# Software Projects Milestone 5

## System Requirements & Technical Specification

## Group Number: 24

## Concept Name: Scale Shelves

1. **Purpose: Purpose of the product**

The purpose of our product is to reduce the time and effort spent on inventory management. We plan to achieve this by collecting current and past stock information and displaying it in an application for easy access. Stock warnings and ability to check stock levels at a glance should reduce the time spent over manual inventory systems.

**2. Scope:** **What can users expect to be able to do with this product**

**Functional requirements:**

We considered which functions were considered essential to users, below is an outline of the features of our Minimum Viable Product (MVP):

1. Add Items to shelf (set up)
   * set up/calibrate item weight
2. Remove items from shelf (delete)
3. Swap items/shelf spaces
4. Update shelf stock level (automatically, set time period)
5. Display stock level:
   * Number of items
   * Absolute weight
   * Percentage weight
     + Manually set 100%
     + Auto Determine 100% (by looking through weight history)
6. Stock level history
7. Low stock warnings within app
8. Form field to store re-supply information
9. Search
   * Search by name or search by 'tags'
10. Amend/ modify item details

These features can be translated to two four key functional components:

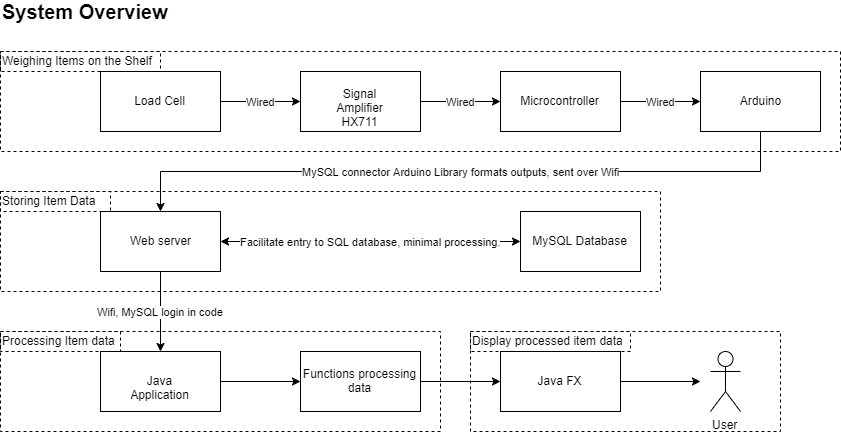
* 1. Weigh items on shelf.
  2. Store item data, both from the shelf and from user input.
  3. Process raw Item data so that is useful to the user.
  4. Display processed item data to user.

**Non-functional requirements:**

After our prototype consultation our stakeholders relayed, they would prefer the app be displayed on a desktop. Development will be taking place over a relatively small time frame of 10 weeks, so we have decided to use Java to code the desktop app, as it is the language most familiar to our team.

1. A desktop app (coded in Java).
2. As much of the data processing and calculations are to take place in the client.
3. The server will host the storage and perform few calculations.

**3. System Overview:**

******

*Load cell -> Signal Amplifier, HX711, microcontroller -> Arduino | all wired communication*

*Arduino convert signal -> formatted SQL command to add converted data | Internal processing*

*Arduino -> MySQL Connector Arduino Library - > MySQL server | direct connection over Wi-Fi*

*Web server – manage logins/ permissible connections. Mainly just perform database operations.*

*MySQL server - > inbuilt java.sql functions | Wifi (SSH?)*

*Java app -> Process data using functions we build -> JAVA FX -> Display to user (using electromagnetic radiation fields)*

**4. References:**

- MySQL connector/Arduino library – Chuck Bell : <https://github.com/ChuckBell/MySQL_Connector_Arduino>

(more references will appear in the proposal, properly formatted)

**5. Definitions:**

None.