Case Study 2- Smart Classroom Monitor

Step 1: Understanding

A Python program is required to be built that monitors a classroom. It needs to keep a track of the room state like the status of the projector, capacity and topic. It also needs to keep an attendance list. It needs to alert if the room is full, if the temperature is too cold or hot (<16 or >28), topic set but projector is off. It should display a menu through which the user can interact and perform various function like toggle projector, setting the topic, adding student to room, removing student from the room, temperature records. So the functions are:

1. Track projector and equipment status
2. Track current attendance
3. Track current room capacity
4. Let the user select the topic of the class
5. Keep the temperature log of the room and provide statistics and report
6. Raise alerts accordingly
7. Display a Menu from where the user can select the aforementioned features to access them and give necessary inputs while getting the necessary outputs

Step 2: Inputs & Outputs

The inputs of the program are as follows:

1. Student Name (str)
2. Room capacity (int)
3. Temperature (float, °C)
4. Topic (str)
5. Projector status (bool)
6. Menu selection (int): select an option from the displayed menu

The outputs of the program are as follows:

1. Attendance count
2. Current room capacity
3. Temperature stats and report (min, max, avg)
4. Topic state
5. Projector state
6. Alerts: “Room full”, Temperature warning, reminder about projector

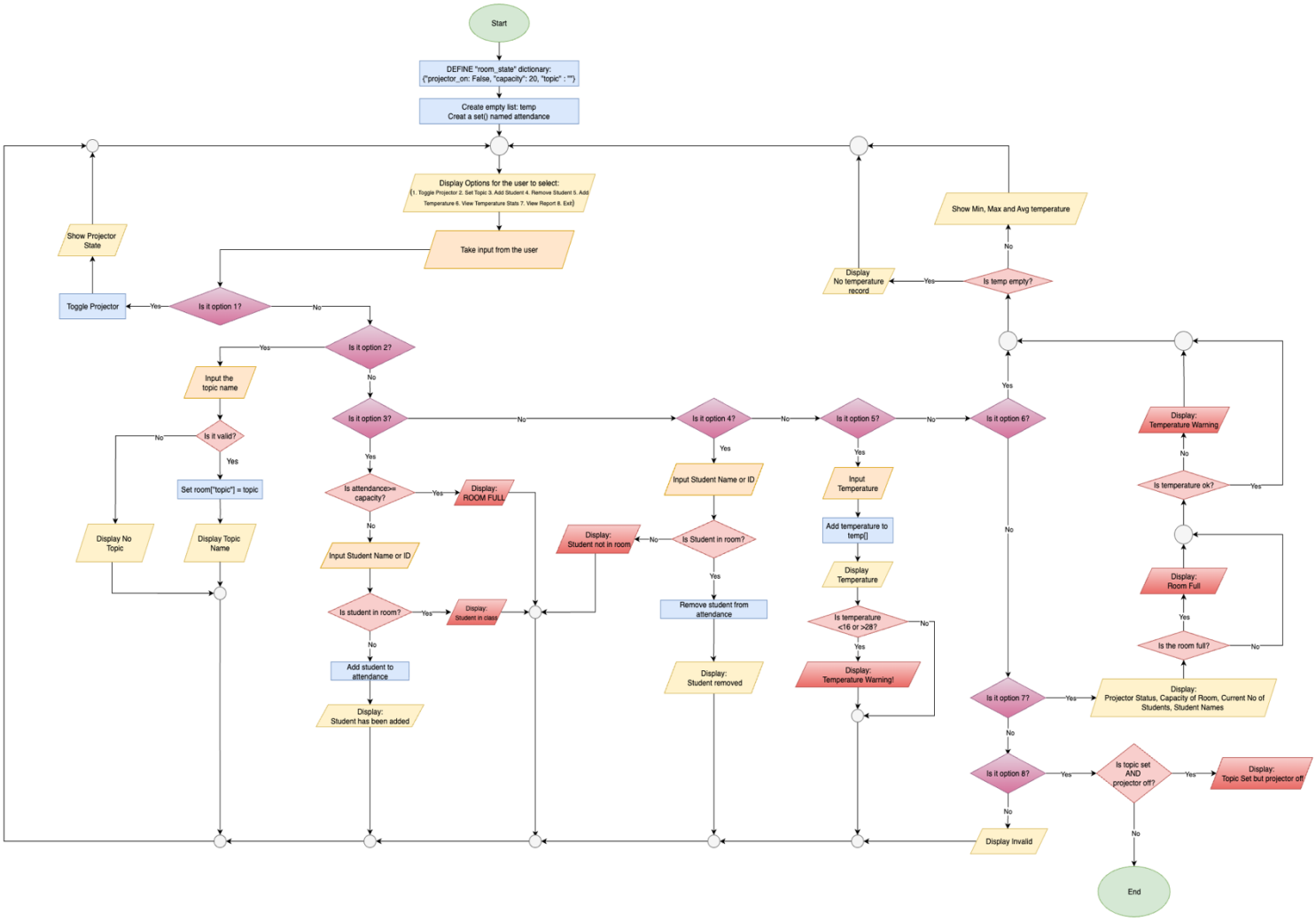
Step 3: Algorithm Design

1. Input:
    - Ask for user actions via a menu loop.
    - Validate numeric inputs
    - Validate strings

2. Process:
    - Toggle projector: ON or OFF
    - Set topic: update string in dict.
    - Add/remove students: update attendance set; check capacity.
    - Add temperature: append to list.
    - Compute stats: min, max, average.
    - Generate alerts:
        - If len(attendance) > capacity then “ROOM FULL”
        - If temp < 16 or > 28 then “Temperature Warning”
        - If topic set AND projector is off when exiting then “Reminder: projector off.”
3. Output:
    - Reports:
        - Room state
        - Attendance
        - Temperatures
        - Alerts

