INDIAN INSTITUTE OF INFORMATION TECHNOLOGY ALLAHABAD

Semester III, B.Tech(IT)

OBJECT ORIENTED METHODOLOGY MINI PROJECT

ATMOSPHERIC MANAGEMENT SYSTEM

GROUP-11

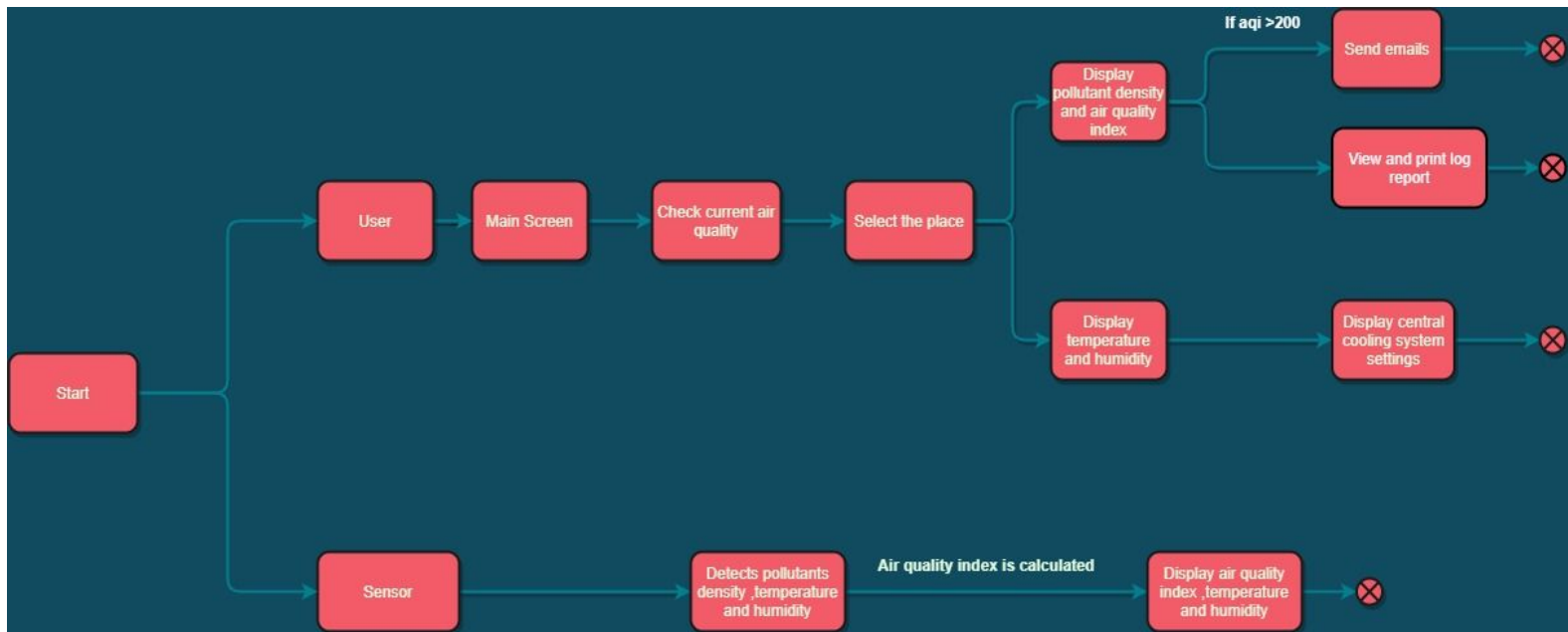| MEMBERS	ROLL NUMBER
ARPIT AGARWAL	IIT2019139
PARAS AGRAWAL	IIT2019145
AMIT JAIN	IIT2019150
AYUSH TIWARI	IIT2019151

