Project Report: Weather Station Interface in Java

Reda El Mansouri & Dekkan Mohammed

December 31, 2024

Contents

Abstract

This report describes the development of a weather station interface using Java and a MySQL database to display real-time weather information. The project also integrates the use of an Arduino microcontroller for climate data collection.

1 Introduction

In this project, we developed a weather station interface that combines modern technologies such as Java, MySQL, and Arduino to provide real-time weather data. This report presents the design details, components used, and tests conducted.

2 Theme

The growing need for accurate climate monitoring motivated this project. The objective was to create a system capable of collecting, storing, and displaying real-time climate data.

3 Materials and Components Used

The main hardware components used in this project are:

- Arduino Microcontroller (model: Arduino Uno)

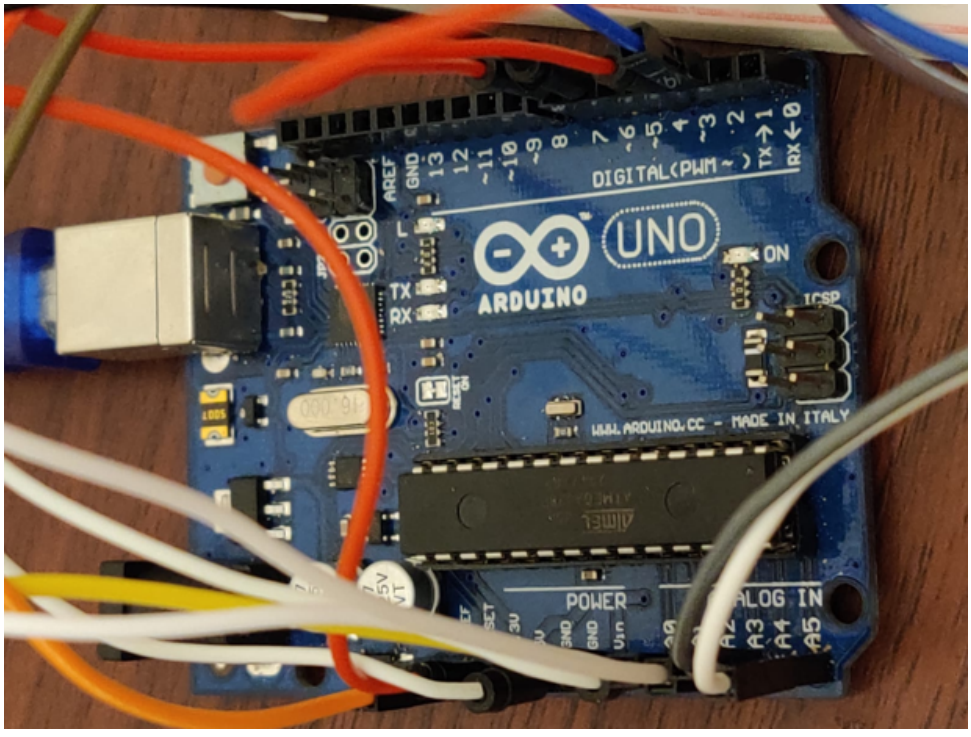

Figure 1: Arduino Uno Microcontroller

- Humidity and temperature sensors

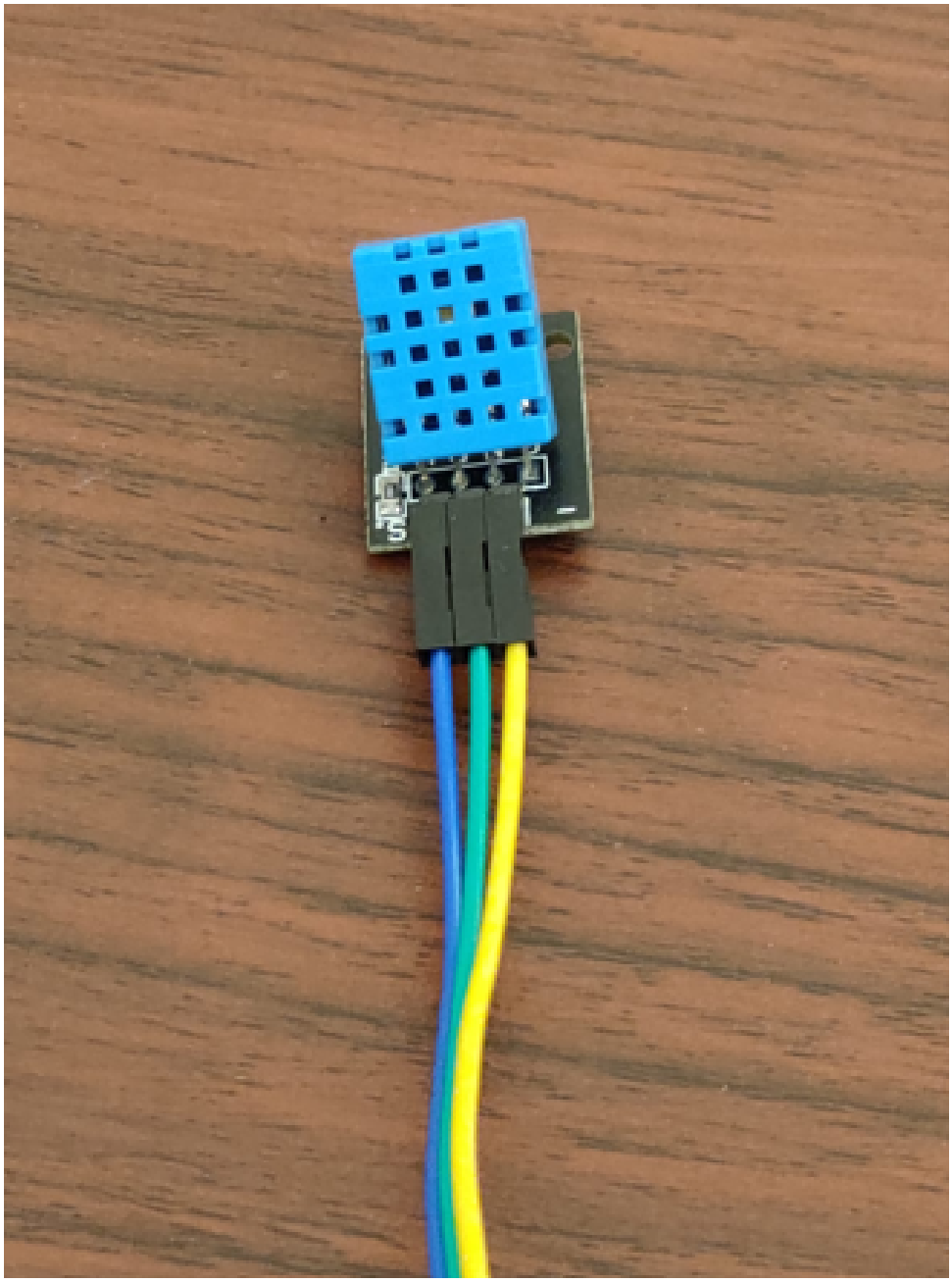

Figure 2: Humidity and Temperature Sensors

- Pressure sensors

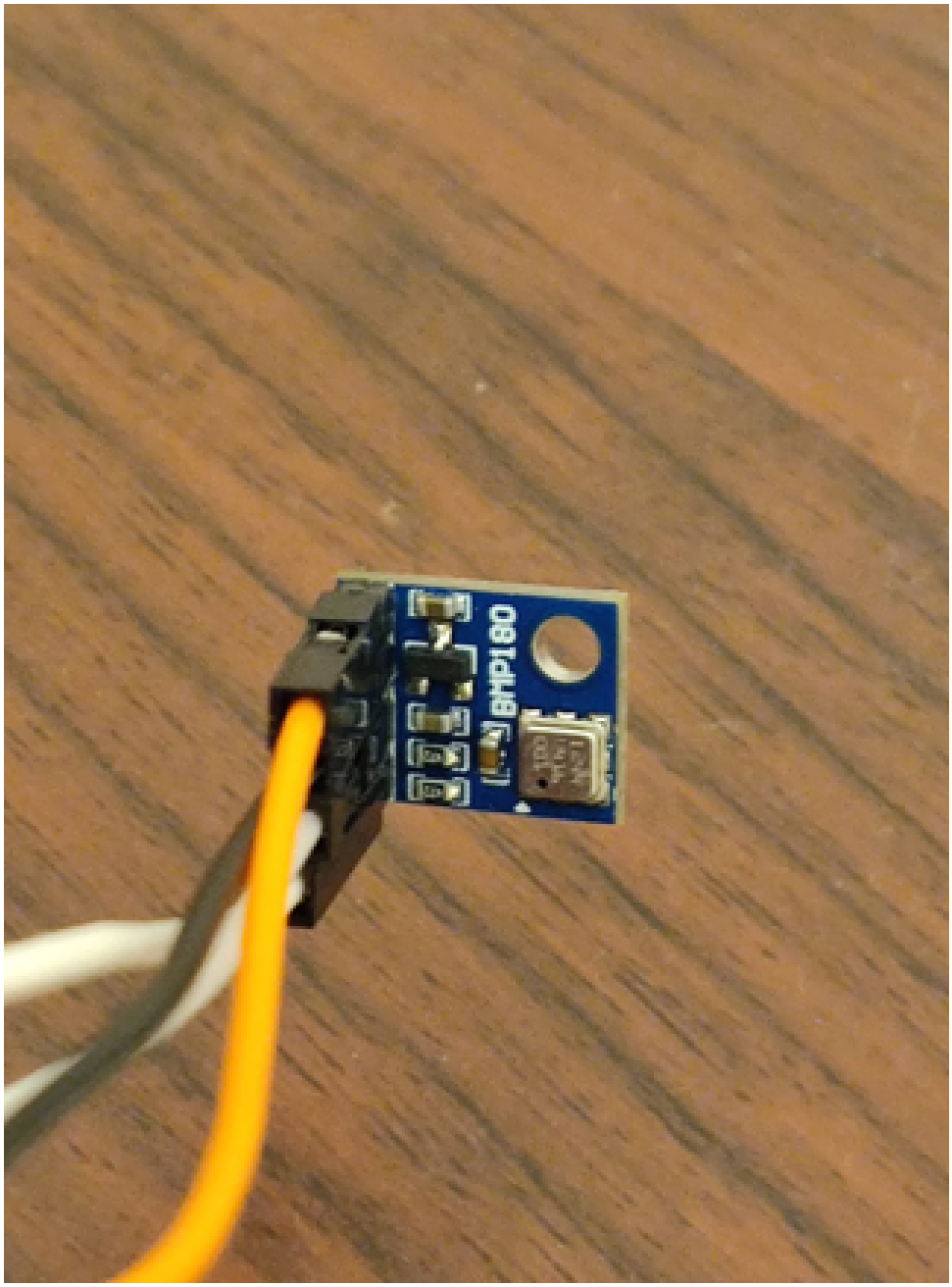

Figure 3: Pressure Sensor

- Light sensors

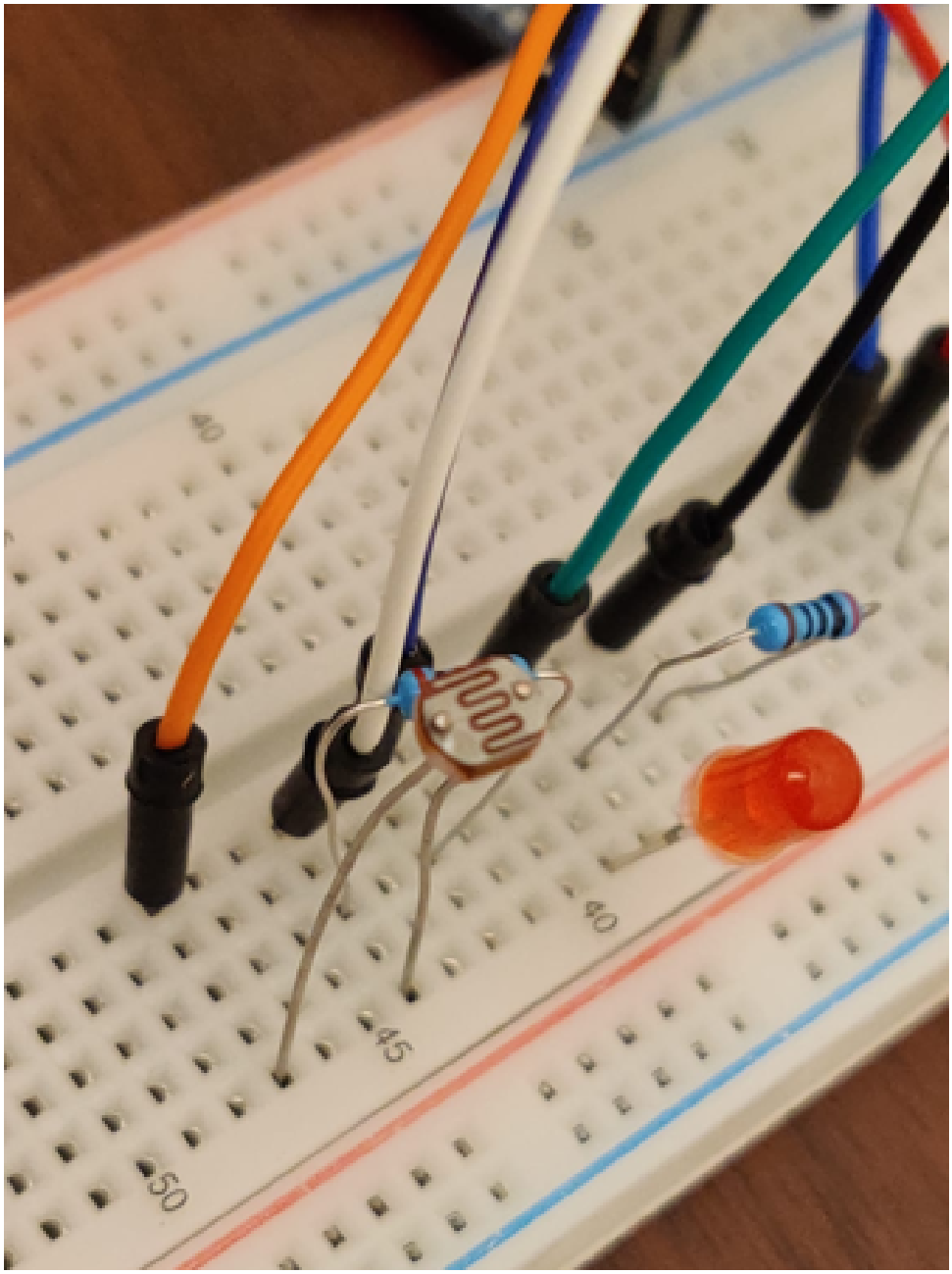

Figure 4: Light Sensor

- Rain sensors

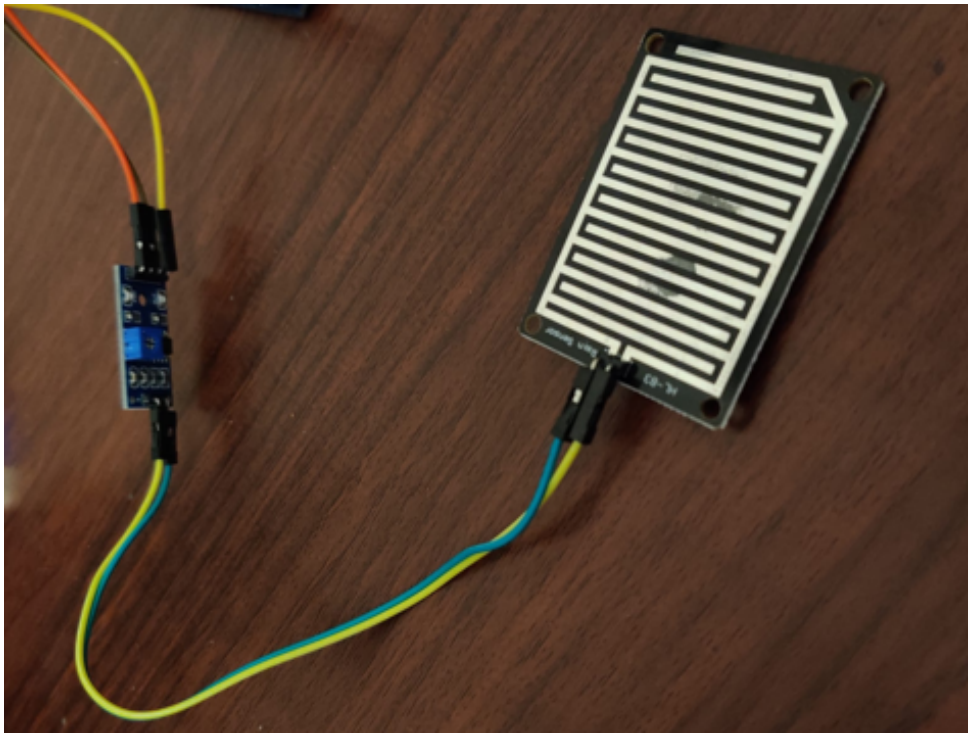

Figure 5: Rain Sensor

- MySQL Database for data storage

	id	humidity	tempretuure	pression	lumniosite	pluie
