1. Project Showcase

Team 23 presented the demo of the current artifact to the other team, prioritizing its value over a set of slides.We showcased the functionality of the prototype which consisted of a MPU6050, ESP32, Python Server, and Capacitive Sensor. Initially, we elucidated the project vision and justified the selection of the ESP32 as the microcontroller of choice (WIFI Bluetooth capabilities). The demo then consisted of tilting the breadboard in various angles to present the idea of the wheelchair moving, not moving, and a crash. All of the respective axis metrics were displayed in the terminal. In the event of a crash, a "GET" request is dispatched to the Python Server, promptly processed to mitigate false positives. The server then showcases the request's status code and triggers an alert notification. It's worth noting that the Python server faced minor challenges attributed to eduroam firewall restrictions and the replacement of specific cables (data/power), impacting only the display of the notification, not the overall functionality. At another level of false positive prevention, we introduced a “deny” capacitive sensor to withstand any unnecessary requests made (more detail in Section 3). After the demo, it seems the peers thoroughly understood the status, affirming that our demo successfully conveyed the success of our vision.

2. Feedback Analysis

After the live demonstration of our project, there was a lot of concern about the reliability of the prototype.The first inquiry addressed the prototype's capability to handle scenarios beyond the wheelchair falling, such as situations where the user falls out of the wheelchair or if the wheelchair becomes stuck without falling.We were able to address these concerns by explaining to the peer evaluators that wheelchairs have seat belts so they will not fall out of the chair and that they could use their phones or call nearby pedestrians for help if they are just stuck, which is just much more reliable and specific than any program can be. Another potential problem that came up was the portability of the prototype. Presently, it's on a large breadboard linked to a laptop. Once sensors and actuators are finalized, we plan to transition it to a smaller breadboard, powered by a battery (leveraging standalone of ESP32s).The peers highlighted the final and most challenging issue: the Midas Touch/false positives problem. This concern has been a focus since the project's start, aiming to avoid sending unnecessary help requests when the user doesn't require assistance. Hence, we currently have 3 different parameters to check before sending a request. We check if the chair has fallen, if the chair is moving, and if the user is touching the “deny” wire. If the chair is moving or the user is “denying” the request, the code will not send a message even if a fall has been detected.However, this still has issues as this prototype relies on the capacitive sensor which has its own problems of false positives. The TA mentioned rain as a potential issue triggering an accidental "deny," prompting us to consider additional sensors for reliability, possibly opting for a button instead of the capacitive sensor.They also inquired about the button or capacitive sensors' placement, and we emphasized the strategic positioning to prevent accidental pressing or triggering by the operator.

3. Subsequent Timeline

In light of the future itinerary, we make an emphasis on that not everyone is affected the same way by the cerebral palsy condition so the design should be tailored to the operator. We plan to incorporate a magnetic contact switch, auditory output, and if feasible, a pulse sensor to add to the integrity of the project and address the false positive concern. Our timeline has been updated accordingly, specifically in the system implementation, signal process and validation phases.