CONTENTS

INTRODUCTION

The project asks us to develop a GUI enabled Air Quality Management system for IIITA campus. The air quality includes the detection of the pollutant density of the gases and pollutants in the atmosphere along with the temperature and humidity levels. Currently we have made the project having four sensors at four different parts of the campus namely, CC3, Visitor's Hostel, Gupta Ji Canteen and Admin Block. Each of the sensors monitors the level of hazardous gases present in the environment via means of random number generation. A user is allowed and given access to check the AQI for all sensors even if they are present elsewhere. Whenever the Air Quality Index becomes *Hazardous* it generates an alarm by sending an auto generated email. To avoid stacking of the authority e-mail inbox, we have also provided a flag (yes/no) for the user whether they want to inform the authorities immediately or after a period of time. The system measures the humidity and temperature present in the atmosphere and provides information about the Central Cooling System settings. Apart from that, we have also provided a log report for viewing history of the sensor data and also implemented a way to print it as a PDF.
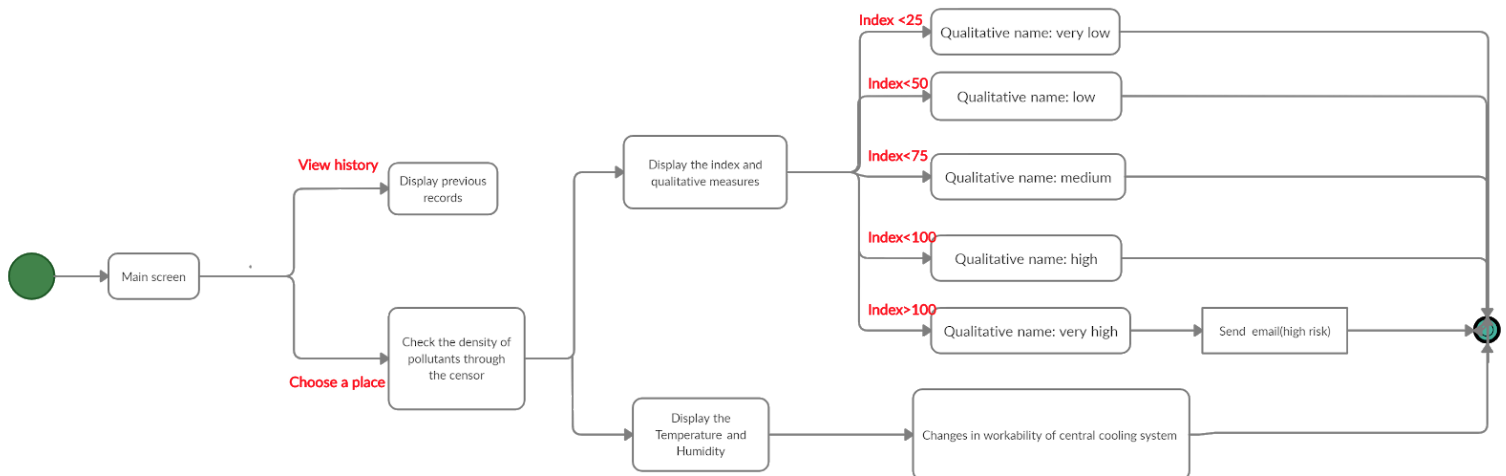
Languages Used - Java(AWT + Swing)

