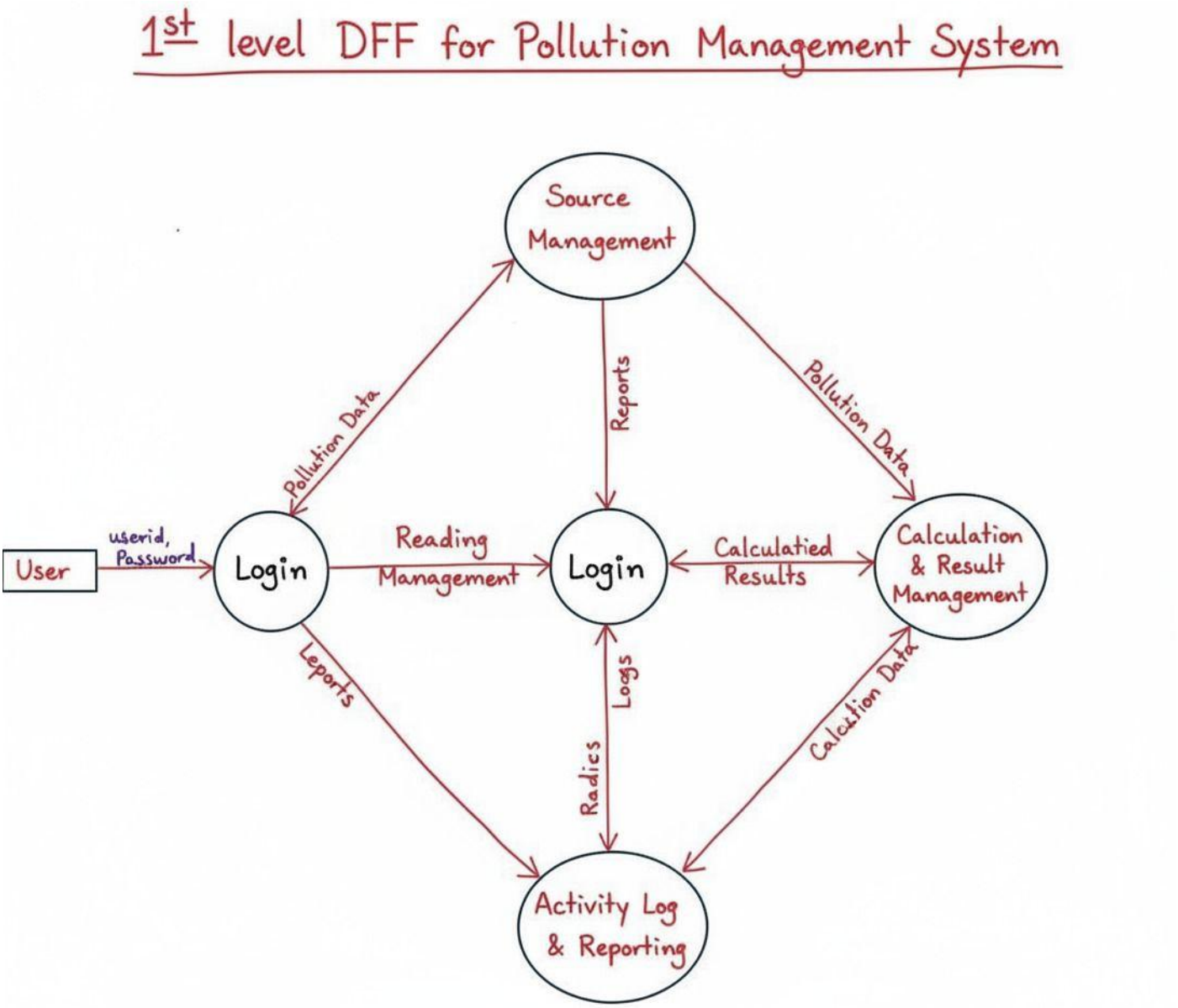


S

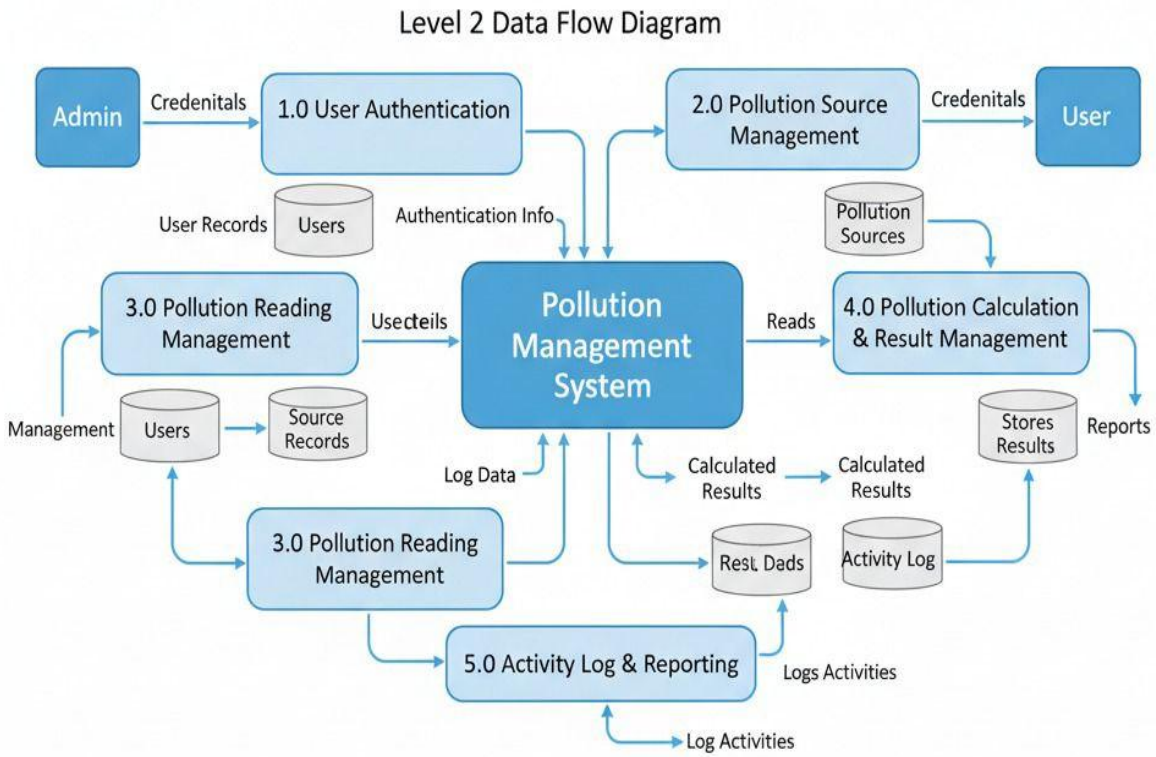
DATA FLOW DIAGRAM

0-level DFD:-

1-level DFD:-


1st level DFF for Pollution Management System
Source Management
Pollution Data
Reports
Pollution Data
User
userid, Password
Login
Reading Management
Login
Calculatied Results
Calculation & Result Management
Leports
Logs
Radics
Calcution Data
Activity Log & Reporting

