This project is called 'stockHome'. I have made a simple weighing scales using a 5kg load cell, as seen in figure 1.
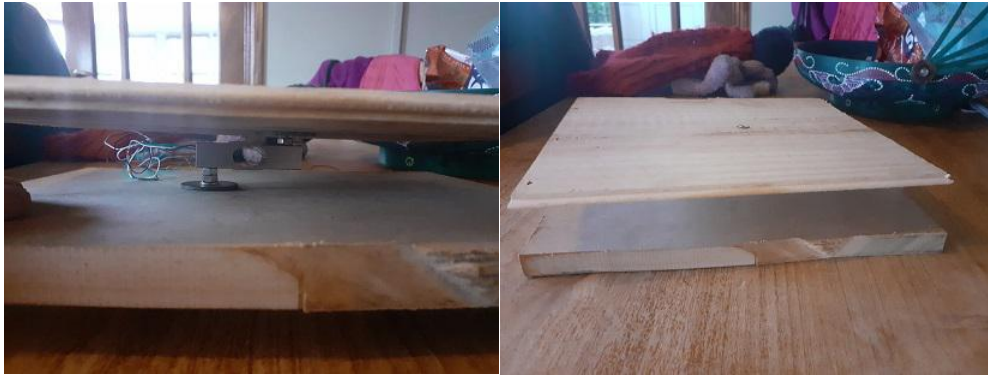

Figure 1 Photos of weighing device

The load cell has strain gauges attached to it, their electrical resistance changes with small deformations in the load cell when an item is placed on top of the device. This change in resistance results in a very small analog voltage signal to be sent through the output wires of the load cell. As this voltage signal is very small, it needs to be amplified. The signal also needs conversion from analog to digital for processing via code. Thankfully, someone has made a small commercially available PCB to do both of these things. I will be using a raspberry Pi and Python programming language to receive and process the digital signals. The hookup is as seen in Figure 2.
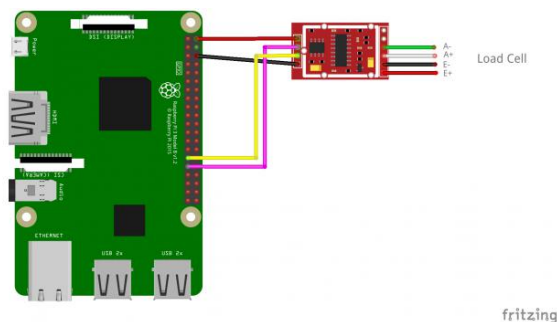

Figure 2: Load Cell, ADC and Amplifier, and Rpi

The goal of the project is to propose a concept for a system of measuring the weight of typical household items and utilities in order to inform the user about how much of a particular item they have left. A user could be in a shopping centre, and can hit a button to display how much of a particular item they have left. For example, the user can use the system to input 10 toilet rolls as 100% full stock, and a display will show them they have 100% stock when they have 10 toilet rolls. When people in the home have used 9 toilet rolls and a person presses the update button, they will see a reading of 10% stock remaining and will know to pick up some more.

The PCB containing the ADC and amplifier is known as a "HX711 Load Cell Amplifier". It converts the small analog signal to a 24 bit digital output. This digital output is in the form of an I2C serial protocol. The Rpi sends a signal when it is ready to receive the next bit, allowing a way to process each individual bit using Python. After some calibration the bytes can be converted into a weight reading. This weight reading is communicated via TCP/IP (SSH) from the Rpi to a laptop. The information will then be sent to a simple web app (developed with Glitch or Webstorm) using HTTP. Collected data (usage trends) will be analysed and presented using ThingSpeak, with a plan to evoke messaging to inform the user of unusual, excessive usage or depletion of stock.