PROCEDURE TO RUN

- ❏ Firstly, see that there are two ways to run the project application.
- ❏ The first one is the "Swing_proj.jar" file in this folder and another one which is present in the folder "dist/Swing_proj.jar".
- ❏ When you open the jar file, the application will begin to run.
- ❏ Also we can open and run the project from the .java files in the "src" folder with the help of any JAVA IDE.
- ❏ The advantage of using .jar files is that it can run on any platform without the need for a separate folder for dependencies or the need of an IDE.
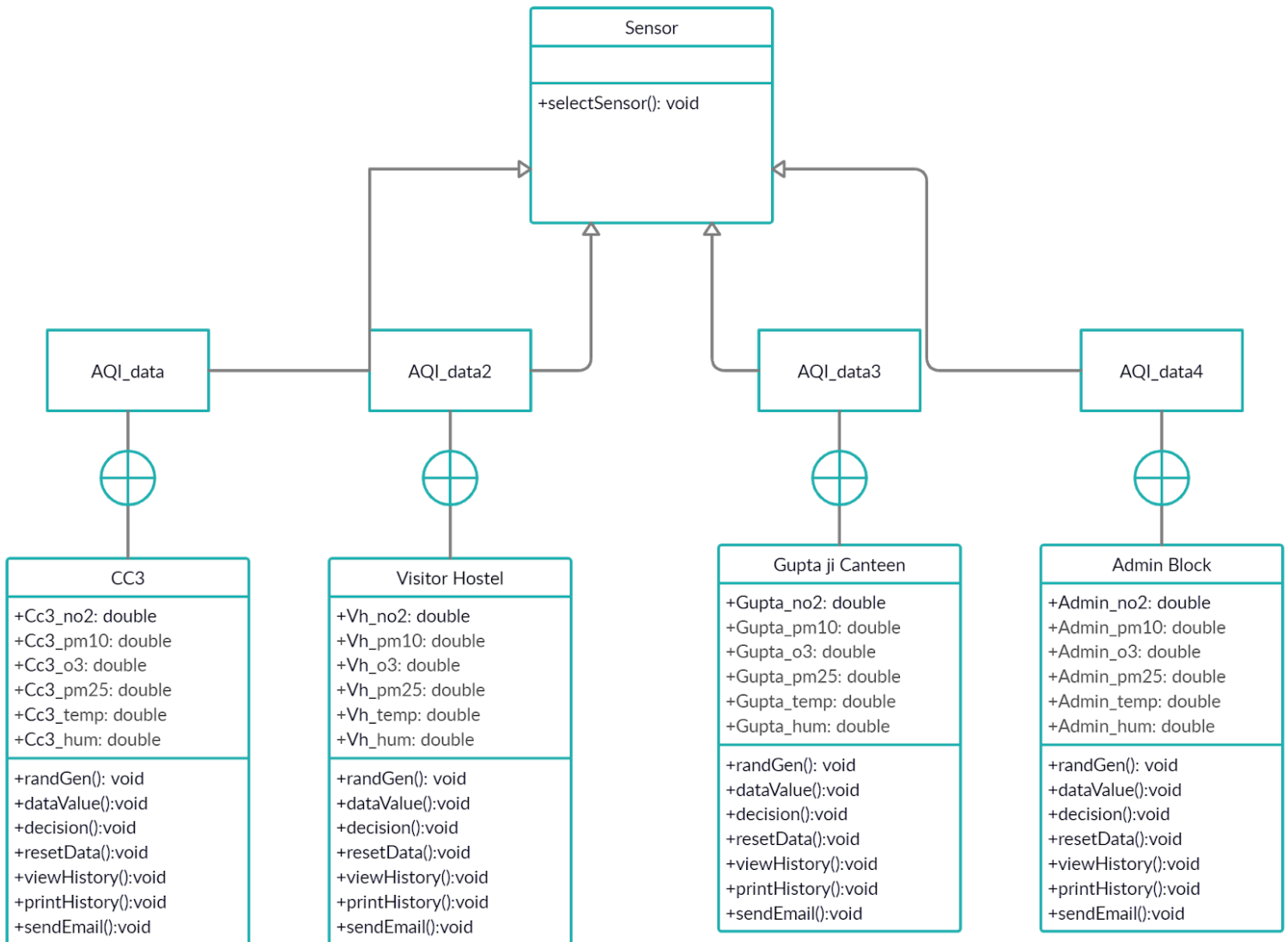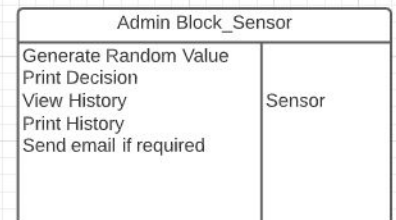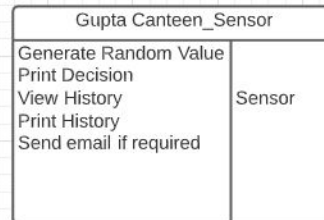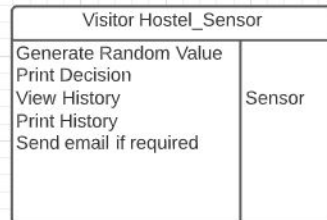
DESIGN PLAN

STATE DIAGRAM

UML CLASS DIAGRAM

Sensor

+selectSensor(): void

AQI_data

AQI_data2

AQI_data3

AQI_data4

CC3

+Cc3_no2: double
+Cc3_pm10: double
+Cc3_o3: double
+Cc3_pm25: double
+Cc3_temp: double
+Cc3_hum: double

+randGen(): void
+dataValue():void
+decision():void
+resetData():void
+viewHistory():void
+printHistory():void
+sendEmail():void

Visitor Hostel

+Vh_no2: double
+Vh_pm10: double
+Vh_o3: double
+Vh_pm25: double
+Vh_temp: double
+Vh_hum: double

+randGen(): void
+dataValue():void
+decision():void
+resetData():void
+viewHistory():void
+printHistory():void
+sendEmail():void