Edit Copy Delete	76	72	17	1013	2	0
Edit Copy Delete	77	72	17	1013	2	0
Edit Copy Delete	78	72	17	1013	2	0
Edit Copy Delete	79	72	17	1013	2	0
Edit Copy Delete	80	72	17	1013	2	0
Edit Copy Delete	81	72	17	1013	2	0
Edit Copy Delete	82	72	17	1013	2	0
Edit Copy Delete	83	72	17	1013	2	0
Edit Copy Delete	84	72	17	1013	2	0
Edit Copy Delete	85	72	17	1013	2	0
Edit Copy Delete	86	72	17	1013	1	0
Edit Copy Delete	87	72	17	1013	2	0
Edit Copy Delete	88	72	17	1013	2	0
Edit Copy Delete	89	72	17	1013	2	0
Edit Copy Delete	90	72	17	1013	2	0
Edit Copy Delete	91	72	17	1013	2	0
Edit Copy Delete	92	72	17	1013	2	0
Edit Copy Delete	93	73	17	1013	12	25
Edit Copy Delete	94	73	17	1013	12	25
Edit Copy Delete	95	73	17	1013	12	25
Edit Copy Delete	96	74	17	1013	25	0
Edit Copy Delete	97	74	17	1013	25	0
Edit Copy Delete	98	74	17	1013	44	0
Edit Copy Delete	99	74	17	1013	45	0
Edit Copy Delete	100	74	17	1013	71	0

Figure 6: MySQL Database

4 Project Architecture

The project architecture is divided into three main modules:

- **Data Collection:** An Arduino microcontroller connected to sensors measures climate parameters such as temperature, humidity, and pressure. The collected data is sent to the host computer via a serial connection.
- **User Interface:** A Java application displays real-time data in the form of graphs and tables. It also includes functionalities to view historical data.
- **Database:** Collected data is stored in a MySQL database for efficient management and future reference.

