Build 2 Report

The purpose of Build 2 was to enhance the basic access control system from Build 1 so that it more closely modelled a real-world customer entry system. The system was designed to detect approaching individuals using the ultrasonic sensor, enforce a health check before entry, track occupancy with entry and exit buttons, and block entry once the maximum capacity was reached. As soon as someone came within range of the ultrasonic sensor, the system signalled that an individual was approaching and communicated the occupancy status as LOW, MEDIUM, HIGH, or FULL. If the entry button was pressed without first pressing the health check button, access was denied and the Red LED lit up. If the health check button was pressed followed by the entry button, the Green LED lit up, "Access Granted" was printed, and the people count increased. The exit button decreased the count by one to reflect individuals leaving. If the people count equalled MAX_CAPACITY, the system immediately denied further entry attempts even before buttons were pressed. In this case, only the Red LED lit up and the system printed "Maximum Capacity Reached. Access Denied."

Build 1 successfully implemented health-based access control with buttons, LEDs, and an OLED display, but it had two shortcomings noted in the professor's feedback. First, there was no clear maximum capacity enforcement; it was not apparent whether the system actually stopped people from entering once the room was full. Second, approaching customers were not signalled with the relative occupancy level before considering entry. Build 2 directly corrected these issues. A fixed MAX_CAPACITY was set in the code, and once reached, new customers were barred from entry. The denial was clearly signalled through the Red LED and system messages. Additionally, the ultrasonic sensor was expanded to serve a dual purpose: it not only detected when a person approached but also triggered the occupancy status printout, informing individuals whether the room was in LOW, MEDIUM, HIGH, or FULL occupancy before they attempted to enter.

In Build 1, the OLED successfully displayed short status messages and people counts. However, in Build 2 the OLED failed to initialize on the I²C bus. At startup, the program stalled when the OLED was not detected at its expected address (0x3C). Attempts to bypass the fatal error still resulted in no device being found on the I²C bus, even when tested with an I²C scanner. This suggested either a hardware fault in the OLED module or an incompatibility with the hardware setup. To confirm, the system was first tested in ANSI C and then again in Arduino code; in both cases, the OLED could not be detected. Because the issue persisted across both environments, the OLED was completely disabled in software to allow the program to run without stalling. As a result, all system outputs—including health check status, access results, people count, and capacity messages—were redirected to the Serial Monitor and captured into a log file. This ensured the system remained fully functional and that all results could be preserved for analysis.

Two full test runs were performed and logged into .txt files to verify the system. The first run, with MAX_CAPACITY set to 10, was used as an initial functional test to confirm that the ultrasonic detection, button sequence, and LED indicators behaved as expected. Occupancy levels transitioned properly from LOW to MEDIUM to HIGH and finally FULL. Once capacity was reached, the system denied entry with the correct message and red LED indication. The second run, with MAX_CAPACITY set to 95, was conducted to fully validate the project requirements. This required a large number of manual button presses to increment the people count step by step, with random “Health Check Failed” and “Exit” cases included throughout. The resulting log contained over 800 lines of output, documenting every single approach, button event, occupancy update, and denial. Occupancy levels were correctly displayed as LOW for counts 0–37, MEDIUM for counts 38–75, HIGH for counts 76–94, and FULL at 95. At full occupancy, the system consistently blocked further entry attempts, as required. Both runs demonstrated that the system logic is stable, the maximum capacity enforcement is reliable, and all events are traceable through the generated .txt logs.

Link to Folder containing code, video and test logs:
https://drive.google.com/drive/folders/1P59muDyoP6vsFCKKahVEjha7I9KH9fV8?usp=sharing