Gupta ji Canteen

+Gupta_no2: double
+Gupta_pm10: double
+Gupta_o3: double
+Gupta_pm25: double
+Gupta_temp: double
+Gupta_hum: double

+randGen(): void
+dataValue():void
+decision():void
+resetData():void
+viewHistory():void
+printHistory():void
+sendEmail():void

Admin Block

+Admin_no2: double
+Admin_pm10: double
+Admin_o3: double
+Admin_pm25: double
+Admin_temp: double
+Admin_hum: double

+randGen(): void
+dataValue():void
+decision():void
+resetData():void
+viewHistory():void
+printHistory():void
+sendEmail():void

CRC DIAGRAM

CRC Diagram

Sensor	
Select Sensor	CC3_Sensor
	Visitor Hostel_Sensor
	Gupta Canteen_Sensor
	Admin Block_Sensor

CC3_Sensor	
Generate Random Value Print Decision View History Print History Send email if required	Sensor

Visitor Hostel_Sensor	
Generate Random Value Print Decision View History Print History Send email if required	Sensor

Gupta Canteen_Sensor	
Generate Random Value Print Decision View History Print History Send email if required	Sensor

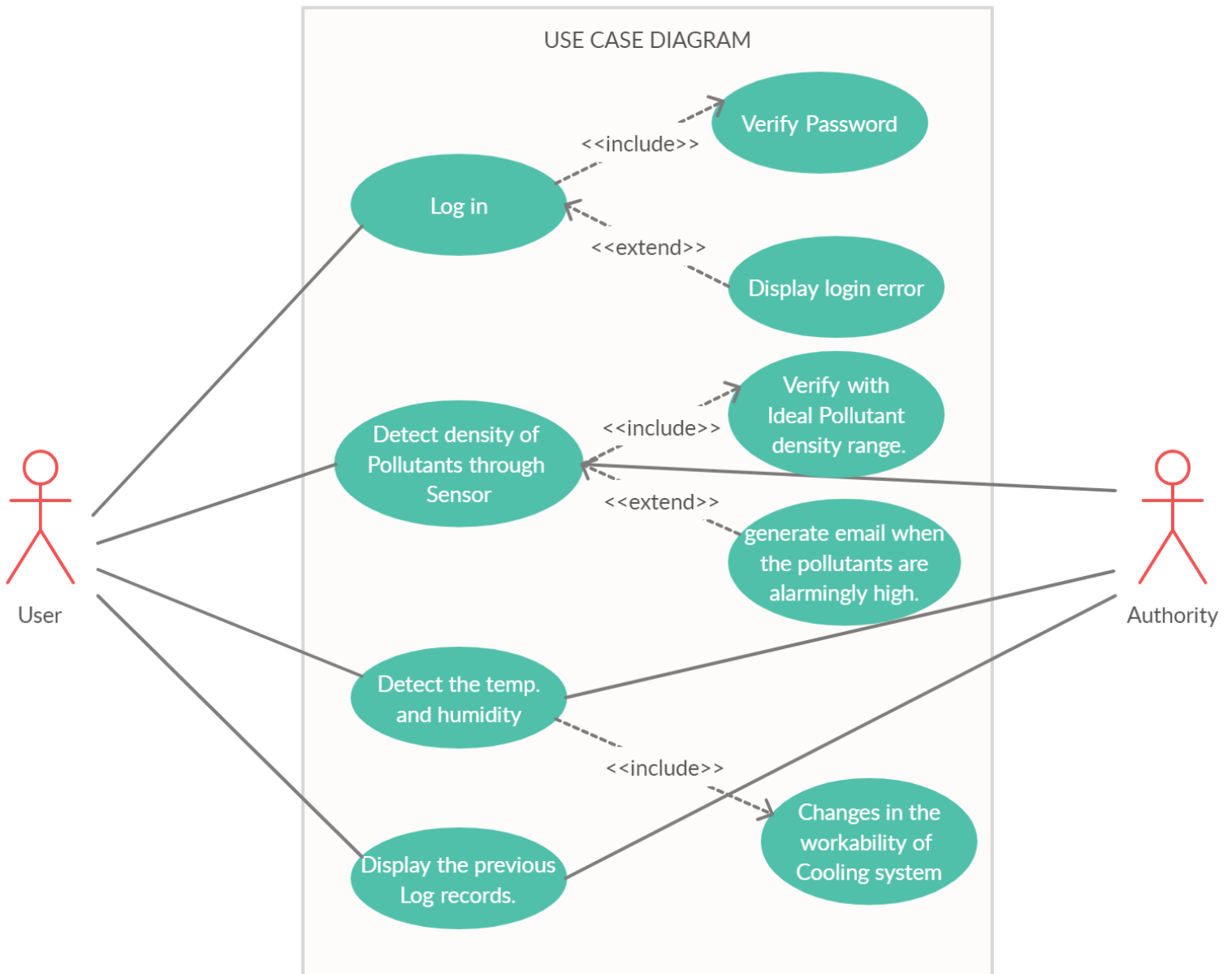| Admin Block_Sensor	
Generate Random Value Print Decision View History Print History Send email if required	Sensor