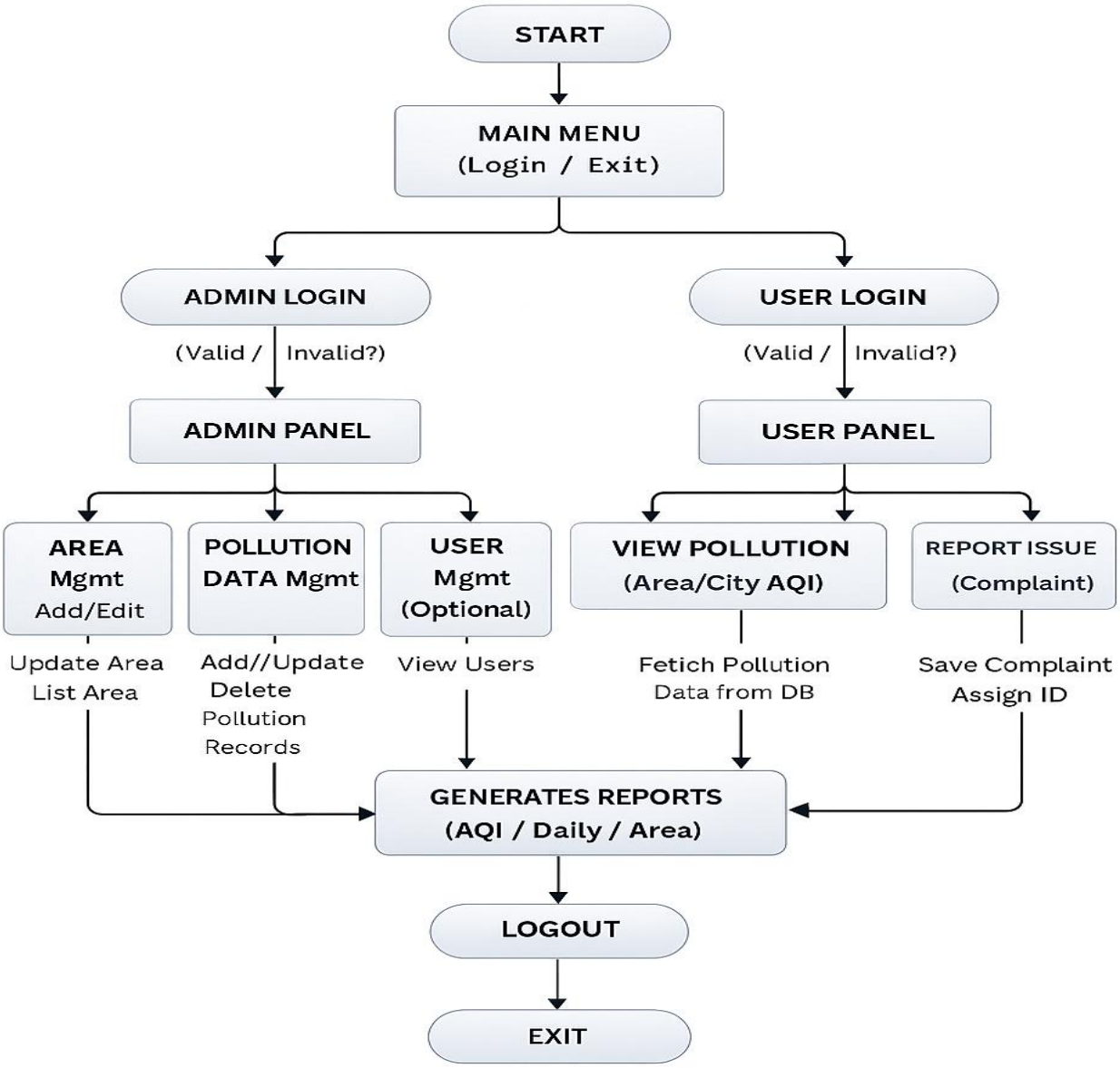
2-level DFD:-


Level 2 Data Flow Diagram
Admin
Credenitals
1.0 User Authentication
2.0 Pollution Source Management
Credenitals
User
User Records
Users
Authentication Info
Pollution Sources
3.0 Pollution Reading Management
Usecteils
Pollution Management System
Reads
4.0 Pollution Calculation & Result Management
Management
Users
Source Records
Log Data
Calculated Results
Calculated Results
Stores Results
Reports
3.0 Pollution Reading Management
Resl Dads
Activity Log
5.0 Activity Log & Reporting
Logs Activities
Log Activities

COMPLETE STRUCTURE

```
PollutionManagementSystem/
|
├── src/
|   ├── dao/            (Database Access Code)
|   |   ├── DBConnection.java
|   |   ├── UserDAO.java
|   |   ├── PollutionRecordDAO.java
|   |   ├── AreaDAO.java
|   |
|   ├── model/          (Java Classes → Table Mapping)
|   |   ├── User.java
|   |   ├── Area.java
|   |   ├── PollutionRecord.java
|   |
|   ├── service/        (Business Logic)
|   |   ├── UserService.java
|   |   ├── PollutionService.java
|   |   ├── ReportService.java
|   |
|   ├── ui/             (Terminal Interaction)
|   |   ├── MainMenu.java
|   |   ├── AdminMenu.java
|   |   ├── UserMenu.java
|   |
|   └── Main.java
|
└── database/
    ├── pollution.sql
```

➢ Process Logical Diagram



PLATFORM USED

➢ HARDWARE REQUIREMENTS:-

- Processor: Intel i5
- RAM: 4GB min
- Hard Disk: 500 free space

➢ SOFTWARE REQUIREMENTS:-

- JDK17
- Eclipse IDE
- MySQL Server

FUTURE SCOPE

- Live pollution data intergration
- Adding a Graphical User Interface (GUI) for better user experience.
- Location based tracking
- Waste and noise modules
- Automatic report generation

BIBLIOGRAPHY

- WEB Resources:-
    1. Oracle Java Documentation
    2. MySQL official documentation
    3. JDBC API Guide
    4. Reference books on DBMS and Java Programming
    5. Online educational resources (GeeksforGeeks, TutorialsPoint, JavaTPoint)