# Component Analysis

**Year: 2023 Semester: Fall Team: 8 Project: Smart Seating System**
**Creation Date: 09/16/2023 Last Modified:** 09/16/2023

**Member1: Roshan Sundar Email: rmsundar@purdue.edu**
**Member2: Giang Nguyen Email: nguye683@purdue.edu**
**Member3: Gabriel Wang Email: wang4340@purdue.edu**
**Member4: Lee Dongeun Email: ldongeun@purdue.edu**

**Assignment Evaluation: See Rubric on Brightspace Assignment**

## 1.0 Component Analysis:

The most important component is the ESP32 microcontroller, which will live on the seat and facilitate the collection of occupancy data. A Force Resistive Sensor will also be used on top of the seat as a contact-based method of occupancy detection. A second sensor will also be used to gather sensor data in a non-contact way, which has yet to be fully determined; for this assignment, we will assume a human presence sensor. Finally, we will discuss the software component of the server.

## 1.1 Analysis of Component 1:

Below is a comparison table between the various micros we considered for the seat-mounted control units.

| Component | Rationale |
|---|---|
| ESP32 [Chosen] | The ESP32 comes integrated with Bluetooth and Wifi capabilities, as well as built-in low power modes. The tech stack of the ESP32 supports VScode, its own version of C called ESP-IDF, along with libraries for networking. |
| nRF52840-DK [Rejected] | The Nordic Brand has good options for connectivity (Zigbee & Bluetooth), as well as low-power modes. However, it is expensive (~$49) and is several times larger than the ESP32. The mounted seat unit should be relatively cheap and small, thus making the Nordic micros less applicable for our project. |
| STM32F091 [Rejected] | The STM32 is very quick, low-power, and cheap. However, it does not have built-in support for wireless connections like the ESP32, and it does not have its own native networking stack. Choosing this option would mean a lot of time would have to be spent writing networking code, which negatively affects our time constraint. |

## 1.1 Analysis of Component 2:

Below is a table comparing the various contact-based sensors we considered for seat occupancy detection.

| Component | Rationale |
|---|---|
| FSR [Chosen] | The FSR is a low-cost and low-power method of weight detection, though it cannot detect precise weight values. However, for our purposes this is fine. We determined that detecting the precise weight on the seat is not necessary, as we only need to determine if the seat is occupied. The FSR is sufficient enough to detect occupancy as a binary: is there a weight or not? |
| Load Cells [Rejected] | Load cells can give accurate weight measurements for a given load, at the cost of higher power consumption. As discussed above, an accurate weight measurement is not necessary for our purpose - negating the advantage of load cells. In addition load cells are more bulky than FSRs and we would likely need multiple since people's weight can get into the 100kg range |
| Capacitive Touch [Rejected] | Capacitive touch sensors are very sensitive compared to the other options. They also tend to have higher power requirements and can handle less weight than the other options. Our application needs a contact-based sensor that is able to handle the weight of a person. |

**1.1 Analysis of Component 3:**

Below is a table comparing the various contactless-based sensors we considered for seat occupancy detection. We haven't fully completed testing on them yet, but it does provide a summary of our thoughts.

| Component | Rationale |
|---|---|
| Human Presence Sensor AK9753 [Consideration] | The AK9753 is basically a multi-directional PIR sensor that is much more sensitive than other PIR sensors. We have tested it by mounting it underneath a seat and seeing whether it can detect motion. It is able to detect motion quite well, perhaps too well. It has a range (at least 1 meter) that exceeds the bounds of the seat by a lot. That means human presence/motion in nearby seats may be detected. We are still trying to find ways to restrict the range of vision on the sensor. |
| Human Presence Sensor LD2140 [Consideration] | The LD2140 only just arrived after ordering, so we haven't tested it yet. As per the datasheet though, it is able to detect both movement and static behavior of a person, as well as a distance measurement; unlike the AK9753. It remains to be seen how accurate/effective it is though. |
| PIR Sensor HC-SR501 | The HC-SR501 PIR sensor can only detect motion in a limited FOV and is not as sensitive as the AK9753. There are potentiometers to adjust the |

| | |
|---|---|
| [Rejected] | sensitivity and trigger time; however, the system in a deployed environment would require them to all be calibrated the same, which is hard to do with potentiometers. |

**1.1 Analysis of Component 4:**

Finally we will consider software options for a change, namely for the Web Server that is collecting the sensor data from all the seat-mounted Control Units.

| Tech Stack | Rationale |
|---|---|
| HTML/CSS Frontend [Unlikely] | HTML/CSS has been integral to webpage development for many years. However, other newer tools such as website creators, and various javascript frameworks, provide a much easier development experience than working with HTML/CSS directly. |
| React.js Frontend [Likely] | The React frontend framework allows for highly modularized and customizable javascript-based web UI development. It removes the need to work with HTML/CSS directly, and is increasingly becoming the industry standard. |
| Website Creators [Unlikely] | There are many website creation tools that can simplify the creation of websites down to drag-and-drop. However, these do not provide a very nice way to plug custom functionality into the page. We need the ability to have custom code in order to calculate occupancy for example, which can't easily be done natively using these template-based website creation tools. |

**2.0 Sources Cited:**

No direct sources were referenced in the creation of this document.