USE-CASE DIAGRAM

INSTRUCTIONS (FRONTEND)

SCREEN-1 : WELCOME

> ➤ Upon launching, the user views a Welcome Screen where they are given two choices, either Quit the running program (close symbol on top of window) or Check Current Air Quality.
> ➤ We have provided a button in order to implement that and when the user clicks the button to check the current AQI, they are redirected to the second screen where they can choose the sensors.

SCREEN-2 : SENSOR CHOICE

> ➤ The next frame asks the user to select the sensor or the place from a drop-down list whose detection values they wish to know.
> ➤ Also there are two buttons, one to go back to the welcome screen and the other to proceed to that particular sensor frame which they have selected.
> ➤ It is also possible for a user to select multiple sensors and simultaneously check their data in order to compare the surroundings inside the campus.

SCREEN-3 : SENSOR DATA SCREEN

- ➢ There are four similar screens for the four sensors placed at different places inside the IIITA campus.
- ➢ Each of them have the labels for the gases and pollutant density, temperature, humidity, AQI, its qualitative name and the central cooling system settings (depending on temperature and humidity).
- ➢ When the user clicks on the detect data button, the gases and pollutant density, temperature, humidity, AQI and its qualitative measurements are displayed automatically via the sensors (random number generation) onto the frame.
- ➢ We have also done colour coding based on the AQI values.
- ➢ When the user clicks the submit button the cooling system settings are displayed and based on the AQI values (if >200 or Hazardous), a pop-up dialog box is displayed with a warning message which also sends an automatic e-mail to the authority alerting them about the situation and to take proper precautions.
- ➢ So once this process is completed the information is dynamically stored in the history with the given date and time when this action was performed, whenever the

user wishes to detect again after a few minutes or hours, they can do so and the data will get stored in a table.

➢ Now, there is also a button for showing the history for the past detections based on the time and date they were done. The history is stored in a table and can be printed as a PDF whenever the user wants to do so.

CC3 LOG DATA REPORT

Date:Time	NO$_2$	PM$_{10}$	O$_3$	PM$_{2.5}$	AQI
2020-11-19::15:00:03	487.08	189.48	100.49	67.34	211.1
2020-11-19::15:00:10	200.93	17.76	13.35	15.82	61.97
2020-11-19::15:00:13	0.74	216.61	237.32	110.0	141.17
2020-11-19::15:00:16	101.67	61.01	189.06	170.16	130.47
2020-11-19::15:00:18	466.59	281.72	250.59	50.2	262.28
2020-11-19::15:00:24	60.78	142.91	8.04	190.26	100.5

🖶 PRINT   ↩ Back

CC3 Log Report

Date:Time	NO$_2$	PM$_{10}$	O$_3$	PM$_{2.5}$	AQI
2020-11-19::15:00:03	487.08	189.48	100.49	67.34	211.1
2020-11-19::15:00:10	200.93	17.76	13.35	15.82	61.97
2020-11-19::15:00:13	0.74	216.61	237.32	110.0	141.17
2020-11-19::15:00:16	101.67	61.01	189.06	170.16	130.47
2020-11-19::15:00:18	466.59	281.72	250.59	50.2	262.28
2020-11-19::15:00:24	60.78	142.91	8.04	190.26	100.5

REFERENCES

★ Designing a SWING GUI in Netbeans

https://netbeans.org/kb/docs/java/quickstart-gui.html

★ Documentation

https://docs.oracle.com/en/

★ Stack OverFlow

★ JavatPoint

★ Java E-Mail System

https://examples.javacodegeeks.com/core-java/send-email-with-gmail-in-java-example/