**Capstone Project Description**

**Team ForceField**

**Goal**

Design an easy-to-use, simple installation anti-theft system that enables construction companies and the common power tool owner the ability to operate and control the use of their equipment. The design should be durable, reliable, and low powered. Users should be able to control authorized and unauthorized use of each tool they own.

**Solution**

The design will be divided into four key parts: 1- Tool, 2- Transponders, 3- Hub, 4- Amazon Web Services (AWS). Each part is interconnected wirelessly through a secure channel. The end product will consist of three prototype devices that include a fully functional hub, transponder and tool. The tool device will contain a customized printed circuit board, integrated circuit components, sensors, a Raspberry Pi controller and a battery. What makes Forcefield unique is the application and utility of common electronics and software in the hub, transponder and power tool.

Diagram

Description automatically generated

Figure 1: System Block Diagram

* **Tool:**

The controller will be able to gather data on the different sensors of the tool such as temperature, vibration, and time of use. This data will be continuously uploaded to the transponder for safe storing (more on transponder in following section). The data will ultimately be gathered by the Hub and uploaded to the internet to be processed (more on Hub and information processing in subsequent sections). If a tool runs out of the timer, the circuitry will be designed to have an LED light to show the user the tool’s status.

* **Transponders:**

The transponders are durable portable devices which communicate to power tools and the hub using SSH (Secure Shell) protocol. A Raspberry Pi Zero will be used and controlled using Python. During the workday timer, the tools will be enabled, once the timer is up, the tools will be deactivated. Transponder contains a timer set by the main hub, allowing the tools to only work during set hours. The timer can be configured by owner/foreman through AWS (more on this in subsequent sections).

* **Hub:**

The main hub is designed to function as the central “brain” of the product by remotely communicating with the transponders when they are within its vicinity—this will be simulated by disabling the AWS to Hub communications. The hub can collect data from the transponder and communicate it to AWS for processing. The user can control the system remotely through a website (AWS) and configure any of the above mentioned transponder features and tool data. The main hub will communicate to the transponder through wifi, which shows that it is in the vicinity of the hub, allowing it to upload its data to the Hub which in turn transfers the data to the internet (AWS application) for processing.

* **AWS:**

We will be using Amazon Web Services to create a cloud based database that will be used to communicate between the tools, transponders/app and the main hub. In the user’s computer or device, the host will have access to the tool ID, and tool and transponder status all within the AWS application. The AWS allows the user to add or remove a tool or transponder from the system as needed. ~~When approaching the main hub, the user’s device will automatically connect to the hub after their first authenticated login. Settings within the main hub will permit the user to add or remove tools and transponders to its inventory by toggling.~~ When the transponder establishes connection with the hub, the transponder downloads over SSH the data gathered from the tools it oversees which in turn uploads the data to AWS

**AWS Flowchart**

This flowchart explains the AWS services used for the project. Once the Hub transfers data to the AWS cloud, it gets processed through the various services below to be cleaned up before being displayed as graphs.