Step 4: Flowchart & Pseudocode

Flowchart:



Pseudocode:

```
Begin
    SET room_state  as dict = {“projector_on” : False, “capacity” : 20, “topic” =””}
    attendance <- empty set
    temp <- empty list

    FUNCTION toggle_projector(room)
        room["projector_on"] = not room["projector_on"]

    FUNCTION set_topic()
        room["topic"] = topic
        IF topic THEN
            PRINT topic
        ELSE
            PRINT “No Topic”

    FUNCTION add_student()
        IF len(attendance) >= room["capacity"]
            PRINT "ROOM FULL"
        ELSE
            ADD name to attendance

    FUNCTION remove_student()
        REMOVE name if exists

    FUNCTION add_temp()
        APPEND value to temp

    FUNCTION show_stats()
        IF temp THEN
            PRINT (min, max, avg)

    FUNCTION show_report()
        PRINT projector status, topic, capacity, current students number and name
        PRINT stats
        IF len(attendance) > room["capacity"]
            PRINT warning
```

```
        IF any temp < 16 or > 28
            PRINT warning
        IF topic set and projector off PRINT reminder

    MAIN FUNCTION
        WHILE True:
            PRINT “1. Toggle Projector 2. Set Topic 3. Add Student 4. Remove Student
            5. Add Temperature 6. View Temperature Stats 7. View Report 8. Exit”

            GET user choice
             IF choice == 1: toggle_projector()
            IF choice == 2: set_topic()
            IF choice == 3: add_student()
            IF choice == 4: remove_student()
            IF choice == 5: add_temp()
            IF choice == 6: show_stats()
            IF choice == 7: show_report()
            IF choice == 8: EXIT LOOP

        ON EXIT:
            IF topic not empty AND projector_on == False:
                PRINT "Reminder: projector is OFF!"

END
```

Step 6: Testing- Handwritten Expected Results + Test Runs & Notes

Test Scenario	Input	Expected Output & Workings
1. Alert "Room full" when only names are given as input	• Capacity: 3 • 4 names added • Show stats* • Exit	• Alert message is shown when the 4th name is entered. • Stats show projector status, capacity, current students name & no. • Stats doesn't show topic as none is set, no temperature recordings • Alert Message "Room Full" is shown
2. Alert "Temperature Warning" with no student inputs	• Temperature is 12°C • Show stats* • Exit	• Alert Message is shown when the temperature is entered • Alert Message states "Temperature Warning" • Stats doesn't show any students name / ID, topic, current no. • Temperature stats shows min: 12°C max: 12°C avg: 12°C
3. Inserting names in different case (Uppercase, lowercase, etc)	~~• Apo~~ • Add "Ari", "aRi", "Bob"	• "Ari" gets added • "aRi" doesn't get added and alert message: "Ari is already in room is shown" • "Bob" gets added
4. Inserting ID number instead of name	• Add B312851	• Works fine
5. Different Temperatures added	• Temp1: 18°C • Temp2: 22°C	• Stats shows min: 18°C, max: 22°C, avg: 20°C

Test Scenario	Input	Expected Output & Workings
6. Program Exitted when topic is set but projector is OFF	• Option 8 (Exit) • Topic: rain • Projector: OFF	• Before exiting program shows "Topic set but projector is off" Reminder
7. Topic set but projector is OFF	• Topic: Random • Projector: OFF	• Alert Message: "Projector is OFF"
~~8. Invalid Temperature~~		
8. Invalid inputs	• Selected Option: '9' or anything apart from (1-8) ~~• Anything apart from (1-8) entered~~	• Error Message Shown
9. Empty Class	• No students added	• Capacity: 3, current: 0 No students' name
10. No topic set & Projector is OFF	• No Topic • Projector's OFF • Option 8 (Exit)	• Program exits with no message showing
11. Invalid temp	• Temperature: 'a'	• Error Message

1. Alert “Room Full” (when only names are given as input and no topic is set and nor is the temperature):
   Notes: Worked as Expected
   Test run:

```
============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 3
Enter Student Name or ID: Mark
Mark added to classroom. Current students in room: 1


============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 3
Enter Student Name or ID: Bob
Bob added to classroom. Current students in room: 2


============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 3
Enter Student Name or ID: Wade
Wade added to classroom. Current students in room: 3


============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 3
The Room Is Full!
Student Can't Be Added!
```

```
============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 7
~~~~~~~~~~~~~~~Classroom Report~~~~~~~~~~~~~~~
Projector: OFF
Capacity: 3, Current: 3
ROOM FULL!
Topic: Not set
Students: bob, mark, wade
No temperature readings found!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

