Gary Szekely & Paige Ruschke
IPD

Documentation

GITHUB LINK: https://github.com/garyszekely/HackUMass_2020

SECTION 1: HARDWARE

- Arduino UNO
- Humidity Sensor (Contamination Level Sensor):
    - We used this sensor to simulate a “contamination level” scan.
- RGB LED:
    - Lights up on different events. Blue when scan is in progress and then either green or red depending on if the employee is above or below the contamination threshold given.
- Ultrasonic Sensor:
    - Determines the distance the employee is from the sensor to check contamination. Would alert the user if they are too far back or too close.
- Breadboard

SECTION 2: SOFTWARE

- Languages:
    - C++
    - Arduino (C++)
- Files:
    - Arduino Code:
        - arduino_code.ino:
            - Contains the main code for the arduino board. Simulates an employee stepping into a chamber to test their contamination levels. Starts by checking to see if the employee is within range of the sensor. It then takes their “contamination level” and alerts them if they should change PPE or if they are good for now.
    - Database Code:
        - database_main.cpp
            - *int main()*:
                - Runs the main part of the code. This includes displaying the options below and calling the functions to execute those options.
            - *void viewAllEmployees()*:
                - Loops through the database and formats the data for better viewing. It displays the data in the console and calculates how many employees there are.
            - *void viewEmployeesByCP(int threshold)*:

        - Loops through the database and finds the employees that are equal or greater than the passed in threshold. The threshold is specified by the user when executing the option. It also counts how many employees match this description.
    - *void viewEmployeesByTR(string desired_test_result)*:
        - Loops through the database and finds the employees have the passed in test result (positive or negative). It also counts how many employees have the test result and displays it for the user.
    - *void addEmployee(string first_name, string last_name)*:
        - Adds an employee with the desired first name and last name to the database and gives it a unique ID #.
    - *void removeByID(string id_to_remove)*:
        - Removes the employee with the same ID as passed in.
    - *void removeByName(string first_name_to_remove, string last_name_to_remove)*:
        - Removes the employee with the same first and last name as the strings passed in.
- testing_center.cpp:
    - *int main()*:
        - Runs the main part of the testing center's functionality. This involves inputting test results for a specified employee.
- database.txt:
    - Text file that contains the data used in the above files.