5 UML Diagrams

This section presents key UML diagrams for the project:

5.1 Use Case Diagram

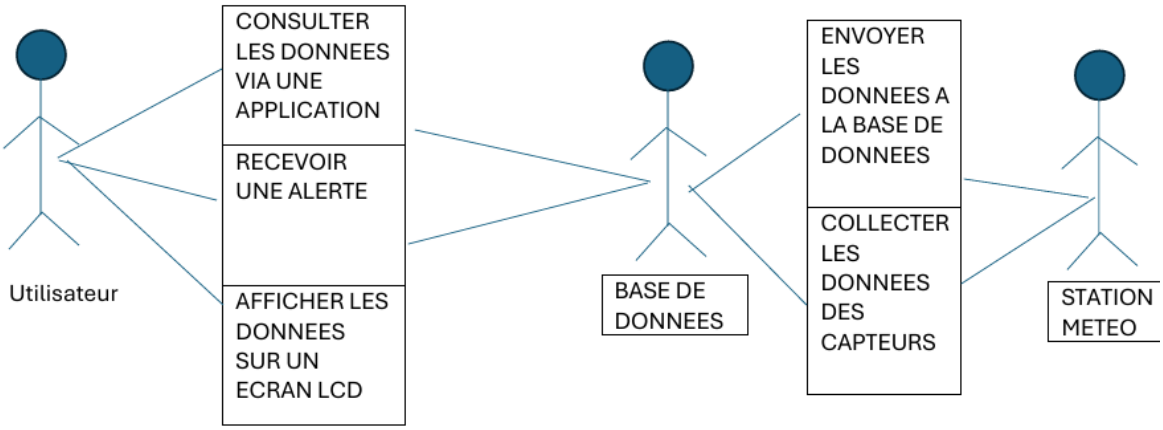

Figure 7: Use Case Diagram

5.2 Class Diagram

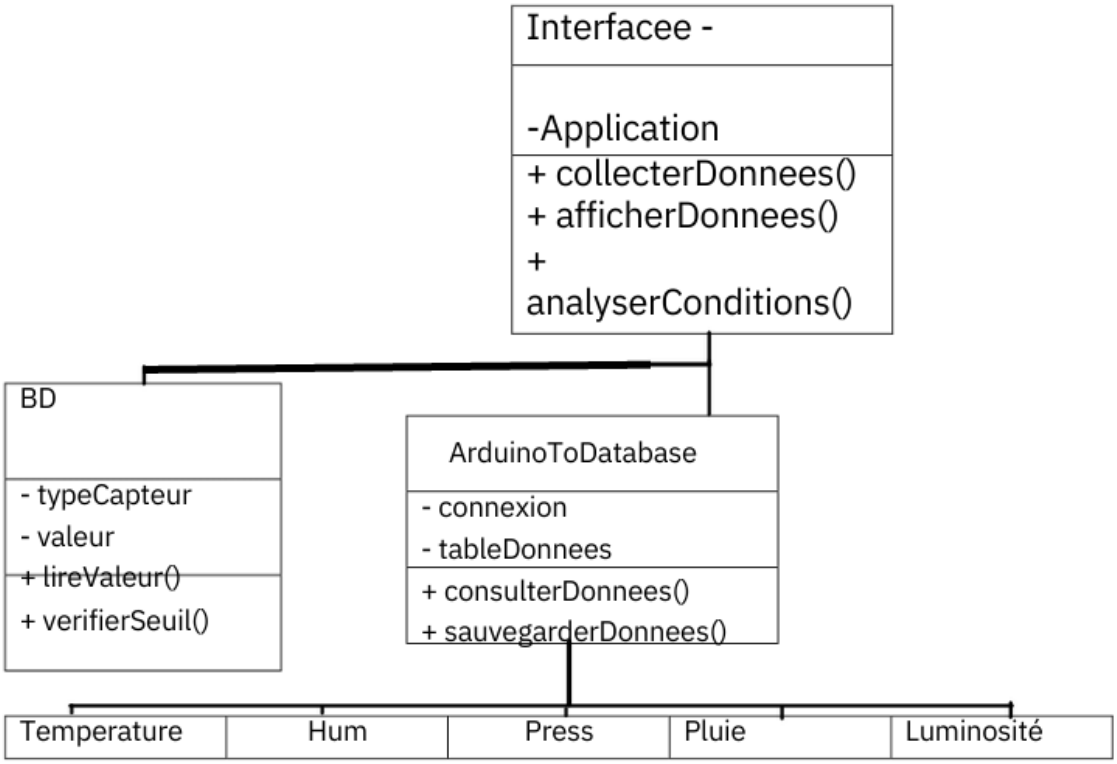

Figure 8: Class Diagram

5.3 Sequence Diagram

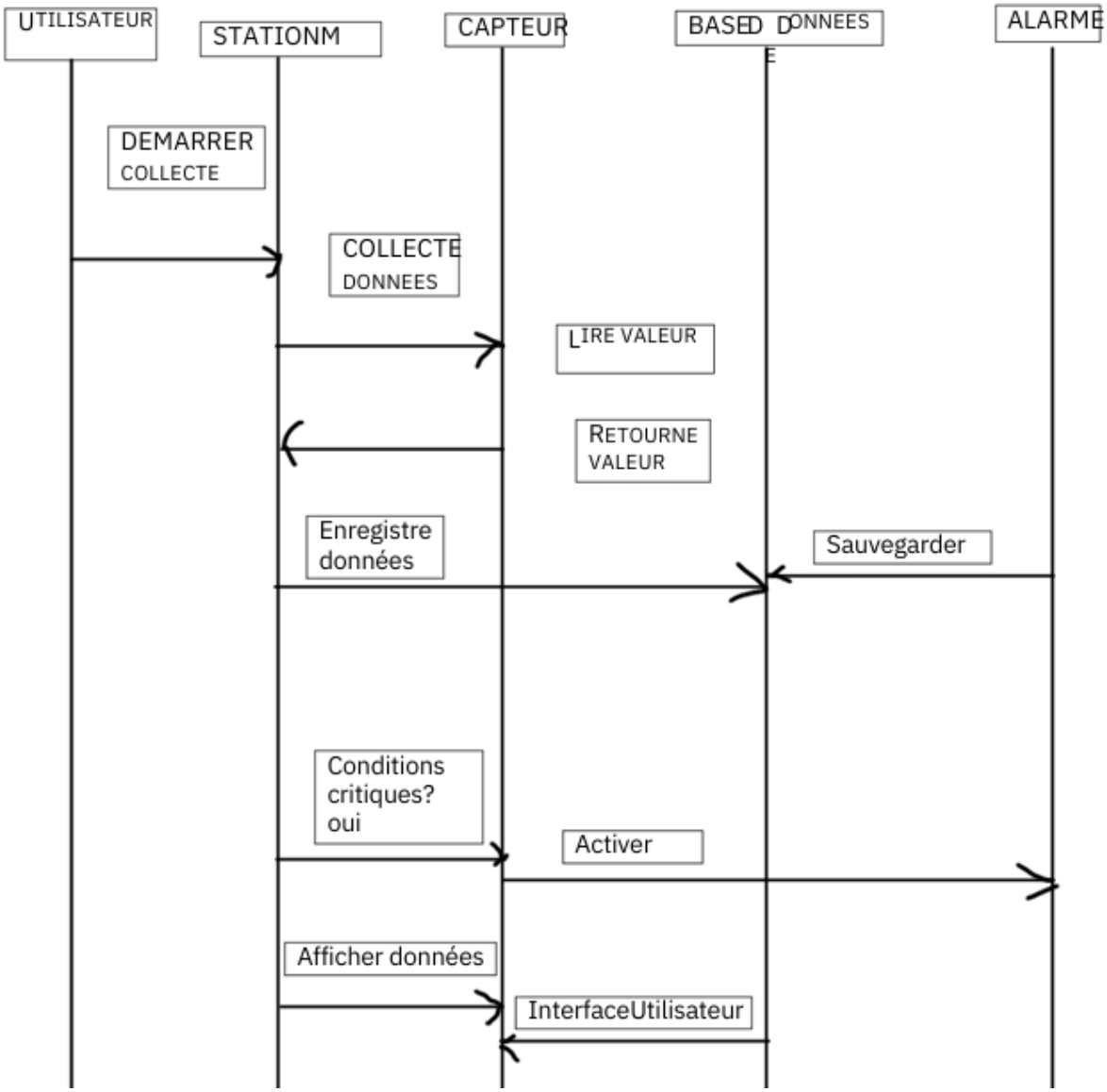

Figure 9: Sequence Diagram

5.4 Activity Diagram

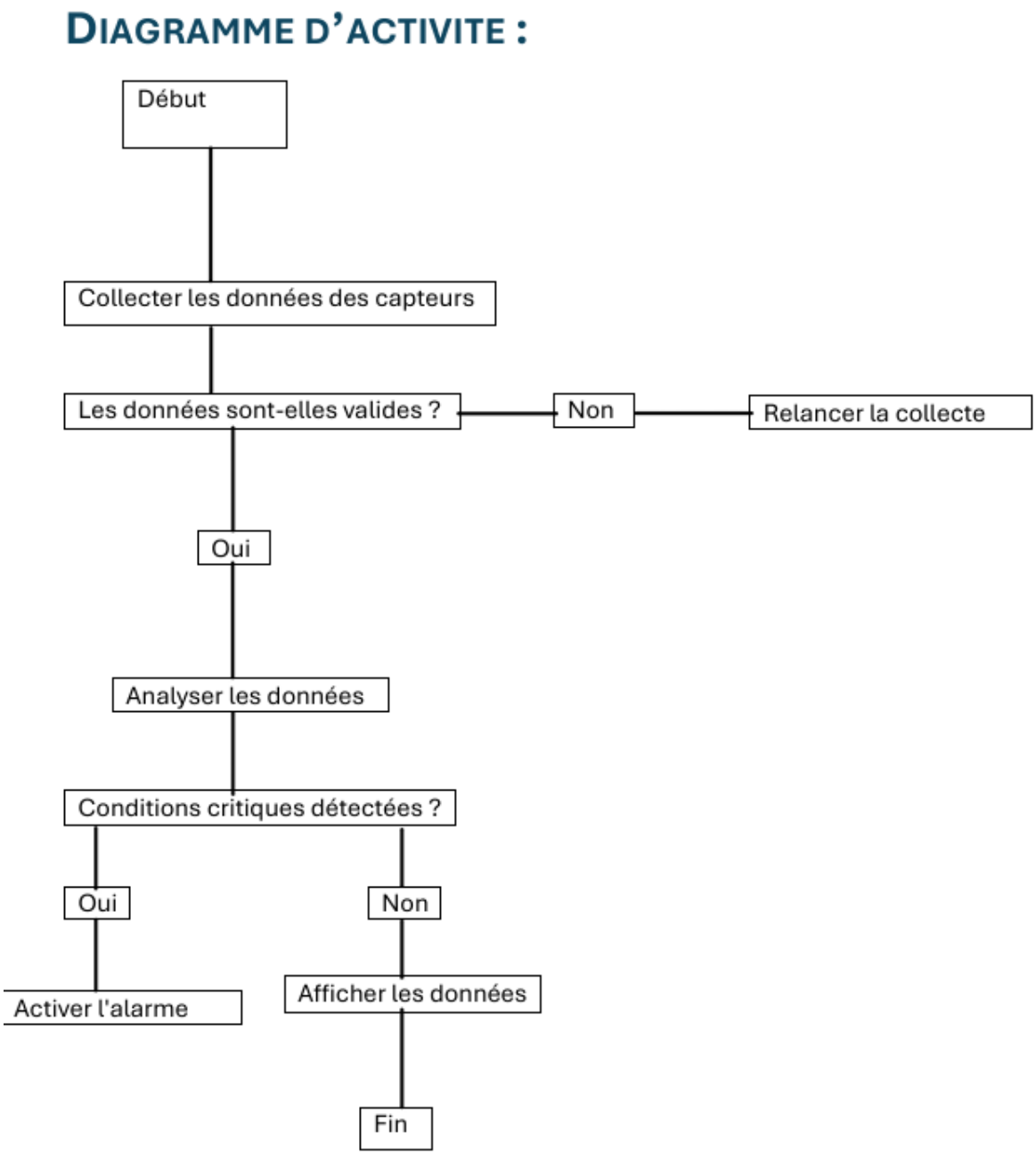

Figure 10: Activity Diagram

6 Unit Test Documentation

