**Living Quality** **Project Documentation**

CTIEN year 3

Team: Condrat Radu, Nica Alexandru

Project Idea:

The objective of this project is to address the disparity in living conditions between urban and suburban areas by developing a comprehensive framework called "Living Quality". The framework incorporates various environmental sensors, including humidity, light, air, and sound sensors, strategically placed in different parts of a city. By comparing the collected sensor data with corresponding metrics outside the city, the Living Quality framework aims to provide valuable insights into the overall quality of living in urban areas. To achieve this, the project employs a cloud service, specifically Firebase, to store and process the gathered data, from which the user can determine whether the quality of living has improved or not.

Main Features:

Sensor Integration: The Living Quality framework integrates multiple sensors, including humidity, light, air, and sound sensors, which are strategically placed in various locations within the city. This allows for the comprehensive measurement of environmental conditions.

Comparison Analysis: The collected sensor data from the urban areas is compared with metrics recorded outside the city. This analysis enables a comprehensive evaluation of the living conditions and facilitates the identification of areas that require improvement.

Cloud Service: Firebase is utilized as the cloud service platform to store, process, and retrieve sensor data. By leveraging the capabilities of Firebase, the Living Quality framework can efficiently handle large volumes of data and ensure real-time access to the information.

Mobile Application: A dedicated mobile application is developed using Android Studio, to present the sensor readings and living quality analysis to users in an intuitive and user-friendly manner. The application allows users to access real-time sensor data and view historical trends.

Architecture:

The Living Quality measurement framework follows a distributed architecture, comprising several interconnected subsystems. These subsystems include IoT nodes with sensors, a cloud service, and a mobile application.

IoT Nodes:

Raspberry Pi: The Raspberry Pi serves as the IoT node, connecting to the humidity, light, air, and sound sensors placed in different areas of the city.

Sensors: The sensors capture and measure the respective environmental metrics, providing crucial data for the Living Quality framework.

Sensors used: MQ-135 air quality sensor, DHT-11 humidity sensor, CZN-15E sound detection sensor, LDR photoresistor.

Arduino: Some sensors provide analog data, so an Arduino board is connected to the main Raspberry Pi, with the data being sent through the USB serial port.

Sensor implementation:

Air quality sensor MQ-135, calibrated for CO2 reading, data is measured in ppm (parts per million): the sensor needs 5V VCC, which is supplied directly from the Raspberry Pi, with the data being sent first to the Arduino from the sensor’s analog output, then being sent to the Raspberry node through the USB port.

CZN-15E sound detection sensor, data is measured in decibels: 3.3V VCC, which is supplied directly from the Raspberry Pi, with the data being sent first to the Arduino from the sensor’s analog output, then being sent to the Raspberry node through the USB port.

DHT-11 humidity sensor: 3.3V VCC, which is supplied directly from the Raspberry Pi, with the data being sent to the Raspberry Pi. It needs a 10kOhm resistance between the output and power pins.

LDR Photoresistor, data is measured in light intensity: 3.3V VCC, which is supplied directly from the Raspberry Pi. It needs a 1microFarad capacitor connected to its ground side, with the output being between the photoresistor and the capacitor. Data is being sent directly to the Raspberry Pi.

Cloud Service:

Firebase: The Firebase platform is employed as the cloud service, offering scalable data storage, efficient data processing, and real-time communication capabilities.

Data Storage: Firebase stores the collected sensor readings and historical data, ensuring data integrity and accessibility.

Data Processing: The cloud service processes and performs necessary computations on the collected data, enabling efficient analysis and prediction.

API: An API facilitates seamless communication and data exchange between the IoT nodes, and the mobile application.

Mobile Application:

User Interface: The dedicated mobile application provides a graphical user interface that allows users to access and visualize the sensor readings, living quality analysis, and other relevant information. The interface is designed to be intuitive and user-friendly, providing an enhanced user experience.

Real-time Updates: The mobile application enables users to monitor changes in living conditions in real-time. Users can view historical trends and receive notifications regarding living quality improvements.

Diagram:

A diagram of a circuit board

Description automatically generated with low confidence

GitHub link:  
<https://github.com/ELKondrado/LivingQuality.git>

Code examples:

A screenshot of a computer code

Description automatically generated with medium confidence

Here can be seen the sensor data gathering from the Raspberry Pi, ready to be sent to the Firebase cloud service.

A screenshot of a computer

Description automatically generated with medium confidence

Presented here is the data gathering from the Arduino board for the sound and gas sensor, from where it is sent to the Raspberry Pi node through the USB serial port.

Future work:

The next step is to build a radiation detection system (Geiger counter) in order to observe how much radiation our living space emits. This can be integrated into the current project, with a little bit of finesse even make it as sellable product.

References:

The IIoTCA labs <3

<https://www.youtube.com/watch?v=qcNfXSe9CTI>

<https://github.com/5cottyD/Projects/blob/master/co2ppm_meter.ino>

<https://www.youtube.com/watch?v=pt4fVZL7iZo>

<https://roboticsbackend.com/raspberry-pi-arduino-serial-communication/>

<https://iotprojectsideas.com/arduino-based-decibel-meter-with-sound-sensor/>

<https://www.youtube.com/watch?v=IOyYQ34C2y0>