Case Study 2 – Steps 6-7

Step 6 – Test

Access Cards:

Expected it to work and log access for students in an attendance tracker.

Expected to update the lecturer in the attendance tracker, and update the unit.

Expected Admin access to work – No logging or additional actions required

- When building ran into errors creating the solution – Most notably was relating to infinite loops
    - Resolved the issue by utilising a loop with continue
- Whilst implementing the logging for teachers and updating the class_unit, I was running into KeyErrors due to how I was attempting to update the unit.
    - Updated the code to reflect that the teacherCards dict is nested.

Equipment status:

Expected roomState() keys to update accordingly
Successful

Expected the reportStatus() to return no errors when calling if the states weren't updated:

- The biggest hurdle was stopping KeyErrors when attempting to call the report for the statusReport() function.
    - Resolved the error by attempting to get the value from the dict, but if there is no value setting it to unknown.

Attendance:

- Currently have an error where if the unit is not set, it returns <class: str>.
    - I have been unable to solve this issue, even by attempting to use the solution that I implemented for the Equipment status.
    - Will be asking AI how to error handle this in my AI integration (Step 7).

Case Study 2

Step 7 – Refine with GenAI

Prompt:

I am trying to add error handling for my code when attempting to call a value from a dict - When a value has not been entered into the dict (Which is "class_unit" : str) - it returns <class: str> This is how I am trying to fix it

```
def attendancePrint():
    currentUnit = roomState.get("class_unit", "Unknown")
    print(f"Lecturer: {lecturer} | Unit: {currentUnit}")
    print("\nStudents: ")
    for i in range(len(attendance)):
        print(f"{attendance[i]}")
    if len(attendance) == 0: #If no students are in the list
        print("No students registered in this classrom.")
```

(Comments removed for this document)

The suggestion from Co-Pilot was to re-write the attendancePrint function.
Co-Pilots solution kept the roomState.get and told me I was on the right track.

It added a check to make sure that the value is a string and not the str type. I have implemented this into my code as it does what I was attempting to do.

An easier solution Co-Pilot provided was centered around updating the class_unit value to “Unknown” when the code starts, which would simplify the code entirely and remove the need to check if it is a type or an actual string.

I also asked Co-Pilot if my projector and computer status checks and report are as optimised as they could be.
Prompt: Is this section of code optimised? Is there anything major you would change while keeping the structure? (*I inserted the code for def projectorStatus(), computerStatus(), equipStatus_report()*)

I only used the suggestion of adding .strip().upper() to the inputs, to reduce errors by users entering a lowercase y or n instead of uppercase.
I have integrated this into the code, and commented that the integration was the suggestion of Co-Pilot.

Stephen James | u3263912 | Intro to IT – 4478/8936 | Assignment II – Python

Response generated using Microsoft Co-Pilot to Stephen James, 10 September 2025, https://copilot.microsoft.com/shares/peAq52vJu6m1z4JqLSJ3d