2. Alert “Temperature Warning” (with no student inputs):
   Notes: Worked as Expected
   Test run:

```
============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 5
Enter the temperature(°C) : 12
Recorded temperature: 12.0°C
Temperature Warning!


============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 7
~~~~~~~~~~~~~~~Classroom Report~~~~~~~~~~~~~~~
Projector: OFF
Capacity: 3, Current: 0
Topic: Not set
Students: None
Temperature stats:
Min:  12.00°C
Max:  12.00°C
Average =  12.00°C
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

3. Inserting names in different case (uppercase, lowercase, etc):
   Notes: Worked as Expected
   Test run:

```
============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 3
Enter Student Name or ID: Ari
Ari added to classroom. Current students in room: 1

============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 3
Enter Student Name or ID: aRi
Ari is already in the room.

============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 3
Enter Student Name or ID: Bob
Bob added to classroom. Current students in room: 2

============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 7
~~~~~~~~~~~~~~~Classroom Report~~~~~~~~~~~~~~~
Projector: OFF
Capacity: 3, Current: 2
Topic: Not set
Students: ari, bob
No temperature readings found!
```

4. Inserting ID number instead of name:
   Notes: Worked as Expected
   Test run:

```
============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 3
Enter Student Name or ID: u3312851
U3312851 added to classroom. Current students in room: 3
```

5. Different Temperature Input:
   Notes: Worked as Expected
   Test run:

```
============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 5
Enter the temperature(°C) : 18
Recorded temperature: 18.0°C


============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 5
Enter the temperature(°C) : 22
Recorded temperature: 22.0°C


============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 6
Temperature stats:
Min:  18.00°C
Max:  22.00°C
Average =  20.00°C
```

6. Program Exited When Topic is Set but Projector is OFF
   Notes: Worked as Expected
   Test run:

```
============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 2
Enter topic: rain
The Topic is: rain


============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 1
Projector status: OFF


============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 8
Reminder: projector is OFF but topic is set!
```

7. Topic Set but Projector is OFF:
   Notes: Worked as Expected
   Test run:

```
============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 2
Enter topic: Random
The Topic is: Random
Projector is OFF
```

8. Invalid Inputs:
   Notes: Worked as Expected
   Test run:

```
============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 9
Invalid choice. Please select 1-8.
```

9. Empty Class:
   Notes: Worked as Expected
   Test run:

```
============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 7
~~~~~~~~~~~~~~~Classroom Report~~~~~~~~~~~~~~~
Projector: OFF
Capacity: 3, Current: 0
Topic: Random
Students: None
No temperature readings found!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

10. No Topic is Set and Projector is OFF
    Notes: Worked as Expected
    Test run:

```
============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 7
~~~~~~~~~~~~~~~Classroom Report~~~~~~~~~~~~~~~
Projector: OFF
Capacity: 3, Current: 0
Topic: Not set
Students: None
No temperature readings found!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 8
```

11. Invalid Temperature
    Notes: Worked as Expected
    Test run:

```
============Smart Classroom Monitor============
1. Toggle Projector
2. Set Topic
3. Add Student
4. Remove Student
5. Add Temperature
6. View Temperature Stats
7. View Report
8. Exit
Choose (1-8): 5
Enter the temperature(°C) : a
Invalid temperature input
```

Step 7: Refinement via GenAI

I have used Microsoft Copilot to refine my code and the prompt that I used to do so:

“This is my Python classroom monitor program. Can you suggest improvements to make it cleaner and more user-friendly? It should accept both student names and ID numbers in attendance (not just letters). I also want stronger input validation (like stopping empty names or invalid temperatures), a clearer report format, and less overloaded functions. Keep the required alerts (room full, projector off, temperature warning).”

Before (issues in my code):

- Attendance didn’t check if the input was empty, so blank names could be added.
- Temperature input allowed invalid entries
- show_report() was overloaded, and the formatting was messy
- Code only displayed “Invalid” message for invalid inputs

After (changes suggested):

- Added input validation: rejects blanks, strips whitespace, and prevents duplicates
- Temperature input: forces valid float input, with reprompt until correct
- Formatted show_report() output into clear sections (Projector status, Topic, Attendance, Temperature Stats, Alerts)
- Broke show_report() into smaller helper functions by adding show_stats to it
- Added clearer messages like “No temperature readings found!” instead of just showing blanks

Justification (why I accepted changes):

- Rejecting blanks prevents useless entries and keeps the data clean.
- Stronger validation avoids crashes and improves user experience.
- Better report formatting makes it easy for the user to scan info quickly.
- Splitting functions makes the program more maintainable and clearer.
- Clearer alert messages are more understandable than just “Invalid”

I rejected one suggestion: GenAI suggested sorting attendance alphabetically, but I didn’t accept it because the order of entry is more natural for attendance logs.