Five unit tests were conducted to validate key application functions. Here is a summary of the tests:

ID	Fonction	Resultat attendu	Statut
TU1	Lecture des données des capteurs	**Affichage correct des données dans le moniteur série et l'écran LCD**	**À valider**
TU2	**Détection de pluie**	**LED rouge clignotante lente**	**À valider**
TU3	Indication de beau temps	LED verte allumée	À valider
TU4	Transmission des données	Données enregistrées avec succès dans la base	À valider
TU5	Affichage	Données correctement récupérées et affichées	À valider

Figure 11: Unit Test Documentation

1. Reading sensor data: Positive result

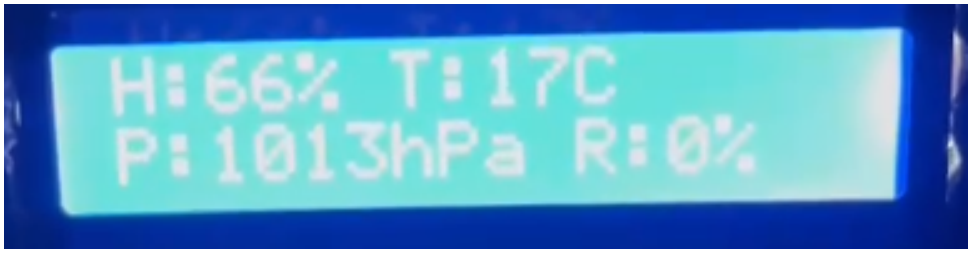

Figure 1: Reading sensor data

2. Rain detection: Positive result

Figure 2: Rain detection

3. Good weather indication: Positive result

Figure 3: Good weather indication

4. Real-time data update: Positive result

```
Data length: 5
data[0] = 66
data[1] = 19
data[2] = 1013
data[3] = 181
data[4] = 0
Data inserted successfully.
```

Figure 4: Real-time data update (Part 1)

	id	humidity	tempretuure	pression	lumniosite	pluie
Edit Copy Delete	76	72	17	1013	2	0
Edit Copy Delete	77	72	17	1013	2	0
Edit Copy Delete	78	72	17	1013	2	0
Edit Copy Delete	79	72	17	1013	2	0
Edit Copy Delete	80	72	17	1013	2	0
Edit Copy Delete	81	72	17	1013	2	0
Edit Copy Delete	82	72	17	1013	2	0
Edit Copy Delete	83	72	17	1013	2	0
Edit Copy Delete	84	72	17	1013	2	0
Edit Copy Delete	85	72	17	1013	2	0
Edit Copy Delete	86	72	17	1013	1	0
Edit Copy Delete	87	72	17	1013	2	0
Edit Copy Delete	88	72	17	1013	2	0
Edit Copy Delete	89	72	17	1013	2	0
Edit Copy Delete	90	72	17	1013	2	0
Edit Copy Delete	91	72	17	1013	2	0
Edit Copy Delete	92	72	17	1013	2	0
Edit Copy Delete	93	73	17	1013	12	25
Edit Copy Delete	94	73	17	1013	12	25
Edit Copy Delete	95	73	17	1013	12	25
Edit Copy Delete	96	74	17	1013	25	0
Edit Copy Delete	97	74	17	1013	25	0
Edit Copy Delete	98	74	17	1013	44	0
Edit Copy Delete	99	74	17	1013	45	0
Edit Copy Delete	100	74	17	1013	71	0

Figure 5: Real-time data update (Part 2)

5. Display: Positive result

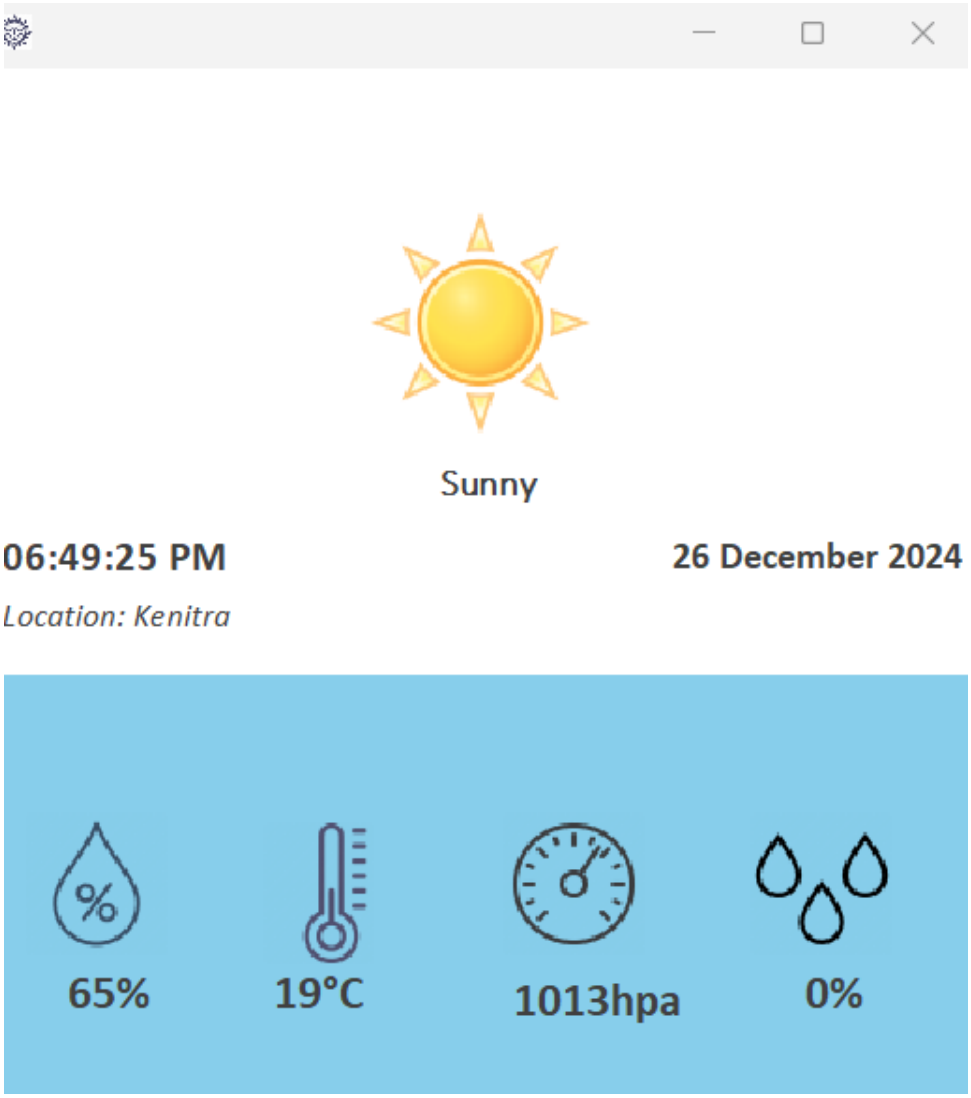

Figure 6: Java GUI

7 Conclusion

This project successfully developed a functional and reliable weather station interface using Java for the main application, MySQL for data management, and Arduino for climate data collection. The tests conducted confirm the robustness and efficiency of the system.